BACHELOROPPGAVE:

Norkart ID - Single sign-on autentication for Norkart

FORFATTERE:
Per Christian Kofstad
Ida F. Granholt
Alf Magnus K. Hammerseth

DATO:
Vår 2015

Sammendrag av Bacheloroppgaven

Tittel:	**Norkart ID - Single sign-on autentisering for Norkart**
Dato:	Vår 2015
Deltakere:	Per Christian Kofstad Ida F. Granholt Alf Magnus K. Hammerseth
Veiledere:	Frode Haug Eigil Obrestad
Oppdragsgiver:	Norkart
Kontaktperson:	Håkon Sagehaug, Einar Tomter
Nøkkelord:	Autentisering, Microsoft Azure AD, Sikkerhet, Single sign-on
Antall sider:	155
Antall vedlegg:	9
Tilgjengelighet:	Åpen

Sammendrag:

Norkart ID er et prosjekt som undersøker nye single sign-on autentiseringsløsninger. Det er et oppstarts prosjekt for å implementere en ny felles autentiseringsløsning for applikasjoner bedriften Norkart leverer til sine kunder. Prosjektgruppen har tatt for seg to ulike løsninger som begge har styrker og svakheter i forhold til hva Norkart trenger. Azure AD ble valgt av løsning. Prosjektet har hatt som mål å samle breddeinformasjon om valgt løsning framfor å gå i dybden på spesifikke funksjoner. Den valgte løsningen er bruker- og funksjonstestet, samt satt opp i et testmiljø for å kontrollere at løsningen svarer til krav gitt av oppdragsgiver. Prosjektgruppen har forsøkt å se på alle relevante aspekter rundt konfigurering av løsningen, med unntak av juridiske spørsmål for lagring utenfor Norge. I tillegg til valg, dokumentasjon og testing er det utarbeidet to brukerveiledninger for hvordan tilknytningen mot Azure AD kan settes opp. Prosjektgruppen konkluderte med at kravene møtes dersom det utvikles en egen brukerportal i tillegg til bruk av Azure AD.

Summary of Graduate Project

Title:	**Norkart ID - Single sign-on autentication for Norkart**
Date:	Vår 2015
Participants:	Per Christian Kofstad Ida F. Granholt Alf Magnus K. Hammerseth
Supervisor:	Frode Haug Eigil Obrestad
Employer:	Norkart
Contact Person:	Håkon Sagehaug, Einar Tomter
Keywords:	Autentication, Microsoft Azure AD, Security, Single sign-on
Pages:	155
Attachments:	9
Availability:	Open

Abstract: Norkart ID is a project that looks into a new single sign-on authentication solution. It is a kick-start project for the implementation of a new authentication solution for applications Norkart offers to costumers. The project has looked into two different solutions that both have strengths and weaknesses according to what Norkart needs. The chosen solution was Azure AD. The goal of the project has been to gather a wide range of relevant information about the solution and not focus on specific areas. The chosen solution has been user- and functional tested and implemented in a testing environment, to ensure it meets the requirements that Norkart has defined. The project has not gone into legal questions regarding storing personal information outside Norway. In addition to choosing a solution, documenting and testing it, the group has written two tutorials on how to connect applications against Azure AD. The project concludes that the chosen solution meets the requirements, if a user-portal is developed.

Forord

Dette prosjektet ble noe helt annet enn det vi i prosjektgruppen først så for oss. Likevel, mener vi alle det har vært svært lærerikt og en god erfaring å ha med seg videre. Vi ser på prosjektperioden som en utfordrende, men bra periode.

Vi vil takke oppdragsgiver, Norkart, som ga oss oppgaven og stilte med både veileder og produkteier for prosjektet. De har vært imøtekommende og hjelpsomme hele perioden. Norkart er en bedrift på 153 ansatte som holder til på Lillehammer, Trondheim, Bergen og Sandvika.

I tillegg vil vi også takke de som har tatt seg tid til å lese gjennom rapporten og komme med gode tilbakemeldinger og innspill. Dette er blant annet veilederene våre og bachelorgruppen BioDemo. Vi retter også en takk til de testpersonene som stilte opp.

Per Christian, Alf og Ida

Innhold

Tabeller

Figurer

Kodeutdrag

Ordliste

- **AAD** Azure Active Directory.
- **AD** Active Directory.
- **GUI** Graphical User Interface.
- **GUID** Globally Unique Identifier.
- **HiG** Høgskolen i Gjøvik.
- **HTTP** Hypertext Transfer Protocol.
- **IaaS** Infrastructure-as-a-Service
- **JSON** JavaScript Object Notation.
- **JWT** JSON Web Token.
- **LDAP** Lightweight Directory Access Protocol.
- **MVC** Model View Controller.
- **MVP** Minimum Viable Product.
- **OIDC** OpenID Connect.
- **PaaS** Platform-as-a-Service
- **PBI** Product Backlog Items.
- **REST** Representational State Transfer.
- **SAML** Security Assertion Markup Language.
- **SLA** Service Level Agreement.
- **SOAP** Simple Object Access Protocol.
- **SSL** Secure Sockets Layer.
- **SSO** Single sign-on.
- **SUS** System Usabillity Scale.
- **WCAG 2.0** Web Content Accessibility Guidelines 2.0.
- **XML** Exstensible Markup Language.

Definisjoner

- **JSON** JavaScript Object Notation. En enkel tekstbasert standard for datautveksling.
- **JWT** JSON Web Token. En sikker måte å over claims mellom to parter.
- **LDAP** Lightweight Directory Access Protocol. En protokoll som brukes til oppslag i en katalogtjeneste på en server.
- **MVP** Minimum Viable Product. En versjon av produktet som dekker kundens behov og går gjennom en Lean Startup iterasjon med minst mulig bruk av ressurser.
- **OASIS** "Advancing open standards for the information security". En non-profit open-source organisasjon.
- **PowerShell** Kommandolinje rammeverk for konfigurering, skripting og automatisering av oppgaver.
- **REST** Representational State Transfer. En software arkitektstil som følger retningslinjer og best practice for å lage skalerbare web tjenester.
- **SLA** Service Level Agreement. En kontrakt som skal sikre at kunde og leverandør har en felles forståelse for hva som skal leveres, og til hvilken kvalitet.
- **SOAP** Simple Object Access Protocol. En protokoll for utveksling av XML-baserte meldinger.
- **SUS** System Usabillity Scale. Et verktøy for å måle brukervennlighet.
- **XML** Exstensible Markup Language. Et universelt og utvidbart markeringsspråk. Brukes for deling av data mellom systemer, spesielt over internett og for koding av dokumenter.

1 Innledning

Oppgaven går ut på at prosjektgruppen skal velge og sette seg inn i en single sign-on (SSO) autentiseringsløsning for Norkart. Dette kapittelet beskriver oppgaven og rapportens struktur. Kapittelet vil også si noe om endringer underveis, og hvordan dette har preget prosjektet og rapporten.

1.1 Oppgaven

Dette er et direkte utdrag fra den opprinnelige oppgaveteksten prosjektgruppen har bygget bacheloroppgaven på:

> Vi ønsker at dere skal lage Norkart sin nye autentiseringstjeneste. Løsningen skal inneholde webgrensesnitt for sluttbruker på web for alle skjermstørrelser, administrasjonsgrensesnitt for lokale superbrukere, Norkart sin kundestøtte og driftstjeneste. Oppgaven innebærer å vurdere aktuelle tekniske rammeverk og benytte disse til å bygge tjenesten.
>
> Løsningen skal driftsettes på dedikerte servere i Norkart sitt driftsmiljø og vil derfra tas i bruk som autentiseringstjeneste for Norkart sine tjenester og produkter på web, native apper og windowsbasert desktop programvare.
>
> Vi er åpne for å diskutere omfang og evt også vinkling dersom dette er ønskelig.

Dette er oppgaveteksten fra bacheloroppgaven prosjektgruppen fikk av Norkart 2. oktober 2014.

Norkart har tidligere hatt et eget autentiseringssystem for hver applikasjon de har levert. Autentiseringsmekanismer og sikkerhet har ikke blitt prioritert like høyt som funksjonalitet i applikasjonene.

1.1.1 Problemområde

Autentisering av brukere for tilgang til beskyttede ressurser er utfordrende. Autentisering over internett krever utprøvde og sikre rammeverk for å garantere at autentiseringen gjøres på en trygg måte.

Når brukere har mange brukerkontoer å forholde seg til, kan det være utfordrende å huske brukernavn og passord for alle kontoene. Dette kan føre til at brukerne ender opp med å ha det samme brukernavnet og passordet på flere kontoer, noe som øker risiko for at kontoer blir kompromittert. Konsekvensen av dette vil ramme både bruker og bedrift.

I tillegg er det en utfordring for tjenestetilbyder av applikasjoner og brukerkontoer å vedlikehode og drifte alle autentiseringssystemene som brukes. Dette fører til stor sannsynlighet for at ikke alle autentiseringssystemene er sikre nok.

En brukerkonto for flere tjenester gjør det enklere å ha kontroll på brukernavn og pass-

ord. I tillegg kan tjenestetilbyderen fokusere på vedlikehold og drift av kun et autentiseringssystem. Et eksempel på en slik tjeneste er FEIDE som er autentiseringsløsningen brukt i utdanningssektoren i Norge [1].

1.1.2 Avgrensning

Oppgaven avgrenses til å omhandle faglig dypdykk. Dette innebærer å kartlegge muligheter og begrensninger opp mot krav prosjektgruppen utarbeidet sammen med oppdragsgiver.

Prosjektgruppen har i samråd med oppdragsgiver valgt å fokusere på autentisering for web og Android applikasjoner.

Oppdragsgiver ønsket at prosjektgruppen ikke skulle ta hensyn til juridiske spørsmål i forhold til lagring av persondata i utlandet.

1.1.3 Problemstilling

Prosjektgruppen har underveis i prosjektperioden måttet endre problemstilling i samarbeid med oppdragsgiver (jfr delkapittel 1.3). Dette ble siste versjon av problemstillingen:

> Prosjektgruppen skal definere, velge, planlegge og teste en ny autentiseringsløsning for Norkart.

Autentiseringsløsningen omtales som Norkart ID. Oppgaven ble delt i to der første del var å definere krav og finne en løsning som tilfredstile disse. Del to beskriver implementasjon, mulige løsninger, utfordringer og problemstillinger ved implementerering av Norkart ID.

Norkart planlegger å gjennomføre et prosjekt for å implementere en autentiseringsløsning mot sine systemer høsten 2015. De ser på denne bacheloroppgaven som en oppstart til dette.

1.2 Prosjektmål

Prosjektmålene er sammensatt av resultatmål og effektmål, i tillegg har vi definert noen læringsmål. Prosjektmålene har bachelorgruppen definert selv, med utgangspunkt i oppgaven og problemstillingen.

1.2.1 Resultatmål

Dette er mål som skal foreligge når prosjektet er ferdig.

- Kartlegge teknologier og muligheter innenfor definerte rammer gitt av Norkart for å løse prosjektoppgaven.
- Introdusere Norkart for Azure AD (AAD) og OpenID Connect (OIDC).
- Kick-starte utviklingsprosjektet Norkart ID.
- Øke sikkerhet i forbindelse med brukerhåndtering og tilgangsstyring av tjenestene til Norkart.

1.2.2 Effektmål

Dette er mål som oppdragsgiver ønsker å oppnå etter løsningen er tatt i bruk.

- Redusere tid brukt på innlogging til Norkart sine tjenester.
- Forenkle innlogging ved å kun ha et brukernavn og passord for alle tjenester.
- Redusere tid brukt på administrasjon av brukere.
- Øke sikkerhet rundt brukerhåndtering av tjenester levert av Norkart.

1.2.3 Læringsmål

Prosjektgruppens ønsket kompetanse etter endt prosjekt.

- Lære om informasjonsikkerhet relatert til autentiseringstjenester, innlogging og utfordringer rundt brukeradministrasjon.
- Bygge kunnskap og erfaringer ved bruk av AAD.
- Bygge erfaringer ved å jobbe på tvers av linjer med personer som har ulik bakgrunn.
- Trene på bruk av arbeidsmetodikk i større prosjekter.

1.3 Endring av oppgaven

I utviklingsfasens første sprint planla gruppen å utvikle et ”proof of concept” på en autentiseringsløsning (jfr vedlegg F.1 Arbeidsplan). Gruppen satte seg inn i allerede eksisterende programvare som kunne benyttes, blant annet AAD fra Microsoft. Det ble da funnet ut at AAD allerede inneholdt funksjonaliteten gruppen hadde planlagt å utvikle. Dette betydde at det ikke ble nødvendig å utvikle en mellomvare, slik både gruppen og oppdragsgiver hadde sett for seg i første omgang. Gruppen brukte derfor sprint en til å undersøke om AAD var et reelt alternativ til den løsningen prosjektgruppen opprinnelig planla å utvikle.

I gjennomgangsmøte for sprint en den 5. mars 2015 (jfr vedlegg A.1.2) ble det bestemt av både oppdragsgiver og prosjektgruppen at problemstillingen skulle endres fra design og utvikling, til å gå i dybden av AAD. Ordlyden i problemstillingen ble endret fra ”velge ut, designe, utvikle og teste” til ”velge ut, planlegge og teste”. Se kapittelet om valgt løsning (jfr 4) for begrunnelse.

Ettersom prosjektets oppgavebeskrivelse (jfr delkapittel 1.1 Oppgaven) var generell, passet den nye problemstillingen selv om den endret seg. Istedenfor å utvikle et ”proof of concept” på en autentiseringsløsning, resulterte endringen i at prosjektgruppen kunne se på en full innføring av Norkart ID som et konsept.

I delkapittelet om oppgaven (jfr 1.1) presiseres det at autentiseringssytemet ”skal driftes på dedikerte servere i Norkart sitt driftsmiljø”. Dette kravet ble strøket da valget om å gå nærmere inn på AAD ble tatt.

Når endringen av problemstillingen ble vedtatt hadde prosjektgruppen allerede jobbet en og en halv måned med kravspesifikasjon og design for IdentityServer3. Prosjektgruppen valgte å legge dette arbeidet ved i rapporten for å vise at det ble gjort en jobb også før oppgaven ble endret. Les mer om hvordan dette er lagt inn i rapporten i delkapittel om organisering av rapporten (jfr 1.8).

1.4 Målgruppe for rapporten

Rapportens målgruppe er sensor, utviklere, medstudenter og ansatte hos Norkart. Varierende kunnskap blant målgruppene i forhold til ulike teknologier som beskrives i prosjektet er en av grunnene til at prosjektgruppen har valgt å lage et teorikapittel (jfr kapittel 3). Dette skal gi leserne mulighet til relevant, oppdatert og konsentrert informasjon om de teknologiene som er beskrevet.

1.5 Valg av oppgaven

Ingen av gruppemedlemmene hadde tidligere fordypet seg i teknologier som brukes for autentisering av brukere. Alle i prosjektgruppen ønsket å lære om autentisering, implementasjon og autentiseringsløsninger. Tverrfaglige grupper ved HiG har selv ansvar for å finne oppgaver. Gruppen sendte ut forespørsler til flere bedrifter, og hadde på et tidspunkt potensielle oppgaver fra flere oppdragsgivere. Norkart sin problemstilling ble valgt med bakgrunn i oppgavens utfordring og det gode inntrykket Norkart ga i forhold til oppfølging, veiledningsmuligheter og faglig kompetanse. Flere av de andre potensielle oppdragsgiverne kunne ikke stille med veiledere til prosjektet på samme måte som Norkart skisserte. Hvordan veiledningen fungerte drøftes i avslutningskapittelet (jfr kapittel 10). Prosjektgruppen var delaktig i prosessen om å definere problemstillingen.

1.6 Prosjektgruppens faglige bakgrunn

Prosjektgruppen besto av tre studenter med ulik bakgrunn og tok selv initiativet til å jobbe på tvers av studieprogram. Dette delkapittelet er lagt inn for å presentere prosjektgruppens sammensetning av faglig bakgrunn og er en kort beskrivelse av hvert gruppemedlem for å tydeliggjøre forskjellene. Gruppemedlemmene er fra tre ulike linjer ved Høgskolen i Gjøvik (HiG).

Alf Hammerseth

Alf har gått på Fagskolen i Innlandet, avdeling Gjøvik før han begynte ved HiG. I prosjektperioden gikk han ingeniørfag data ved HiG og har fordypet seg i programmering. Han tilbrakte et semester ved University of Wollongong i Australia. Gjennom studietiden ved HiG har han tatt en rekke fag utover læreplanen. Alf sine interessefelter er drift av servermiljøer og programmering.

Ida Granholdt

Ida har en bachelor i grafisk design fra Metropolitan University. I prosjektperioden tok hun bachelor i webutvikling ved HiG og har fordypet seg i programmering. Interesserfeltene til Ida er design, webutvikling og programmering. Under studieteiden jobbet hun deltid som selvstendig næringsdrivende og fagassistent i en rekke fag.

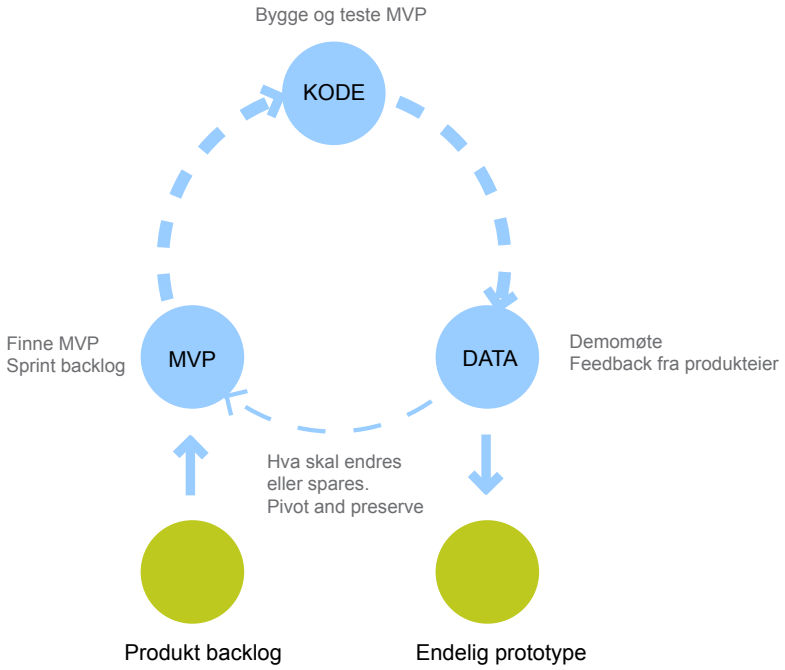
Per Christian Kofstad

Per Christian jobbet tidligere 6 år i forsvaret som sambandsspesialist med fokusområde på sikkerhet. I prosjektperioden gikk han informasjonssikkerhet ved HiG og har fordypet seg i programmering. Han er interessert i programvareutvikling og sikkerhet. Per Christian har under studietiden hatt deltidsjobb som sambandsansvarlig i en innsatsstyrke i Heimevernet og fagassistent i rekke fag.

1.7 Prosjektgruppens valgte arbeidsform

Ettersom prosjektet skulle ferdiggjøres på kun noen måneder ble det valgt å bruke en smidig utviklingsmodell. Problemstillingen skulle løses med teknologier som var ukjente både for oppdragsgiver og prosjektgruppen, derfor kunne endringshyppigheten være stor. På grunn av dette ble gruppen anbefalt av oppdragsgiver å bruke elementer fra Lean Startup. Det ble derfor valgt å bruke Minimum Viable Product (MVP) og varierende lengde på iterasjoner. I tillegg ble det brukt elementer fra Scrum ettersom både oppdragsgiver og gruppen var kjent med dette. I hovedsak ble det brukt roller, møter og product backlog fra Scrum.

Product backlog besto av Product Backlog Items (PBI) knyttet til elementer i rapporten. For å dokumentere utviklingsprosessen ble det skrevet møtereferater fra planleggingsmøter, vurderingsmøter, retrospektive møter og demomøter. Møtereferater fra alle disse møtene ligger som vedlegg (jfr vedlegg A).



Figur 1: Utviklingsmodell for prosjektet.

1.7.1 Iterasjoner

Hver iterasjon skulle fungere tilsvarende iterasjonsirkelen som vist i figur 1. Sirklen ble definert av prosjektgruppen med inspirasjon fra Lean Startup og Scrum.

MVP

Første fase i iterasjonen var MVP. I denne fasen ble det holdt et planleggingsmøte hvor målet var å definere en MVP for oppkommende sprint.

KODE

Kodefasen er for å utvikle og teste MVP.

DATA

Data var den siste fasen i iterasjonen. Her ble det holdt et gjennomgangsmøte hvor resultatet av iterasjonen ble presentert for oppdragsgiver som ga tilbakemelding. I tillegg ble det holdt et retrospektivt møte.

Etter datafasen i iterasjonen ble det holdt et endringsmøte hvor gruppen bestemte om utviklingen kunne gå videre i neste iterasjon eller om MVP måtte endres eller gjøres på nytt.

1.7.2 Roller

Gruppen brukte roller fra Scrum. Oppdragsgiver stilte med produkteier og Scrum master (omtalt som veileder), mens medlemmene i gruppa var team medlemmer.

1.7.3 Møter

Møter ble holdt for å skape oversikt og dokumentere utviklingsprosessen.

Planleggingsmøte

Når:	I MVPfasen
Deltagere:	Gruppen
Hensikt:	Fastsette en MVP og hvilke tasker som trengs for å utvikle denne i kommende sprint.

Gjennomgangsmøte

Når:	I datafasen
Deltagere:	Gruppen og kontaktpersoner hos Norkart
Hensikt:	Presentere utviklet MVP for oppdragsgiver. Oppdragsgiver gir tilbakemelding på MVP.

Retrospektivtmøte

Når:	I datafasen
Deltagere:	Gruppen
Hensikt:	Gjennomgang av utviklingsperioden som har vært, samt finne eventuelle endringer som bør gjøres før neste iterasjon. Dette har til hensikt å skape overblikk over fremdrift.

Endringsmøte

Når:	Mellom to iterasjoner
Deltagere:	Gruppen
Hensikt:	Bestemmer om utviklingen kan gå videre eller om MVP må endres eller gjøres på nytt.

Demomøte

Når:	Under hvert beslutningspunkt
Deltagere:	Gruppen og kontaktpersoner hos Norkart
Hensikt:	Oppdragsgiver får en demo av Norkart ID.

I tillegg til møtene nevnt ovenfor kunne gruppen møte med veiledere en gang i uken. Gruppen skulle benyttet seg også av Scrums daily standup og statusmøter hver mandag.

1.7.4 Ansvarsforhold og roller

Gruppeleder er Alf og hans rolle var å holde oversikt over gruppen, samt se til at det som skulle gjøres ble gjort. Ved konflikter og uenighet skulle veileder ta avgjørelsen sammen med gruppen. Ida var informasjonsforvalter og håndterte informasjonsflyten innad i gruppen. Per Christian fungerte som møteleder ved gruppemøter og var teknologiansvarlig. Han skulle passe på at gruppen forholdt seg til den satte agendaen.

Overordnet ansvar for at punktene ovenfor ble overholdt falt på gruppeleder. Ved eventuelle behov for signaturer skulle gruppeleder signere.

Oppdragsgiver

Navn: Håkon Sagehaug, Norkart, Veileder
Epost: hakon.sagehaug@norkart.no

Navn: Einar Tomter, Norkart, Produkteier
Epost: einar.tomter@norkart.no

Veiledere

Navn: Frode Haug
Epost: frode.haug@hig.no

Navn: Eigil Obrestad
Epost: Eigilo@hig.no

1.8 Organisering av rapporten

Første kapittel (Innledning) har til hensikt å sette rapportleser inn i oppgaven, prosjektarbeidet og rapportens oppbygning. Kapittel 2 inneholder den siste kravspesifikasjonen prosjektgruppen lagde. Kapittel 3 (Teoridel) inneholder en introduksjon av ulike teknologier og begreper som er relatert med problemstillingens løsning. Teorikapittelet kan sees på som et oppslagsverk.

I kapittelene 4 (Valg av løsning), 5 (Konfigurasjon) og 7 (Testing) drøftes oppgavens problemstilling. Her kombineres teori, drøfting, og argumentasjon for å belyse valg prosjektgruppen tok i løpet av prosjektet. Kapittel 6 (Veiledninger) består av to brukerveiledninger for implementasjon av autentisering fra web- og Androidapplikasjoner. Kapittel 10 (Avslutning) har til hensikt å sammenfatte og konkludere med funn vi gjorde i prosjektperioden.

Todelt rapport

Prosjektgruppen har valgt å ta med to kapitler gruppen jobbet mye med før problemstillingen ble endret. Disse ligger som kapittel 8 (Første utarbeidede kravspesifikasjon) og 9 (Arkitektur og design av IdentityServer3) i rapporten. Kapittel 8 (Første utarbeidede kravspesifikasjon) er den opprinnelige kravspesifikasjonen som ble laget med tanke på at prosjektgruppen skulle utvikle en mellomvare. Kapittel 9 (Arkitektur og design av IdentityServer3) beskriver hvordan gruppen planla å designe og implementere ulik funksjonalitet i mellomvaren. Disse to kapitlene er ikke direkte relatert til problemstillingen,

men er tatt med for å vise at prosjektgruppen jobbet med et annet fokus før oppgaven endret seg. Les mer om endringen i delkapittelet endring av oppgaven (jfr 1.3).

1.9 Terminologibruk

Beskrivelser og betegnelser som brukes synonymt i rapporten.

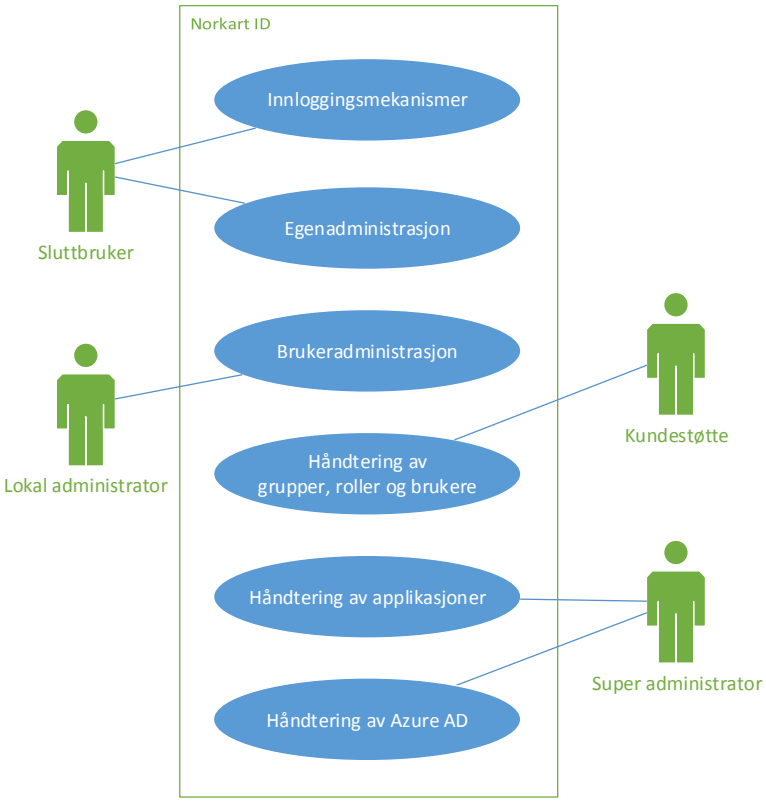
- **AAD portalen:** Administrering for AAD i Azure portalen.
- **AAD Graph API, Graph API** API protokoll for AAD.
- **Bruker, sluttbruker:** Betegnelse for en bruker av systemet Norkart ID.
- **Brukerportal:** En portal tilhørende Norkart ID for egen- og brukeradministrasjon.
- **Claim, claims:** Et utsagn om at bruker faktisk er bruker.
- **Datasystem:** Et system som behandler, lagrer eller overfører data.
- **Identitetstilbyder:** Et system som tilbyr bruker å lagre identiteter.
- **Maskinressurs:** Fil, tilgang eller tjeneste i et datasystem.
- **Norkart:** Oppdragsgiver for prosjektet.
- **Norkart ID, Autentiseringsløsning, løsningen:** En portal, et system eller server som lar brukere logge seg på med brukernavn og passord for å få tilgang til en eller flere applikasjoner.
- **Nøkkel, passord:** En autentiseringshemmlighet.
- **Produkteier, Veileder, Kontaktpersoner hos Norkart:** Produkteier er ansvarlig for at løsningen inneholder den funksjonaliteten som viktigst for Norkart. Veileder er gruppens faglig rådgiver. Norkart kontaktpersoner er ressurspersoner som er knyttet til prosjektet og som har deltatt på demo- og gjennomgangsmøter.
- **Prosjektgruppen, gruppen:** Studentene som jobber med prosjektet.
- **Tjenestetilbyder:** En applikasjon.
- **Token, tokens:** Et objekt som åpner for utføring av en operasjon.
- **Veiledninger, brukerveiledninger:** Oppskrift på hvordan noe skal gjøres.

2 Kravspesifikasjon

Dette kapitlet inneholder funksjonelle og operasjonelle krav for Norkart ID.

2.1 Funksjonelle krav til løsning

Oppgavens funksjonelle krav har til hensikt å forklare hvilke funksjonalitet og roller løsningen skal tilby.



Figur 2: Use case for Norkart ID

2.1.1 Use case

For å beskrive hva Norkart ønsket i løsningen ble det laget et use case (jfr figur 2). Use caset viser rollene og deres knytning til Norkart IDs hovedfunksjonalitet.

Sluttbruker

Sluttbruker benytter seg av Norkart IDs innloggingsmekanismer og har tilgang til egenadministrasjon i brukerportalen.

Lokal administrator

Denne rollen har til hensikt å administrere sluttbrukere innenfor en gitt gruppe. Lokal administrator vil jobbe i brukerportalen til Norkart ID. Personer som har denne rollen kan enten jobbe hos Norkart eller i en bedrift som benytter seg av Norkart ID. Rollen har til hensikt å redusere press på Norkart kundestøtte.

Kundestøtte

Dette er en rolle ment for Norkart kundestøtte som skal håndtere brukere, grupper og roller i AAD portalen.

Super administrator

Denne rollen er hovedsakelig ment for utviklere og driftere i Norkart og skal håndtere applikasjoner, i tillegg til generell håndtering av AAD. Rollen er øverste nivå og skal derfor ha mulighet til å bruke alle tjenestene levert i AAD samt all funksjonalitet i Norkart ID.

2.1.2 Høynivå use case beskrivelser

For å skape forståelse rundt hovedfunksjonalitet i løsningen er det utviklet seks høynivå use case beskrivelser (jfr tabell 1 til 6).

Use case:	Innloggingsmekanismer
Aktører:	Sluttbruker
Mål:	Logge seg inn og få sikker tilgang til ønsket applikasjon. Dette skal være SSO slik at aktør i tillegg blir autentisert for andre applikasjoner. Aktør skal også kunne logge seg ut av en applikasjon og resette sitt eget passord.
Beskrivelse:	For å skape oversikt deles innloggingsmekanismer opp i tre underkategorier.
Logg inn	Når aktør skriver inn brukernavn og passord skal aktøren autentiseres og få tilgang. Da skal aktøren også bli autentisert for andre applikasjoner uten å måtte oppgi brukernavn og passord.
Logg ut	Når aktør klikker logg ut skal aktøren bli utlogget fra applikasjonen.
Glemt passord	Om aktør glemmer sitt passord skal det være mulig å gjenopprette dette selv. Aktøren skal gjøre dette ved å trykke på en glemt passord lenke på innloggingsiden. Det sendes så en verifiseringkode, enten via e-post eller telefon. Ved å oppgi denne koden skal aktøren kunne sette nytt passord.

Tabell 1: Use case - Innloggingsmekanismer

Use case:	Egenadministrasjon
Aktører:	Sluttbruker
Mål:	Kunne registrere og endre egne opplysninger i brukerportalen.
Beskrivelse:	Aktør skal kunne registrere og endre sitt telefonnummer, mobilnummer og adresse. I tillegg skal aktør kunne registrere og endre eksternt mobilnummer og e-post til bruk av glemt passord funksjonalitet. Aktør skal også kunne endre sitt passord ved å oppgi sitt gamle passord. Egenadministrasjon skal foregå i Norkart IDs brukerportal.

Tabell 2: Use case - Egenadministrasjon

Use case:	Brukeradministrasjon
Aktører:	Lokal administrator
Mål:	Håndtere brukere innenfor en gitt gruppe i brukerprotalen.
Beskrivelse:	Aktør skal kunne opprette og fjerne brukere, samt knytte disse til og fra en gruppe. Det skal i tillegg kunne resettes passord og endres brukerdata for medlemmene i gruppen. Brukeradministrasjon skal skje i Norkart IDs brukerportal.

Tabell 3: Use case - Brukeradministrasjon

Use case:	Håndtering av brukere og grupper.
Aktører:	Kundestøtte
Mål:	Registrere og håndtere brukere og grupper i AAD portalen.
Beskrivelse:	Funksjonalitet for brukere og grupper beskrives hver for seg.
Brukere	Aktør skal kunne registrere og slette brukere i AAD, samt endre brukerdata, passord og roller for disse. Dette skal gjøres i AAD portalen.
Grupper	Aktør skal kunne opprette, fjerne og redigere grupper i AAD portalen, samt legge til og fjerne brukere i grupper. Aktør kan i tillegg gi grupper tilgang til applikasjoner.

Tabell 4: Use case - Håndtering av brukerer og grupper

Use case:	Håndtering av applikasjoner
Aktører:	Super Administrator
Mål:	Registrere og administrere applikasjoner i AAD Portalen.
Beskrivelse:	I AAD portalen skal aktør kunne registrere web og native applikasjoner for bruk av autentiseringløsningen. Aktør skal også kunne konfigurere applikasjonsdata og applikasjonenens tilganger til andre applikasjoner.

Tabell 5: Use case - Håndtering av applikasjoner

Use case:	Håndtering av AAD
Aktører:	Super Administrator
Mål:	Aktør skal kunne konfigurere og administrere AAD.
Beskrivelse:	Aktør skal kunne konfigurere passord policy, lisens- og domeneoppsett for AAD. I tillegg skal aktør kunne sette opp profilering av Norkart, administrere roller og sette standardregler for grupper. Aktør skal kunne masseregistrere brukere fra en ekstern kilde.

Tabell 6: Use case - Håndtering av AAD

2.1.3 Roller og tilganger

I dette delkapittelet har prosjektgruppen utformet en oversikt over hvilke roller som har tilgang til hvilke funksjonalitet. Figur 7 viser hvilke av Norkart IDs systemer rollene har tilgang til. Figur 8 til 10 viser tilganger i brukerportalen og figur 11 til 13 viser tilganger i AAD portalen.

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Tilgang til brukerportalen for Norkart ID	JA	JA	JA	JA
Tilgang til AAD Portalen for Norkart ID	-	-	JA	JA

Tabell 7: Rolleoversikt - Systemer

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Logg inn	JA	JA	JA	JA
Logg ut	JA	JA	JA	JA
Glemt passord	JA	JA	JA	JA

Tabell 8: Rolleoversikt - Innloggingsmekanismer

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Registrere brukerdata	JA	JA	JA	JA
Endre egen brukerdata	JA	JA	JA	JA
Endre egen autentiseringsdata	JA	JA	JA	JA
Endre eget passord	JA	JA	JA	JA

Tabell 9: Rolleoversikt - Egenadministrasjon i brukerportalen

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Opprette og slette brukere	-	JA	JA	JA
Resetting av passord for brukere i en gruppe	-	JA	JA	JA
Legg til og fjern brukere i en gruppe	-	JA	JA	JA
Endre brukerdata for brukere i en gruppe	-	JA	JA	JA

Tabell 10: Rolleoversikt - Brukeradministrasjon i brukerportalen

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Opprette og slette brukere	-	-	JA	JA
Registrere og endre brukerdata for brukere	-	-	JA	JA
Endre passord for brukere	-	-	JA	JA
Tildele roller til brukere	-	-	JA	JA
Opprette og slette grupper	-	-	JA	JA
Legge til og fjerne brukere fra grupper	-	-	JA	JA
Redigere eier av en gruppe	-	-	JA	JA
Tildele tilgang til applikasjoner for en gruppe	-	-	JA	JA

Tabell 11: Rolleoversikt - Håndtering av brukere, grupper og roller i AAD portalen

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Registrere web og native applikasjoner i AAD	-	-	-	JA
Konfigurere registrerte applikasjoner	-	-	-	JA
Administrere tilganger til andre applikasjoner	-	-	-	JA
Slette applikasjoner fra AAD	-	-	-	JA

Tabell 12: Rolleoversikt - Håndtering av applikasjoner i AAD portalen

	Sluttbruker	Lokaladmin	Kundestøtte	Superadmin
Sette opp profilering av Norkart	-	-	-	JA
Definere autentiseringspolicy	-	-	-	JA
Definere standardregler for grupper	-	-	-	JA
Administrere lisenser	-	-	-	JA
Konfigurere domene	-	-	-	JA
Opprette nye roller	-	-	-	JA
Konfigurere eksisterende roller	-	-	-	JA
Slette roller	-	-	-	JA
Masseregistrering av brukere fra eksterne kilder	-	-	-	JA

Tabell 13: Rolleoversikt - Håndtering av AAD i Azure portalen

2.2 Operasjonelle krav til løsning

Dette delkapittelet beskriver hvilke krav som er satt til hvordan systemet opererer og hvilken spesifikasjoner det skal ha.

2.2.1 Ytelse

- Løsningen skal som minimum takle 10 000 brukere innlogget samtidig.
- Løsningen skal håndtere pålogging av 100 brukere i minuttet.
- Løsningen skal bygges for å være skalerbar.
- Verifiseringskode for resetting av passord skal mottas innen 20 sekunder for både e-post og telefon.
- Etter innloggingsknappen er trykt skal det ta mindre enn 2 sekunder før brukeren ser progresjon.

2.2.2 Brukervennlighet

- Løsningen skal følge regler for universell utforming (jfr delkapittel 2.2.4 Lover og regler)
- Bruker skal kunne benytte samme påloggинginformasjon mot alle systemene til Norkart.
- Brukergrensesnittet (GUI) på innloggingsiden skal være så intuitivt at det tar mindre enn 2 sekunder å skjønne hvor brukerid felt, passord felt og innloggingknappen er plassert.
- Bruker skal selv forstå at utlogging er utført.

- Når bruker oppretter eller endrer sitt passord skal det forstås om passordkravet oppfylles (jfr delkapittel 2.2.3).
- Egenadministrasjon skal være oversiktlig og bruker skal selv skjønne hvordan brukerdata kan endres.
- Brukeradministrasjon skal være så intuitivt at det kan benyttes etter en kort innføring.
- GUI for kundestøtte og lokal administrator skal være oversiktlig nok til at denne kan tas i bruk etter en innføring.
- GUI skal skape gjenkjennelighet til Norkart for sluttbrukere.
- GUI for innloggingsiden, egenadministrasjon og brukeradministrasjon skal være like intuitivt på alle skjermstørrelser.

2.2.3 Sikkerhet og autentiseringskrav

- Det skal kunne registreres følgende brukerdata: Fullt navn, e-post, passord, mobiltelefonnummer.
- Ingen passord skal sendes eller lagres i klartekst.
- Minimumskrav for passordlengde er åtte tegn.
- Minimumskrav for passordkompleksitet er en stor bokstav, en liten bokstav og et tall.
- Ved feil passord eller brukernavn skal det stå at innlogging feilet.
- En bruker som logger inn via en nettleser autentiseres for alle web applikasjoner brukeren har tilgang til.
- En bruker som logger inn i en mobil applikasjon autentiseres kun for denne.

2.2.4 Lover og regler

- Løsningen skal følge diskriminerings- og tilgjengelighetsloven, §13 og §14 om universell utforming.
- Løsningen utformes i samsvar med standard Web Content Accessibility Guidelines 2.0 (WCAG 2.0) og skal dermed følge norsk forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger, § 4 Krav til utforming av IKT-løsninger.

3 Teoridel

Dette kapittelet er et oppslagsverk for leserne av rapporten. Oppdragsgiver ytret ønske om en oppsummert innføring i ulike teknologier og begreper som omtales i rapporten. Kapittelet er skrevet med egne ord for å framheve det som vurderes som relevant.

3.1 Autentisering

I denne rapporten er autentisering knyttet mellom menneske-til-maskin og maskin-til-maskin. Definisjon på autentisering:

> Å bevise at man er den man utgir seg for å være. Autentisering skal bekrefte en påstått identitet. Dette kan skje gjennom noe du vet (passord), noe du er (fingeravtrykk/ biometri) eller noe du har (nøkkelkort). Kombinasjoner av disse er også mye brukt. Den som autentiseres kan være en person som bruker en datamaskin, kun en datamaskin eller et program.

Kilde: Norsk senter for informasjonssikring (NorSIS) [2].

Prinsipp

Menneske-til-maskin autentisering foregår mellom et menneske og en maskinressurs. Som bruker av en maskinressurs må det bekreftes for maskinen at man er den man utgir seg for å være. For å gjøre dette opprettes det brukerkontoer. En bruker kan dermed oppgi brukernavn og passord for å autentiseres mot en maskinressurs. Om maskinen gjenkjenner brukernavn og passord vil tilgang tillates. En maskinressurs kan eksempelvis være en smarttelefon, en webapplikasjon eller et operativsystem. I teorien kan det autentiseres for å få tilgang til hva som helst av beskyttet innhold.

Maskin-til-maskin autentisering gjøres på nesten samme måte som menneske-til-maskin, men det brukes nøkler i steden for passord. Det er ulike former for metoder og teknologier som kan brukes for å beskytte mot misbruk av nøklene. Dette løses av rammeverk og protokoller brukt i autentiseringssystemer.

Utfordringer

Gitt at brukernavnet skal være enkelt for en bruker å huske, velges det ofte et brukernavn som er relativt logisk bygd opp i forhold til eget navn, stilling eller rolle. Dette gjør det lettere for andre å gjette brukernavnet. Dersom e-post adressen brukes som brukernavn er det kun passordet som fungerer som nøkkel. Dette stiller krav til nøkkelen.

Det som brukes for å bekrefte brukerens identitet bør være enkelt å huske, men vanskelig for andre å gjette eller simulere. Trustwave sin Global Security Report fra 2014 [3], påstår at en tredjedel av alle saker knyttet til uautorisert tilgang var som følge av et svakt passord. En god autentiseringsmekanisme bør derfor kreve at brukeren lager et sterkt passord og eventuelt må autentiseres med andre metoder enn bare passord. Eksempel på dette kan være engangskoder på SMS, fingeravtrykk eller ansiktsgjenkjenning.

Det er mulig å designe et system som hjelper brukere å velge sterkere nøkler. I tillegg kan autentiseringssystemet designes til å kreve fler-faktor autentisering for å bekrefte en identitet. Dette kan for eksempel være passord i tillegg til engangskode på mail eller sms. Fler-faktor kan brukes på hvert autentiseringsforsøk, eller bare ved tilfeldig utvalgte autentiseringer.

3.2 Autorisasjon

Definisjon på autorisering:

> Autorisering er prosessen med å beslutte å gi en person, en datamaskin eller et program tillatelse til å bruke bestemte IT-ressurser. Eksempler på en IT-ressurs kan være filer, nettverksstasjoner og prosesser.

Kilde: Norsk senter for informasjonssikring (NorSIS) [4].

Prinsipper

Autorisasjon i forhold til datasystemer dreier seg om å avgjøre hva en bruker skal få tilgang til eller ikke. En autorisasjon skjer etter at en bruker er vellykket autentisert (jfr 3.1 Autentisering). Autorisasjonsprosessen kan deles inn i to faser. I første fase, definisjonsfasen, defineres tilgangen til en bruker. I fase to, godkjenningsfasen, godkjennes eller nektes tilgang til ressurser basert på første fase. Alle tilganger ligger i en form for brukerdatabase. Denne kan være implementert på svært ulike måter avhengig av system og bruk.

Datasystemer bruker autorisasjon for å skille brukere fra hverandre, slik at brukere og brukergrupper kun har tilgang til ressurser de skal ha. Autorisasjon kan eksempelvis være at bruker har mulighet til å se egne private filer, mens andre brukere ikke har tilgang til disse.

Utfordringer

I definisjonsfasen av en autorisasjon kan det være utfordrende å ha kontroll over tilganger for ulike grupper og medlemmer. Et godt autoriseringsystem gir spesielt utvalgte brukere i ulike grupper, ofte kalt gruppeeiere, tilgang til å fjerne eller legge til rettigheter for andre brukere. Om brukeradministrasjonssystemet gir mulighet for slike gruppeeiere, bør systemet designes for å hindre uønskede brukerfeil.

3.3 SSO

Dette er et begrep for å beskrive at en innloggingsøkt, som allerede er vellykket, kan benyttes for å få tilgang til andre applikasjoner [5]. Typisk blir dette håndtert av Lightweight Directory Access Protocol (LDAP) [6] og en database med brukere.

To eksempler på selskaper som tilbyr SSO mot andre tjenester, er Facebook [7] og Google [8]. Disse gir mulighet for å benytte brukerkontoer lagret hos dem til autentisering mot andre applikasjoner. Ved bruk av SSO reduseres antall ulike kontoer en bruker må forholde seg til.

3.4 Security Assertion Markup Language (SAML)

Dette er en åpen XML basert autentiserings protokoll. SAML støtter SSO (jfr 3.3 SSO) og fungerer ved at den definerer tre roller:

- Bruker
- Identitetstilbyder
- Tjenestetilbyder

En beskrivelse av autentiseringsflyten i SAML protokollen kan leses i konfigureringskapitlet (jfr delkapittel 5.1.1 Hva skjer ved innlogging). Kommunikasjonmetoden SAML bruker er XML kombinert med andre kommunikasjonprotokoller som HTTP og SOAP [9]. SAML ble definert av OASIS Security Services Technical Committee i januar 2001. Protokollen har blitt revidert flere ganger og endt opp i SAML versjon 2.0 som en standard fra 2005 [10].

SAML er fortsatt en aktuell autentiseringprotokoll for en rekke tjenester. Våren 2015 ble det lagt inn støtte for SAML i nye Office 365 fra Microsoft [11].

3.5 OAuth 2.0 & OAuth 1.0

OAuth 2.0 er en åpen protokoll for autentisering og autorisering og er basert på bruk av tokens med ulik funksjonalitet. OAuth 2.0 kan brukes som en SSO protokoll. Den bruker ”Refresh tokens”, ”Access tokens” og ”Access code” for å oppnå en sikker knytning mellom identitetstilbydere, autentiserte brukere og applikasjoner. I konfigurasjonskapittelet (jfr delkapittel 5.1.1 Hva skjer ved innlogging) beskrives autentiseringsflyten og tokens nærmere.

OAuth 1.0 ble utviklet da OpenID skulle implementeres mot Twitter i 2007 [12]. I OAuth 1.0 blir alt kodet og dekodet med en rekke signaturer som førte til behandlingstid for hver forespørsler. OAuth 2.0 ble lansert i 2012 og gjør det samme som sin forgjenger uten signaturer grunnet bruk av Secure Sockets Layer (SSL). Dette resulterer i høyere ytelse. [13]

Protokollen er basert på REST og JSON og åpner for enkel bruk av i datamaskiner, mobile enheter, smartklokker og annet utstyr. OAuth 2.0 har lav overhead i forhold til SAML.

3.6 OIDC

OIDC er en åpen identitetsprotokoll som ligger på toppen av OAuth 2.0. Den muliggjør verifisering av identiteten til en sluttbruker ved hjelp av en id token. OIDC kan bekrefte identiteten på brukeren som er autentisert med svært lav overhead i forhold til andre autentiseringssystemer [14]. OIDC er utviklet med mål om å ”making simple things simple and complicated things possible” [15]. Utviklere kan bruke OIDC for å autentisere sine brukere på tvers av nettsider og applikasjoner.

3.7 WS-Federation

Dette er en autorisasjonsprotokoll utviklet av blant annet IBM og Microsoft. Protokollen kan benyttes for å oppnå SSO på samme måte som OAuth 2.0 og SAML. WS-Federation kan brukes mellom en identitetstilbyder og en applikasjon, eller mot flere tjenester i et

større nettverk. Den kommuniserer over JSON og håndterer sikker transport av autentiseringstokens. Dette brukes for autorisering av brukere fra en identitetstilbyder til en applikasjon. Protokollen håndterer ikke autentisering. Den overfører tilgangene i form av innkapslede tokens, eller claims, i egne JSON pakker. Et eksempel på dette er at WS-Federation kan overføre en SAML claim i en WS-Federation pakke for å bekrefte identiteten til en bruker. Dette betyr derimot ikke at SAML og WS-Federation kan snakke direkte med hverandre, men WS-Federation overfører SAML autentisering, for å kunne autorisere brukere.

3.8 Microsoft Active Directory

Microsoft Active Directory (AD) er en katalogtjeneste fra Microsoft. Den brukes for å tildele ressurser til brukere og brukergrupper både eksternt og internt i en bedrift. AD kan sees på som en spesialdesignet database som er optimalisert for små lese og søke operasjoner. Katalogtjenesten er bygd opp av objekter og disse representerer systemer, ressurser eller tjenester [16].

3.9 Microsoft Azure

Dette er skyplattformen til Microsoft. Deler plattformen og funksjonaliteten kan også bygges på egne servere. Azure levere tjenester som enten er Infrastructure-as-a-Service(IaaS) eller Platform-as-a-Service(PaaS). Dette muliggjør både managed og unmanaged tjenester. En managed tjeneste betyr at den kan settes opp fra bunnen og bedriften har ansvar i alle ledd. Eksempler på dette er konfigurasjon av operativsystemer, programvare, oppfølging av oppdateringer og løpende vedlikehold. En unmanaged tjeneste betyr at tjenesten kan brukes direkte i Azure portalen som en abstrakt entitet. Microsoft tar seg av vedlikehold, overvåking, oppetid og oppdateringer for disse.

Azure ble lansert i 2010 og har siden hatt stødig vekst [17]. Azure kan brukes for å løse utfordringer rundt infrastruktur ved å tilby tjenester på en skalerbar måte. Når data er lagret i skyen vil det tidvis være uklart hvilke lovverk som gjelder. Microsoft etterstreber å møte nasjonale og internasjonale standarder [18].

3.10 AAD

AAD er skyversjonen av Microsoft AD og er designet for å dekke identitet- og tilgangshåndtering for bedrifter. Den gir administratorer en mulighet til å bruke en egen privat katalogtjeneste for håndtering av tilganger både mot eksterne og lokale ressurser. Eksempeler på funksjonalitet i AAD er integrerte påloggingprotokoller for SSO, dynamiske grupper, avanserte sikkerhetsmekanismer og detaljerte sikkerhetsrapporter [19]. Det er mulig å knytte egenutviklede applikasjoner til AAD, i tillegg til de forhåndsregistrerte. AAD lisensieres på ulike måter, les mer om dette i konfigurasjonskapittelet (jfr delkapittel 5.8.2 Lisensmodell).

AAD kobles opp mot et sett av ulike tjenester med ferdig konfigurerte endpoints eller API'er for tilknytning. Dette gjelder blant annet OAuth 2.0, OIDC, SAML og WS-Federation. Ved å bruke AAD har man tilgang til sikkerhet og beskyttelsesmekanismer fra Microsoft.

3.11 AAD Graph API

Dette er en API protokoll som lar autoriserte brukere og applikasjoner gjøre operasjoner på objekter i AAD. Objektene er brukere, grupper og applikasjoner som er knyttet til hverandre, disse kan også kalles noder. Graph API tillater autoriserte brukere å gjøre spørringer mot AAD for å oppdatere eller navigere mellom objektene. Protokollen kommuniserer ved bruk av REST og JSON meldinger. Denne funksjonaliteten bidrar til å skape et stort skille mellom hva en tradisjonell AD og AAD kan gjøre. Dette gjelder for sluttbrukere, utviklere og administratorer. Graph API gjør det mulig å lage webapplikasjoner som kan bruke objektene i AAD uten ekstra mellomvare.

3.12 AAD Application Proxy

Dette er en tjeneste i AAD som gjør det mulig å eksponere applikasjoner i et lukket nettverk for brukere tilknyttet internett på en sikker måte. Dette betyr at brukere utenfor det lukkede nettverket kan koble seg på nett-tjenester uten bruk av andre sikkerhetsmekanismer.

For å bruke AAD Application Proxy må det installeres en Application Proxy Connector applikasjon på en maskin i det lukkede nettverket som oppretter knytningen mot AAD. I Azure vil dette dukke opp som en applikasjon med både ekstern og intern URL konfigurert, samt autentiseringalternativer. Når en bruker åpner den eksterne URL'en blir spørringen videresendt til Connector applikasjonen som sender spørringen videre til applikasjonsserveren i det lukkede nettverket. Både AAD og sluttbrukeren vil oppfatte det som at applikasjonen ligger i Azure.

3.13 MyApps

MyApps, også kalt Microsoft App Access Panel, er en ferdigutviklet brukerportal fra Microsoft. Portalen er webbasert og gir sluttbrukere oversikt over applikasjoner. Dette er applikasjoner som brukerne har fått tilgang til fra AAD. Tjenesten er laget for at brukerne skal kunne være selvadministrerende angående passord resett og gruppehåndtering. Tjenesten er separert fra Azure portalen og krever ikke at bruker har et Azure abonnement.

3.14 IdentityServer3

Dette er et open source .NET basert rammeverk for implementering av OIDC. Rammeverket må settes opp med tilknytning til brukerdatabaser og deler av GUI'en. Det muliggjør implementering av SSO og tilgangkontroll for web applikasjoner og API'er. Rammeverket kan brukes på ulike klienter deriblant mobil, web og skrivebordsapplikasjoner.

Det tyske selskapet Thinktecture, med Dominick Baier i spissen, driver prosjektet. IdentityServer3 1.0 ble tilgjengelig i Februar 2015, mens IdentityServer3 2.0 beta har vært tilgjengelig siden April 2015 [20]. Rammeverket blir testet i betautgivelser før det slippes offisielle nye versjoner. For en minimumsløsning uten tilknytning til brukerbibliotek må det følges en lengre veiledning for å sette opp serveren i Visual Studio [21].

3.15 OWIN og Katana

OWIN er en standard for .NET applikasjoner og er et open source prosjekt. Standarden spesifiserer hvordan kommunikasjon mellom klient og server gjøres. Hensikten med prosjektet er å skille ut komponenter Microsoft tidligere hadde bygget inn i .NET kjernen, for å gjøre dem tilgjengelige for alle webservere. Tidligere var det kun mulig å kjøre .NET på IIS webservere. Dette gir utviklere mulighet til å bygge inn de modulene de trenger. Eksempel på dette kan være et skript som kjører i en nettleser som kommuniserer med webserver både automatisk og når brukeren ber om det.

Katana er prosjektnavnet på Microsoft sitt eget open source prosjekt for implementasjon av OWIN standarder.

3.16 Single og multi tenant applikasjon

Single og multi tennant er to ulike måter å gjøre en applikasjon tilgjengelig for brukere tilhørende domener eller identitetstilbydere.

Single tenant applikasjon

En single tenant applikasjon betyr at den kun kan brukes av en bedrift eller i et domene. For at flere bedrifter skal kunne bruke applikasjonen må det lages en kopi av applikasjonen og databasen. Dette gir tilbyder av programvaren enklere mulighet til å tilpasse hver enkelt leveranse og sikrer adskillelse av data. Dersom en applikasjon skulle feile under en oppdatering eller ved vedlikehold, påvirker det kun de som er knyttet til denne. Ulemper vil være at det vil ta tid å vedlikeholde koden for mange like applikasjoner parallelt, spesielt dersom det er gjort store tilpasninger. I tillegg til dette blir det flere databaser, applikasjoner og potensielt servere å vedlikeholde. Det er ikke gitt at tilbyder av en applikasjon drifter alle instanser av applikasjonen for brukerne.

Multi tenant applikasjon

En multi tenant applikasjon betyr at den ligger et sted, men er satt opp mot flere bedrifter hvor dataen oppleves som adskilt fra hverandre. Dette gjøres ved at data blir merket, noe som gjør det mulig å skille hvilke data som tilhører hvilke bedrifter. Vedlikehold og drift blir annerledes å forholde seg til på grunn av dette. Om en applikasjon er konfigurert med multi tenant oppsett kan den oppnå single point of failure. Dette er vil si at om systemet skulle feile ville dette gått ut over hele systemet. Et realistisk scenario kan være at applikasjonen blir spredd ut over flere servere. Dette gjøres for å redusere faren for at single point of failure skal inntreffe og i tillegg øke ytelse og sikkerhet.

4 Valg av løsning

Prosjektgruppen valgte å fokusere på to mulige løsninger for å løse problemstillingen. Disse var AAD (jfr delkapittel 3.10) og IdentityServer3 (jfr delkapittel 3.14). I dette kapittelet identifiseres det styrker og svakheter i disse og argumenteres for valgt løsning. IdentityServer3 er ikke en ferdig løsning men designet som ble utarbeidet i kapittel om arkitektur og design av IdentityServer3 (jfr kapittel 9) brukes når den sammenlignes med AAD. Gruppen satt seg inn i IdentityServer3 ved å gjennomføre en veiledning fra hjemmesiden deres [22].

4.1 IdentityServer3

Nedenfor er det en punktliste over hva prosjektgruppen har identifisert som styrker og svakheter basert på veiledning.

Styrker

- Utviklet med OIDC i fokus
- Gir mulighet for egen passordpolicy
- Kan bygges helt etter design og spesifikasjonsønsker fra Norkart
- Fribrukslisens

Svakheter

- Må bruke ressurser på implementasjon og utvikling
- Usikker fremtid da prosjektet er open source
- Må driftes og vedlikeholdes av Norkart
- Støtter kun et begrenset antall autentiseringsprotokoller.

4.2 AAD

Nedenfor er det en punktliste over hva prosjektgruppen har identifisert som styrker og svakheter i AAD.

Styrker

- Krever ingen oppsett eller installasjon før bruk
- Pris på tjeneste skalerer etter bruk av ressurser
- Driftes og vedlikeholdes av Microsoft
- Innebygd ”glemt passord” funksjonalitet
- Støtte for flere autentiseringsprotokoller
- Brukerportal for egenadministrering

Svakheter

- Applikasjoner må knyttes til Azure for å kunne brukes i AAD.
- Kan ikke endre passordkompleksitet.
- Brukerdatabasen må ligge skyen.

4.3 Drøfting av løsninger

Ved å ta i bruk IdentityServer3 vil Norkart kunne implementere løsningen etter egne krav. Dette gir en annen oversikt enn ved bruk av AAD og potensialet vil kun begrenses av behov. Alle applikasjoner som har støtte for OIDC kan tilknyttes og valg av database er fritt.

Norkart er knyttet til Microsoft og applikasjonene som leveres i dag er levert på Microsoft systemer. Knytning av applikasjoner til AAD kan dermed antas som realistisk å gjøre uten store utfordringer. AAD er allerede mulig å ta i bruk og det trengs mindre utvikling for å møte krav satt i kravspesifikasjonen (jfr kapittel 2)

Både IdentityServer3 og AAD er nytt, og har begrenset oppdatert støttedokumentasjon på Internett. Norkart er allerede Gold Partner [23] med Microsoft. Det resulterer i et tett samarbeid hvor de vil få tilgang til støtten de ønsker. AAD fremstår som en stor modul mens IdentityServer3 hadde bestått av minst 3 moduler (jfr kapittel 9 Arkitektur og design av IdentityServer3).

Det er en vurderingssak om løsningen skal settes bort til Microsoft for å redusere implementasjonskostnader eller om det skal brukes tid og ressurser på å utvikle store deler av løsningen selv. Oppdragsgiver har uttalt at de er forberedt på å betale for en autentiseringsløsning.

4.3.1 Driftsaspekter og vedlikehold

Norkart har allerede flere applikasjoner de drifter idag. Drift av IdentityServer3 vil trolig inngå som en del av de etablerte driftsrutinene Norkart allerede har. Begge løsningene er avhengig av vedlikehold av innholdet i brukerdatabasen. Denne type hendelser må Norkart håndtere uansett hvilket valg de tar. Drifter de løsningen selv er de ansvarlig for oppetid og alle hendelser på egenhånd. Dersom IdentityServer3 viser seg å være stabil, og inneholde lite feil, vil denne kunne kjøre over lengre tid uten behov for store grep. Dette kan medføre at vedlikehold skjer sjeldnere enn ved AAD hvor det skjer fortløpende.

4.4 Valg

Norkart har allerede tette bånd til Microsoft og i begynnelsen av mars (jfr vedlegg E), ytret oppdragsgiver et ønsket om at prosjektgruppen skulle undersøke AAD dypere. Grunnlaget for dette var tilgangen til kundestøtte, mengden funksjonalitet og muligheter som allerede var i tjenesten. I tillegg mente oppdragsgiver at de ville få mer ut av prosjektet dersom gruppen fokuserte på AAD (jfr vedlegg A.1.2 Gjennomgangsmøte).

4.4.1 Utfordringer med valgt løsning

Norkart vil med AAD få en totalløsning med begrensninger. Det vil si at Norkart er begrenset til de tilpasningene som er tilgjengelig i AAD. Det er kun datasentere i Nederland og Irland som tilbyr AAD i Europa. Dette betyr at selve brukerdatabasen for applikasjonene Norkart knytter til sin AAD vil ligge på en server i en av disse landene. Norkart bør undersøke den juridiske problemstillingen angående persondata i Azure. Prosjektgruppen valgte å ikke gå i dybden på dette etter oppfordring fra oppdragsgiver(jfr vedlegg A.3.2).

5 Konfigurasjon

Dette kapittelet viser hvordan Norkart kan bruke AAD som en autentisering- og autoriseringløsning i henhold til kravspesifikasjonen (jfr kapittel 2), samt hvilke hensyn som må tas. Det skal også informere om hvordan løsningen fungerer, og brukes.

Prosjektgruppen har valgt å fokusere på to hovedeksempler i forhold til implementering og konfigurasjon. Første eksempel er en web applikasjon (jfr delkapittel 5.5.3), det andre eksempelet er en Android applikasjon (jfr delkapittel 5.5.4). Begge eksempelapplikasjonene beskriver minimumskrav for å oppnå autentisering og tilgang til beskyttet innhold.

5.1 Innloggingsmekanismer

Dette delkapittelet skal beskrive hvordan AAD løser use case for Innloggingsmekanismer, beskrevet i kravspesifikasjonen, (jfr 2.1.1 Use case).

5.1.1 Logg inn

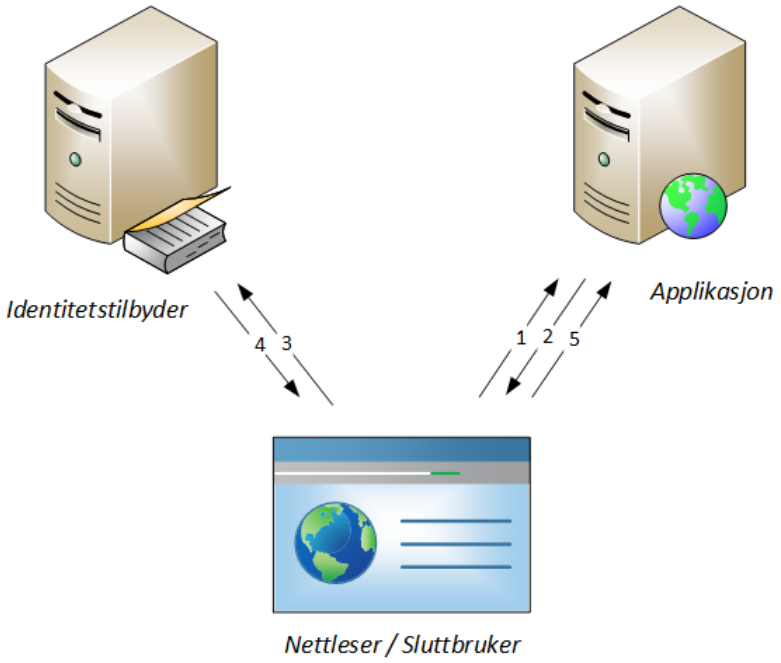
Logg inn funksjonalitet tillatter brukere å benytte autentiseringsløsnigen mot applikasjoner med beskyttet innhold.

Hva skjer ved innlogging

Protokollene som kan brukes for autentisering mot AAD er SAML, (jfr 3.4), OAuth 2.0 (jfr 3.5), OIDC (jfr 3.6) og WS-Federation (jfr 3.7). OIDC protokollen bygger på OAuth 2.0 og vil omtales som det samme videre i kapittelet.

SAML

Autentiseringsflyten i SAML er overordnet beskrevet i listen nedenfor. Pilene i figur 3 korresponderer med punktene i listen.



Figur 3: Autentiseringsflyt for SAML protokoll

1. Sluttbruker spør en applikasjon om tilgang til beskyttede ressurser.
2. Applikasjonen håndterer ikke autentisering selv så sluttbruker videresendes til en identitetstilbyder (AAD).
3. Dette skjer automatisk og oppfattes som en videresending til en ny innloggingsside hos identitetstilbyder. Nettleser sender automatisk med en RelayState adresse til identitetstilbyder. Denne adressen hjelper identitetstilbyder å finne hvilken applikasjon sluttbruker ønsker tilgang til. Sluttbruker må oppgi autentiseringsdata før dette sendes tilbake til identitetstilbyder. Denne kommunikasjonen er ikke oppgitt i figur 3 men skjer mellom pil 3 og 4.
4. Ved vellykket innlogging hos identitetstilbyder opprettes det en identifikasjonskode i form av en signatur. Denne signaturen sendes videre til applikasjonen og gjør at applikasjonen kan godkjenne sluttbruker som bruker av systemet.
5. Signaturen sendes til applikasjonens RelayState adresse, og sjekker om signaturen er gyldig. Om applikasjonen godkjenner signaturen vil innloggingen fullføres og brukeren vil få tilgang til de beskyttede ressursene.

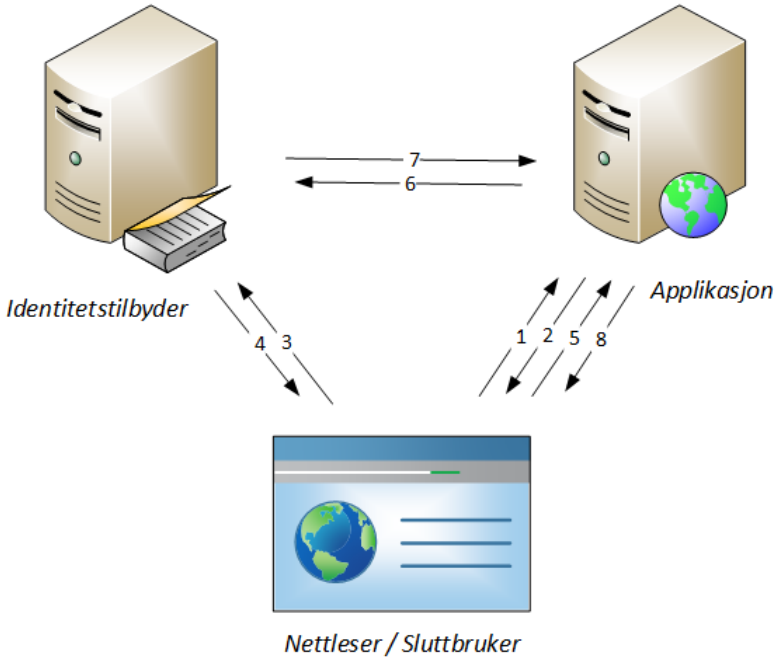
OIDC

OIDC består av tre typer tokens:

- Access token gir tilgang til en spesifikk applikasjon og har en gitt levetid.
- Refresh token brukes for å få nye Access tokens uten å måtte oppgi innloggingsdetaljer på nytt. Tokenet gis ut av identitetstilbyder og kan ha evig varighet. Dette betyr at så lenge bruker ikke trekker tilbake tilgangen til en applikasjon, eller logger ut, vil sluttbruker være innlogget i applikasjonen. Refresh token kan mottas samtidig med Access tokens eller spørres etter separat.

- Identity token er en samling av personlige data en bruker kan etterspørre fra identitetstilbyder. Tokenet er designet slik at data kan etterprøves.

I listen nedenfor blir bruken av disse forklart og autentiseringsflyten beskrevet. Pilene i figur 4 korresponderer med punktene i listen.



Figur 4: Autentiseringsflyt for OAuth 2.0 protokoll

1. Sluttbruker spør en applikasjon om tilgang til beskyttede ressurser.
2. Applikasjonen håndterer ikke autentisering selv så sluttbruker videresendes til en identitetstilbyder (AAD).
3. Dette skjer automatisk og oppfattes som en videresending til en ny innloggingsside hos identitetstilbyder. Det som også skjer er at sluttbruker sender med en applikasjons ID til identitetstilbyder. Denne applikasjons IDen hjelper identitetstilbyder å finne ut hvilken applikasjon sluttbruker ønsker tilgang til. Denne kommunikasjonen er ikke oppgitt i figur 4, men skjer mellom pil 3 og 4.
4. Om identitetstilbyder godkjenner autentiseringen sendes en autentiseringkode tilbake til sluttbruker. Autentiseringskoden har kort levetid og er ment for å videresendes til applikasjonen.
5. Applikasjonsserveren mottar forespørsel om å autentisere seg for andre gang og med autentiseringskoden.
6. Applikasjonsserveren sender så autentiseringskoden til identitetstilbyder direkte for å bekrefte at applikasjonen er den bruker har spurt om å autentiseres for.
7. Om identitetstilbyder ser at autentiseringskoden er den samme som ble sendt til sluttbruker, generes og sendes brukerens Access token og Refresh token til applikasjonsserveren.
8. Applikasjonsserveren mottar tokens, validerer og sender begge videre til sluttbru-

ker som nå har tilgang til beskyttede ressurser i applikasjonen.

5.1.2 Logg ut

Autentiseringsprotokollen brukt ved innlogging brukes også ved utlogging. Ved bruk av en SSO løsning er det mulig å ha single sign-out. For at single sign-out skal fungere må identitetstilbyder vite om alle pågående autentiserte applikasjoner og få beskjed om LogoutURL. LogoutURL brukes for å finne ut hvilken applikasjon det logges ut fra og den videresender bruker til en url. SAML og OAuth 2.0 løser utlogging på nesten samme måte, med unntak av at pakkene som sendes er i ulike formater og med ulikt innhold. Applikasjoner som ikke er koblet til identitetstilbyder når utlogging pågår vil få en feilmelding neste gang bruker forsøker å benytte disse. Bruker vil da bli bedt om å logge inn igjen. Hvordan dette utføres kan sees i listen nedenfor:

- Sluttbruker sender en forespørsel til en applikasjon om å logge ut av sin konto.
- Applikasjonen sender en utloggingsforespørsel til identitetstilbyder via sluttbruker.
- Identitetstilbyder logger ut bruker og sender en broadcast til alle applikasjoner tilknyttet seg selv og den autentiserte sesjonen.
- Identitetstilbyder sender et utloggingssvar tilbake til applikasjonen via sluttbruker og denne responsen inneholder en LogoutURL.
- Applikasjonen mottar denne og fullfører utloggingen.

5.1.3 Glemt passord

Om en bruker har glemt passordet til sin Norkart ID konto, kan brukeren trykke på lenken ”Får du ikke tilgang til kontoen?” som ligger på innloggingsiden. Denne siden er levert av AAD, og benyttes hver gang en bruker skal logge på en applikasjon. Når denne linken blir trykket på vil brukeren bli sendt til AAD sin side for tilbakestilling av passord. Her vises en CAPTCHA test som må fylles inn for å bevise at brukeren ikke er en maskin. Bestås testen blir bruker bedt om å velge ønsket autentiseringsmetode. Alternativene er enten å svare på sikkerhetsspørsmål, bli oppringt av Microsoft eller velge å motta en bekreftelsekode på ekstern e-post eller sms. Etter valgt autentiseringsmetode er godkjent, kan det skrives inn et nytt passord. Alle brukere må ha registrert autentiseringsdata for å kunne bruke glemt passord funksjonalitet.

5.2 Egenadministrasjon

Egenadministrasjon i AAD kan håndteres via MyApps (jfr delkapittel 3.13). Registrere eller endre brukerdata er ikke mulig i MyApps, men sluttbruker kan endre passord og registrere autentiseringsdata. Endring av passord kan gjøres ved å trykke på ”endre passord” i MyApps. Først må bruker oppgi gammelt passord, deretter må nytt passord skrives inn to ganger. Passordstyrken vises mens det skrives inn. For at bruker skal kunne endre sitt passord må kontoen være Azure Premium eller Basic, (jfr delkapittel 5.8.2 Lisensmodell).

For å kunne registrere autentiseringsdata må sluttbruker velge ”Registrering for tilbakestilling av passord” i MyApps. Brukeren vil da få valget om å registrere en ekstern e-post adresse, telefonnummer eller oppgi sikkerhetsspørsmål. Dette avhenger av hvilke autentiseringsmetoder som er konfigurert i AAD på forhånd.

For at sluttbruker skal kunne registrere eller endre brukerdata må det utvikles en løsning som benytter PowerShell eller Graph API.

5.3 Brukeradministrasjon

I MyApps kan lokal administrator håndtere gruppemedlemskap. Muligheten til å opprette og fjerne brukere fra AAD er ikke tilgjengelig, heller ikke endring av brukerdata for brukere.

Lokal administrator kan håndtere alle brukerne i AAD. Det er ikke mulig å begrense rettighetene til kun brukerne i en gruppe. Etterspurt funksjonalitet (jfr delkapittel 2.1.2 Høynivå use case beskrivelser) som ikke er i MyApps kan løses ved å sette opp en egen løsning som bruker Graph API eller PowerShell mot AAD.

5.4 Håndtering av brukere og grupper

Dette delkapitlet tar for seg ulike konfigureringer rundt håndtering av brukere og grupper i AAD.

5.4.1 Brukere

Enkeltregistrering, endring av brukerdata og resett av passord kan gjøres i AAD portalen.

Det er også mulig å masse-registrere brukere. Dette kan gjøres ved bruk av AAD Sync, Graph API eller PowerShell. AAD Sync er et verktøy som legges inn på en lokal AD og synkroniserer inn brukere etter ønsket intervall. Ved å bruke PowerShell eller Graph API i denne prosessen kan importen håndteres helt etter egne krav og ønsker.

5.4.2 Grupper

En gruppe i AAD er en samling av brukere og grupper som kan håndteres som en singel enhet. Når et objekt legges inn i gruppe får det tilgangen og rettighetene tilhørende gruppen. Dette medfører at en applikasjon kan tildeles på gruppenivå.

I AAD kan det legges til dynamiske grupper. Dette er grupper som legger til medlemmer automatisk basert på attributter som skrives inn av bruker.

5.5 Håndtering av applikasjoner

Denne seksjonen forklarer hva Norkart må ta hensyn til angående applikasjoner som skal benytte seg av Norkart ID. Applikasjonene må være registrert i en AAD. I tillegg må det implementeres logikk i applikasjonene.

5.5.1 Registrere applikasjoner

For at AAD skal klare å kommunisere med applikasjonene og vite hvilke rettigheter de har trengs det en del informasjon. Hva slags informasjon og hvorfor den trengs er beskrevet nedenfor [24]:

Applikasjons URI

Applikasjons URI er identifikatoren til en applikasjon. URI'en benyttes når en bruker skal autentiseres for applikasjonen. Den blir sendt til AAD og forteller at det er denne applikasjonen bruker ønsker tilgang til.

Reply URL og redirect URL
Dette er URL'er AAD sender autentiseringrespons og token til, dersom autentisering lykkes.

Klient id og nøkkel
Klient id blir generert av AAD når applikasjonen registreres og er applikasjonens id. Denne brukes sammen med en nøkkel, client secret, når applikasjonen trenger tilgang til AAD gjennom Graph API eller andre API'er. Dette gjøres ved at både AAD og applikasjonen kjenner til klient id og nøkkel og kan derfor verifisere forespørsler fra hverandre.

For brukerveiledning til hvordan applikasjoner kan registreres i en AAD, se veiledningskapittelet (jfr 6.1.1 Registrering av web applikasjoner i AAD). Veiledningene viser registrering av single tenant applikasjoner (jfr delkapittelet 3.16).

5.5.2 Konfigurere applikasjoner

Denne seksjonen forklarer hvordan eksponere et web API, få tilgang til Graph API og hvilke regler som kan settes for brukeradgang til applikasjoner.

Denne seksjonen er basert på Microsoft sine forklaringer rundt håndtering av applikasjoner i en AAD [25].

Web API
Om oppdragsgiver ønsker å eksponere et web API til andre applikasjoner i AAD kan dette gjøres ved å laste ned manifestet til API'et og endre OAuth 2.0 rettighetene. Manifestet er en konfigurasjonsfil og kan lastes ned ved å trykke på "Manage manifest" i bunnmenyen i AAD portalen. Bytt ut innholdet i JSON manifestet til å passe API'et (jfr kodeutdrag 1).

```
1   ''oauth2Permissions'': [
2     {
3       ''adminConsentDescription'': ''Allow the application full access to the
            Todo List service on behalf of the signed-in user'',
4       ''adminConsentDisplayName'': ''Have full access to the Todo List service''
            ,
5       ''id'': ''b69ee3c9-c40d-4f2a-ac80-961cd1534e40'',
6       ''isEnabled'': true,
7       ''origin'': ''Application''
8       ''type'': ''User'',
9       ''userConsentDescription'': ''Allow the application full access to the
            todo
10      service on your behalf'',
11      ''userConsentDisplayName'': ''Have full access to the todo service'',
12      ''value'': ''user_impersonation''
13    }
14  ]
```

Kodeutdrag 1: JSON manifest

ID'en, i kodeutdraget, er en unik identifikator for rettighetene som gis av API'et. Denne ID'en kan generes selv og må møte kriteriene for å være en Globally Unique Identifier (GUID). Etter endring lastes manifestet opp i AAD portalen og web API'et vil da bli tilgjengelig for andre applikasjoner.

Få tilgang til Graph API
Hvis applikasjoner trenger tilgang til Graph API kan dette konfigureres under "permissions to other applications". Tilgang til Graph API, er satt som standardinnstilling for alle

applikasjoner i AAD og kalles Windows Azure Active Directory. Tilgang som applikasjoner kan trenge fra Graph API, som lese og skrive rettigheter, settes her.

Adgangsregler

I AAD er det mulig å sette en del adgangsregler for hvilke brukere som skal ha tilgang til applikasjonen. Adgangsreglene kan gjelde for alle brukere, spesifikke brukere eller spesifikke grupper. Det kan konfigureres om multifaktor autentisering alltid skal brukes eller kun dersom brukere logger inn via usikre nettverk. Det er også mulig å la brukere utsette multifaktor autentisering ved å tillate AAD å huske enheter i opptil 60 dager.

5.5.3 Implementering av Webapplikasjon

Applikasjoner som skal bruke AAD som autentiseringsverktøy må implementere elementer. For webapplikasjoner må det legges inn noen biblioteker for å kunne benytte OIDC protokollen og informasjon som AAD trenger for å autentisere brukere opp mot applikasjonen. Utgangspunktet for delkapittelet er veiledningen for MVC ASP.NET webapplikasjon (jfr delkapittel 6.1).

For at applikasjonen skal bruke AAD autentisering via OIDC protokollen må den være Katana basert med OWIN pakker (jfr delkapittel 3.15 OWIN og Katana). Dette gjøres ved å installere OWIN system.web og OWIN security. OWIN security pakken (jfr kodeutdag 17) gjør det mulig å velge OIDC som autentiseringsprotokoll [26].

```
Install-Package Microsoft.Owin.Security.OpenIdConnect -Pre
Install-Package Microsoft.Owin.Security.Cookies -Pre
Install-Package Microsoft.Owin.Host.SystemWeb -Pre
```

Kodeutdrag 2: Katana elementer

I tillegg til Katana elementene må det implementeres logikk i applikasjonen. For å sette OIDC som autentiseringsprotokoll må det lages en klasse som bruker OWIN mellomvaren. Her kan det defineres egenskaper for OIDC (jfr kodeutdrag 3). Egenskapene som trengs er applikasjonens klient id, innloggingsurl og logg ut url.

```
app.UseOpenIdConnectAuthentication(
    new OpenIdConnectAuthenticationOptions
    {
        ClientId = clientId,
        Authority = authority,
        PostLogoutRedirectUri = postLogoutRedirectUri
    });
```

Kodeutdrag 3: Egenskaper for OIDC

I applikasjoner som ikke bruker innlogging fra før må det legges til controller og view som håndterer dette. For at AAD skal kunne autentisere brukere opp mot applikasjonen må det settes fire verdier i web.config filen under ”AppSettings” (jfr kodeutdrag 4).

- ClientId blir forklart nærmere under delkapittel 5.5.1 (Registrere applikasjoner)
- AADInstance vil si hvilken instanse av AAD som brukes. Her brukes https://login.windows.net/0.
- Tenant er navnet, eller domene til den aktuelle AAD tenanten. Dette kan være et domene gitt av Azure eller et eget domene som er registrert for den aktuelle AAD.
- PostLogoutRedirectUri er url’en brukerne vil bli viderekoblet til etter utlogging.

```
1
2     <add key=''ida:ClientId'' value=''1ef05302-c640-4459-a2b1-3c660c9854db'' />
3     <add key=''ida:AADInstance'' value=''https://login.windows.net/{0}'' />
4     <add key=''ida:Tenant'' value=''NorkartIDDevelopment.onmicrosoft.com'' />
5     <add key=''ida:PostLogoutRedirectUri'' value=''https://norkartidman.
          azurewebsites.net/'' />
```

Kodeutdrag 4: Verdier for AAD konfigurasjon

For å sikre kommunikasjon mellom webapplikasjon og AAD må SSL aktiveres.

Kodeeksemplene presentert i denne seksjonen er i hovedsak hentet fra et kodeeksempel i GitHub under AzureAD Samples som heter WebApp-OpenIDConnect-DotNet [27]. Microsoft referer til dette kodeeksemplet flere steder på deres sider når det gjelder implementering av AAD som autentiseringsverktøy for webapplikasjoner.

5.5.4 Implementering av Android-applikasjon

For å knytte Android applikasjoner til AAD er det flere opsjoner som må settes. Dette delkapitlet vil forklare overordnet hvilke elementer som må knyttes sammen og hvorfor. Utgangspunktet for delkapittelet er en demoapplikasjon som lages under veiledning for Android applikasjoner (jfr delkapittel 6.2 Android Brukerveiledning).

Mobile Services

Dette er en Azure tjeneste som muliggjør å koble mobile applikasjoner mot en skalerbar backend server med database og serverlogikk. Demoapplikasjonen tar utgangspunkt i en mobile-service backend for å teste lagring i en Azure database.

Norkart har applikasjoner og tjenester som bruker data fra flere eksterne databaser. I AAD er det mulig å sette opp backend logikk for applikasjoner som jobber direkte mot disse databasene.

AAD

For å bruke AAD som autentisering for en applikasjon må man først opprette en applikasjon i Mobile Services, for deretter å knytte den til AAD. Applikasjonene henger sammen ved hjelp av en applikasjons URL og en klient ID. Hvordan dette gjøres beskrives i veilednigen (jfr delkapittel 6.2.2 Del 2 - Legg inn autentisering i demoapplikasjonen).

Autorisasjonnivåer internt i en applikasjon håndteres av applikasjonens egen database og logikk. Siden AAD støtter OpenID kan applikasjonen få vite hvilken bruker som har fått tilgang og kan derfor autorisere internt i applikasjonen.

Android kode

For å få en Android applikasjon til å støtte autentisering kreves det at hele applikasjonen designes for å oppnå dette. Demoapplikasjonen spør etter autentisering ved oppstart. Dersom autentiseringsprosessen ikke blir vellykket, vil applikasjonen avsluttes. Det brukes biblioteker som benytter både lokale funksjoner og webview mot AAD. Det er klare likheter på innloggingsiden til en Androidapplikasjon og en webapplikasjon.

5.6 Bruk av Graph API

I dette delkapitlet er det noen eksempler på hvordan Graph API kan brukes av en webapplikasjon. Kodeutdrag 5 til 11 viser enkle brukeradministrasjons oppgaver og er hentet

fra Microsoft AAD Graph Team sin blogg [28]

```
public static ActiveDirectoryClient GetActiveDirectoryClientAsApplication() {
    Uri servicePointUri = new Uri(''https://graph.windows.net'');
    Uri serviceRoot = new Uri(servicePointUri, ''norkart.no'');

    ActiveDirectoryClient activeDirectoryClient = new ActiveDirectoryClient(
        serviceRoot, async () => await AcquireTokenAsyncForApplication());

    return activeDirectoryClient;
}
```

Kodeutdrag 5: Graph API: Tilkobling til AAD.

```
AzureActiveDirectoryClient activeDirectoryClient;
Task<IPagedCollection<IUser>> getGraphObjectsTask = activeDirectoryClient.Users.
    ExecuteAsync(); % Hente ut objekter

IPagedCollection<IUser> graphObjects = await getGraphObjectsTask; % Eksempel
```

Kodeutdrag 6: Graph API: Hente brukere.

```
List<IUser> users = activeDirectoryClient.Users.Where(user =>

                    user.UserPrincipalName.Equals(  jon@Norkart  .no )).
                    ExecuteAsync().Result.CurrentPage.ToList();
```

Kodeutdrag 7: Graph API: Hente en spesifikk bruker.

```
Group retrievedGroup = new Group();
string searchString = "US";
List<IGroup> foundGroups = null;

try {
foundGroups = activeDirectoryClient.Groups.Where(group =>
group.DisplayName.StartsWith(searchString))
.ExecuteAsync().Result.CurrentPage.ToList();
} catch (Exception e) {
Console.WriteLine("\nError␣getting␣Group␣{0}␣{1}",
e.Message, e.InnerException != null ? e.InnerException.Message :"");
}

if (foundGroups != null && foundGroups.Count > 0) {
retrievedGroup = foundGroups.First() as Group;
} else {
Console.WriteLine("Group␣Not␣Found");
}
```

Kodeutdrag 8: Graph API: Finne en gruppe og vise innholdet.

```
{IEntity} entity = new {Entity}();
% Definer hvilke entitet som skal legges til.
% Sett properties til entitet.

% Legg til entitet.
await activeDirectoryClient.{Entity}.Add{Entity}Async();

% Eksempel
IUser userToBeAdded = new User();
userToBeAdded.DisplayName = "Test␣Bruker";
userToBeAdded.UserPrincipalName = "testBruker@" + defaultDomain.Name;
userToBeAdded.AccountEnabled = true;
userToBeAdded.MailNickname = "TestBruker";
userToBeAdded.PasswordProfile = new PasswordProfile {
```

```
Password = "TempP@ssw0rd!",
ForceChangePasswordNextLogin = true
};

userToBeAdded.UsageLocation = "US";
await activeDirectoryClient.Users.AddUserAsync(userToBeAdded);
```

Kodeutdrag 9: Graph API: Legge til bruker.

```

% Oppdatere entitet.
await {entityObject}.UpdateAsync();

% Eksempel.
user.City = ''Updated City'';
user.Country = ''New Country'';

await user.UpdateAsync();
```

Kodeutdrag 10: Graph API: Oppdatere bruker.

```
% Gruppeobjektet som bruker skal legges til.
{groupObject}.Members.Add({entityObject} as DirectoryObject);

% Eksempel.
myGroup.Members.Add(userToBeAdded as DirectoryObject);

await myGroup.UpdateAsync();
```

Kodeutdrag 11: Graph API: Legge til bruker i en gruppe.

Kode for å bruker i en gruppe (jfr kodeutdrag 12).

```

% Objektet som skal slettes.
await {EntityObject}.DeleteAsync();

% Eksempeler.
await user.DeleteAsync();
await group.DeleteAsync();
```

Kodeutdrag 12: Graph API: Slette objekt.

5.7 Bruk av PowerShell

I dette delkapitlet er det noen eksempler på hvordan PowerShell kan brukes. For å kunne ta i bruk PowerShell må Microsoft Online Services Sign-In Assistant for IT Professionals RTW installeres. Deretter må Azure Active Directory Module for Windows PowerShell (64-bit version) legges inn. Kodeutdrag 13 til 16 viser enkle brukeradministrasjons oppgaver og er hentet fra nettsidene til Micrsoft Azure [29].

```
% Kobler til Azure AD og innloggingsdetaljer skrives inn.
connect-msolservice
```

Kodeutdrag 13: PowerShell: Koble til Azure

```
% Opprette bruker.
New-MsolUser -UserPrincipalName Test@Norkart.no -DisplayName "Jamie Warner" -
    FirstName "Jamie" -LastName "Warner"

```

```
% Finner bruker.
Get-MsolUser -UserPrincipalName Test@Norkart.no

% Fjerner bruker.
Remove-MsolUser -UserPrincipalName Test@Norkart.no

% Henter fram slettede bruker
$DelUser = Get-MsolUser -UserPrincipalName FSlattery@Norkart.no -
    ReturnDeletedUsers

% Legger bruker tilbake i Azure AD.
Restore-MsolUser -ObjectId $DelUser.ObjectId

% Random passord vil bli generert og velger nytt ved innlogging.
Set-MsolUserPassword -UserPrincipalName Test@Norkart.no

% Satt nytt passord og bruker velger nytt ved innlogging.
Set-MsolUserPassword -userPrincipalName Test@Norkart.no -NewPassword Nula8787
```

Kodeutdrag 14: PowerShell: Administrere brukere

```
% Finner bruker.
$FornavnEtternavn = Get-MsolGroup | Where-Object {$_.DisplayName -eq "Fornavn␣
    Etternavn"}

% Finner gruppen bruker skal legge i.
$Grp = Get-MsolGroup | Where-Object {$_.DisplayName -eq "Norkart␣ID"}

% Legger til medlemmet i gruppen.
Add-MsolGroupMember -groupObjectid $Grp.ObjectId -GroupMemberType "User" -
    GroupMemberObjectId $FornavnEtternavn.ObjectId

% Viser medlemene i gruppen.
Get-MsolGroupMember -GroupObjectId $MktGrp.ObjectId

% Gir bruker rolle.
Add-MsolRoleMember -RoleName "Administrator" -RoleMemberEmailAddress "
    Test@Norkart.no"

% Returnerer alle grupper. Kan bruke ''Get-MsolGroup -GroupType Security |
    format-list'' i tillegg.
Get-MsolGroup

% Returnerer alle medlemmer i en gruppe. Dette er brukere og eventuelle grupper.
Get-MsolGroupMember -groupObjectid <id>

% Opprette ny gruppe.
New-MsolGroup -DisplayName "Support" -Description "Kundeservice␣hos␣Norkart"

% Finn gruppe ID og fjern gruppe.
$groupId = Get-MsolGroup -searchString "Utvilkere"
Remove-MsolGroup -objectid $groupId

% Finn gruppe ID, finne bruker ID og fjerne medlem.
$groupId = Get-MsolGroup -searchString "Norkart␣ID"
$userid = get-msoluser -userPrincipalName Test@Norkart.no
Remove-MsoLGroupMember -groupObjectId $groupid -GroupMemberType User -
    groupmemberobjectid $userid

% Finne gruppe ID og oppdaterer gruppe. I dette tilfellet beskrivelsen av
    gruppen.
$MktGrp = Get-MsolGroup -searchstring Administrator
Set-MsolGroup -ObjectId $MktGrp.ObjectId -Description "Administrator␣over␣hele␣
    systemet."
```

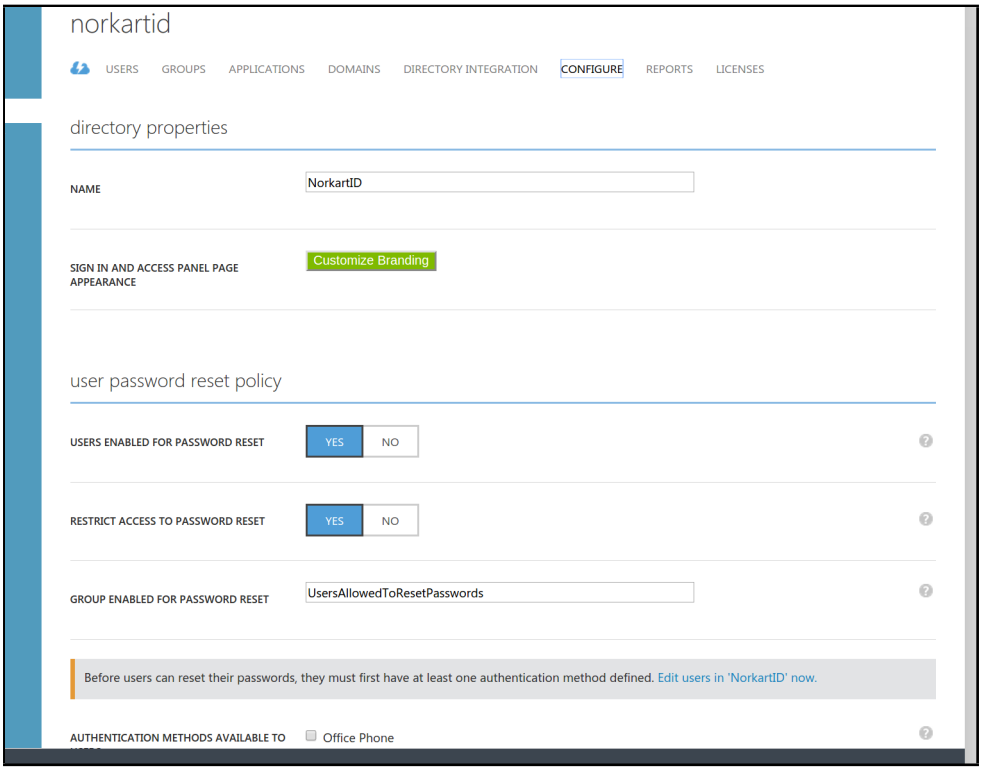
Kodeutdrag 15: PowerShell: Administrere grupper

```
% Returnerer alle administrator roller.
Get-MsolRole

% Finner rolle ID
$role = Get-MsolRole -RoleName "Administrator"

% Finner alle brukere med gitt rolle ID.
Get-MsolRoleMember -RoleObjectId $role.ObjectId

% Fjerne rolle fra bruker.
Remove-MsolRoleMember -RoleName "Administrator" -RoleMemberType User -
    RoleMemberEmailAddress Test@Norkart.no

% Returnerer rollen til en bruker.
Get-MsolUserRole -UserPrincipalName Test@Norkart.no
```

Kodeutdrag 16: PowerShell: Administrere roller

5.8 Generell konfigurasjon av AAD

Konfigurasjon av AAD gir administrator mulighet til å endre og tilpasse en rekke parametere. Det er blant annet mulig å konfigurere tilbakestilling av passord, grupper, applikasjonsproxy og brukeradgang. Delkapittelet vil ta for seg dette og håndtering av roller. Prosjektgruppen anser mye av innstillingene som kan settes som selvforklarende (jfr figur 5).



Figur 5: Konfigurasjon av AAD

5.8.1 Roller

I AAD kan det tildeles roller med ulike rettigheter til brukere og disse er forhåndsdefinert. Det kan ikke opprettes, endres eller slettes roller. Rollene Norkart trenger er:

- Super administrator
- Kundestøtte
- Lokal administrator
- Sluttbruker

Rollene i AAD som passer Norkart er:

- Super administrator passer med ”Global administrator”
- Kundestøtte passer med ”User administrator”
- Lokal administrator passer med ”User” som har eierskap over en gruppe.
- Sluttbruker passer med ”User”

Global administrator kan utføre alle administrative oppgaver i AAD. Det er kun denne rollen som kan tildele andre brukere administratorroller. User administrator kan resette passord og håndtere brukere og grupper. Begrensningen til user administrator er mulighet til håndtering av applikasjoner og generell konfigurasjon av AAD [30].

5.8.2 Lisensmodell

AAD har en egen lisensmodell i forhold til Azure tjenester. Tar Norkart utgangspunkt i den som er tilgjengelig i April 2015 må alle brukere ha premium lisens.

Lisensmodellen i AAD er knyttet til hver enkelt bruker i brukerdatabasen [31]. Første nivå kalles ”Free” som er gratis for de første 500 000 brukerene. Andre nivå heter ”Basic” og prisen må forhandles fram mellom bedriften og Mircosoft for hver samarbeidsavtale. Siste nivået kalles ”premium” og lisensieres med en fast pris i måneden. Premium brukere vil få tilgang til det AAD har av funksjonalitet. Brukere som allerede benytter Microsoft Office 365 har premium lisens.

Ved å ha premium lisens har bruker blant annet mulighet til å endre passord, benytte flerfaktor autentisering og kan tilknyttes til mer enn 10 applikasjoner. Nasos Kladakis, ”Product Marketing Manager” for AAD, sier under en konferanse i November 2014 at det vil komme flere lisensmodeller i løpet av 2015 [32].

Autentiseringspolicy i AAD

For at brukerne skal kunne egenadministrere passord for sin konto må dette aktiveres under konfigurasjonsfanen i AAD portalen. Det kan velges mellom flere autentiseringmetoder og det kan være jobbtelefon, mobiltelefon, ekstern e-post eller sikkerhetsspørsmål. Om sikkerhetsspørsmål er valgt metode kan antall spørsmål defineres.

Profilering

Det er mulig å legge inn logoer og bilder som skal knytte AAD innlogging og annen funksjonalitet til bedriften. Dette gjøres under generell konfigurasjon av AAD. Krav til størrelser på bilder og logoer står beskrevet der dette lastes opp.

6 Veiledninger

Dette kapittelet består av brukerveiledninger for implementering av autentisering i webapplikasjon og Android ved hjelp av AAD. Disse veiledningene er laget av prosjektgruppen og er ment for utviklere ansatt hos Norkart. Alle figurer er laget av prosjektgruppen.

6.1 Brukerveiledning for web applikasjon

Denne veiledningen viser hvordan en web applikasjon kan registreres i en AAD og hvordan implementere autentisering i en ASP.NET Web applikasjon. Applikasjonen er utviklet i Visual Studio 2013 og autentisering er satt til No Authentication.

Kodeeksemplene presentert er hentet fra et kodeeksempel i GitHub under AzureAD Samples som heter WebApp-OpenIDConnect-DotNet [27].

6.1.1 Registrering av web applikasjoner i AAD

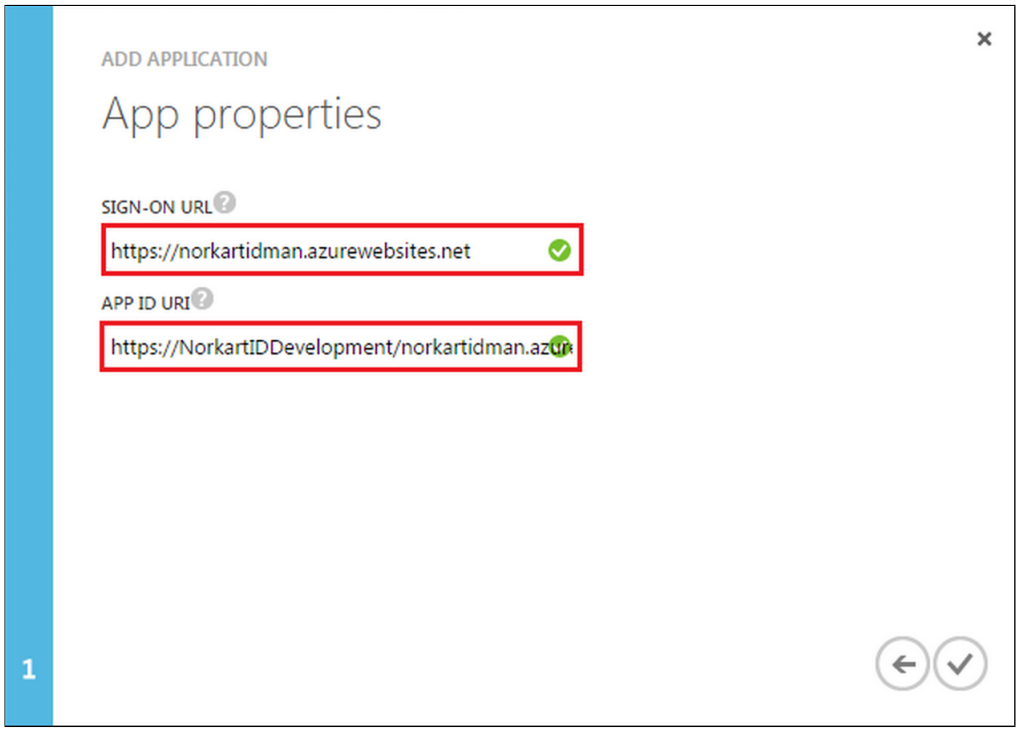
For å bruke AAD som autentiseringsverktøy må applikasjonene registreres. Dette gjøres ved å logge inn i Azure portalen og navigere til aktuell AAD. Velg applikasjoner i toppmenyen og trykk på ADD i bunnmenyen. En dialogboks vises.

1. Velg ”Add an application my organization is developing”.

2. Skriv inn navnet til applikasjonen. Velg WEB APPLICATION AND/OR WEB API og trykk på pilen for å gå videre.

3. Fyll inn sign-on url og App ID URI (jfr figur 6). Sign-on url er base urlen til web applikasjonen eller web API’et. Her må https:// eller http:// være med i urlen. App ID URI er applikasjonen sin ID og brukes når en bruker skal autentiseres for applikasjonen.

4. Klikk på ferdig. Nå er applikasjonen registrert og kan benyttes av brukere i AAD.



Figur 6: WebApp: Registrere applikasjon i AAD

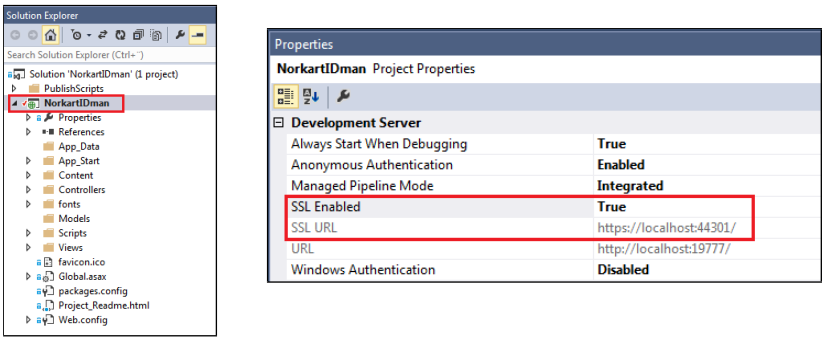
6.1.2 Implementering av AAD i web applikasjon

Når AAD skal brukes som autentiseringsverktøy for en web applikasjon må det legges til elementer i applikasjonen. Det må legges informasjon AAD benytter for å autentisere brukere og biblioteker for å kunne benytte OIDC protokollen.

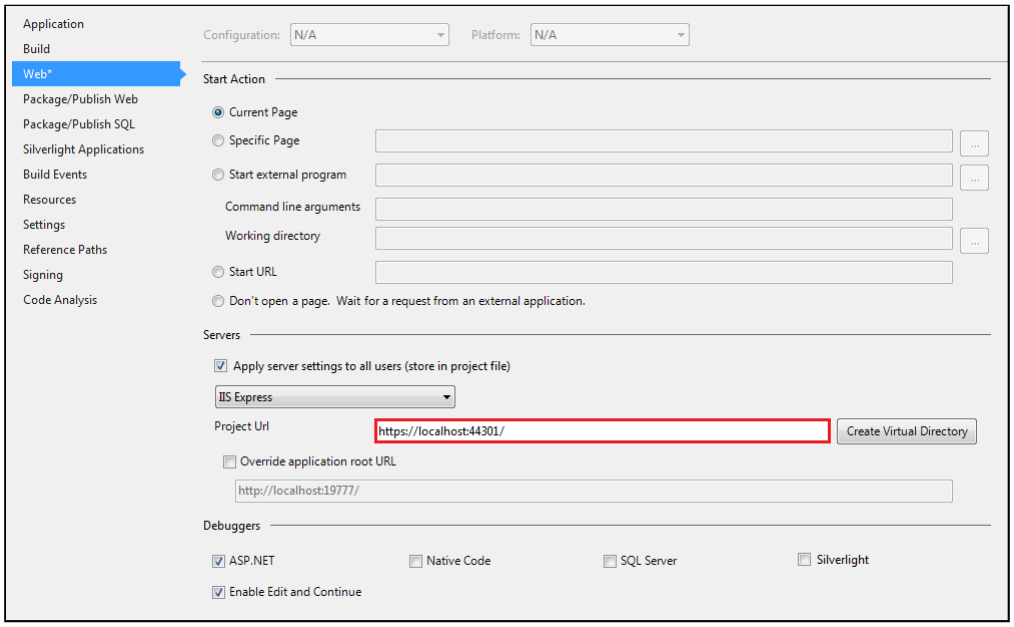
SSL

SSL brukes for å sikre kommunikasjonen mellom applikasjon og AAD. Dette kan aktiveres i Visual Studio.

1. Klikk på applikasjonen i solution explorer, da vil properties vinduet vises. Sett SSL-enabled til true og noter ned SSL url'en (jfr figur 7).
2. Høyreklikk på applikasjonen i solution explorer og velg properties. Velg web og endre Project URL til SSL URL'en (jfr figur 8).



Figur 7: WebApp: Aktiver SSL



Figur 8: WebApp: Registrere SSL url

Implementere OWIN pakker

For å bruke OIDC som autentiseringsprotokoll må det legges til OWIN pakker i applikasjonen. Åpne NuGet konsollen i Visual Studio og installer Katana elementer vist i kodeutdrag 17:

```
1 Install-Package Microsoft.Owin.Security.OpenIdConnect -Pre
2 Install-Package Microsoft.Owin.Security.Cookies -Pre
3 Install-Package Microsoft.Owin.Host.SystemWeb -Pre
```

Kodeutdrag 17: WebApp: Katana elementer

Implementere autentiseringslogikk

1. Naviger til App_Start mappen i prosjektet og legg til en C# klasse som heter Startup.Auth.cs. Fjern App_Start fra namespace og legg til referanser og kode vist i kodeutdrag 18.

```
        using Owin;
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OpenIdConnect;
using System.Configuration;
using System.Globalization;
using System.Threading.Tasks;

namespace WebApp_OpenIDConnect_DotNet
{
    public partial class Startup
    {
        private static string clientId =
            ConfigurationManager.AppSettings["ida:ClientId"];
        private static string aadInstance =
            ConfigurationManager.AppSettings["ida:AADInstance"];
        private static string tenant =
            ConfigurationManager.AppSettings["ida:Tenant"];
        private static string postLogoutRedirectUri
            ConfigurationManager.AppSettings["ida:PostLogoutRedirectUri"];

        string authority =
        String.Format(CultureInfo.InvariantCulture, aadInstance, tenant);

        public void ConfigureAuth(IAppBuilder app)
        {
            app.SetDefaultSignInAsAuthenticationType(
            CookieAuthenticationDefaults.AuthenticationType);

            app.UseCookieAuthentication(new CookieAuthenticationOptions())
                ;

            app.UseOpenIdConnectAuthentication(
                new OpenIdConnectAuthenticationOptions
                {
                    ClientId = clientId,
                    Authority = authority,
                    PostLogoutRedirectUri = postLogoutRedirectUri,
                    Notifications = new
                        OpenIdConnectAuthenticationNotifications
                    {
                        AuthenticationFailed = context =>
                        {
                            context.HandleResponse();
                            context.Response.Redirect("/Error?message="
                                + context.Exception.Message);
                            return Task.FromResult(0);
                        }
                    }
                });
        }
    }
}
```

Kodeutdrag 18: WebApp: Startup.Auth.cs

2. Høyreklikk på prosjektet, velg add, class og velg OWIN Startup Class. Kall klassen for Startup.cs. Legg til koden vist i kodeutdrag 19.

```
using System;
using System.Threading.Tasks;
using Microsoft.Owin;
using Owin;

[assembly: OwinStartup(typeof(WebApp_OpenIDConnect_DotNet.Startup))]
```

```
7
8  namespace WebApp_OpenIDConnect_DotNet
9  {    {
10     public partial class Startup
11     {
12         public void Configuration(IAppBuilder app)
13         {
14             ConfigureAuth(app);
15         }
16     }
17 }
```

Kodeutdrag 19: WebApp: Startup.cs

3. Naviger til Views, Shared og lag et nytt partial view som heter _LoginPartial.cshtml. Legg til kode i _LoginPartial.cshtml vist i kodeutdrag 20. Denne koden vil legge til en logg inn og logg ut knapp i applikasjonen.

```
1  @if (Request.IsAuthenticated)
2  {
3      <text>
4          <ul class="nav␣navbar-nav␣navbar-right">
5              <li class="navbar-text">
6                  Hello, @User.Identity.Name!
7              </li>
8              <li>
9                  @Html.ActionLink("Sign␣out", "SignOut", "Account")
10             </li>
11         </ul>
12     </text>
13 }
14 else
15 {
16     <ul class="nav␣navbar-nav␣navbar-right">
17         <li>@Html.ActionLink(
18         "Sign␣in", "SignIn", "Account", routeValues: null,
19         htmlAttributes: new { id = "loginLink" })</li>
20     </ul>
21 }
```

Kodeutdrag 20: WebApp: LoginPartial.cs

4. Naviger til Views, Shared og åpne _Layout.cshtml. Legg til:

```
1      @Html.Partial("_LoginPartial").
```

Dette vil legge til innholdet i _LoginPartial.cshtml i applikasjonens design.

5. Legg til en ny controller, velg MVC 5 Controller Empty og kall den AccountController. Legg til kode i AccountController vist i kodeutdrag 21.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using Microsoft.Owin.Security.Cookies;
7  using Microsoft.Owin.Security.OpenIdConnect;
8  using Microsoft.Owin.Security;
9
10 namespace WebApp_OpenIDConnect_DotNet.Controllers
11 {
12     public class AccountController : Controller
13     {
```

```
14          public void SignIn()
15          {
16              // Send an OpenID Connect sign-in request.
17              if (!Request.IsAuthenticated)
18              {
19                  HttpContext.GetOwinContext()
20                      .Authentication.Challenge(
21                          new AuthenticationProperties {
22                              RedirectUri = "/"
23                          },
24                          OpenIdConnectAuthenticationDefaults
25                          .AuthenticationType
26                      );
27              }
28          }
29          public void SignOut()
30          {
31              // Send an OpenID Connect sign-out request.
32              HttpContext.GetOwinContext()
33                  .Authentication.SignOut(
34                      OpenIdConnectAuthenticationDefaults
35                      .AuthenticationType,
36                      CookieAuthenticationDefaults
37                      .AuthenticationType
38                  );
39          }
40          }
41  }
```

Kodeutdrag 21: WebApp: AccountController.cs

6. Åpne Web.config og legg til AAD nøkler i <appSettings> vist i kodeutdrag 22.

```
1  <add key="ida:ClientId" value="[Enter␣client␣ID␣from␣Azure␣Portal]" />
2  <add key="ida:Tenant" value="[Enter␣tenant␣name]" />
3  <add key="ida:AADInstance" value="https://login.microsoftonline.com/{0}"
       />
4  <add key="ida:PostLogoutRedirectUri" value="https://localhost:44320/" />
```

Kodeutdrag 22: WebApp: Web.config

- ClientId er en applikasjon id som opprettes når applikasjoner registreres i en AAD. Navigerer til applikasjoner i AAD portalen og velg gjeldende applikasjon. ClientId finnes under Configure.
- AADInstance vil si hvilken instanse av Azure som brukes. Her brukes https://login.windows.net/0
- Tenant er navnet til den aktuelle AAD tenanten. Navnet finnes ved å velge domains i AAD portalen.
- PostLogoutRedirectUri er url'en brukere vil viderekobles til etter utlogging.

Nå er applikasjonen klar for å bruke AAD som identitetstilbyder. Applikasjonen har en logg inn knapp som viderekobler bruker til AAD sin innloggingportal. I applikasjonen kan [Authorize] attributten brukes for å tvinge brukerautentisering på hele eller deler av applikasjonen. Det viser seg å være noen problemer rundt dette hvis brukeren ikke bruker https for å navigere til applikasjonen.

6.2 Android Brukerveiledning

Denne veiledningen viser hvordan en Android applikasjon kan settes opp med AAD som identitetstilbyder. Brukerveiledningen har tatt utgangspunkt i følgende brukerveiledning-

ger fra Microsoft:

- Mobile Services Android get started [33]
- Mobile Services Android get started Users [34]
- Mobile Services how to register Active directory authentication [35]

Veiledningene refererer til hverandre og løser hver sin del. Denne veiledningen er basert på AAD og Android Studio.

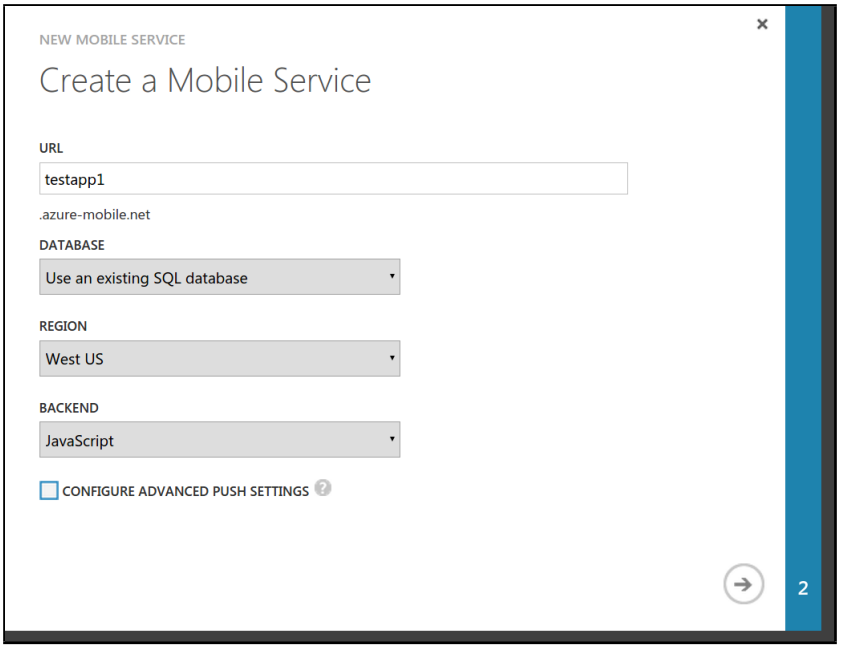
6.2.1 Del 1 - Lag en mobile service i azure og demoapplikasjon

Denne delen viser hvordan en Mobile Service kan opprettes, samt det som må gjøres for å få demoapplikasjonen til å kjøre.

1. Begynn med å logge inn i Azure portalen og opprett en ny mobile service. Mobile service skal fungere som backend, og er det som knytter applikasjonen mot databasen og AAD.

2. Opprett applikasjonsnavnet i Mobile Service, gjør dette så beskrivende som mulig.

3. Velg databasetype. Det kan brukes en eksisterende SQL database eller opprettes ny. Eksisterende databaser må være tilknyttet til Azure før den kan brukes (jfr figur 9).



Figur 9: AndroidApp: Opprette en Mobile Service i Azure

Punkter som må fylles ut:

- Region sier hvor mobile service skal lagres.
- Backend er for å legge inn type backend API. Dette kan være et tillegg til de spørringene applikasjonen allerede kan gjøre mot databasen.
- Sett parametere for databasen.

4. Etter at applikasjonen er opprettet, gå til Mobile Services i Azure portalen. Velg Android og klikk deretter på ”create a new android app”

5. Klikk på create table. Denne oppretter det som trengs for å kjøre demoapplikasjonen. Last ned demoapplikasjonen.

6. Åpne demoapplikasjonen i Android Studio. Om Android Studio ikke godtar sertifikatet som følger prosjektet er dette tilknyttet formatet på https url'en. Denne laster ned siste versjon av gradle og kan hentes over http.

   Slik gjøres dette:

   - Gå inn i filen gradle-wrapper.properties og endre distributionUrl fra https:// til http:// (jfr kodeutdrag 23). Filen ligger under /gradle/wrapper/
   - Kjør en gradle-sync for å oppdatere prosjektet.

```
1  #Sun Jan 11 21:44:32 PST 2015
2  distributionBase=GRADLE_USER_HOME
3  distributionPath=wrapper/dists
4  zipStoreBase=GRADLE_USER_HOME
5  zipStorePath=wrapper/dists
6  distributionUrl=http://services.gradle.org/distributions/gradle-2.2.1-
       all.zip
```

Kodeutdrag 23: AndroidApp: Utdrag fra gradle-wrapper.properties

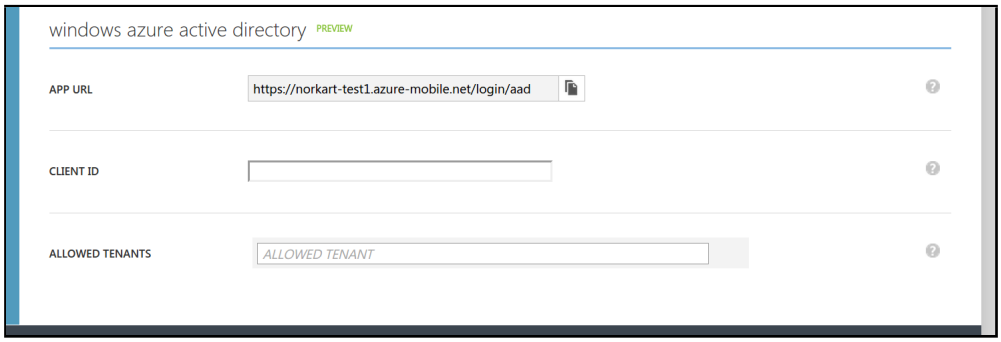
Nå skal applikasjonen kunne kjøres i Android Studio, men den har ikke autentisering. Pakken som ble lastet ned har automatisk lagt inn knytning til mobile service med URI og nøkkel. Applikasjonen leser fra en database og får opp punkter brukerne har lagret. Disse punktene er ikke skilt på brukere og er derfor lik for alle. Kontroller gjerne om punktene som er lagret i applikasjonen fortsatt ligger i databasetabellen i Azure.

6.2.2 Del 2 - Legg inn autentisering i demoapplikasjonen.

For å kunne bruke AAD som identitetstilbyder i applikasjonen må dette registreres både i koden, Azure Mobile Service og AAD. Denne veiledningen forutsetter at det er registrert en applikasjonen i Azure Mobile Service.

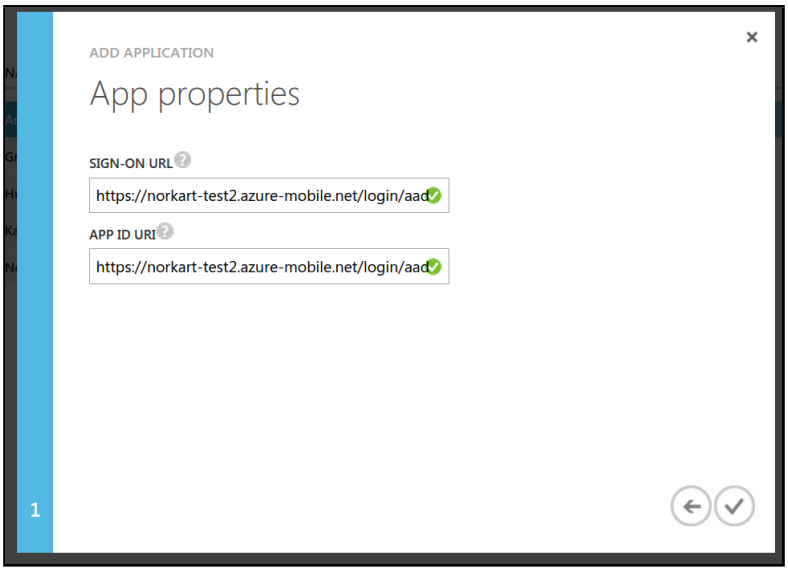
1. Gå inn i Mobile Services og velg applikasjonen som skal knyttes til AAD.

2. Mobile Service skal klargjøres for tilknytning til AAD. Velg identity i Mobile Service og naviger ned til windows azure active directory. Kopier her URL fra applikasjonen (jfr figur 10).



Figur 10: AndroidApp: Mobile Service Identity

3. Applikasjonen i Mobile Service skal derretter legges til i AAD. Gå til applications i AAD portalen og gjør følgende:
   - Klikk ADD nederst på skjermen for å legge til en ny applikasjon.
   - Velg ”Add an application my organization is developing” i vinduet som kommer opp.
   - Angi navnet applikasjonen skal ha i AAD.
   - Velg ”WEB APPLICATION AND/OR WEB API”
   - Klikk neste og angi den kopierte APP URL’n i begge boksene (jfr figur 11).

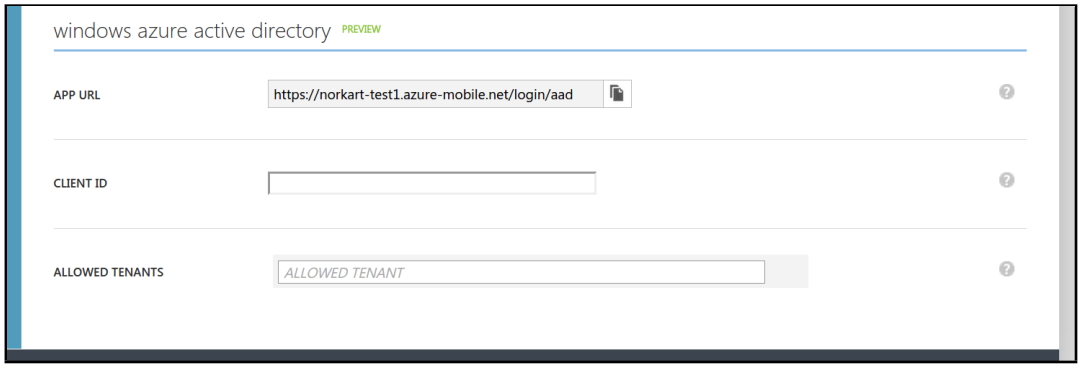


Figur 11: AndroidApp: Registrere applikasjon i AAD

4. Mobile Service er nå knyttet mot AAD og AAD skal nå knyttes mot Mobile Service. Dette gjøres ved å hente ut klient id for applikasjonen og legges inn i Mobile Service. Dette gjøres slik:
   - Klikk på configure og naviger ned til CLIENT ID og kopier denne.
   - Gå tilbake til Azure Mobile Services og velg applikasjonen som ble lagt til tidligere.

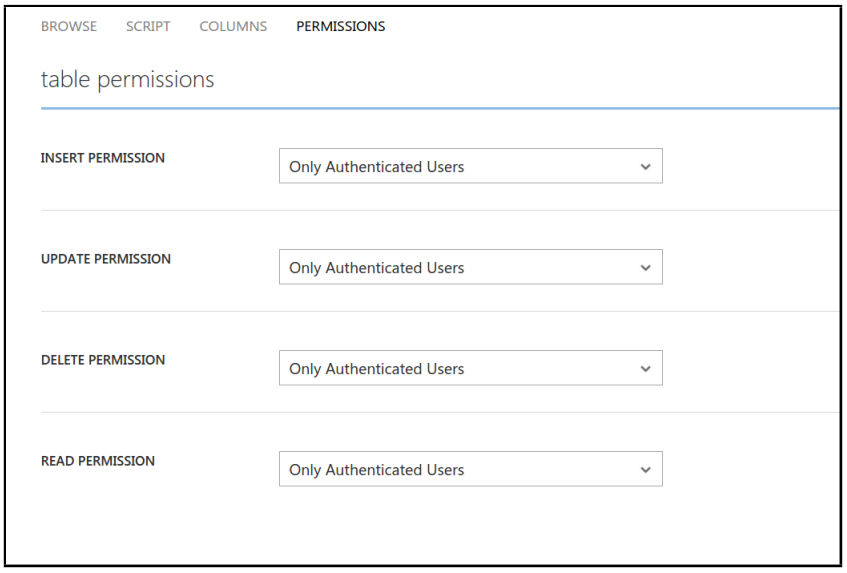
- Velg identity og naviger ned til windows azure active directory og lim inn CLIENT ID i boksen under APP URL.

5. Applikasjonen kan begrenses eller åpnes for spesifikke domener. For å gjøre dette må domenene hentes i AAD og legges inn i Mobile Service. Dette gjøres slik:
   - Gå til AAD portalen og klikk på DOMAINS.
   - Om det ikke er opprettet egne domener vil det være angitt et standard domene. Kopier de domenene som skal ha tilgang til applikasjonen.
   - Gå til Azure Mobile Services, velg applikasjon, identity, nagiver til windows azure active directory og lim inn domenene i ALLOWED TENANTS (jfr figur 12).
   - Gå deretter tilbake til AAD portalen og gi brukere tilgang til applikasjonen.



Figur 12: AndroidApp: Legg til domener

6. Selv om applikasjonen nå vil kreve autentisering ved oppstart er det ikke lagt inn noen begrensning i databasen. For å hindre ikke-autentiserte brukere tilgang til dataene i databasen, endres innstillinger i Azure Mobile Services. Dette gjøres slik:
   - Gå til Azure Mobile Services og velg applikasjonen som skal endres. Velg DATA og deretter PREMISSIONS.
   - Endre alle rettighetene til “Only Authenticated Users” (jfr figur 13).
   - Husk å klikk SAVE nederst ved endring.



Figur 13: AndroidApp: Endre databaserettigheter

7. Det kan nå legges til autentisering i applikasjonen. Endringene som gjøres her refererer til filnavn i demoapplikasjonen. Begynn med å legge til følgende biblioteker for import (jfr kodeutdrag 24).

```
1
2  import java.util.concurrent.ExecutionException;
3  import java.util.concurrent.atomic.AtomicBoolean;
4
5  import android.content.Context;
6  import android.content.SharedPreferences;
7  import android.content.SharedPreferences.Editor;
8
9  import com.microsoft.windowsazure.mobileservices.authentication.
       MobileServiceAuthenticationProvider;
10 import com.microsoft.windowsazure.mobileservices.authentication.
       MobileServiceUser;
```

Kodeutdrag 24: AndroidApp: Biblioteker som bør importeres.

8. Det trengs en funksjon som autentiserer bruker når applikasjonen starter. Legg til ”authenticate()” klassen i samme filen som bibliotekene (jfr kodeutdrag 25). Legg merke til at denne koden kan avvike fra veiledninger på nettet, ettersom det her er spesifisert at AAD er identitetstilbyder.

```
1
2  private void authenticate() {
3      // Login using the Azure AD provider
4
5      ListenableFuture<MobileServiceUser> mLogin =
6      mClient.login(
7          mobileServiceAuthenticationProvider.WindowsAzureActiveDirectory);
8
9      Futures.addCallback(mLogin,
10         new FutureCallback<MobileServiceUser>() {
11
12         @Override
13         public void onFailure(Throwable exc) {
```

```
14              createAndShowDialog((Exception) exc, "Error");
15          }
16
17          @Override
18          public void onSuccess(MobileServiceUser user) {
19              createAndShowDialog(String.format(
20                  "You are now logged in - %1$2s",
21                  user.getUserId()), "Success");
22              createTable();
23          }
24      });
25  }
```

Kodeutdrag 25: AndroidApp: Hvordan authenticate klassen kan se ut.

9. For at funksjonen skal kjøre ved oppstart legges authenticate() inn på egnet sted i applikasjonen. Om det brukes et annet prosjekt enn demoapplikasjonen kan dette legges inn etter initiering av alle objekter, men før andre logikk operasjoner. For demoapplikasjonen legges dette inn etter at MobileServiceClient er initiert i onCreate(); klassen (jfr kodeutdrag 26).

```
1
2   try {
3      // Create the Mobile Service Client instance, using the provided
4
5      // Mobile Service URL and key
6      mClient = new MobileServiceClient(
7           "https://norkart-test1.azure-mobile.net/",
8           "nSxoADeTLiHClFQShuomNyfOeSGAQL23",
9           this).withFilter(new ProgressFilter());
10
11     authenticate();
12     } catch (MalformedURLException e) {
13          createAndShowDialog(new Exception("There was an error creating the
                Mobile Service. Verify the URL"), "Error");
14  }
```

Kodeutdrag 26: AndroidApp: Hvordan onCreate() klassen kan se ut etter at kode er flyttet

Kodelinjene som lå under MobileServiceClient initieringen flyttes ned i en egen klasse som kalles createTable() (jfr kodeutdrag 27).

```
1
2   private void createTable() {
3
4       // Get the Mobile Service Table instance to use
5       mToDoTable = mClient.getTable(ToDoItem.class);
6
7       mTextNewToDo = (EditText) findViewById(R.id.textNewToDo);
8
9       // Create an adapter to bind the items with the view
10      mAdapter = new ToDoItemAdapter(this, R.layout.row_list_to_do);
11      ListView listViewToDo = (ListView) findViewById(R.id.listViewToDo);
12      listViewToDo.setAdapter(mAdapter);
13
14      // Load the items from the Mobile Service
15      refreshItemsFromTable();
16  }
```

Kodeutdrag 27: AndroidApp: Hvordan createTable() klassen kan se ut

10. Etter at kodeutdragene er lagt inn kan applikasjonen kjøres og det må oppgis brukernavn og passord for å få logget inn. Det er fortsatt mulig å gå ut av påloggingsvinduet, men applikasjonen vil feile. Dette skjer fordi applikasjonen ikke er programmert til å ta i annet enn vellykkede spørringer fra databasen. Det kan legges

inn en egen sjekk om bruker er autentisert eller applikasjon kan avslutte seg selv. Kodeeksempel på hva som kan gjøres er å legge til en finish(); i callBack exception i authenticate() klassen (jfr kodeutdrag 28).

```
1
2  Futures.addCallback(mLogin, new FutureCallback<MobileServiceUser>() {
3    @Override
4    public void onFailure(Throwable exc) {
5        createAndShowDialog((Exception) exc, "Error");
6        finish();
7    }
8
9    @Override
10   public void onSuccess(MobileServiceUser user) {
11       createAndShowDialog(String.format(
12           "You␣are␣now␣logged␣in␣-␣%1$2s",
13           user.getUserId()), "Success");
14       createTable();
15   }
16 });
```

Kodeutdrag 28: AndroidApp: Hvordan onFailure() kan se ut om det legges til finish();

11. Applikasjonen skal nå fungere med autentisering mot AAD, men det er ikke lagt til noen logg ut knapp. For å teste innlogging med ulike brukere må det fjernes applikasjonsdata i Android systemet.

7 Testing

Dette kapitlet presenterer hvordan prosjektgruppen testet valgt løsning mot kravene i kravspesifikasjonen og består av fire deler (jfr kapittel 2). Del en presenterer funksjonstester gruppen har gjort mot systemet for å svare på om funksjonaliteten i løsningen samsvarer med kravene. Del to består av en brukervennlighetsanalyse og avdekker om løsningen virker intuitiv og brukervennlig. I del tre testes AAD portalen mot de operasjonelle kravene. I fjerde og siste del av kapittelet gjør prosjektgruppen en samlet vurdering rundt testresultatene.

7.1 Funksjonstester

Prosjektgruppen testet funksjonalitet knyttet til innloggingsmekanismer, MyApps og AAD Portalen for å avklare om løsningen leverer etter det kravspesifikasjonen presiserer.

7.1.1 Metode

For å teste funksjonaliteten på de forskjellige systemene utførte prosjektgruppen funksjonstester i form av black box testing. Det ble testet om forventet resultat ble oppnådd når en handling ble gjort i løsningen [36]. Under black box testing er koden ukjent og ble brukt fordi prosjektgruppen ikke hadde innsyn i koden til hverken MyApps eller AAD portalen. Testene som ble gjort baserer seg på oversikten over hvilke funksjonalitet de forskjellige rollene skal kunne gjøre i kravspesifikasjonen (jfr delkapittel 2.1.3 Roller og tilganger).

7.1.2 Innloggingsmekanismer

Dette ble testet på både web og native applikasjoner gruppen utviklet. Resultatet av testene viser at funksjonalitet innen innloggingsmekanismer fungerer som forventet. Testene gjort på innloggingsmekanismer kan sees i vedlegget funksjonstester (jfr B.1).

7.1.3 MyApps

For å teste rollene til sluttbruker og lokal administrator ble det tatt utgangspunkt i funksjonaliteten som finnes i MyApps. Resultatet av testene (jfr vedlegg B.2 Tester for MyApps) viser at sluttbruker kun har mulighet til å endre sine autentiseringsdata og sitt passord. Det vil si at MyApps ikke har støtte for at sluttbruker kan registrere eller endre sine egne data. Testene viser også at lokal administrator har mulighet til å legge til og fjerne brukere i en gruppe, men ikke endre brukeres data eller passord. Det er heller ikke mulig å registrere eller fjerne brukere fra AAD via MyApps.

7.1.4 AAD portalen

Testresultat kan leses i vedlegg om funksjonstester av AAD portalen (jfr vedlegg B.3 Tester for AAD Portalen). Testene viste at all funksjonalitet definert for kundestøtte i kravspesifikasjonen er mulig. For super administrator viser testene at all funksjonalitet tilfredsstiller kravene utenom håndtering av roller. Det er ikke mulig å opprette, endre eller slette roller i AAD. Prosjektgruppen fikk ikke testet endring av domene i AAD, men det skal være mulig å knytte Norkart sitt domene til AAD når det opprettes kontoer [37].

Dette vil da kunne gi brukernavn som ”bruker@norkart.no” eller ”bruker@norkartid.no”.

7.2 Brukervennlighetsanalyse

Det var viktig for oppdragsgiver at Norkart ID skulle være brukervennlig for alle brukere. ”En brukervennlig løsning skal føre brukeren til målet på en effektiv og forståelig måte” [38]. Brukervennlighetskrav i denne analysen er basert på kravene i kravspesifikasjonen (jfr delkapittel 2.2.2 Brukervennlighet) og at hovedfunksjonalitet beskrevet i høynivå use case beskrivelser (jfr delkapittel 2.1.2) kan utføres på en intuitiv måte.

7.2.1 Metode

For å analysere brukervennligheten til Norart ID ble det gjort brukertester og analyse av innloggingsmekanismer, MyApps og AAD portalen. Brukertestene ble utført ved at testbrukere gjorde oppgaver samtidig som de ble observert av prosjektgruppen. For å kunne måle hvor godt testbrukerne utførte hver oppgave ble det satt karakterer i forhold til karakterskala (jfr tabell 14).

Lett	Fullført på 1 forsøk
Middels	Fullført på 2 eller 3 forsøk. Observert vanskeligheter
Vanskelig	Fullført på 3. eller 4 forsøk. Uttrykt vanskeligheter
Fullført med hjelp	Måtte få hjelp for å fullføre
Ikke fullført	Klarte ikke oppgaven

Tabell 14: Karakterskala for brukertester

I tillegg til brukertestene svarte testbrukerne på et System Usabillity Scale (SUS) spørreskjema. SUS er et verktøy laget av John Brooke, for å måle brukervennlighet [39]. Brukervennlighetsanalysen ble utført på bagrunn av notatene fra observasjonene gjort under brukertestingen, karakterene på hver oppgave og resultatet av SUS. Metode brukt for brukertesting og analyse er inspirert av boken Praktisk brukertesting, skrevet av E. Andersen og J.G. Wold [40]. For å få oversikt over resultatene fra brukertestene ble observasjoner og karakterer registrert i verktøyet Datalogger. Datalogger gir statistikk på utføring av oppgaver og resultat av SUS. Datagrunnlaget vil ikke være statistisk holdbar ettersom brukertestene ble gjennomført av få brukere, men den vil likevel gi innsikt.

7.2.2 Innloggingsmekanismer og MyApps

Dette delkapittelet viser tester, testresultater og analyse av innloggingsmekanismer og MyApps.

Bruketesting

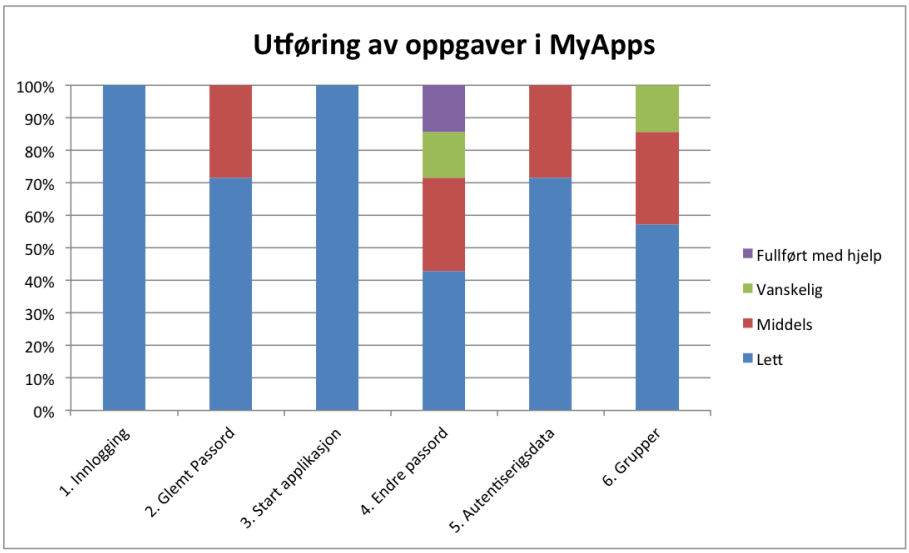
For å teste innloggingsmekanismer og MyApps ble syv studenter fra HiG observert en etter en mens de fullførte brukertester. Testbrukerne hadde varierende IT-kunnskaper og ingen hadde erfaring med MyApps fra før. Brukertestene gikk ut på å teste innloggingsmekanismer, egenadministrasjon og håndtering av grupper. Nedenfor er forkortet versjon av oppgavene de fikk. Oppgavene i sin helhet kan lese i vedlegget med testplaner (jfr vedlegg C.1.1).

1. Logg inn på demoapplikasjon med ditt brukernavn og passord.
2. Du vil logge inn på demoapplikasjon men har glemt ditt passord. hva gjør du?
3. Start en applikasjon fra MyApps.
4. Endre ditt passord via MyApps.
5. Endre telefonnummer registrert for resett av passord.
6. Legg til og fjern en bruker fra en gruppe.

I tillegg til disse oppgavene ønsket prosjektgruppen å teste sluttbrukers registrering og endring av brukerdata og lokal administrators håndtering av brukere. Funksjonstestene gruppen gjorde (jfr delkapittel 7.1.3 MyApps) viser at denne funksjonaliteten ikke er tilgjengelig i MyApps og kunne derfor ikke testes.

Resultat av testing

Diagrammet i figur 14 viser en oversikt over hvordan oppgavene ble utført av alle testbrukerne. Testene viste at innlogging var lett forståelig ettersom alle testbrukerne klarte denne oppgaven på første forsøk. Det kom også frem at noen brukte mer enn ett forsøk på å utføre glemt passord funksjonaliteten. Diagrammet viser at alle oppgavene som ble utført i MyApps, unntatt å starte en applikasjon, ikke virket logisk for alle. Litt over 40% av testbrukerne klarte ikke å endre passord på første forsøk. Observasjoner gjort under brukertestene støtter resultatene i diagrammet. Et sammendrag av observasjoner på tvers av testbrukerne kan leses i vedlegget om observasjoner for sluttbruker og brukeradministrator (jfr vedlegg C.2.1).

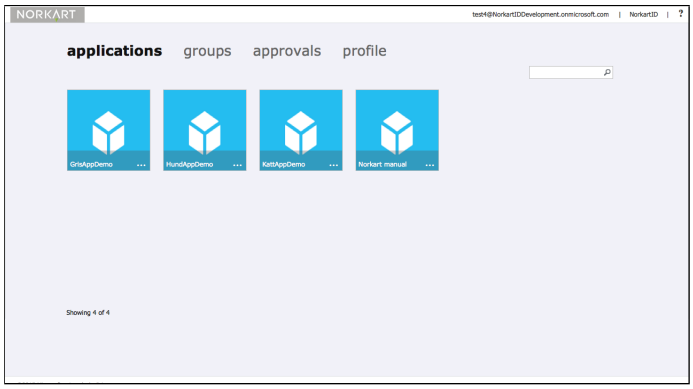


Figur 14: Diagram for testresultater

Resultatet av SUS analysen viser at MyApps og innloggingssiden til sammen fikk en SUS score på 71,4. En SUS score kan maksimalt være 100 og verdier fra 68 og oppover vurderes som over gjennomsnittet [41]. En oversikt over svarene på SUS spørreskjema kan leses i vedlegget SUS resultat for innloggingsmekanismer og MyApps (jfr vedlegg C.2.3).

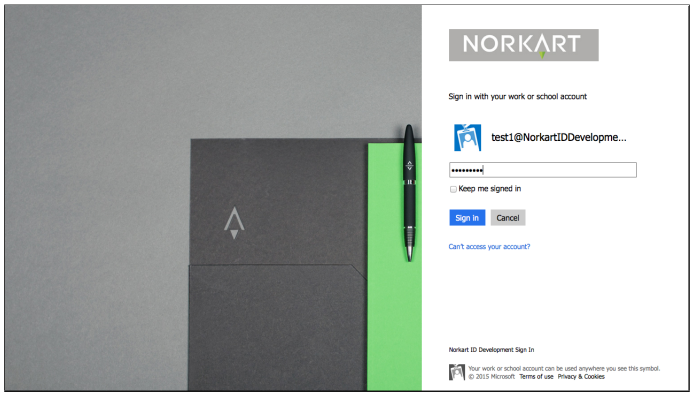
Analyse

I observasjoner gjort under bukertestene (jfr vedlegg C.2.1 Observasjoner fra brukertesting av MyApps og innlogging) kom det tydelig frem at navigasjonen i MyApps var uvant for flere av testbrukerne (jfr figur 15). Det var flere som ikke forsto at hovedmenyen var en meny. Mange prøvde å lete etter funksjonsvalg øverst til høyre under brukernavn. Dette var trolig hovedårsaken til at flere av testbrukerne hadde trøbbel med å utføre oppgavene. Prosjektgruppen så en tydelig forbedring i selvsikkerhet ved bruk av MyApps etter at navigasjonen ble forstått.



Figur 15: Hovedsiden til MyApps

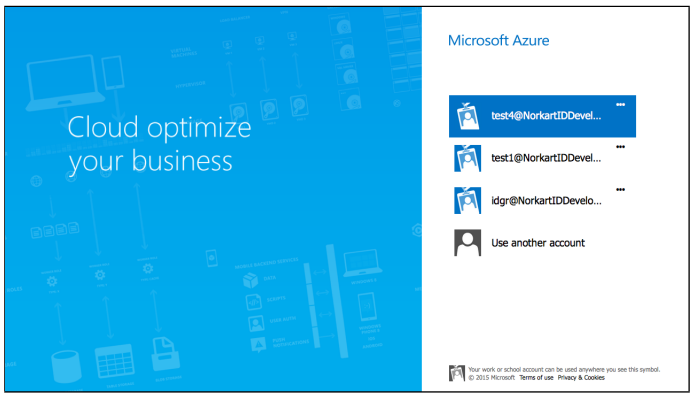
Kun 40% av testbrukerne forsto at det var Norkart ID de brukte for å logge seg inn (jfr figur 16. Dette bryter med kravet ”GUI design skal skape gjenkjennelighet til Norkart for sluttbrukere.” fra kravspesifikasjonen (jfr delkapittel 2.2.2 Brukervennlighet).



Figur 16: Norkart ID Innlogging

Før brukere kommer til innloggingssiden ser de en oversikt over brukere som har logget seg inn på applikasjonen tidligere (jfr figur 17). For å se glemt passord lenken må det

først velges en brukerkonto. Dette er trolig grunnen til at flere av testbrukerne brukte tid på å lokalisere lenken.



Figur 17: Norkart ID Innlogging, brukerkonto

7.2.3 AAD portalen

Dette delkapittelet viser tester, testresultater og analyse av AAD portalen.

Brukertesting

AAD portalen ble testet av to ansatte hos Norkart. En har vært ansatt som utvikler i 2 år og hadde sett AAD portalen før men ikke brukt den. Den andre har jobbet i kundestøtte i 8 år og hadde aldri hørt om AAD portalen, men kjente til AD prinsippet.

Testbrukeren fra kundestøtte gjorde oppgaver relatert til håndtering av brukere og grupper i AAD portalen. Se testplanen for kundestøtte (jfr vedlegg C.1.2) for mer informasjon rundt brukertesten. Listen nedenfor er en forkortet oversikt over oppgavene:

1. Opprett en ny gruppe og gi den tilgang til en applikasjon.
2. Opprett en ny bruker med rollen som brukeradministrator og sett persondata og autentiseringsdata for brukeren.

Super administrator utførte oppgaver som omhandlet generell håndtering av AAD og applikasjoner i denne. For mer infromasjon rundt disse oppgavene og brukertest for super administrator se testplanen (jfr veldegg C.1.3). Listen nedenfor er en forkortet oversikt over oppgavene:

1. Endre autentiseringsvalg for resett av passord.
2. Gjør alle brukere i AAD om til premium.
3. Registrer en ny web applikasjon i AAD og opprett en klient nøkkel for denne.
4. Gi en applikasjon skrive- og leserettigheter til Graph API.

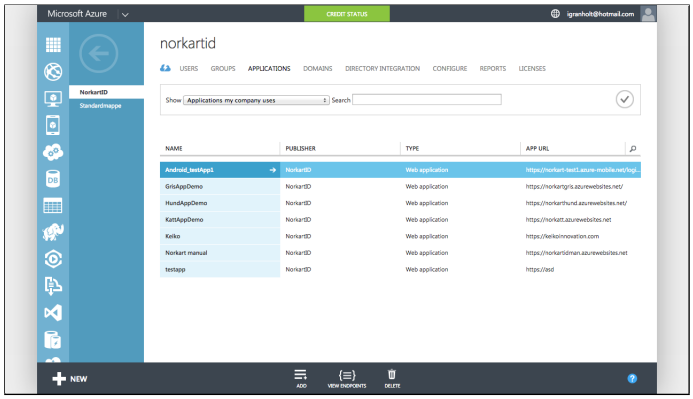
Håndtering av roller ble ikke testet ettersom funksjonstestene avklarte at det ikke finnes funksjonalitet for å gjøre dette i AAD (jfr delkapittel 7.1.4 - AAD portalen).

Resultat av testing

Prosjektgruppens observasjon av testbrukerne (jfr vedlegg C.2.2) viser at alle oppgavene ble fullført. Testpersonen for kundestøtte brukte flere forsøk på begge oppgavene og hadde problemer med å nagivere. Utvikleren, som hadde rollen super administrator, brukte to forsøk på alle oppgavene utenom å gi en applikasjon tilgang til Graph API. Dette ble gjort på ett forsøk. SUS analysen viser at AAD portalen fikk en SUS score på 57,5, altså under gjennomsnittet. Resultatet av SUS undersøkelsen kan leses i SUS resultat for AAD portalen (jfr vedlegg C.2.4).

Analyse

Tilbakemelding fra testbrukerne og prosjektgruppens observasjoner viser at AAD portalen krever opplæring før bruk. Det tok testpersonene flere forsøk å klare flere av oppgavene. Dette var et forventet resultat ettersom testpersonene kun fikk en liten innføring i AAD portalen før de utførte oppgavene. Begge testpersonene ytret at en bunnmeny med mye funksjonalitet var uvant og dette kan være grunnen til at de brukte flere forsøk på å navigere til riktig sted(jfr figur 18). Begge ønsket gjerne å bli henvist til videre funksjonalitet som kunne gjøres med en nyopprettert bruker eller gruppe. Dette tilsier at arbeidsflyten i systemet kan forbedres.



Figur 18: AAD Portalen

7.3 Gjennomgang av operasjonelle systemkrav mot AAD portalen

Prosjektgruppen har gått gjennom de operasjonelle kravene spesifisert i kravspesifikasjonen (jfr delkapittel 2.2) for å finne ut om AAD portalen overholder disse. Delkapitlet har fokus på å besvare om løsningen tilfredsstiller kravene om ytelse, sikkerhet og lover. Kravene om brukervennlighet er ikke gjennomgått her, ettersom de ble testet i brukervennlighetsanalysen (jfr delkapittel 7.2).

7.3.1 Ytelse

I dette delkapittel presenteres metode og resultat for testing av ytelse.

Metode

Ytelseskravene som omhandler innloggingsmekanismer testet prosjektgruppen ved hjelp av black box testing. Hvilke tester som ble gjennomført kan leses i vedlegget om funksjonstesting (jfr vedlegg B.4). Gjennomgang av resten av kravene om ytelse er basert på

dokumentasjon fra Microsoft.

Resultat

Kravene om antall brukere som skal kunne autentiseres samtidig blir bekreftet av Microsoft [42]. Resultatet av prosjektgruppens testing av ytelse rundt innloggingsmekanismer viser at kravene er overholdt (jfr vedlegg B.4).

7.3.2 Sikkerhet og autentiseringskrav

I dette delkapittelet presenteres metode og resultat for testing av sikkerhet og autentiseringskrav.

Metode

Det ble gjennomført black box tester for å avklare kravene rundt sikkerhet og autentisering. Autentisering i forhold til innlogging i nettleser ligger i vedlegget om funksjonstester av innloggingsmekanismer (jfr B.1). Resterende tester kan leses i vedlegget om funksjonstester av sikkerhet og autentisering (jfr B.5).

Resultat

Testene viser at alle kravene rundt sikkerhet og autentisering blir oppfylt (jfr vedlegg B.5 Sikkerhetstester i forhold til kravspesifikasjon).

7.3.3 Lover og regler

I dette delkapittelet presenteres metode og resultat for testing av lover og regler.

Metode

> Forskrift om universell utforming av IKT-løsninger stiller krav om at nettsider må oppfylle 35 av 61 suksesskriteriene i standarden Retningslinjer for tilgjengelig webinnhold (WCAG) 2.0."

Kilde: Direktoratet for forvaltning og IKT [43].

For å finne ut om AAD portalen overholder reglement om universell utforming har prosjektgruppen gått gjennom WCAG 2.0 kravene (jfr vedlegg D Krav til nettløsninger i WCAG 2.0). Om et nettsted innfrir disse kravene blir det AA verifisert, som betyr at det er universelt utformet. For å teste kravet om kontrast ble to nettsteder tatt i bruk, WebAIM [44] og RGB to Hex [45]. WCAG 2.0 krav som omhandler kildekoden til AAD Portalen, som språk på siden og deler av innhold, har gruppen ikke fått testet ettersom kildekoden ikke er offentlig tilgjengelig.

Resultat

Resultatet av gjennomgangen av WCAG 2.0 krav viser at AAD Portalen møter alle kravene prosjektgruppen testet, utenom kravet om kontrast. Kontrasttestene viser at den ikke er god nok for å skape leselighet hos alle type brukere. Testen avdekket en differanse på 1.96:1 der kravet er minimum 4.5:1 for å få godkjent.

7.4 Oppsummering og vurdering av testresultater

I dette delkapittelet gjøres det en samlet vurdering av testresultatene.

Innloggingsmekanismer og MyApps

Resultatet av testene gjort viser at det er noe funksjonalitet i kravspesifikasjonen som ikke er tilgjengelig. Sluttbrukere får kun endret passord, ikke brukerdata. Lokal administrator kan kun håndtere grupper i MyApps, ikke sluttbrukere. I tillegg kommer det frem i brukervennlighetsanalysen at kravet om at GUI design som skal skape gjenkjennelighet til Norkart ikke overholdes. Den manglende funksjonaliteten blir sett på som viktig for Norkart. Det er mulig for Norkart å utvikle en egen brukerportal ved hjelp av Graph API eller PowerShell. MyApps tilbyr altså mye av ønsket funksjonalitet, men leverer ikke alt som er spesifisert i kravspesifikasjonen.

AAD portalen

I testene gjort på AAD portalen kom det frem at noen av kravene ikke oppfylles men hovedfunksjonaliteten i kravspesifikasjonen overholdes. Oppretting, endring og sletting av roller er ikke mulig. Denne funksjonaliteten blir ikke sett på som avgjørende for Norkart da AAD allerede har forhåndsdefinerte roller som kan brukes. AAD portalen møter ikke WCAG 2.0 sine krav om kontrast. Brukervennlighetsanalysen viste at løsingen ikke har optimal arbeidsflyt når det kommer til oppretting av brukere og grupper. AAD portalen tilbyr altså hovedfunksjonaliteten Norkart etterspør.

8 Første utarbeidede kravspesifikasjon

Kravspesifikasjonen gjelder for SSO løsningen gruppen skulle utvikle. Ved behov gjelder også kravspesifikasjonen for eventuelle prototyper som lages for å bevise eller utdype ulike tekniske valg og løsninger.

8.1 Hvordan kravspesifikasjon er utarbeidet

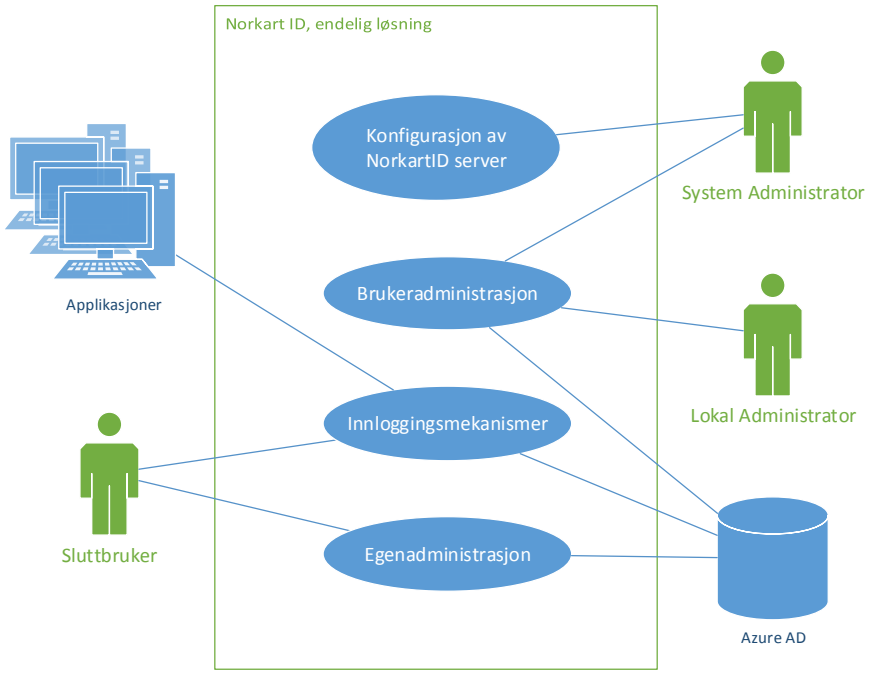
Prosjektgruppen satt seg inn i ulike autentiseringsløsninger før kravspesifikasjonen ble utarbeidet. Ingen på prosjektgruppen hadde innsikt på fagområdet autentisering og autorisering så et faglig dypdykk ble gjort i en kort periode. Etter dette ble en kravspesifikasjon for SSO løsningen utarbeidet etter Norkart og produkteier sine krav. Prosjektgruppen hadde jevnlig dialog med produkteier for å sikre at oppgaven og kravspesifikasjon hang sammen. I forhold til målbare data oppgitt i delkapitlet om operasjonelle krav (jfr delkapittel 8.3 Operasjonelle krav til løsning) ble det tatt utgangspunkt i hva Norkart så for seg som et ekstremscenario for løsningen om de skulle ha høy aktivitet.

8.2 Funksjonelle krav til løsningen

Dette er krav som sier hva systemet skal utføre, og hvordan systemet skal reagere på ulike situasjoner.

8.2.1 Overordnet use case diagram

For å beskrive hva endelig løsning skal inneholde av funksjonalitet og roller ble det laget et overordnet use case (jfr figur 19). Dette use caset er en skisse av Norkart ID som endelig løsning. For å begrense oppgaven fokuseres det på egenadministrasjon og innloggingsmekanismer. Nedenfor kan det leses om en kort forklaring av roller og funksjoner.



Figur 19: Overordnet Use Case for endelig løsning av Norkart ID

System Administrator

Dette er en rolle som setter opp hvordan Norkart ID systemet skal konfigureres sammen med applikasjonserverne. System administrator har også mulighet til å administrere alle brukere som er tilkoblet systemet.

Lokal Administrator

Denne rollen har til hensikt å gjøre passordgjennoppretting og administrasjon av brukeretilgang tilgjengelig for bedriften som benytter seg av programvaren. Dette resulterer i redusert press på Norkart kundestøtte.

AAD

Dette er en brukerdatabasen Norkart ID skal benytte for å lagre brukerdata og tilgangsstyring. Den skal også kunne vedlikeholdes uten å gå via Norkart ID om dette skulle være ønskelig. Ettersom all brukerdata lagres i AAD må alle roller som har mulighet til å endre brukerdata også kunne skrive eller oppdatere data igjennom Norkart ID. AAD brukes for å hente ut oppslag for hver eneste forespørsel om autentisering og autorisasjon av brukere mot applikasjoner.

Applikasjoner

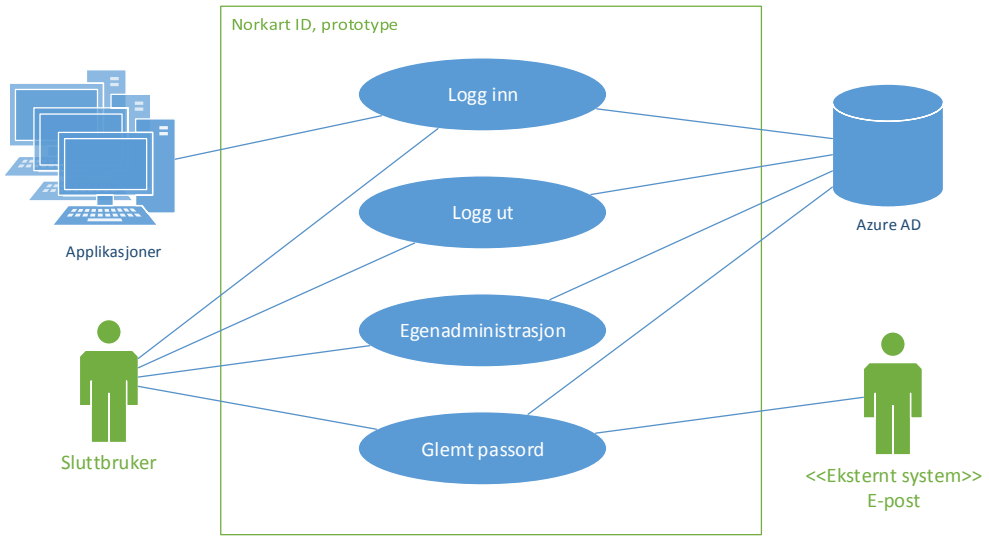
Dette er det Norkart ID skal autorisere brukere for. Norkart ID må kjenne til hvor og hvordan brukerne som autoriseres skal kontakte applikasjonene de kan bruke. Etter at en bruker logger seg inn med Norkart ID vil bruker få informasjonen som trengs for å knytte seg til applikasjonene.

Sluttbruker

Dette er brukere av Norkart ID som autentisering og autoriseringtjeneste og logger inn for å få tilgang til tjenestene Norkart tilbyr.

8.2.2 Use case diagram for prototype

For å få et dypere innblikk av hva selve prototypen skal inneholde er det laget et use case diagram (jfr figur 20), som inneholder roller og funksjoner basert på innloggingsmekanismer og egenadministrasjon i overordnet use case (jfr figur 19). Innlogggingmekanismer er delt opp i tre funksjoner og e-post er tatt med som eksternt system.



Figur 20: Use Case Diagram for prototypen av Norkart ID

Applikasjoner

For at bruker skal kunne logges inn må Norkart ID vite hvilken applikasjon bruker vil logge inn på. Derfor trenger applikasjoner (jfr figur 20) kun tilgang til logg inn funksjonalitet.

Sluttbruker

Bruker som skal kunne benytte all funksjonalitet i prototypen.

AAD

Ettersom bruker må autentiseres for å logge inn, logge ut og bruke glemt passord funksjonalitet må Norkart ID ha tilgang til AAD for å utføre disse funksjonene. Norkart ID må også ha tilgang til AAD for å endre brukerdata under egenadministrasjon.

E-post

For at Norkart ID skal kunne gjennomføre glemt passord funksjonalitet trenger den tilgang til en e-post server. Om en bruker kjører glemt passord funksjonen skal Norkart ID sende en resett passord link på e-post til brukeren.

8.2.3 Høynivå use case beskrivelser

Fire høynivå use case beskrivelser er definert for å beskrive funksjonalitet i løsningen.

Use case:	Logg inn
Aktører:	Sluttbruker
Mål:	Aktør skal kunne få tilgang til ønsket applikasjon.
Beskrivelse:	Når aktør skriver inn brukernavn og passord skal aktøren autentiseres og få tilgang til applikasjonen den ønsker å jobbe på.

Use case:	Logg ut
Aktører:	Sluttbruker
Mål:	Aktør skal kunne få logget ut av ønsket applikasjon og fra alle andre tjenester
Beskrivelse:	Når aktør klikker logg ut skal aktøren bli utlogget i alle systemer, både web, mobil og applikasjoner. En single sign out metode for å slippe å måtte logge ut individuelt i alle systemer.

Use case:	Glemt passord
Aktører:	Sluttbruker
Mål:	Aktør skal kunne få mulighet til resette passord og få logget inn igjen.
Beskrivelse:	Aktør skal kunne utføre glemt passord funksjonalitet for å få tilgang til brukerprofil uten å måtte kontakte kundestøtte. Det skal resultere i at det mottas en e-post med en resettlink som leder direkte til hvor det skal opprettes nytt passord.

Use case:	Egenadministrasjon
Aktører:	Sluttbruker
Mål:	Aktør skal kunne endre på registrerte opplysning inne på sin brukerprofil.
Beskrivelse:	Når aktør er autentisert skal det kunne endres på brukerdata. Ved å gjøre dette vil aktør få tilgang til e-post, telefon og generelle opplysninger som er registrert. Aktøren har mulighet til å endre de attributtene som kan endres og dette vil være en egen nettside som krever autentisering.

8.2.4 User Stories

For å få en dypere forståelse av hva slags funksjonalitet som trengs for å overholde kravene til oppgaven er det utarbeidet user stories som viser hva de forskjellige brukerne av applikasjonen skal kunne å utføre (jfr vedlegg H). De er basert ut i fra et use case diagram for prototypen (jfr figur 20), og oppgavens operasjonelle krav (jfr delkapittel 8.3).

8.3 Operasjonelle krav til løsning

Dette er krav som brukes for å beskrive kvaliteten på systemet og dette kan være i form av standarder som benyttes, målinger som skal være innenfor en gitt grense og kostnader.

8.3.1 Ytelse

- Løsningen skal som minimum takle 10 000 brukere innlogget samtidig.
- Løsningen skal håndtere pålogging av 100 brukere i minuttet.
- Løsningen skal bygges for å være skalerbar.
- Det skal ta mindre enn et sekund å autentisere en bruker for de systemene brukeren er autorisert for å bruke.
- Om belastning på Norkart ID serversystemet skulle bli så høy at overordnede krav til innloggingtid ikke overholdes vil operasjoner og forespørsler fra brukere behandles etter FIFO-kø prinsippet.

8.3.2 Implementasjon

- Norkart ID skal fungere som en autentiserings- og autoriseringsledd mellom brukerne og tjenestene Norkart tilbyr.
- Løsningen skal kjøres på en Microsoft Windows Server 2012 eller nyere.
- Løsningen skal benytte AAD som brukerdatabase.
- Innloggingsløsningen skal baseres på OIDC biblioteker.
- Norkart ID skal designes for eksterne brukere.

8.3.3 Standarder

- Løsningen skal tilfredstille kravene til OIDC standardene.
- Løsningen skal igjennom bruk av OIDC tilfredstille sikkerhetsmekanismene gitt ved å bruke OAuth 2.0.
- All kommentering og dokumentasjon av kildekode skal gjøres i henhold til normer gitt for programmeringspråkene som brukes i prosjektet.

8.3.4 Pålitelighet

- Påliteligheten til Azure skyplattformen medregnes ikke i dette prosjektet.
- AAD skal håndteres slik at den aldri skal trenge å tas ned.
- Systemet skal designes for å ha en oppetid på minimum 99,9 % som resulterer i 10 minutter nedetid i uken.
- Norkart ID serveren, OIDC og AAD skal kunne vedlikeholdes individuelt.
- Systemet stiller ingen krav til å kunne oppdateres mens det kjører.
- Kundestøtte for Norkart ID vil implementeres hos eksisterende kunderstøtte.

8.3.5 Brukervennlighet

- Løsningen skal følge regler for universell utforming (jfr delkapittel 8.3.6).
- Bruker skal slippe å skrive inn samme påloggininformasjon mot alle applikasjoner fra Norkart og kun logge inn en gang.
- Bruker skal selv kunne administrere brukerprofil og passord.
- GUI skal være så intuitivt at det tar mindre enn to sekunder å skjønne hvor brukerid felt, passord felt og innloggingknapp er.
- GUI på løsningen skal være skrevet på norsk.
- GUI designet skal skape gjenkjennelighet til Norkart.
- Fremdriftsindikator skal brukes der det er hensiktsmessig for å gi brukeren tilbakemelding.
- Når bruker lager passord skal det gis indikasjon på om passordkravene tilfredstilles.

8.3.6 Lover og regler

- Systemet skal håndtere data i samsvar med lov for Norsk personvern
- Systemet skal følge forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger

8.3.7 Intraoperabilitet

- Systemets API skal kunne kommunisere over standard http protokoll.
- Systemets API skal støtte autentisering, glemt passord funksjonalitet og fornying av økt.
- GUI på løsningen skal ha høy responsivitet.
- Systemet skal ha støtte for SSO på Microsoft Windows 8 (eller nyere Windows systemer).

8.3.8 Sikkerhet og autentiseringskrav

- Brukerdata løsningen trenger: Fullt navn, selskap, mail, passord, mobil, sist innlogget og gjeldende autentiserte enheter.
- All lagring av brukerdata skal beskyttes i forhold til trusselbilde.
- Ingen passord skal sendes eller lagres i klartekst.
- Systemet skal følge generelle normer innenfor autentisering.
- Minimumskrav for passordlengde er åtte tegn.
- Minimumskrav for passordkompleksitet er minimum en stor og en liten bokstav, og minimum et tall.
- Etter fem feilede innloggingforsøk mot en bruker id skal det legges inn ventetid på et minutt før det tillattes nytt innloggingforsøk.
- Ved feil passord eller brukernavn skal det kun stå at innlogging feilet.
- Ved glemt passord skal det sendes link til bruker for generering av nytt passord. Denne linken skal kun være gyldig i 20 minutter.
- En sesjon er kun gyldig seks timer før den må fornyes.
- En bruker som logger inn via en nettleser autentiseres for alle web applikasjoner brukeren har tilgang til i den nettleseren.
- En bruker som logger inn i en mobil applikasjon autentiseres kun til denne applikasjonen
- En bruker som logger inn i en desktop applikasjon autentiseres kun til denne applikasjonen.

8.3.9 Klientkrav

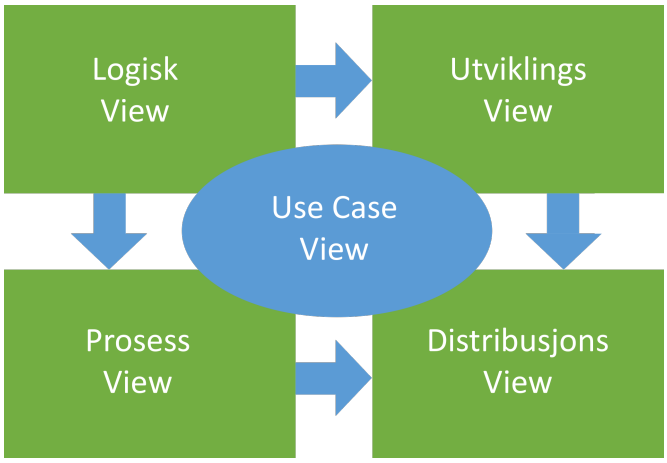
- Det kreves at klienten har tilgang på en nettleser som leser HTML5, CSS3 og JavaScript.
- Klienten må gi tilgang til cookies for å bruke SSO funksjonaliteten i nettleser.
- Klienten må ha skriverettigheter i maskinens register for å bruke SSO funksjonaliteten på skrivebordsapplikasjoner.
- Klienten må være tilkoblet Internett eller i nettverk med Norkart ID serveren.
- Klienten må ha en nettverkshastighet tilsvarende 0,4 Mbit eller høyere for garantert stabil tilkobling til tjenesten (stabil Edge eller høyere tilkobling).

9 Arkitektur og design av IdentityServer3

Prosjektgruppen begynte prosjektet med å se på mulig implementasjon av en løsning basert på rammeverket IdentityServer3. Dette kapittelet beskriver hvordan en implementasjon av autentiseringssystemet kunne sett ut om dette var basert på IdentityServer3. Alle figurene er utarbeidet av prosjektgruppen.

Arkitekturvalg

For å få et arkitekturisk overblikk over systemet brukes SAD fra RUP med Philippe Kruchten's 4+1 view modell (jfr figur 21). 4+1 tar utgangspunkt i at det overordnede use case diagrammet (jfr figur 19) er definert. Det overordnede use case diagrammet regnes som +1 av de 4+1 og befinner seg i midten av modellen (jfr figur 21). De fire views'ene som ligger rundt scenarioene i use caset har til hensikt å illustrere ulike synsvinkler (jfr delkapittel 9.1.1 Logge inn).



Figur 21: 4+1 illustrasjonsskisse

9.1 Use case View

Det er laget fire extended use caser: logg inn, logg ut, glemt passord og egenadministrasjon. Ved å gjøre dette fikk gruppen større innblikk i det som ble vurdert som den viktigste funksjonaliteten.

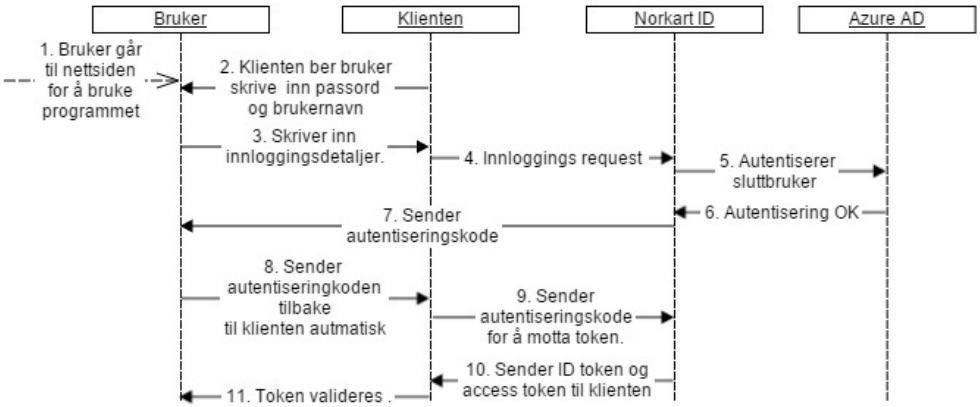
9.1.1 Logge inn

Use case: Logg inn på en webtjeneste
Aktør: Sluttbruker
Mål: Aktør skal kunne få tilgang til ønsket applikasjon.
Beskrivelse: Når aktør skriver inn brukernavn og passord skal bruker autentiseres og får tilgang til applikasjonen den ønsker å jobbe på.
Pre-betingelser: Aktør finnes allerede i systemet.
Post betingelser: Autentisert i alle webtjenester via Norkart ID.

Spesielle krav: Har tilgang til Norkart ID via Internett.
Detaljert hendelsesforløp:

Brukerhandling
1. Use casen begynner når aktøren går til nettsiden for å bruke programmet.
2. Klient ber aktør om å logge inn med brukernavn og passord.
3. Aktør skriver inn innloggingsdetaljer.
4. Klient sender innloggingspørring til OIDC
8. Aktør sender autentiseringkoden til OIDC automatisk fra klienten.
9. Klient sender inn autentiseringkode for å motta token.
11. Token valideres i klienten.

Systemrespons
5. OIDC autentiserer aktør mot AAD.
6. AAD autentiserer aktør.
7. Norkart ID sender autentiseringkode til aktør.
10. OIDC sender en ID token og Access token til klienten.



Figur 22: Sekvensdiagram 'logge inn'

Figur 22, viser sekvensdiagrammet for når en aktør logger inn, og hvordan systemet kommer til å håndtere en slik spørring. Dette er en litt forenklet representasjon av hva som står under detaljert hendelsesforløp.

Nedenfor vises det ulike hendelsesforløp der det er brukerfeil eller systemfeil.

Brukerfeil

- Aktør skriver feil passord
  Aktør bes om å skrive brukernavn og passord på nytt. Beskjed til aktør skal være "Innlogging feilet".
- Aktør skriver feil brukernavn
  Aktør bes om å skrive inn brukernavn og passord på nytt. Beskjed til aktør skal være: "Innlogging feilet".

- Gjentatt forsøk på innlogging med ugyldige opplysninger
  Aktør bes etter fem forsøk om å vente ett minutt før det prøves igjen. Skulle innlogging feile igjen vil det gå 20 minutter til neste gang aktør kan prøve.

Systemfeil

- Lang innloggingtid
  Om innlogging skulle ta mer enn to sekunder skal det gis beskjed til aktør at det tar unormalt lang tid å logge inn. Skulle det feile bes aktør om å prøve en gang til. Om det skulle feile enda en gang bes aktør om å ta kontakt med kundestøtte, enten lokalt eller eksternt.
- Innlogging ikke mulig å gjennomføre
  Skulle innlogging feile bes aktør om å prøve på nytt i første omgang. Om dette ikke løser problemet bes aktør om å ta kontakt med kundestøtte, enten lokalt eller eksternt.

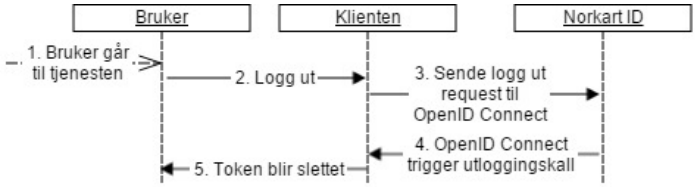
9.1.2 Logge ut

Use case: Logg ut via en webtjeneste
Aktør: Sluttbruker
Mål: Aktør skal kunne få logget ut av ønsket applikasjon og fra alle andre tjenester.
Beskrivelse: Når aktør klikker logg ut skal aktøren bli utlogget i alle systemer, både web, mobil og applikasjoner. En single sign out metode for å slippe å måtte logge ut individuelt i alle systemer.
Pre-betingelser: Er allerede innlogget.
Post betingelser: Blir logget ut i alle tjenester og sletter token slik at aktør er ute av alle webtjenester
Detaljert hendelsesforløp:

Brukerhandling
1. Use casen begynner når aktøren går til tjenesten for å logge ut.
2. Aktør klikker logg ut.
3. Klient sender spørringen til OIDC.
5. Token blir slettet og viser i klienten at man har blitt logget ut.

Systemrespons
4. OIDC mottar spørring og trigger utloggingskall til alle tjenester

Aktør kommuniserer med Norkart ID igjennom en klient som viser hvordan man skal gå frem for å kunne logge ut (jfr figur 23).



Figur 23: Sekvensdiagram 'logg ut'

Brukerfeil

- Aktør logger ikke ut før man forlater arbeidsstasjonen
  Aktør bli automatisk logget ut etter et tidsintervall på seks timer og det må autentiseres på nytt.

Systemfeil

- Aktør blir ikke logget ut av alle webtjenestene
  Aktør får beskjed om at det ikke var mulig å gjennomføre forespørsel og bes om å gå inn direkte på Norkart ID tjenesten å prøve på nytt. Om dette ikke skulle fungere normalt skal det opprettes et hendelsesflagg om at det er problem med utlogging.

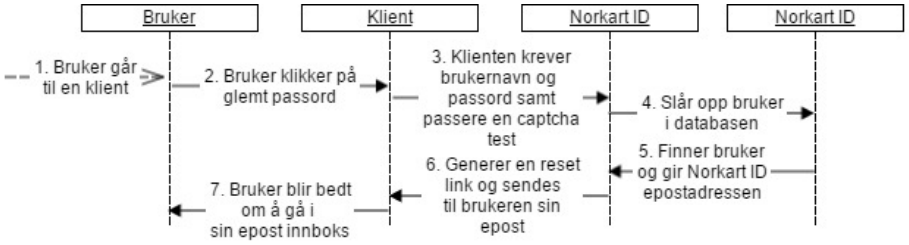
9.1.3 Glemt passord

Use case: Glemt passord på websiden
Aktør: Sluttbruker
Mål: Aktør skal kunne få mulighet til sette nytt passord og få logget inn igjen.
Beskrivelse: Aktør skal kunne hente ut passord selv til sin brukerprofil uten å måtte kontakte kundeservice. Her er tanken at det skal mottas en e-post med en resettlink hvor aktør kommer direkte til oppretting av nytt passord.
Pre-betingelser: Aktør er allerede registrert med e-post.
Detaljert hendelsesforløp:

Brukerhandling
1. Use casen begynner når aktøren går til nettsiden for å et få nytt passord.
2. Aktør klikker på glemt passord.
3. Aktør skriver inn e-post og bekrefter en CAPTCHA test.
6. Aktør blir bedt om å henvende seg til sin e-post innboks for å forsette.

Systemrespons
4. Systemet slår opp aktør i databasen.
5. Det genereres en resettlink og den blir sendt til registrert e-post.

Nedenfor vises det ulike hendelsesforløp der det er brukerfeil eller systemfeil.



Figur 24: Sekvensdiagram 'glemt passord'

Brukerfeil

- Aktør oppgir ugyldig format på e-post
  Det gis en beskjed til aktør at det er brukt feil format på e-post som er skrevet inn og presenterer a@b.c som format.
- Aktør feiler på CAPCHA testen
  Oppdaterer siden, si at sjekken feilet og ber brukeren prøve på nytt.

Systemfeil

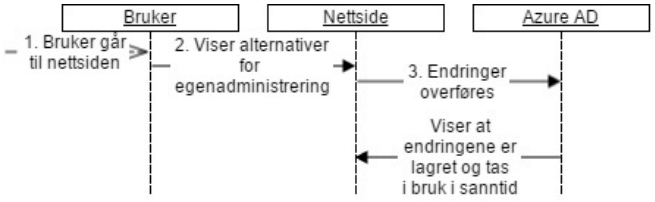
- E-post blir ikke sendt ut
  Det gjøres ingen sjekk av systemet annet enn at linken som blir sendt ut er gyldig i 20 minutter. Om den ikke kommer fram til bruker får bruker selv prøve på nytt.

9.1.4 Egenadministrasjon

Use case: Egenadministrasjon av brukerprofil
Aktør: Sluttbruker
Mål: Aktør skal kunne endre på registrerte opplysning på sin brukerprofil.
Beskrivelse: Når aktør er autentisert skal det kunne gjøres endringer på sin egen brukerdata. Det er her aktør vil få tilgang til epost, telefon, generelle opplysninger som er registrert. Aktøren skal kunne endre de attributtene som går an. Dette er en egen nettside som aktør får tilgang til etter autentisering.
Pre-betingelser: Er allerede innlogget i systemet
Post betingelser: Aktør og brukerprofil tar i bruk endringer i sanntid.
Detaljert hendelsesforløp:

Brukerhandling
1. Use casen begynner når aktøren går til nettsiden for å endre brukerdata.
2. Viser siden for egenadministrasjon.
4. Ved lagring kan aktør forsatt gjøre endringer eller logge ut.

Systemrespons
3. Endringer gjort fra klient blir overført til database og klient.



Figur 25: Sekvensdiagram 'egenadministrasjon'

Denne sees på som viktig for å kunne muliggjør en form for selvadministrering for brukerne som igjen resulterer til mindre trykk på Norkart kundestøtte.

Nedenfor vises det ulike hendelsesforløp der det er brukerfeil eller systemfeil.

Brukerfeil

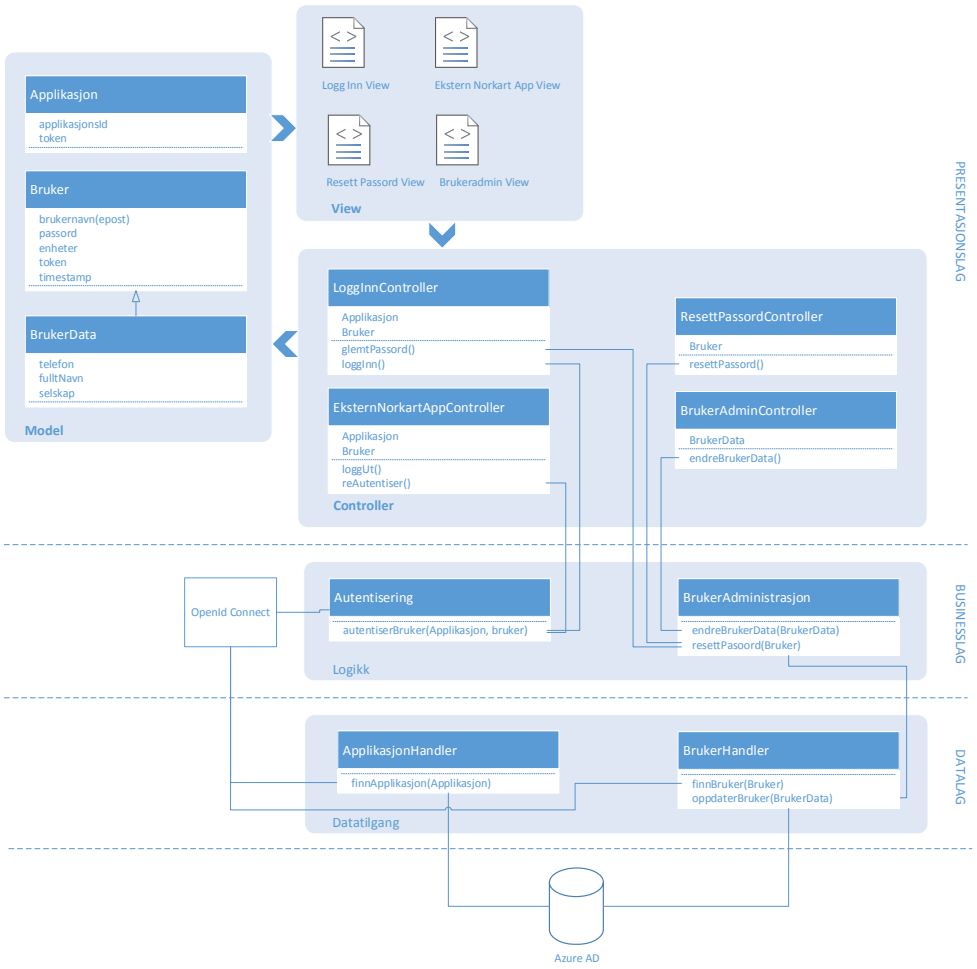
- Ikke innlogget
  Aktør skal bli henvist til innloggingsiden da systemet ikke mottar valid token eller er innlogget fra før.
- Endrer til ugyldig informasjon i brukerprofilen
  Når aktør endrer opplysninger skal disse sjekkes ved hjelp av enten Javascript eller biblioteker for å sjekke at det er riktig format på dataen før den går videre inn i systemet.

Systemfeil

- Om systemet ikke klarer å behandle endringsforespørsel
  Systemet skal gi beskjed tilbake om endringer er OK eller ikke. Ved tilfeller der det ikke er OK bes aktør å prøve på nytt og om feilen vedvarer om å ta kontakt med Norkart.

9.2 Logisk View

I forhold til 4+1 modellen skal det logiske viewet illustrere sluttbrukers perspektiv på Norkart ID. For å øke skalerbarheten og sikkerheten til systemet er det brukt en trelags arkitektur hvor Model View Controller (MVC) er brukt som arkitekturmønster. MVC er brukt for å skape mindre avhengigheter i systemet og redusere kodekompleksiteten. For å få en oversikt over lag og klasser er det utviklet et design klassediagram (jfr figur 26).



Figur 26: Klassediagram

Trelagsarkitektur

Klassediagrammet (jfr figur 26) viser at systemet er delt opp i et trelags system og muliggjør at hvert lag kan oppgraderes eller skiftes ut individuelt. Dette gjør det enklere å vedlikeholde systemet om det blir behov for forandringer i kravene, eller det dukker opp ny teknologi som er hensiktsmessig å bruke. Lagene som er brukt er presentasjonslag, businesslag og datalag. Under står det beskrevet hva de tre lagene har ansvar for og hva de inneholder.

9.2.1 Presentasjonslag

Presentasjonslaget består av GUI og har som hovedmål å videreformidle input fra brukeren til resten av systemet. Det skal også oppdatere GUI når input er behandlet. Det er bygd opp av et MVC pattern med fire views og fire tilhørende controllere. Systemet har i tillegg tre modell klasser.

Views

Viewene viser data til sluttbruker. Hvert view har sin controller som videreformidler input fra bruker nedover i arkitekturen.

- Logg Inn View inneholder en glemt passord link og et logg inn skjema.
- Ekstern Norkart App View er en ekstern Norkart applikasjon som brukere har logget seg inn på via Norkart ID og har en logg ut knapp.
- Resett Passord View inneholder muligheten til å resette passord. Sluttbruker kan kun komme til dette viewet fra en link mottatt i e-post under glemt passord funksjonalitet.
- Egenadmin View inneholder et skjema med brukerdata som brukeren kan endre på og oppdatere.

Controllere

Disse informerer model og view om endringer basert på input fra bruker. Klassediagrammet (jfr figur 26) viser hvilke klasser de forskjellige controllerne tilkaller på forskjellig input.

- Logg inn Controller starter glemt passord funksjonalitet og informerer businesslaget om at bruker vil logge seg inn.
- Ekstern Norkart App Controller setter i gang logg ut funksjonalitet.
- Resett Passord Controller sier i fra til businesslaget at bruker ønsker å resette sitt passord.
- Egenadmin Controller sender ny brukerdata nedover i systemet for å oppdatere bruker.

Modeller

Systemet har tre modeller som brukes til å holde på data.

- Bruker inneholder brukerens autentiseringsdata.
- BrukerData er et barn av Bruker modellen og inneholder brukerens administrative data.
- Applikasjon inneholder applikasjonsdata til bruk i autentiseringsprosessen.

9.2.2 Businesslag

Dette laget har ansvar for logikken i systemet og mottar henvendelser fra presentasjonslaget. Det kommuniserer også med klassene i datalaget for henvendelser mot AAD. Systemet har to hovedklasser som styrer logikken i systemet.

- Autentisering har ansvar for all logikken som må gjøres for at en bruker skal autentiseres mot sine applikasjoner.
- EgenAdministrasjon utøver all logikk som skal til for endring og oppdatering av brukerdata.

OIDC

I tillegg til de to logikk klassene vil også OIDC ligge i businesslaget og brukes for å autentisere brukere.

9.2.3 Datalag

Datalaget har ansvaret for alle henvendelser mot AAD og består av to hovedklasser.

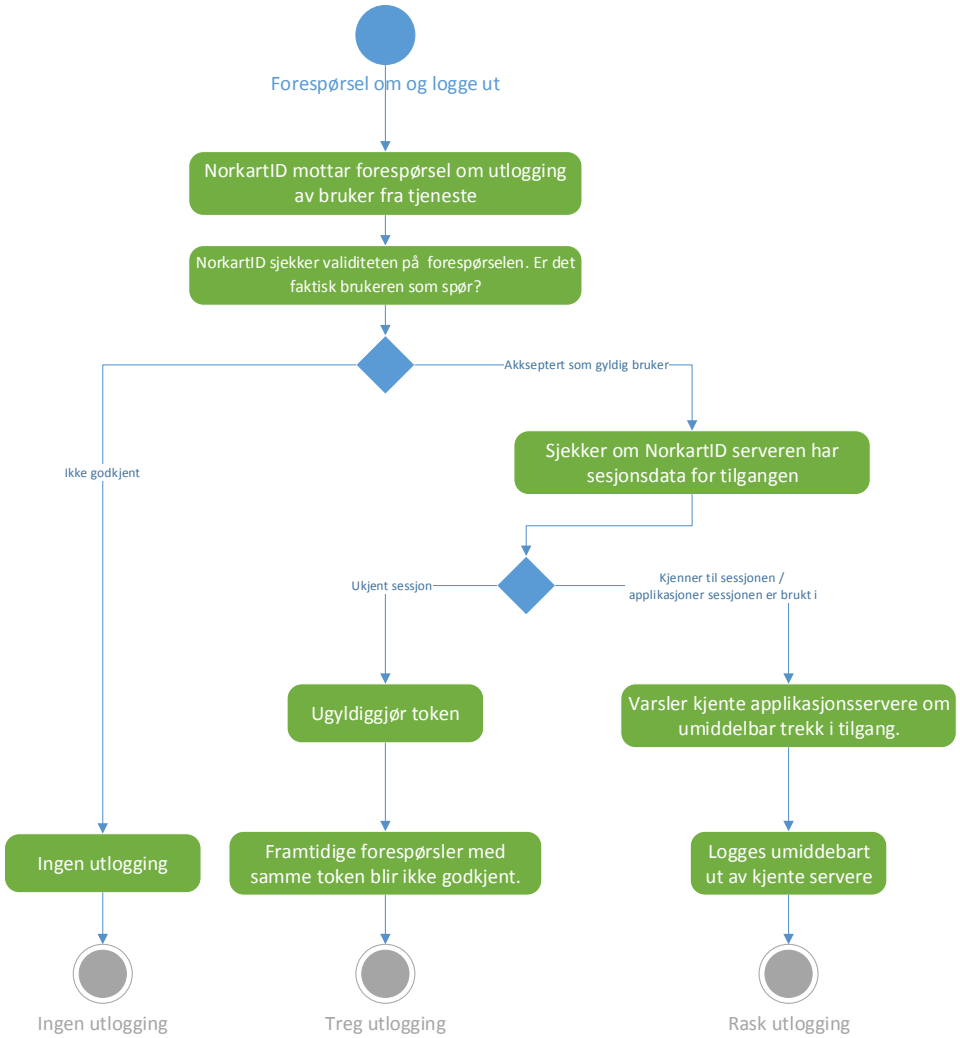
- ApplikasjonHandler sjekker om applikasjoner eksisterer i AAD
- BrukerHandler sjekker om brukere eksisterer og oppdaterer brukerdata i AAD

9.3 Prosess View

Som prosess view ble det valgt å lage activity diagram for de fire overordnede use casene (jfr delkapittel 9.1). Activity diagrammet illustrerer arbeidsflyt, etter hendelser og valg når en oppgave skal gjennomføres. Det ble videre utarbeidet activity diagrammer for de fire kjernefunksjonene prosjektet hovedsaklig skal fokusere på. Begrepet applikasjonsserver brukes for å illustrere serveren som kjører applikasjoner Norkart ID autentiserer og autoriserer brukerne for.

9.3.1 Logge ut

Activity diagrammet “Logge ut” (jfr figur 27) begynner når en bruker sender en forespørsel til Norkart ID serveren om å logge ut. For å kunne logge ut må brukeren bekrefte at det faktisk er brukeren som ønsker å logge ut. Dette gjøres for å beskytte mot at angripere kan kaste ut brukere fra applikasjonene de er logget inn på. Avhengig av måten applikasjoner er koblet opp mot Norkart ID vil utgangsverdiene etter diagrammet ende opp i tre ulike tilstander.



Figur 27: Activity diagram 'logge ut'

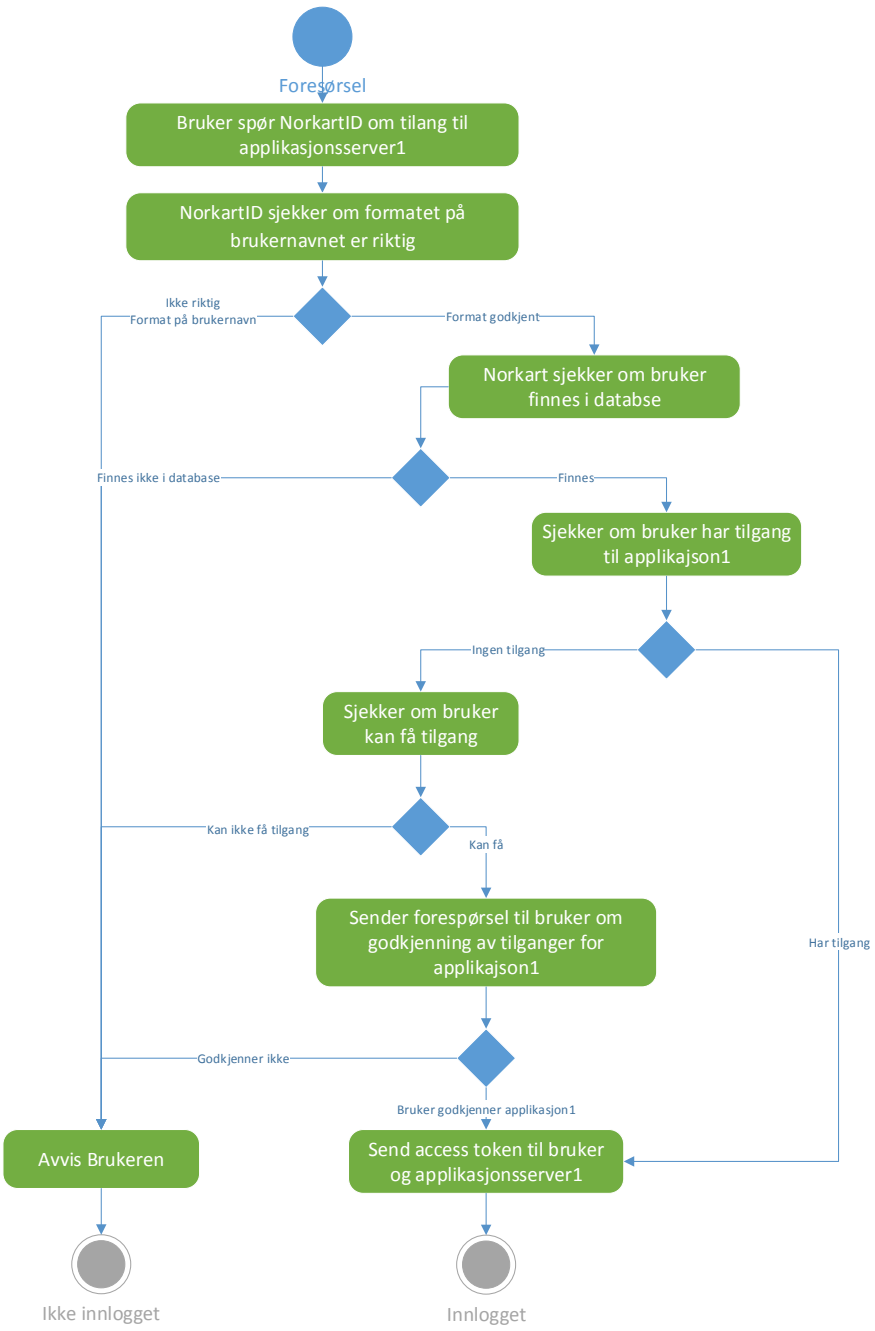
Kommentar

Gruppen ser for oss at det kun er applikasjoner som støtter rask utlogging fra applikasjonserveren som får lov til å dele Access token med andre prosesser. For de applikasjonene som kun støtter treg utlogging, anbefales bruk av egen Access token, uten mulighet for å dele dette tokenet med andre applikasjoner. Dette vil føre til en egen pålogging før bruk, men vil øke sikkerheten for bruker og Norkart.

9.3.2 Logge inn

Activity diagrammet for påloggingsforespørselen (jfr figur 28) begynner inngangsverdien etter at en bruker har sendt inn brukernavn og passord for å logge på en spesifikk tjeneste. Utgangsverdiene fra diagrammet er at brukeren blir logget inn eller avvist. Activity diagrammet tar ikke hensyn til eventuelle sikkerhetsmekanismer som er lagt inn for å hindre brute-force angrep på serveren. Dette blir det likevel implementert mekanismer for og vil trolig bygges inn som en del av de to første hendelsene etter inngangsverdien i

diagrammet.

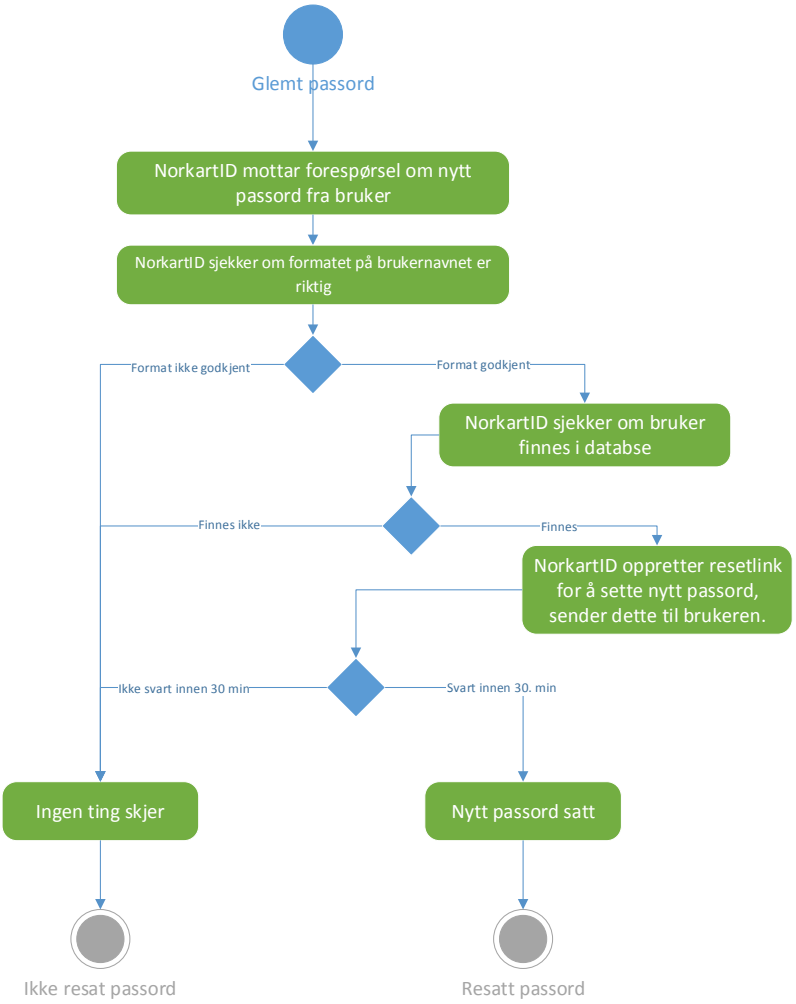


Figur 28: Activity diagram 'logge inn'

9.3.3 Glemt passord

Activity diagrammet "glemt passord" (jfr figur 29) begynner etter at en bruker har skrevet inn en brukerid, bekreftet at det faktisk er en bruker som forsøker å resette passordet, og ikke en datamaskin igjennom en CAPTCHA test. Prosessen innebærer at det sendes

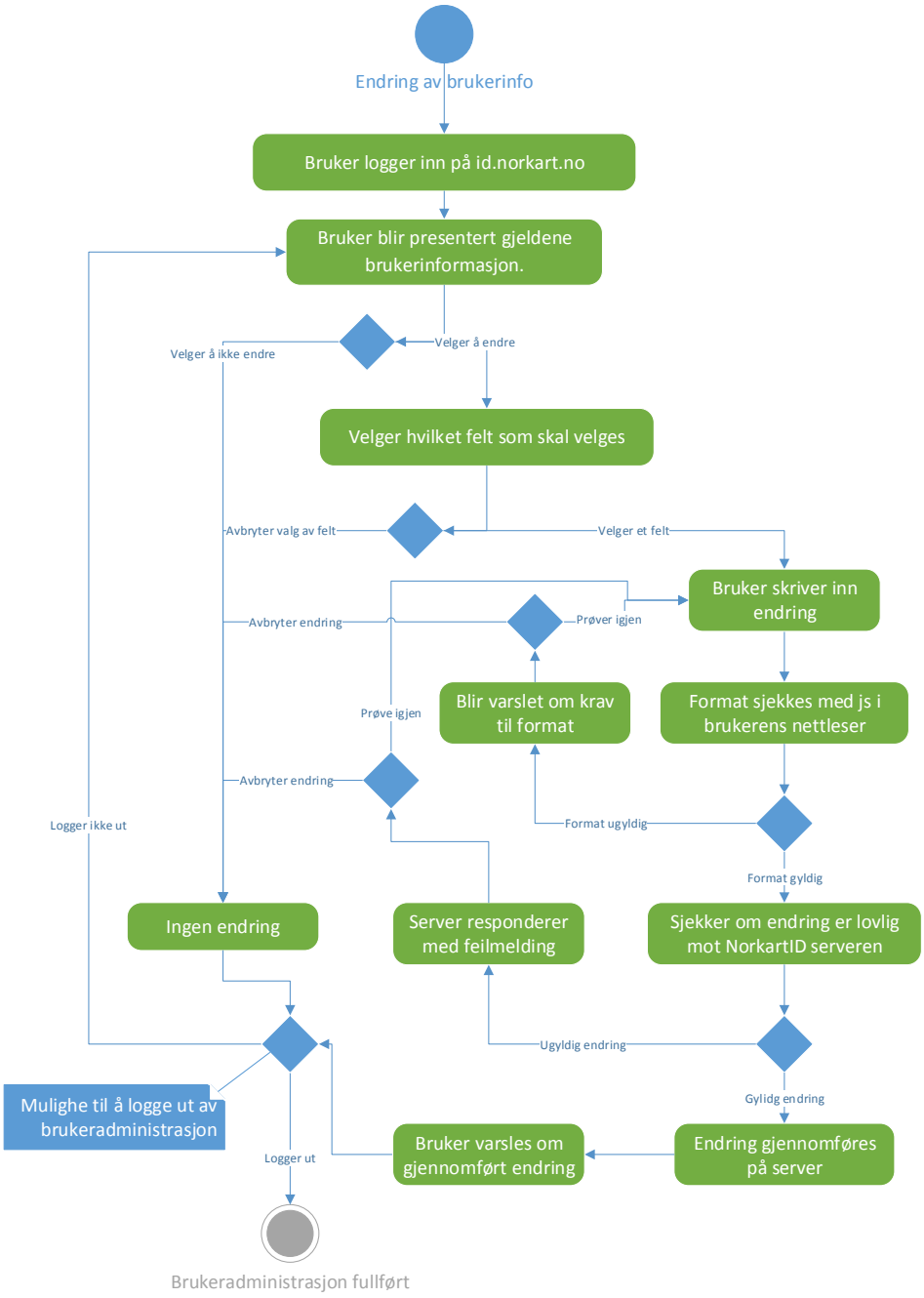
en e-post til bruker som må benyttes innen 20 minutter for at resett av passordet skal godkjennes. Dersom samme bruker klikker på glemt passord flere ganger i løpet av satt tidsintervall, uten å bekrefte på e-post i mellomtiden, vil det kun være den siste e-posten som ble sendt fra serveren som er gyldig. De foregående e-postene fra serveren vil ugyldiggjøres.



Figur 29: Activity diagram 'glemt passord'

9.3.4 Egenadministrasjon

Activity diagrammet Egenadministrasjon (jfr figur 30) illustrerer når en bruker selv kan gjøre endringer av egne brukerprofildata. Inngangsverdien er at bruker skal være innlogget for å endre. Første handling er at bruker logger på id.norkart.no. Gruppen forutsetter at dette fører til en vellykket innlogging. Utgangsverdien er at brukeren, enten bruker har endret brukerprofildata eller ikke, logger ut, eller forlater id.norkart.no.

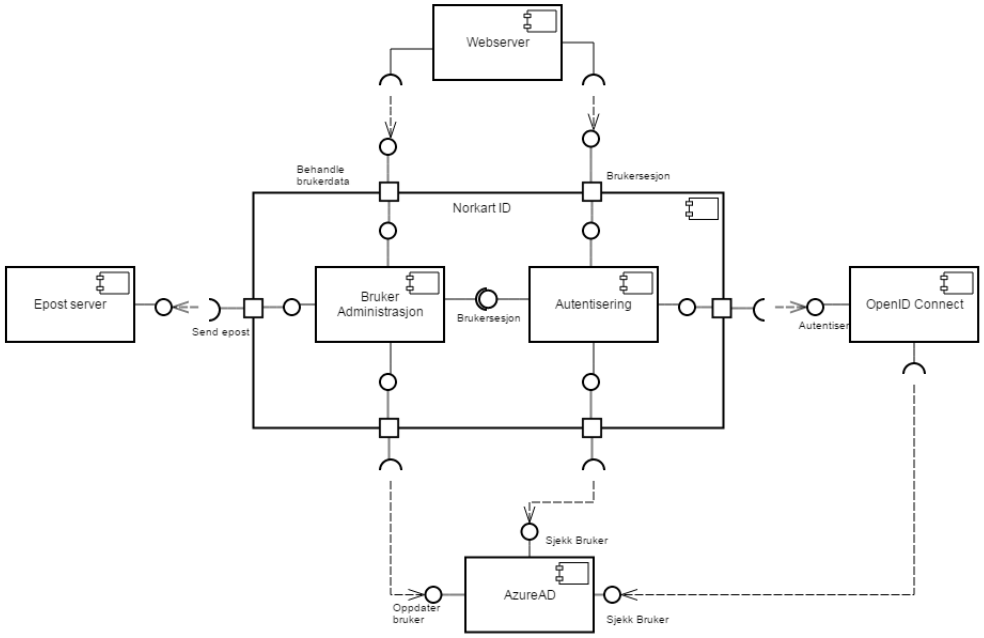


Figur 30: Activity diagram ’egenadministrasjon’

9.4 Utviklings View

Dette viewet illustrerer systemet fra en programmerers sitt perspektiv. Komponent diagrammet for systemet (jfr figur 31) viser at systemet består av flere komponenter som er avhengig av hverandre for å fungere. For å kunne bruke systemet er webserveren avhengig av Norkart ID. Norkart ID vil fungere som hovedkomponent. For å kunne logge inn og autentisere brukere er Norkart ID avhengig av OIDC for autentisering og en AAD for

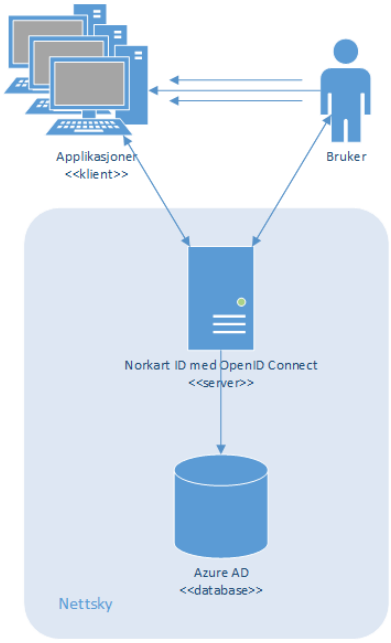
å finne brukere og applikasjonene de har tilgang til. Norkart ID er i tillegg avhengig av en AAD for å administrere brukere. Hvis en bruker har glemt sitt passord må Norkart ID kontakte en e-post server. Norkart ID inneholder i tillegg to komponenter, nemlig Brukeradministrasjon og Autentisering. Brukeradministrasjon er avhengig av at brukeren er autentisert for å kunne utføre sine oppgaver.



Figur 31: Komponent diagram over Norkart ID

9.5 Distribusjons View

Hensikten med distribusjons view er å illustrere hvordan systemet skal settes opp i forhold til hardwareimplementasjon. Ettersom hele Norkart ID er planlagt å kjøre på virtuelle maskiner i Norkart sin nettsky, illustreres bare distribusjons view med tre elementer i diagrammet (jfr figur 32). Det regnes at bruker er en betegnelse på brukere av systemet. Applikasjoner i diagrammet er de Norkart ID skal autentisere og autorisere tilganger for. Den markerte rammen rundt ”Azure AD” og ”Norkart ID med OpenID Connect” skal illustrere at systemet ligger virtuelt på en sky. Når applikasjoner ligger utenfor skyen betyr dette at det stilles ingen krav til at applikasjonene må være en del av den samme skyen, eller kjøre på de samme serverne.



Figur 32: Distribusjons diagram for Norkart ID

10 Avslutning

Dette kapittelet oppsummerer de faglige resultatene i rapporten og trekker fram prosjektgruppens erfaringer relatert til prosessen. Det trekkes også fram anbefalinger for videre arbeid med prosjektet. Kapittelet avsluttes med en konklusjon av prosjektets problemstilling og læringsmål.

10.1 Valg underveis i oppgaven

Valg av løsning (jfr kapittel 4) og konfigurasjon (jfr kapittel 5) er et resultat av vurderinger tatt av gruppen sammen med oppdragsgiver. I et møte med Norkart i begynnelsen av mars (jfr delkapittel 4.4) ble det valgt å fokusere på AAD.

10.2 Drøfting av oppgaven

I dette avsnittet drøftes resultat- og effektmålene for oppgaven (jfr delkapittel 1.2 Prosjektmål). Et av resultatmålene for prosjektet var å kartlegge teknologier og muligheter for å løse oppgaven. Dette målet ble nådd og er en del av leveransen til oppdragsgiver som et eget kapittel, teorikapittelet (jfr 3). Veiledningene (jfr kapittel 6) oppfyller målet om å gi Norkart en introduksjon til bruk av AAD og OIDC. Målet om økt sikkerhet i forbindelse med brukerhåndtering og tilgangsstyring av tjenestene til Norkart løses ved å samle alt i en robust løsning. Denne rapporten når målet om å være et godt grunnlag for Norkart sitt utviklingsprosjekt høsten 2015.

Effektmålene for prosjektet omhandler forenkling av innlogging mot tjenestene til Norkart via en sikrere og mer tidseffektiv løsning. Om Norkart bruker anbefalingene gitt i denne rapporten vil effektmålene nås. Bruk av SSO vil forenkle innloggingen og redusere tid brukt på autentisering for Norkart tjenester.

10.3 Kritikk av oppgaven

Oppgaven som opprinnelig ble gitt av Norkart var vid og hverken prosjektgruppen eller oppdragsgiver hadde innsyn i teknologien som skulle undersøkes. Prosjektgruppen forsto det som at oppgaven i utgangspunkt var å utvikle en middelvare, selv om dette ikke nevnes i oppgaveteksten. Dette førte til begrenset tankesett før undersøkelser av teknologi og systemer rundt autentiseringsløsninger ble gjort.

Som en konsekvens av dette ble det brukt mye tid på å utvikle en kravspesifikasjon (jfr kapittel 8) og arkitektur (jfr kapittel 9 Arkitektur og design av IdentityServer3) for en løsning som skulle utvikles. Hadde oppgaven blitt sett på med et mer overordnet blikk fra starten, mener prosjektgruppen at det ville vært mer tid til å fokusere på valgt løsning. Det kunne vært fordelaktig å bruke iterativ metode tidligere i prosessen og ikke kun i utviklingsfasen. Prosjektet ville da vært mer mottakelig for endringen som skjedde og kunne oppdaget dette tidligere i prosessen. Endringen forklares nærmere i delkapittelet endring av oppgaven (jfr delkapittel 1.3). I begynnelsen av prosessen var det planlagt å utføre et kunnskapsdykk slik at prosjektgruppen kunne sette seg inn i teknologier rundt

autentisering. I ettertid ser gruppen at det burde blitt definert konkrete problemstillinger om hva som skulle undersøkes under kunnskapsdykket. Dette ville trolig resultert i at prosjektgruppen ville hatt et sterkere grunnlag for videre arbeid.

10.4 Veien videre

Prosjektgruppen vil trekke fram disse fokusområdene for videre arbeid med prosjektet.

- Utvikle brukerportal.
- Tilknytning til eksisterende databaser.
- Juridiske aspekt om lagring av persondata i utlandet.

Resultatene av testene prosjektgruppen gjennomførte på MyApps viser at den mangler viktig funksjonalitet fra kravspesifikasjonen (jfr kapittel 2). Dette medfører at det bør kjøres et utviklingsprosjekt ved siden av implementering av AAD hvor det utvikles en brukerportal spesielt for Norkart. Ved å gjøre dette kan de få funksjonaliteten de har behov for.

Prosjektgruppen anser det som nyttig for Norkart å undersøke deler av backend logikken og API'en for databaser i Azure skyen. Etter det gruppen har funnet ut peker Azure Mobile Services og Azure Biztalk seg ut som aktuelle Azure tjenester. Bakgrunnen for dette er at AAD jobber godt med alt som er tilknyttet Azure og både Mobile Services og Biztalk bidrar til dette. Begge tjenestene gjør det mulig å knytte lokale databaser til Azure.

Gruppen anbefaler Norkart å se på juridiske aspektet ved bruk av Azure, ettersom de leverer tjenester til staten og kommuner. Grunnlaget for denne anbefalingen baseres på forskjeller mellom utenlandske og norske lover om lagring av sensitiv data.

10.5 Evaluering av gruppens arbeid

Dette delkapitlet omhandler gruppens prosess, samarbeid og hva prosjektgruppen har lært i løpet av prosjektet.

10.5.1 Organisering

Prosjektgruppen lagde en fremdriftsplan for prosjektarbeidet under forprosjektet (jfr F.1 Arbeidsplan). Mot avslutningen av prosjektet ble det satt opp et Gantt diagram for hvordan perioden faktisk ble (jfr F.2). Dette vil adresseres utover i delkapitlet.

Allerede under forprosjektet brukte gruppen en uke lengre enn planlagt. Det ble tidlig oppdaget at det var mye prosjektgruppen måtte sette seg inn i for å kunne ta gode valg. Prosjektgruppen begynte derfor å lese på teori og mulige løsninger ved prosjektperiodens oppstart. Underveis ble det tydelig at hele prosjektet kunne regnes som et kunnskapsdykk, framfor bare noen uker i begynnelsen av perioden.

Kravspesifikasjon og designdokumentasjon ble ferdig som planlagt, på tross av forsinkelser med forprosjektet. I begynnelsen av utviklingsperioden ble det gjort et funn som i praksis førte til at gruppen gikk bort fra å utvikle en middelvare. Det ble derfor store forskjeller mellom fremdriftsplanen og faktisk tidsbruk.

Etter at endringen ble vedtatt (jfr vedlegg A.1.2 Gjennomgangsmøte) gjennomførte gruppen en kort sprint for å bekrefte påstander og underbygge valget som ble tatt om å fokusere på en eksisterende løsning. Dette ble presentert på et demomøte i midten av mars (jfr vedlegg A.3 Planleggingsmøte). Deretter ble det gjennomført noen lengre sprinter for å komme i gang med datainnsamlingen som måtte gjøres for å kunne løse oppgaven etter den nye problemstillingen.

Prosjektgruppen var bestemt på å bli ferdig med alt innhold til 1/5-2015. Etter mye jobbing de siste ukene i april ble de to siste ukene i prosjektperioden brukt til gjennomlesning og utarbeidelse av avslutningskapittelet og vedlegg.

Prosjektgruppen har lært at det alltid skjer endringer eller andre ting som gjør at estimert tidsplan ikke holder. Gruppen synes det var nyttig at det ble definert faser for å kunne se progresjon i forhold til gjenstående tid.

Ansvarsforhold og roller

I begynnelsen av prosessen fordelte prosjektgruppen administrative roller seg i mellom (jfr delkapittel 1.7.4 Ansvarsforhold og roller). Disse rollene ble overholdt og fungerte bra gjennom prosessen. Utover dette har det vært en flat gruppestruktur, hvor alle har hatt like mye ansvar. Prosjektgruppen ser i ettertid at det kunne vært stilt større krav til hvert enkelt gruppemedlem. Dette for å skape økt eierforhold til oppgavene som skulle gjøres.

Norkart stilte med en produkteier og en veilder. De har opptrådd imøtekommende og villige til å hjelpe prosjektgruppen underveis. Norkart har fulgt opp og kommet med anbefalinger og innspill til hvilken retning prosjektet skal ta og hva de ønsker prosjektgruppen skal fokusere på. Det ble gitt spillerom til å gå så bredt og dypt som prosjektgruppen vurderte som hensiktsmessig.

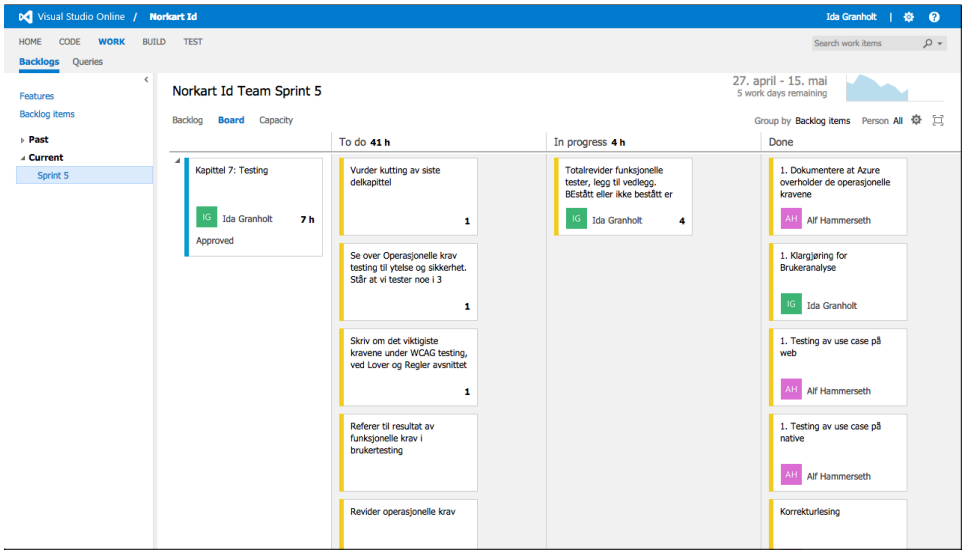
Veilederne gjennom ukentlige møter har fungert som gode sparringspartnere for prosess og arbeidsmetode.

10.5.2 Fordeling av arbeid

I de første sprintene ble oppgaver fordelt til alle medlemmer av gruppen for en uke av gangen. Det viste seg at det var vanskelig å estimere hvor lang tid oppgavene tok. Resultatet av dette var at noen oppgaver ble fort ferdig og det ble vanskelig å ta på seg nye oppgaver ettersom alle oppgavene var fordelt. Dette ble endret i løpet av prosessen ved at medlemmene av gruppen fikk hovedansvar for hele kapitler. Alle oppgavene ble liggende ufordelt og medlemmene kunne påta seg nye oppgaver etter eget ønske. Denne type arbeidsfordeling fungerte bra og medlemmene av gruppen følte eierskap til kapitlene de hadde ansvaret for, samt oppgavene de selv valgte å ta. Dette medførte til et sterkere samarbeid, økt motivasjon og høyere produktivitet.

I gruppereglementet ble det definert at alle skulle jobbe gjennomsnittlig 30 timer i uken, dette ble ikke overholdt. Etter prosjektendringen fikk gruppen et fall i motivasjon, noe

det var tatt høyde for i risikoanalysen. Tiltaket som var utarbeidet innebærte å ha daily standup hver dag og å gjennomføre sosiale samlinger utenfor bacheloroppgaven. Innføringen av daily standup førte til økt motivasjon og ble en arena for å dele utfordringer i arbeidet på en ryddig måte. I begynnelsen av prosessen jobbet prosjektgruppen mye adskilt mellom møter, noe som ikke var spesielt effektivt. Gruppen bestemte derfor tidlig i april at alle skulle sitte sammen og jobbe fra mandag til torsdag hver uke. Figur 33 er et utdrag fra den siste sprinten i prosjektperioden og viser verktøyet gruppen brukte for å fordele oppgaver og estimere tid.



Figur 33: TFS oppgaveoversikt

10.5.3 Prosjekt som arbeidsform

Å løse en oppgave av denne størrelsen for en reell oppdragsgiver har vært interessant og lærerikt for hele prosjektgruppen. Ettersom oppgaven fra Norkart var åpen fikk gruppen være med på å definere problemstillingen og finne kursen i prosjektet. Dette er noe gruppen har hatt nytte av og vil ta med seg videre i arbeidslivet. Gruppesamarbeid og utnyttelse av hverandres forskjeller har vært god læring for hele gruppen. Bruken av hverandres styrker og svakheter har ført til et godt samarbeid og har skapt kvalitet i arbeidet.

I starten av prosessen ble det brukt tid på å sette sammen og planlegge en arbeidsmetode (jfr delkapittelet 1.7). Valget om å bruke en agile arbeidsmetode med iterasjoner har fungert godt for et prosjekt av denne størrelsen. Etter at prosjektet skiftet fokus til fordypning ble det gjort noen endringer i arbeidsmetoden. Gruppen bestemte seg for å gå bort fra MVP (jfr vedlegg A.3.1 Endrings og retrospektivt møte) da ingenting skulle utvikles. Det ble i tillegg avgjort å gå bort fra varierende iterasjonslengde på de to siste sprintene (jfr vedlegg A.4.3 Endrings og retrospektivt møte) slik at det ble lettere å sette arbeidsmål. Det var planlagt og utføre fem møter relatert til hver iterasjon. Etter første møte avgjorde gruppen (jfr vedlegg A.1.3 Retrospektivt møte) at det ble brukt for mye

tid på møter. Retrospektivt- og endringsmøte ble derfor slått sammen til et møte. Selv om arbeidsmetoden ble endret underveis i prosessen har gruppen hatt god bruk av den opprinnelige planen og synes det var nyttig å planlegge arbeidsstruktur og møter så nøye som det ble gjort.

I begynnelsen av prosessen oppfattet gruppen det som tungvint med mange planlagte møter i hver sprint. Det viste seg i etterkant at møtene var bra for gruppen. Møtene gjorde det enkelt å se hva som skjedde underveis, identifisere hvilke beslutninger som ble tatt når og skapte oversikt over arbeid og prosess. Spesielt fikk gruppen mye ut av retrospektivt møte som åpnet for mulighet til å endre arbeidsmetode og gruppesamarbeid for hver sprint. Daily standup møter ble ikke holdt i begynnelsen av prosessen, istedet ble det holdt statusmøter hver mandag. Dette ble endret i sprint fire og fem hvor daily standup ble utført hver mandag til torsdag (jfr vedlegg A.4.3). Denne avgjørelsen økte motivasjonen i gruppen og poenget med disse møtene kom klart frem.

10.6 Konklusjon

Konklusjonen vil presentere resultatet av oppgaven opp mot problemstilling, hva som ble levert til oppdragsgiver, og om prosjektgruppens læringsmål ble holdt.

Prosjektgruppen mener at rapporten svarer på problemstillingen (jfr delkapittel 1.1.3). Det ble definert og utarbeidet teori rundt aktuelle teknologier og systemer. I tillegg ble det produsert en begrunnelse for valg av løsning for å vise prosessen bak valget. Gruppen planla bruk av løsningen ved å sette opp en oversikt over konfigurasjonshensyn og lage veiledninger på hvordan dette gjøres. Til slutt ble løsningen testet og vurdert opp mot oppdragsgivers krav til løsningen.

På bakgrunn av funnene i denne rapporten anbefaler prosjektgruppen at Norkart bruker AAD som autentiseringsløsning mot sine systemer. I tillegg anbefaler gruppen at det utvikles en egen brukerportal.

Prosjektgruppen overleverer følgende til Norkart:

- Teoretisk innføring i AAD, tjenester og protokoller.
- Begrunnelse rundt valg av løsning.
- Konfigurasjon og hensyn som må tas ved bruk av AAD.
- To veiledninger som viser bruk av AAD mot web og native applikasjon.
- Resultat av testing gjennomført mot AAD, brukerportalen MyApps og innloggingsfunksjonalitet.
- Tre nettsider som demonstrerer OIDC og AAD i bruk.
- En Android applikasjon som demonstrerer innlogging ved bruk av AAD.

Etter fullført bacheloroppgave har gruppen lært mye både faglig og om gruppesamarbeid. De faglige læringsmålene til gruppen omhandlet sikkerhet rundt autentiseringsmekanismer, utfordringer rundt brukeradministrasjon og erfaringer rundt bruk av AAD (jfrdelkapittel 1.2.3). Disse målene ble nådd ved at gruppen gjorde dypdykk i flere teknologier og systemer som omhandlet autentisering. Ved å jobbe i Azure lærte gruppen i tillegg

mye om brukeradministrasjon og AAD. Medlemene i gruppen lærte også om arbeidsmetodikker i større prosjekter og å jobbe i projekt med personer med ulik faglig bakgrunn. Prosjektgruppen mener at alle læringsmålene for prosjektet er nådd og sitter igjen med en følelse av å ha lært mye.

Bibliografi

[1] FEIDE. 2015. Virkemåte. https://www.feide.no/virkemate. [Online; lest 08-Mai-2015].

[2] Norsis. 2012. Norsis leksikon - autentisering. https://norsis.no/leksikon/autentisering/. [Online; lest 25-Mars-2015].

[3] Trustwave. 2014. Trustwave global security report 2014. http://www2.trustwave.com/rs/trustwave/images/2014_Trustwave_Global_Security_Report.pdf. [Online; lest 25-Mars-2015].

[4] Norsis. 2012. Norsis leksikon - autorisering. https://norsis.no/leksikon/autorisering/. [Online; lest 25-Mars-2015].

[5] of Guelph, U. 2015. Sso benefits. https://www.uoguelph.ca/ccs/security/internet/single-sign-sso/benefits. [Online; lest 20-Mars-2015].

[6] Authenticationworld. 2006. Sso and ldap authentication. http://www.authenticationworld.com/Single-Sign-On-Authentication/SSOandLDAP.html. [Online; lest 20-Mars-2015].

[7] Facebook. 2015. Technical documentation. https://developers.facebook.com/docs. [Online; lest 12-Mai-2015].

[8] Google. 2015. Set up single sign-on (sso) for google apps accounts. https://support.google.com/a/answer/60224?hl=en. [Online; lest 12-Mai-2015].

[9] saml.xml.org. 2009. Assertions and protocols for the oasis security assertion markup language (saml) v2.0 – errata composite. https://www.oasis-open.org/committees/download.php/35711/sstc-saml-core-errata-2.0-wd-06-diff.pdf. [Online; lest 20-Mars-2015].

[10] saml.xml.org. 2007. History of saml. http://saml.xml.org/history. [Online; lest 20-Mars-2015].

[11] Team, O. . 2014. Announcing support for saml 2.0 federation with office 365. http://blogs.office.com/2014/03/06/announcing-support-for-saml-2-0-federation-with-office-365/. [Online; lest 20-Mars-2015].

[12] Site, O. O. 2015. Introduction to oauth 1.0. http://oauth.net/about/. [Online; lest 23-Mars-2015].

[13] Brail, G. 2010. Top differences between oauth 1.0 and oauth 2.0 for api calls. https://blog.apigee.com/detail/oauth_differences. [Online; lest 23-Mars-2015].

[14] Connect2id. 2015. Openid connect explained. http://connect2id.com/learn/openid-connect. [Online; lest 20-Mars-2015].

[15] OpenID. 2015. Openid connect faq and qas. http://openid.net/connect/faq/. [Online; lest 20-Mars-2015].

[16] Microsoft. 1999. Active directory. https://msdn.microsoft.com/en-us/library/bb742424.aspx. [Online; lest 25-Mars-2015].

[17] Blog, T. O. M. 2010. Windows azure general availability. http://blogs.microsoft.com/blog/2010/02/01/windows-azure-general-availability/. [Online; lest 23-Mars-2015].

[18] Microsoft. 2015. Microsoft trust center. http://azure.microsoft.com/en-us/support/trust-center/privacy/. [Online; lest 26-Mars-2015].

[19] Azure, M. 2015. Azure active directory. http://azure.microsoft.com/nb-no/services/active-directory/. [Online; lest 12-Mai-2015].

[20] Baier, D. 2015. Identityserver3 vnext. http://leastprivilege.com/2015/03/29/identityserver3-vnext/. [Online; lest 25-Mars-2015].

[21] team, I. D. 2015. Creating the simplest oauth2 authorization server, client and api. http://identityserver.github.io/Documentation/docs/overview/simplestOAuth.html. [Online; lest 29-April-2015].

[22] Azure, M. Ukjent. Password policy in azure ad. https://identityserver.github.io/Documentation/docs/overview/mvcGettingStarted.html. [Online; lest 11-Mai-2015].

[23] Hellier, T. 2015. Benefits of being a microsoft partner - registered, certified and certified gold. http://blogs.technet.com/b/tara/archive/2009/06/24/benefits-of-being-a-microsoft-partner-registered-certified-and-certified-gold.aspx. [Online; lest 29-April-2015].

[24] Azure, M. 2015. Basics of registering applications in azure ad. https://msdn.microsoft.com/en-us/library/azure/dn499820.aspx#BKMK_Registering. [Online; lest 15-April-2015].

[25] team, M. A. 28-10-2014. Adding, updating and removing applications. https://msdn.microsoft.com/en-us/library/azure/dn132599.aspx#BKMK_Exposing. [Online; lest 21-April-2015].

[26] Bertocci, V. 2014. Protecting an asp.net webforms app with openid connect and azure ad. http://www.cloudidentity.com/blog/2014/07/24/protecting-an-asp-net-webforms-app-with-openid-connect-and-azure-ad/1. [Online; lest 13-April-2015].

[27] Stuart Kwan, Danny Strockis, V. B. 2015. Webapp-openidconnect-dotnet. https://github.com/AzureADSamples/WebApp-OpenIDConnect-DotNet/commits/master. [Online; lest 13-April-2015].

[28] MSDN. 2014. Announcing azure ad graph api client library 2.0. http://blogs.msdn.com/b/aadgraphteam/archive/2014/12/12/announcing-azure-ad-graph-api-client-library-2-0.aspx. [Online; lest 10-Mai-2015].

[29] Azure, M. 2014. Manage azure ad users. https://msdn.microsoft.com/en-us/library/azure/jj151815.aspx. [Online; lest 10-Mai-2015].

[30] Team, M. A. 2015. Assigning administrator roles in azure ad. https://msdn.microsoft.com/library/azure/dn468213.aspx. [Online; lest 29-April-2015].

[31] team, M. A. A. 2015. Azure ad pricing. http://azure.microsoft.com/nb-no/pricing/details/active-directory/. [Online; lest 17-April-2015].

[32] Kladakis, N. 2014. Cloud identity: Microsoft azure active directory explained. http://channel9.msdn.com/Events/TechEd/Europe/2014/CDP-B210. [Foredrag filmet; 29-Oktober-2014].

[33] Rick Saling, M. A. T. 2015. Get started with mobile services. http://azure.microsoft.com/en-us/documentation/articles/mobile-services-android-get-started/. [Online; lest 29-April-2015].

[34] Rick Saling, M. A. T. 2015. Add authentication to your mobile services app. http://azure.microsoft.com/en-us/documentation/articles/mobile-services-android-get-started-users/. [Online; lest 29-April-2015].

[35] wesmc7777, M. A. T. 2015. Register your apps to use an azure active directory account login. http://azure.microsoft.com/en-us/documentation/articles/mobile-services-how-to-register-active-directory-authentication/. [Online; lest 29-April-2015].

[36] Williams, L. 2006. Testing overview and black-box testing techniques. http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf. [Online; lest 23-April-2015].

[37] Bertocci, V. 2013. Adding a custom domain to your windows azure ad. http://www.cloudidentity.com/blog/2013/04/14/adding-a-custom-domain-to-your-windows-azure-ad/. [Online; lest 11-Mai-2015].

[38] Tronerud, J. 2014. Brukervennlighet. http://brukertest.com/brukeropplevelser/interaksjonsdesign/brukervennlighet. [Online; lest 21-April-2015].

[39] Usabillity.gov. 2015. System usability scale (sus). http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. [Online; lest 01-Mai-2015].

[40] Eli Toftøy-Andersen, J. G. W. 2011. *Praktisk brukertesting*. Cappelen Damm AS.

[41] Sauro, J. 2011. Measuring usability with the system usability scale (sus). http://www.measuringu.com/sus.php. [Online; lest 01-Mai-2015].

[42] Simons, A. 2015. What's the best way to connect to office 365 and azure? http://blogs.technet.com/b/ad/archive/2015/03/26/what-s-the-best-way-to-connect-to-office365-and-azure.aspx. [Online; lest 11-Mai-2015].

[43] difi, D. f. f. o. I. 2014. Krav til nettløsninger(wcag). http://uu.difi.no/veiledning/nettsider/krav-til-nettlosninger/krav-wcag. [Online; lest 27-April-2015].

[44] WebAIM. 2014. Color contrast checker. http://webaim.org/resources/contrastchecker/. [Online; lest 11-Mai-2015].

[45] to Hex, R. 2015. Rgb to hex converter. http://www.rgbtohex.net. [Online; lest 11-Mai-2015].

A Møtereferater

A.1 Møtereferater sprint 1

A.1.1 Planleggingsmøte

24 /2 - 25/2

Saksliste

Del 1:

- Velge PBI for oppkommende sprint
- Opprette tasker til PBI
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Ingen tasks over 8 timer
- Finne the definition of donefor tasker

Del 2:

- Prioritere tasks
- Finne MVP basert på tasker
- Fastsette hvor lenge sprinten skal vare

Referat

Del 1: 24/2-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 5 timer

- Valgt PBI for sprint 1: Logg Inn
- Lagd tasks til PBIen
- Definert hva doneer for hver task
- Estimert tasks

Del 2: 25/2-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 2 timer

- Prioritert tasks
- Finne MVP
- - MVP skal inneholde et proof of concept på å logge inn med OpenID Connect ved bruk av Thinctecture
- - IdentityServer 3 og skal bruke Azure AD som AD
- Sprint 1 varer i 2 uker 24/2 - 9/3
  MVP Fasen: 24/2 og 25/2
  Kode Fasen: 26/2 til 8/3
  Data Fasen: 9/3

A.1.2 Gjennomgangsmøte

5/3-2015
Oppmøte: Ida, Per Christian, Alf, Einar og Håkon
Lengde: 1 time

Saksliste

- Presentere MVP (demo av Azure AD i bruk med OpenID Connect)
- Diskutere funn fra sprint
- Få tilbakemelding fra oppdragsgiver

Referat

- Gruppen har funnet ut at Azure AD inneholder mye av funksjonaliteten Norkart ønsker. Det vil derfor ikke bli nødvendig å programmere en middelvare
- Norkart ønsker derfor at vi skal konsentrere oppgaven rundt anvendelse av Azure AD for Norkart
- Norkart mente prosjektet ville gi dem mer verdi om prosjektgruppen fokuserte på AAD
- Norkart bekreftet for oss at punktet om lokal hosting bortfaller fra oppgaven

Endringsmøte 9/3-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 1 time

Saksliste

- Bestemme om MVP kan fortsettes på, må endres eller om den skal forkastes basert på gjennomgangsmøte med Norkart 5/3 og gruppens funn under sprint 1.

Referat

- MVP ble utviklet som planlagt, uten bruk av Thinctecture IdentityServer 3
- MVP kan fortsettes på slik den er i dag, men kommer til å spille en mye mindre rolle i endelig løsning enn planlagt.
- MVP skal bygges videre på ved å teste om all funksjonalitet fungerer som den skal.

A.1.3 Retrospektivt møte

9/3-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 1 time

Saksliste

- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.
- Bestemme om noe skal endres, legges til eller fjernes til neste sprint

Referat

- Gruppen er usikre på om det er hensiktsmessig å bruke MVP nå som det ikke skal programmeres et produkt.
- Det ble bestemt og holde på variende iterasjoner og bruken av MVP i neste sprint.
- Rollene til alle i gruppen fungerer og skal fortsatt holdes.
- Det er for mange møter i hver sprint, derfor ble det bestemt at endringsmøte og retrospektivt møte slåes sammen til et møte.

- Istedet for å opprette tasker for en hel PBI på planleggingsmøte skal det kun opprettes tasker på planlagte MVP.

A.2 Møtereferater sprint 2

Planleggingsmøte

9/3-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 2 timer
Saksliste

- Velge PBI for oppkommende sprint
- Finne MVP
- Opprette tasker til MVP
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Ingen tasks over 8 timer
- Finne the definition of donefor tasker
- Fastsette hvor lang sprinten skal

Referat

- Valgt PBI for MVP: Proof of concept og dokumentasjon
  Proof of concept skal inneholde:
  - Logg inn
  - Logg ut
  - Resette passord
  - Brukeradm
  - Registrering
- Opprettet tasker for MVP, se TFS
- Sprint 2 skal vare i 6 dager

A.2.1 Endrings og retrospektivt møte

17/3-2015
Oppmøte: Alf, Ida, Per Christian
Lengde: 1 timer

Saksliste

- Bestemme om MVP kan fortsettes på, må endres eller om den skal forkastes basert på demomøte med Norkart 16/3.
- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.
- Bestemme om noe skal endres, legges til eller fjernes til neste sprint.

Referat

- MVP ble ikke helt ferdig, men den kan fortsettes på i neste sprint
- Neste sprint må være lenger
- Vi går brort fra MVP ettersom utviklingen har tatt en annen vending, vi bruker nå heller sprint mål.
- Tasks skal ikke tildeles til personer på planleggingsmøte, men heller taes når folk

skal jobbe med dette.
- Vi skal prøve å legge til så mange estimerte tasks som trengs for å kommendesprint, men det er også lov til å legge til tasks mitt i sprinten.

A.2.2 Gjennomgangsmøte - Demomøte 1

16/3-2015
Oppmøte: Alle
Oppmøte Norkart: Einar, Håkon, Yngve, Grete og Sven
Lengde: 1 time

Saksliste

- Demonstrasjon av proof of consept på
  - Logg inn
  - Logg ut
  - Resette passord
  - Brukeradm
  - Registrering
- Presentasjon av Azure, Azure AD, lagring og linsensløsninger

Referat

- Oppdragsgiver er fornøyd med det vi har funnet ut.
- Oppdragsgiver godkjenner bruk av Azure AD selv om lagring ikke skjer på norsk jord.
- Demonstrasjon:
  - Logg inn, sliter med reautentisering ved bruk av [Authorize]
  - Logg ut, ikke single sign-out
  - Resett passord, fungerer fint, ikke vist på demo
  - Brukeradministrasjon må sjekkes ut mer, se om Azure graph Api kan brukes her
  - Registrering, ikke vist
- Avtalt møte i løpet av uken for å se om oppgaven bør gjøres større ettersom Azure AD løser mye av kravene.

A.3 Møtereferater sprint 2

Planleggingsmøte

9/3-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 2 timer
Saksliste

- Velge PBI for oppkommende sprint
- Finne MVP
- Opprette tasker til MVP
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Ingen tasks over 8 timer
- Finne the definition of donefor tasker

- Fastsette hvor lang sprinten skal

Referat

- Valgt PBI for MVP: Proof of concept og dokumentasjon
  Proof of concept skal inneholde:
  - Logg inn
  - Logg ut
  - Resette passord
  - Brukeradm
  - Registrering
- Opprettet tasker for MVP, se TFS
- Sprint 2 skal vare i 6 dager

A.3.1 Endrings og retrospektivt møte

17/3-2015
Oppmøte: Alf, Ida, Per Christian
Lengde: 1 timer

Saksliste

- Bestemme om MVP kan fortsettes på, må endres eller om den skal forkastes basert på demomøte med Norkart 16/3.
- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.
- Bestemme om noe skal endres, legges til eller fjernes til neste sprint.

Referat

- MVP ble ikke helt ferdig, men den kan fortsettes på i neste sprint
- Neste sprint må være lenger
- Vi går brort fra MVP ettersom utviklingen har tatt en annen vending, vi bruker nå heller sprint mål.
- Tasks skal ikke tildeles til personer på planleggingsmøte, men heller taes når folk skal jobbe med dette.
- Vi skal prøve å legge til så mange estimerte tasks som trengs for å kommendesprint, men det er også lov til å legge til tasks mitt i sprinten.

A.3.2 Gjennomgangsmøte - Demomøte 1

16/3-2015
Oppmøte: Alle
Oppmøte Norkart: Einar, Håkon, Yngve, Grete og Sven
Lengde: 1 time

Saksliste

- Demonstrasjon av proof of consept på
  - Logg inn
  - Logg ut
  - Resette passord
  - Brukeradm
  - Registrering

- Presentasjon av Azure, Azure AD, lagring og linsensløsninger

Referat

- Oppdragsgiver er fornøyd med det vi har funnet ut.
- Oppdragsgiver godkjenner bruk av Azure AD selv om lagring ikke skjer på norsk jord.
- Demonstrasjon:
    - Logg inn, sliter med reautentisering ved bruk av [Authorize]
    - Logg ut, ikke single sign-out
    - Resett passord, fungerer fint, ikke vist på demo
    - Brukeradministrasjon må sjekkes ut mer, se om Azure graph Api kan brukes her
    - Registrering, ikke vist
- Avtalt møte i løpet av uken for å se om oppgaven bør gjøres større ettersom Azure AD løser mye av kravene.

A.4 Møtereferater sprint 3

A.4.1 Planleggingsmøte

17/3-2015
Oppmøte: Ida, Alf, Per Christian
Lengde: 3 timer

Saksliste

- Finne ut hva som faktisk skal leveres ved endt utviklingstid etter funnet om Azure AD.
- Finne sprint mål
- Opprette tasker til MVP
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Finne the definition of donefor tasker
- Fastsette hvor lenge sprinten skal være med

Referat

- Hva skal leveres etter endt utvikling (dette utgjør hoveddelen av rapporten):
    - Teori - En oversikt over hovedteori som må kunnes for å skjønne resten av rapporten
    - Valg og hvorfor (teknologi)
    - Implementasjon (hvordan)
    - Hoved: Grupper og forskjellige plattformer
    - Testing (Svare på kravspek)
    - Drøfting av testing (Drøfte testresulttater i forhold til kravspek)
- Sprint mål
    - Kapittel 3 Teori - done = skrevet det så godt at det er klart for rapport
    - Kapittel 4 Valg - done = samme
    - Kapittel 5 Implementasjon

        - Grupper og roller - research single/multi tentant done - funnet ut hvem vi skal ha
        - Legg til applikasjon i Azure - done - når forklart dette
        - Web browser mot web app -> Web applikasjon - done - skrevet om hvordan dette ble implementert
        - Liste av Einar av applikasjoner/plattformer
        - Resett passord - done - dokumentere hvordan resette passord
        - Endring av brukerprofil - research - done når vet om hva som kan endres i myapps
    - Kapittel 6 Testing
        - Testing av implentasjon av web app - done når testing er planlagt, gjennomført og diskutert.
- Sprint 3 skal vare fra 17.3 - 8.4

A.4.2 Gjennomgangsmøte - Demomøte 2

9/4-2015
Oppmøte: Alle
Oppmøte Norkart: Einar og Håkon
Lengde: 1 time

Saksliste

- Presentere MVP (demo av Azure AD i bruk med OpenID Connect)
- Diskutere funn fra sprint
- Få tilbakemelding fra oppdragsgiver

Referat

- Gruppen har funnet ut at Azure AD inneholder mye av funksjonaliteten Norkart ønsker. Det vil derfor ikke bli nødvendig å programmere en middelvare
- Norkart ønsker derfor at vi skal konsentrere oppgaven rundt anvendelse av Azure AD for Norkart
- Norkart bekreftet for oss at punktet om lokal hosting bortfaller fra oppgaven

A.4.3 Endrings og retrospektivt møte

9/4-2015
Oppmøte: Alle
Lengde: 1 timer

Saksliste

- Status for sprintmål og eventuelle endringer i rapporten
- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.
- Bestemme om noe skal endres, legges til eller fjernes til neste sprint.

Referat

- Vi kom i mål med sprintmålene for kapittel 3 og 4. I kapittel 5 ble de fleste målene nådd utenom brukeradministrasjon. Kapittel 6 ble målene ferdig.
- Til neste sprint skal det legges til rette for å gå tilbake til det man har skrevet og se over igjen.

- Gruppa har sliti med motivasjon, dette må gjøres noe med
- Gruppa må forvente mer av hverandre
- De neste sprintene kjøres med fast iterasjon på 14 dager
- I neste sprint skal vi være strenge på å ha daily stand up møter hver dag, dette skal gjøres for å skape motivasjon til å jobbe mer og gjøre at gruppa kommuniserer mer med hverandre. Klokka 10 hver dag fra mandag til torsdag.
- Møtes mer på skolen
- Gruppa skal bruke risikoanalysen og finne på noe sosialt sammen.

A.5 Møtereferater sprint 4

A.5.1 Planleggingsmøte

9/3-2015
Oppmøte: Alle
Lengde: 3 timer

Saksliste

- Finne sprint mål
- Hente tasks fra forrige sprint?
- Opprette tasker for sprint
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Ingen tasks over 8 timer
- Finne the definition of donefor tasker
- Fastsette hvor lenge sprinten skal vare

Referat

- Sprint mål
    - Strukturere det som allerede er laget
    - Korrekturlese det som allerede er laget
    - Omstrukturere kapittel 1 for å forklare oppgaven på de første sidene i rapporten
    - Endre kravspesifikasjonen i henhold til nye endringer
    - Skrive om utfordringer de må ta hensyn til og kommer til å møte på.
    - Hvordan har vi valgt å konfigurere azure og mener at Norkart bør gjøre det
    - Forklare og vise autentiseringsløpet og hva slags data som sendes hvor, vise hvilken protokoll som brukes.
    - Hvilke muligheter har Norkart i Azure med tanke på grupper og roller.
    - Ha testmetode klart for testing og eventuelt begynne på noen tester.
    - Testing av use case - dokumentere at Azure overholder de operasjonelle kravene (Testing av operasjonelle krav)
    - Brukeranalyse av Azure scenarier
    - Begynne å skisse opp siste kapittelet
    - Lage en native app for å forstå og forklare konfigurering av native mot Azure AD og teste at dette fungerer.
    - Skrive om proxy
    - Skrive om graph api i henhold til brukeradministrasjon
    - Lage tutorials for å legge til rette for å gi utviklerne i Norkart innsyn i konfi-

gurering av Azure AD.

- Ansvarsfordeling:
  - Per Christian - native app - kapittel 1
  - Alf - Konfigurasjonskapittelet
  - Ida - Testing

- Sprint 4 skal vare fra 9/3-2015 til 27/2-2015

A.5.2 Endrings og retrospektivt møte

27/4-2015
Oppmøte: Alle
Lengde: 30 min

Saksliste

- Status for sprintmål og eventuelle endringer i rapporten
- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.
- Bestemme om noe skal endres, legges til eller fjernes til neste sprint

Referat

- Status for sprint mål:
  - Vi kom ikke i mål med kapittel 6 og 7
  - Vi har brukt mer tid på endring av eksisterende materiale enn planlagt.
  - Målene for resten av kapitlene er nådd
- Gruppesamarbeid:
  - Gruppen synes det har vært veldig fornuftig å ha daily stand up, føler vi har fått mye mer oversikt over hva alle gjør og hvordan vi ligger an.
  - Motivasjonen har økt etterhvert som gruppen begynner å komme igang med ny problemstilling.
- Endringer:
  - Beskrivelsen på taskene bør være mer tydelig, definition of done må være klarere for hver tasks.

A.5.3 Gjennomgangsmøte - Demomøte 3

29/4 - 2015
Oppmøte: Alle
Fra Norkart: Einar, Håkon
Varighet: 1 time

Saksliste

- Statusmøte, hvordan ligger vi an?
- Foreløpige funn og konklusjoner

Referat

- Mail dem om at vi trenger tilbakemelding for å ha i rapporten
- Norkart er fornøyd med det vi har funnet ut, de gleder seg ti å lese rapporten og er nysgjerrig på veiledningene.

- De synes det er fornuftig at vi har et teori kapittel. Einar ble også veldig glad for drøfting kapittelet.

A.6 Møtereferater sprint 5

A.6.1 Planleggingsmøte

27/4-2015
Oppmøte: Alle
Lengde: 3 timer

Saksliste

- Finne sprint mål
- Opprette tasker for sprint
- Hva skal leveres, ikke hvordan vi skal gjøre det
- Estimer taskene i timer
- Ingen tasks over 8 timer
- Finne the definition of donefor tasker

Referat

- Målet for sprint
    - Gjøre + Ferdig med innhold og revisjon av kap 6
    - Gjøre + Ferdig med innhold i kap 9 -> konklusjon av hele rapporten / Broren til kapittel 1":D
    - Planlegge samt legge opp vedlegg
    - Revidering av kapittel 1 etter siste tilbakemelding fra veileder
    - Revidering/korrekturlesing/lesing av hele rapporten
- Opprettet tasker for sprint, estimere timer, finne the definition of done"
- Oprettet tasker for begynnelsen av sprinten. Planla at det blir legget til tasks etterhvert. Opprettet definition of done inne i hver task i TFS

A.6.2 Endrings og retrospektivt møte

15/5-2015
Oppmøte: Alle
Lengde: 1 timer

Saksliste

- Status for sprintmål og eventuelle endringer i rapporten
- Gå gjennom hvordan gruppen har jobbet sammen i forhold til utviklingsmetode og roller.

Referat

- Rapporten er klar til levering.
- Gruppen avsluttet prosjektperioden med samlet høytlesning av rapporten.

A.7 Andre møter med Norkart

Representanter fra Norkart har deltatt på flere av sprintmøtene. Dette er referatene fra møtene som falt utenfor møtene tilhørende sprintene.

A.7.1 Oppstartsmøte - 21/1-2015

Oppmøte: Alle
Fra Norkart: Einar, Håkon og Yngve
Varighet 2 timer

Saksliste

- Del 1
  Alf, Ida og Per Christian presenterer sin definering av oppgaven. Kort diskusjon og beslutning. Vi banker metodikk og diverse rundt arbeidsform om ikke dette allerede er klart.
- Målsetning: Begge parter være enige om hva som skal gjøres, hvem som skal gjøre hvilke oppgaver, hvordan på hvilken måte det skal jobbes, i tillegg til når og hvor.

Referat

- Forprosjekt
  - Autentisering av hele maskinen er svært aktuelt/ønskelig
  - API testing mot android app for å bevise at dette virker
  - Forprosjekt - Tilbakemeldinger
  - Pressisering at dette skal øke sikkerhet
  - Fagområde: autentiseringsserver
  - Ikke testing av serverkjerne
  - Forprosjekt ga et veldig godt bilde av oppgaven
- Gant
  - Ha kortere periode med kunnskapsdykk, kanskje fler
- Utviklingsmodell
  - Vanskelig med tanke på å teste backend
  - Kan teste helt banale ting backend, in/out
  - Eventuelt se på en annen metodikk som passer bedre for backend
  - Fordeler med lean, finner begrensninger tidlig
  - Kan lage to like MVP med forskjellig teknologi

A.7.2 Norkartmøte 2 - 4/2-2015

Oppmøte: Alle
Fra Norkart: Einar, Håkon, Yngve, Sverre
Varighet 1 time

Saksliste

- Presentere første utkast av kravspesifikasjon
- Single sign-on på android? Plattform?
- Windows 7 sikkerhet? Vedlikehold
- Har dere applikasjoner for vanlige brukere som trenger innlogging? Må de kunne registrere seg selv?

- Kan vi bruke mail som brukerid?
- Hva slags brukerinformasjon skal lagres?
- Hva skal bruker kunne endre selv?
- Logger inn på desktop, også være logga inn på nettleser?
- Native app, fungerer kun på en og en app.
- Kommer Norkart ansatte til å bruke NorkartID på samme måte som de eksterne brukerne?
- Hvordan tolker dere trusselbilde?

Referat

- Tilbakemedlinger på kravsepsifikasjon
    - Norkart har behov for admin grensesnitt for proffbrukere (Sverre)
    - Fokuser på nettleser og tabs når det gjelder reautentisering
    - Desktop og mobil kommer langt ned på liste når det gjelder dette
    - Dere bør beskrive hva som er hårete rundt reautentisering på mobil og desktop og hvorfor det er hårete
    - Bonus på å få innsyn i hvorfor mobil og desktop er tricky.
    - Ytelse - er det problemstillinger rundt lastbalansering?
    - Implementasjon - I hovedsak er løsningen for eksterne brukere
    - Sikkerhet - Trusselbilde - ingen kommentarer
    - Brukervennlighet - Sjekk tankekartet til Einar for hva slags informasjon brukeren skal kunne endre selv
    - Skal kunne endre mobil - AD skal ikke inneholde fødselsnummer
    - Autentiseringslengde - kan være varierende for de forskjellige applikasjonene.
- Kan være lurt å skrive litt rundt dette med tanke på informasjonssikkerhet
- Dere bør belyse temaer rundt inn/utlogging med single sign-in og OpenID Connect
- Problemstillinger om OpenID Connect og lastbalansering
- Fokus på eksterne brukere
- Ønsker å kunne sette egen tid for lockou retta mot direkte applikasjoner.
- De vil at vi skal overføre vår kunnskap om OpenID Connect til dem
- Vi skal sende forslag på møteplan til Einar

A.7.3 Norkartmøte 3 - 2/3-2015

Oppmøte: Alle
Fra Norkart: Einar, Håkon, Yngve
Varighet 1 time

Saksliste

- Presentere arkitektur
- Presentere MVP
- Eventuelt

Referat

- Gruppen presenterte arkitektur og første MVP
- Endre på komponentdiagrammet, openID Connect skal være med inn i Norkart ID
- Er IdentityServer 3 eneste valg? Se på muligheter for og kun bruke Azure AD

A.7.4 Norkartmøte 4 - 5/3 - 2015

Telefonmøte
Oppmøte: Alle
Fra Norkart: Einar, Håkon
Varighet: 30 minutter

Saksliste

- Presentere funn om muligheter for kun å bruke Azure AD
- Avklare veien videre

Referat

- Presentert hva vi har funnet ut om Azure AD.
- Norkart ønsker vi nå skal fokusere på anvendelse av Azure AD.
- Norkart bekreftet for oss at punktet om lokal hosting bortfaller dette gjelder også juridiske spørsmål rundt dette.

B Funksjonstester

Skjemaene brukt til funksjonstesting er inspirert av testplaner laget av vår veileder, Frode Haug.

B.1 Tester for innloggingsmekanismer

Test: Logg inn, Web applikasjoner							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er ikke logget inn fra før	Åpner webapp 1	Trykker på sign in	Viderekobles til innloggingssiden	OK	28.04.2015	
2	Er ikke logget inn fra før	Er på innloggingssiden	Gyldig brukernavn og passord	Autentiseres og viderekobles til nettsted	OK	28.04.2015	
3	Er ikke logget inn fra før	Er på innloggingssiden	Ugyldig brukernavn og passord	Får feilmelding om at innlogging feilet	OK	28.04.2015	
4	Er logget inn fra før	Åpner webapp 1	Trykker på logg inn	Får tilgang til nettside	OK	28.04.2015	Trenger ikke å klikke på logg inn
5	Er ikke logget inn fra før	Er på innloggingssiden	Ugyldig brukernavn og passord tastes inn 5 ganger på rad	Får feilmelding om at ikke har tilgang til innlogging før om 1 minutt	IKKE OK	28.04.2015	
6	Er ikke logget inn fra før	Er på innloggingssiden	Ugyldig passord men riktig brukernavn tastes inn 5 ganger på rad	Får feilmelding om at ikke har tilgang til innlogging før om 1 minutt	IKKE OK	28.04.2015	
7	Er ikke logget inn fra før	Er på innloggingssiden	Gyldig brukernavn og tabulatorhopp	Innloggingssiden endres til Norkart innlogging	OK	28.04.2015	Kun mulig via MyApps
8	Er allerede logget inn i webapp 1	Åpner webapp 2	Trykker på logg inn og deretter show image	Får tilgang til beskyttet innhold	OK	28.04.2015	Kan ikke klikke direkte på Show Image
9	Er allerede logget inn i webapp 1	Åpner webapp 3	Ingen	Får tilgang til nettside	OK	28.04.2015	
10	Er ikke logget inn fra før	Åpner webapp 3	Gyldig brukernavn og passord	Autentiseres og viderekobles til nettsted	OK	28.04.2015	
11	Åpnet nettleser	Skrive inn nettadressen	http før adressen	https etter innlogging	OK	28.04.2015	
12	Er ikke logget inn fra før	Åpner webapp 1 og trykker på katt	Ingen	Får ikke tilgang til beskyttet innhold	OK	28.04.2015	

Test: Logg ut, Web applikasjoner							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er innlogget	Åpner webapp 1	Trykker på sign out knapp	Logges ut og viderekobles til "postLogOut" url	OK	28.04.2015	
2	Er innlogget i webapp 3, og logger ut i fra webapp 1	Åpner webapp 3	Ingen	Har blitt logget ut og viderekobles til innloggingsside	OK	28.04.2015	
3	Er innlogget	Logger ut	Klikker logg ut i hvilken som helst applikasjon	Er utlogget i alle applikasjonene	IKKE OK	28.04.2015	Må logge ut av hver enkelt applikasjon
4	Er innlogget i 2 ulike applikasjoner i 2 ulike faner men i samme nettleser	Logge ut av den ene applikasjonen	Klikker logg ut	At man blir logget ut i begge	IKKE OK	28.04.2015	Den andre fanen oppfører seg som logget inn men alle nye faner må logges inn på nytt

Test: Reset passord, web applikasjon							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er ikke innlogget	Er på Innloggingssiden	Trykker på glemt passord	Får CAPTCHA test	OK	8.4	
2	Er på captchasiden	Skrevet inn riktig CAPTCHA	Trykker på neste	Får valg om e post, sms eller ringe	OK	8.4	
3	Valgt e post	Går til e post	Ingen	Har fått e post	OK	8.4	
4	Fått e-post med kode	Er på resett passord side	Taster inn kode og nytt passord	Får endret passord	OK	8.4	
5	Valgt sms	Venter på sms	Ingen	Har fått melding	OK	28.04.2015	
6	Fått sms	Er på resett passord side	Taster inn kode, nytt passord og CAPTCHA	Får endret passord	OK	28.04.2015	
7	Valgt ringe	Venter på telefon	Ingen	Telefonen ringer	OK	28.04.2015	
8	Fått telefon	Er på resett passord side	Taster inn kode, nytt passord og CAPTCHA	Får endret passord	OK	28.04.2015	

Test: Logg inn, Native							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er ikke logget inn fra før	Åpner applikasjon	Ingen	Viderekobles til innloggingssiden	OK	28.04.2015	
2	Er ikke logget inn fra før	Er på innloggingssiden	Gyldig brukernavn og passord	Autentiseres og får tilgang til web api via applikasjon	OK	28.04.2015	
3	Er ikke logget inn fra før	Er på innloggingssiden	Ugyldig brukernavn og passord	Får feilmelding om at innlogging feilet	OK	28.04.2015	
4	Er logget inn fra før	Åpner applikasjonen	Ingen	Får tilgang til web apiet via applikasjonen	OK	28.04.2015	
5	Er ikke logget inn fra før	Åpner applikasjonen	Ugyldig brukernavn og passord tastes inn 5 ganger på rad	Får feilmelding om at ikke har tilgang til innlogging før om 1 minutt	IKKE OK	28.04.2015	
6	Er ikke logget inn fra før	Er på innloggingssiden	Gyldig brukernavn og tab	Innloggingssiden endres til Norkart innlogging	OK	28.04.2015	

B.2 Tester for MyApps

Test: Sluttbruker, MyApps							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er innlogget i Myapps	Registrere brukerdata	Registrere telefonnummer mobilnummer og adresse	Informasjonen blir lagret	IKKE OK	01.05.2015	Støttes ikke av MyApps
2	Er innlogget i Myapps og har registret brukerdata	Endre brukerdata	Endre telefonnummer, mobilnummer eller adresse	Informasjonen blir endret	IKKE OK	01.05.2015	Støttes ikke av MyApps
3	Er innlogget i MyApps	Endre autentiseringsdata	Ny ekstern e-post adresse og eksternt mobilnummer	Får endret autentiseringsdata	OK	01.05.2015	
4	Er innlogget i Myapps	Endre passord	Gammelt passord en gang og nytt passord to ganger	Får bekreftet endring av passord	OK	01.05.2015	

Test: Lokal Administrator, MyApps							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Logget inn i MyApps.	Legge til brukere i AAD via MyApps	Legger til en bruker	Bruker blir lagt til	IKKE OK	28.04.2015	Kan ikke registrere nye brukere i AAD va MyApps
2	Logget inn i MyApps.	Slette brukere i AAD via MyApps	Fjerner en bruker	Bruker blir fjernet	IKKE OK	28.04.2015	Kan ikke fjerne brukere fra AAD via MyApps
3	Logget inn i MyApps.	Legge til grupper	Legg til gruppe i Azure AD	Gruppe blir lagt til	IKKE OK	28.04.2015	Kan ikke lage grupper
4	Logget inn i MyApps, og er eier av gruppene du skal fjerne.	Fjerne grupper	Fjerner en gruppe i fra Azure AD	Gruppen blir fjernet	OK	01.05.2015	
5	Logget inn i MyApps, og er eier av gruppene brukeren er i.	Resette passord for en bruker i en bestemt gruppe	Ber om å få resatt passord for bruker i Azure AD	Passord for brukeren blir resatt	IKKE OK	01.05.2015	Ingen tilgang til bruker data
6	Logget inn i MyApps, og er eier av gruppenen du skal legge brukere til.	Legge til bruker i gruppe	Legger til bruker i gruppe i Azure AD	Bruker blir lagt til	OK	01.05.2015	
7	Logget inn i MyApps, og er eier av gruppenen du skal fjerne bruker fra.	Fjerne bruker fra gruppe	Fjerner bruker fra gruppe	Bruker fjernes fra gruppe	OK	01.05.2015	
8	Logget inn i MyApps, og er eier av gruppenen brukerne du skal endre for er i.	Endre brukerdata for bruker i gruppe	Endre	Endre registrerte opplysninger i en brukerprofil	IKKE OK	01.05.2015	Ingen tilgang til bruker data

B.3 Tester for AAD Portalen

Test: Kundestøtte, AAD Portalen							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Innlogget i Azure portalen	Opprette nye brukere	Type bruker, brukernavn, fulltnavn, rolle	Ny bruker registrert i AAD	OK	01.05.2015	
2	Innlogget i Azure portalen	Slette brukere	Trykke på slett bruker	Bruker blir fjernet fra AAD	OK	01.05.2015	
6	Innlogget i Azure portalen. Bruker eksisterer	Endre brukerdata for brukere	Endrer brukerdata, jobbdata og atentiseringsdata	Endrede data blir registrert om brukeren	OK	01.05.2015	
3	Innlogget i Azure portalen. Bruker eksisterer	Resette passord for en bruker	Trykker på resett passord	Bruker får nytt passord	OK	01.05.2015	Passord blir resatt, men brukeren får ingen beskjed om dette.
7	Innlogget i Azure portalen	Opprette en gruppe	Trykker på legg til gruppe. Gir gruppen navn og beskrivelse	En ny gruppe blir opprettet	OK	01.05.2015	
8	Innlogget i Azure portalen. Det finnes en gruppe	Slette en gruppe	Trykker på slett gruppe	Gruppen blir fjernet fra AAD	OK	01.05.2015	
4	Innlogget i Azure portalen. Det finnes en gruppe	Legge til en bruker i en gruppe	Trykker på legg til bruker. Velger bruker fra en liste over eksisterende brukere	Brukeren blir lagt tli i gruppen	OK	01.05.2015	
5	Innlogget i Azure portalen. Det finnes en gruppe	Fjerne en bruker fra en gruppe	Trykke på fjern bruker. Velger bruker fra en liste over brukere i gruppen	Brukeren blir fjernet fra gruppa	OK	01.05.2015	
9	Innlogget i Azure portalen. Det finnes en gruppe	Redigere eier av en gruppe	Trykker på legg til eier. Velger eier fra en liste av brukere	Ny eier blir lagt til	OK	01.05.2015	Kan gi sluttbrukere rettigheter til en gruppe.
10	Innlogget i Azure portalen. Det finnes en gruppe. Det finnes en applikasjon	Tildele tilgang til en applikasjon for en gruppe	Trykker på legg til gruppe i applikasjonen	Brukere i gruppen får tilgang til applikasjonen	OK	01.05.2015	
Test: Super administrator, AAD Portalen							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Innlogget i Azure portalen	Registrere applikasjoner	Registrere en ny applikasjon i AAD	Applikajsonen blir registrert i AAD	OK	01.05.2015	Testet ved å registrere tre web applikasjoner og en native applikasjon
2	Innlogget i Azure portalen. Det finnes en applikasjon	Konfigurere registrerte applikasjoner	Endre navn, urler og sette krav om tofaktor autentisering	Navn og urler endres. Det kreves tofaktor autentisering for å få tilgang til applikasjonen	OK	01.05.2015	
3	Innlogget i Azure portalen. Det finnes en applikasjon	Administrere tilganger til andre applikasjoner	Trykke på legg til applikasjon under tilgangsstyring. Gi lese og skrive rettigheter	Applikasjonen får skrive og lese rettigheter til applikasjonen som ble lagt til	OK	01.05.2015	
4	Innlogget i Azure portalen. Det finnes en applikasjon	Slette applikasjoner fra AAD	Trykker på slett appllikasjon	Applikasjonen blir fjernet fra AAD	OK	01.05.2015	
5	Innlogget i Azure portalen	Sette opp profilering av Norkart	Legge inn logo, bedriftsnavn, bilde og bakgrunnsfarge	Profilering synes på innloggingssiden og i MyApps	OK	01.05.2015	Synes kun når det er skrevet inn et brukernavn relatert til bedriften
6	Innlogget i Azure portalen	Sette opp passord policy	Velge at brukere kan resette eget passord. Sette en autentiseringsmetode som enten skal være e-post eller telefon.	Opprettet passord policy for brukere i AAD	OK	01.05.2015	
7	Innlogget i Azure portalen. Det finnes en gruppe	Sette opp standardregler for grupper	Velge hvilke brukere som kan lage nye grupper. Velge å bruke "All Users" gruppen	Satt standardregler for gruppene i AAD	OK	01.05.2015	
8	Innlogget i Azure portalen	Lisensoppsett	Sette alle brukere i e gruppe til premium brukere	Alle brukerne i gruppa får premium lisens	OK	01.05.2015	
10	Innlogget i Azure portalen	Opprette nye roller	Trykke på legg til ny rolle	Ny rolle blir lagt til	IKKE OK	01.05.2015	Det er ikke mulig å opprette flere administrator roller enn det som allerede er definert i AAD
11	Innlogget i Azure portalen. Det finnes en rolle	Konfigurere eksisterende roller	Endre rettigheter for eksisterende roller	Rettighetene til rollen er endret	IKKE OK	01.05.2015	Rettiighetene til administratorrollene i AAD kan ikke endres
12	Innlogget i Azure portalen. Det finnes en rolle	Slette roller	Trykke på slett rolle knapp	Rollen blir slettet	IKKE OK	01.05.2015	Definerte roller i AAD kan ikke slettes. Som minimum må en AAD alltid ha en Global Administrator, utover det stilles ingen krav til andre roller.

B.4 Ytelsestester i forhold til kravspesifikasjon

Test: Ytelse							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Er ikke logget inn fra før	Åpner applikasjon	Ingen	Viderekobles til innloggingssiden. Ser at det skjer noe på skjermen inne 2 sekunder, enten hvit skjerm eller at noe lastes inn.	5 av 5 tester OK	28.04.2015	
2	Er ikke logget inn fra før	Er på innloggingssiden	Gyldig brukernavn og passord	Autentiseres og får tilgang til web api via applikasjon. Ser at det skjer noe på skjermen inne 2 sekunder, enten hvit skjerm eller at noe lastes inn. applikasjon	5 av 5 tester OK	28.04.2015	
2	Skal resette passord. Har bestått capcha testen og valgt epost	Ber om å motta epost med kode.	Ingen	Får epost i inboks, med gyldig kode, innen 20 sekunder.	5 av 5 tester OK	28.04.2015	

B.5 Sikkerhetstester i forhold til kravspesifikasjon

Test: Sikkerhet							
Nr	**Forutsetning**	**Aksjon**	**Input**	**Forventet resultat**	**Status**	**Dato**	**Merknad**
1	Innlogget i AAD portalen	Skal legge inn data under følgende attributter: Fullt navn, selskap, e-post, passord, mobiltelefonnummer.	Legger inn de ulike attributtene.	Får lagt inn attributtene	OK	28.04.2015	
2	Innlogget i MyApps portalen	Skal endre passord	Forsøker å legge inn passord kortere enn 8 tegn.	Ska ikke få godkjent passordet på grunn av lengde.	OK	28.04.2015	
3	Innlogget i MyApps portalen	Skal endre passord	Forsøker å lage passord med bare små bokstaver og tall.	Skal ikke få godkjent passord fordi det ikke inneholder stor bokstav.	OK	28.04.2015	
4	Innlogget i MyApps portalen	Skal endre passord	Forsøker å lage passord med bare store og små bokstaver	Skal ikke få godkjent passord fordi det ikke inneholder tall.	OK	28.04.2015	
5	To applikasjoner tilknyttet AAD installert på samme telefon. Autentisert på den ene applikasjonen, ikke den andre.	Åpner den uautentiserte applikasjonen	Ingen	Blir bedt om å autentisere seg på nytt.	OK	28.04.2015	

C Brukertesting for brukervennlighetsanalyse

For å utføre en brukervennlighetsanalyse av Norkart ID ble det gjort brukertester av løsningen. Tesplaner og resultater fra disse er presentert i dette vedlegget

C.1 Testplaner

Det ble utformet testplaner for brukertesting av hovedfunksjonaliteten til alle rollene som skal benytte Norkart ID.

C.1.1 Testplan for sluttbruker og lokal administrator

Problem

Norkart ID bruker innloggingsmekanismer fra AAD for alle applikasjoner, men vi vet ikke om disse er brukervennlig. Norkart ID skal i tillegg ha en egen brukerportal for egenadministering og brukeradministrering. Azure AD leverer en brukerportal som heter MyApps, men vi vet ikke om denne er intutiv og brukervennlig.

Formål

Finne ut om innloggingsmekanismer og MyApps i Norkart ID er så intuitivt at sluttbtuker og lokal administrator kan benytte seg av disse systemene uten opplæring.

Funksjonalitet

- Logge seg inn via innloggingssiden
- Resette passord via glemt passord funksjonalitet
- Registrere egne autentiseringsdata
- Endre eget passord
- Legge til bruekre i en gruppe
- Fjerne brukere fra en gruppe
- Profilere Norkart på MyApps og innloggingssiden

Testbrukere

Syv bachelor og master studenter ved HiG

System

Brukerportalen MyApps og Innloggingsfunksjonalitet tilhørende AAD

Lokasjon

HiG

Innføring

Ingen innføring

Oppgaver

Oppgave 1

Scenario:	Testbruker ønsker å benytte seg av en av Norkart sine applikasjoner
Oppgave:	Logg deg inn på demo applikasjonen NorkartHund med ditt brukernavn og passord. Logg deg deretter ut.
Startpunkt:	https://norkarthund.azurewebsites.net/
Suksesskriterier:	Returneres til websiden ferdig innlogget Skjønner at det er en Norkart ID applikasjon.

Oppgave 2

Scenario:	Testbruker ønsker å bruke en av Norkart sine applikasjoner, men har glemt sitt passord
Oppgave:	Klikk på logg-inn knappen. Du har glemt ditt passord. Hva gjør du?
Startpunkt:	https://norkarthund.azurewebsites.net/
Suksesskriterier:	Får kode på sms, telefon eller e-post Får resatt passord.

Oppgave 3

Scenario:	Testbruker ønsker å bruke NorkartHund applikasjonen.
Oppgave:	Logg deg inn på http://myapps.microsoft.com. Start applikasjonen NorkartHund fra MyApps
Startpunkt:	Nettleser lukket.
Suksesskriterier:	Får logget inn på MyApps Får kommet inn på NorkartHund.

Oppgave 4

Scenario:	Testbruker ønsker å resette sitt passord.
Oppgave:	Endre ditt passord fra Trello2000 til Trello2001.
Startpunkt:	Ferdig logget inn på MyApps
Suksesskriterier:	Skjønner hvor man skal klikke for å endre passord Skjønner hva som er kravene til passordet Får endret passord.

Oppgave 5

Scenario:	Testbruker ønsker å endre telefonnummer for resetting av passord
Oppgave:	Endre telefonnummerregistrert for resetting av passord.
Startpunkt:	Ferdig logget inn på MyApps
Suksesskriterier:	Når testbruker får endret eksternt telefonnummer.

Oppgave 6

Scenario:	Testbruker ønsker å legge til og fjerne brukere i grupper.
Oppgave:	Fjern bruker nr 2 fra Norkatt applikasjonsgruppen. Legg til bruker nr 2 i NorkartHund applikasjonsgruppen.
Startpunkt:	Får fjernet bruker fra gruppe. Får lagt til bruker i gruppe.
Suksesskriterier:	Når testbruker får endret eksternt telefonnummer.

Tidsplan

30. mai, 2015.

Testbruker 1:	09.45
Testbruker 2:	10.00
Testbruker 3:	10.15
Testbruker 4:	10.30
Testbruker 5:	11.00
Testbruker 6:	11.30
Testbruker 7:	11.45

Spørsmål til bruker før test

- Hvordan er din IT kunnskap?

- Har du erfaring med MyApps?

Spørsmål til bruker etter test

- Hva synes du om systemet?
- Har du noen forbedringsforslag?

Testbrukeren svarer til slutt på en SUS undersøkelse.

Testteam

Testleder:	Ida F. Granholt
Hovedobservatør:	Per Christian Kofstad.
Observatør:	Alf Hammerseth.

C.1.2 Testplan for kundestøtte

Problem

Kundestøtte hos Norkart ID skal kunne håndtere grupper, roller og brukere. For Norkart ID skjer dette i Azure AD portalen.

Formål

Finne ut om kundestøtte hos Norkart klarer å gjennomføre funksjonalitet for håndtering av brukere og grupper etter en kort innføring i AAD portalen.

Funksjonalitet

- Opprette en ny bruker
- Slette en bruker
- Endre brukerdata
- Resette passordet til en bruker?
- Gi en bruker en rolle
- Opprette e ny gruppe
- Slette en gruppe
- Legge til brukere i en gruppe
- Fjerne en bruker fra en gruppe
- Konfigurere gruppeingormasjon
- Redigere gruppeeier
- Gi en gruppe tilgang til en applikasjon

Testbruker

Testbruker 8, Kundestøtte hos Norkart. Kundestøtte rollen.

System

Azure AD Portalen

Lokasjon

Norkart, avdeling Lillehammer

Innføring

Kort presentasjon og demo av AAD portalen

Oppgaver

Oppgave 1

Scenario:	HiG ønsker å bruke applikasjonen Norkart Manual hos Norkart.
Oppgave:	Lag en ny gruppe for bedriften, kall gruppen HiG. Gi HiG tilgang til applikasjonen Norkart Manual.
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Lager en ny gruppe. Gir gruppen tilgang til applikasjonen.

Oppgave 2

Scenario:	HiG ønsker at kundestøtte skal registrere en ny lokal administartor som skal ha tilgang til Norkart Manual.
Oppgave:	Opprett en ny bruker med dataene: Brukernavn: perper Navn: Per Persen Jobbtelefon: 33 44 33 44 Mobiltelefon: 99 88 99 77 Sett autentiseringsdata: Ekstern Telefon: 99 88 99 77 Ekstern epost: per@persen.no Gi Per rollen som User Admin Legg Per til i HiG gruppen.
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Klarer å opprette en ny bruker Klarer å registrere data om brukeren Klarer å gi brukeren korrekt rolle Klarer å legge til brukeren i en gruppe.

Tidsplan

29.April, 2015 fra kl.12.00 - 13.00

Spørsmål til bruker før test

- Hva er dine arbeidsoppgaver?
- Hvor lenge har du jobbet i Norkart?
- Har du vært borti Active Directory før?
- Kjenner du godt til Azure AD portalen?

Spørsmål til bruker etter test

- Hva synes du om systemet?
- Har du noen forbedringsforslag?

Testbrukeren svarer til slutt på en SUS undersøkelse.

Testteam

Testleder:	Ida F. Granholt
Hovedobservatør:	Per Christian kofstad.
Observatør:	Alf Hammerseth.

C.1.3 Testplan for super administrator

Problem

Superadministratorer hos Norkart ID skal kunne registrere og håndtere applikasjoner som skal bruke Norkart ID som autentiseringsløsning, samt generell konfigurering av AAD.

Formål

Finne ut om Superadministratorer hos Norkart klarer å gjennomføre funksjonalitet for registrering og håndtering av applikasjoner i AAD portalen og generell håndtering av AAD etter en kort innføring i AAD portalen

Funksjonalitet

- Registrere web applikasjon
- Registrere mobil applikasjon
- Endre applikasjonsdata
- Konfigurere en applikasjons tilgang til andre applikasjoner
- Konfigurere applikasjoner registrert i AADF
- Konfigurere passord policy
- Sette standardvalg for grupper
- Endre lisensoppsett for brukere registrert i AAD
- Endre domene
- Konfigurere AAD

Testbruker

Testbruker 9, Utvikler hos Norkart. Superadministrator rollen.

System

Azure AD Portalen

Lokasjon

Norkart, avdeling Lillehammer

Innføring

Kort presentasjon og demo av AAD portalen

Oppgaver

Oppgave 1

Scenario:	Kunder som skal resette sitt passord autentiserer seg i dag med sikkerhetsspørsmål. Norkart ønsker at de heller skal gjøre dette via ekstern e-post.
Oppgave:	Endre slik at brukere kun kan autentisere seg med ekstern e-post for å resette sitt passord for å stramme innbruker-kravene i AAD.
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Setter kun e-post som autentiseringsdata.

Oppgave 2

Scenario:	Norkart ønsker at alle brukerne i AAD skal være premium brukere
Oppgave:	Gjør alle brukere i AAD til premium brukere
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Finner riktig sted for å utføre dette. Forstår at kan velge gruppen med alle brukere å enable premium.

Oppgave 3

Scenario:	Norkart har lagd en ny web applikasjon. For at dere skal kunne bruke AAD autentisering på denne må den registreres i AAD.
Oppgave:	Bruk urlen “https:/keikoinnovation.com” og registrer den nye web applikasjonen i Azure AD med navnet Keiko. Vis gruppen hvor applikasjons id til den nyregistrerte applikasjonen ligger og hvordan du oppretter en klient secret for 1 år.
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Viser applikasjons id til testteamet Har opprettet en klient secret.

Oppgave 4

Scenario:	Norkart ønsker at applikasjonen Norkart manual skal ha tilgang til Graph api.
Oppgave:	Naviger til applikasjonen Norkart Manual og gi den skrive og lese rettigheter til Graph Api. Graph Api er kalt Windows Azure Active Directory
Startpunkt:	Forsiden av portalen for Norkart ID AAD.
Suksesskriterier:	Endrer applikasjonens tilgang til Graph Api.

Tidsplan

29.April, 2015 fra kl.13.00 - 14.00

Spørsmål til bruker før test

- Hva er dine arbeidsoppgaver?
- Hvor lenge har du jobbet i Norkart?
- Har du vært borti Active Directory før?
- Kjenner du godt til Azure AD portalen?

Spørsmål til bruker etter test

- Hva synes du om systemet?
- Hva var vanskeligst?
- Hva var lettest?
- Har du noen forbedringsforslag?
- Har du andre tilbakemeldinger?

Testbrukeren svarer til slutt på en SUS undersøkelse.

Testteam

Testleder:	Ida F. Granholt
Hovedobservatør:	Per Christian kofstad.
Observatør:	Alf Hammerseth.

C.2 Testresultater

Testresultatene fra brukertestene består av prosjektgruppens observasjoner av testpersonene og Resultatet fra SUS undersøkelsene.
ObservasjonMyApps1

C.2.1 Observasjoner fra brukertesting av MyApps og innlogging

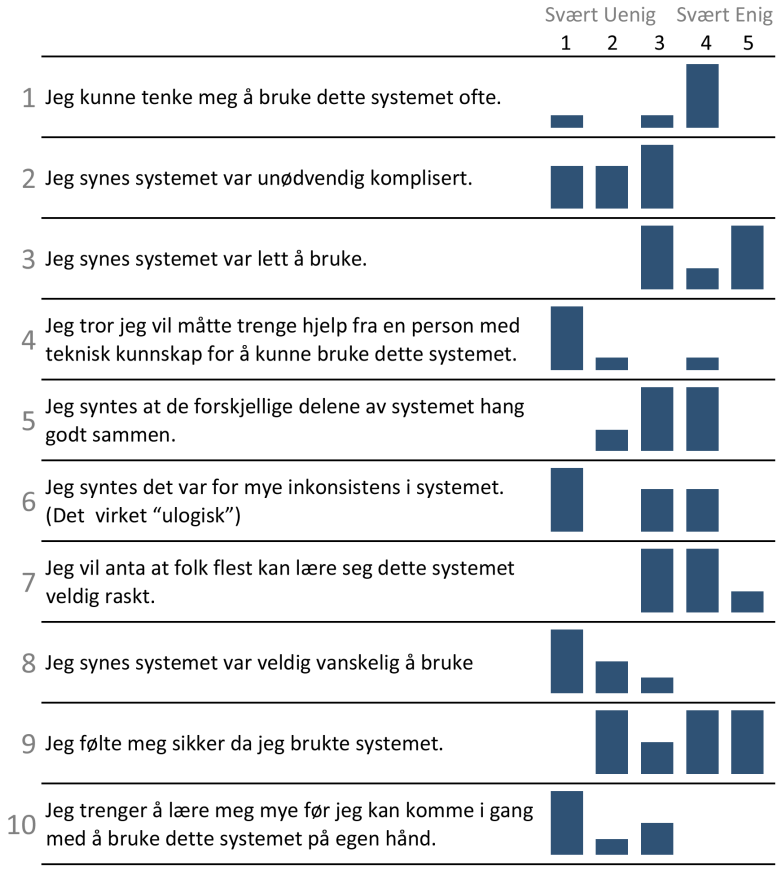
	Norkart ID
1	Logg deg inn på demo applikasjonen https://norkarthund.azurewebsites.net/ med ditt brukernavn og passord. Logg deg deretter ut
1 Testbruker 1	Så ikke Norkart branding, mente det var Microsoft
2 Testbruker 2	Så Norkart branding
3 Testbruker 3	Så ikke Norkart branding, mente det var Microsoft sin side. Så på adressefelt.
4 Testbruker 4	La ikke merke til at Norkart branding
5 Testbruker 5	La ikke merke til Norkart branding. Sitat: "Jeg liker at den husker at jeg har logga meg inn en gang før"
6 Testbruker 6	La merke til norkart branding
7 Testbruker 7	Så Norkart branding.Litt liten innloggingsboks, vanskelig å se bokstavene om man skriver feil. Sitat "Jeg la merke til det grønne i logoen"
2	Naviger til siden https://norkarthund.azurewebsites.net/ Klikk på logg-inn knappen Du har glemt ditt passord. Hva gjør du?
1 Testbruker 1	Indikerer at captcha er tungvint. Prøvde å ringe, funket fint. Leter etter glemt passord link der man velger bruker. Klikker på konto og får ikke logget inn
2 Testbruker 2	Usikkerhet rundt hvor glemt passord knappen er. Syntes captcha var vanskelig. Forsto ikke at man skulle taste inn kode fra mail for å bekrefte resetting av passord
3 Testbruker 3	Kløner litt på Captcha test
4 Testbruker 4	Slet med Captcha test. Vanskelig å se når skrevet feil brukernavn, liten tekst.
5 Testbruker 5	Synes det var vanskelig at man måtte velge bruker for å få tilgang på resett passord link. Nysgjerrig på løsning og spør hva som skjer hvis han trykker på administrator. Sitat "Ikke så intutiv den glem knappen"
6 Testbruker 6	Leter etter glemt passord link der man velger brukere. Hovrer over og ser et "glemt" valg. Usikker på om det er denne. Sitat "Ikke enig at passordet Trello2000 er strekt"
7 Testbruker 7	Leter etter passord link på velg bruker siden. Velger så "en annen bruker" og finner så link. Under bytting av passord så hun at passordet ikke var sterkt nok og kom på at hun hadde glemt stor forbokstav.
3	Logg deg inn på http://myapps.microsoft.com Start applikasjonen NorkartHund fra MyApps
3 Testbruker 3	Klarte oppgaven svært rsakt
4 Testbruker 4	Sitat: "Ser ikke hvilket firma det logges inn til"
4	Logg deg inn på http://myapps.microsoft.com Endre ditt passord fra Trello2000 til Trello2001
1 Testbruker 1	Prøver øverst til høyre først. Skjønte fort at det ikke var riktig valgte så rett. Sitat "Der er profil"
2 Testbruker 2	Skjønte med en gang hvor det var
3 Testbruker 3	Skjønner hvor man skal gå med en gang
4 Testbruker 4	Prøver først på brukernavn oppe til høyre
5 Testbruker 5	Velger først å se øverst til høyre. Syntes det var vrient å forstå at kateoriene var linker(Hovedmenyen). Sitat: "Burde hatt valg om å endre brukerprofil øverst under brukernavn." "Jeg skjønte ikke at menyen var en meny"
6 Testbruker 6	Prøver først valgene øverst til høyre. Går til en applikasjon for så å endre passord via glemt passord funksjonaliteten. Forsto ikkeat dette kunne gjøres i MyApps
7 Testbruker 7	Leter først øverst, finner ikke det hun leter etter. Klikker på hjelp. Ser en stund på hjelp siden. Fikk ingenting ut av hjelp siden. Måtte til slutt spørre hvor hun kunne endre passord. Når hun fant riktig sted gikk det uten problemer.
5	Logg deg inn på http://myapps.microsoft.com Endre telefonnummerregistrert for resetting av passord til 99887766.
1 Testbruker 1	Usikker på hvor det skulle gjøres, men letet seg frem til riktig sted. Sitat "Vi prøver os her" "Hvor er det a?"
2 Testbruker 2	Lurte på tilbakeknappen på bekreftet nummer, men gjorde det ikke
3 Testbruker 3	Usikkerhet rundt hvilket menyvalg som skal trykkes på
5 Testbruker 5	Brukte litt tid på å finne riktig valg. Synes det var irriterende at ikke all teksten synes i valgboksen.
6 Testbruker 6	Klikker først på profilnavnet for å se om det går å endre
7 Testbruker 7	Fant hvor hun skulle endre. Trykte på tilbake før endringen var ferdig, måtte begynne på nytt.

6	Logg deg inn på http://myapps.microsoft.com Fjern bruker nr 2 fra Norkatt applikasjonsgruppen. Legg til bruker nr 2 i NorkartHund applikasjonsgruppen.
1 Testbruker 1	Valgte grupper med en gang. Brukte søk funksjonen for å legge til bruker. Venter på beskjed om at bruker er lagt inn, men får ingen bekreftelse på dette.
2 Testbruker 2	Prøver å trykke på pil ned for å velge bruker, dette fungerer ikke, brukte da søksfelt. Letter litt etter fjern knapp. Velger edit, men går så tilbake.
3 Testbruker 3	Forsto ikke at dette skulle gjøres under grupper. Prøvde flere valg først. Sitat "Dette var tungvint"
4 Testbruker 4	Bruker søk funksjonen for å finne bruker. Usikker på om det man gjør er riktig, får ikke bekreftelse. Sitat: "Oppdaterer seg ikke automatisk"
5 Testbruker 5	Prøver å trykke seg inn på brukeren for å se valg. Etterspør agreknapp eller melding om hvem bruker som blir fjernet. Sitat "Det skulle vært en tilbakeknapp her." "Burde kanskje hvert en angreknapp eller informasjon om at fjerning er skjedd og hvem bruker som ble fjernet."
6 Testbruker 6	Velger grupper, finner ikke valg, går til edit, så tilbake til grupper.
7 Testbruker 7	Valgte riktig menyvalg, forsto ikke at dette var riktig, tilbake til programmer, trykte på websiden og åpnet den. Gikk så tilbake og valgte grupper. Bruker søke feltet for å finne bruker, skriver hele brukernavnet, trykker på add uten å velge bruker, ingen bruker blir lagt til. prøver igjen og får det til. Bruker litt tid på å finne ut hvor man kan fjrne bruker.
7	Hva synes du om systemet?
1 Testbruker 1	Regner med at det er lett å bruke etter man har brukt det en stund. Synes ikke det var vanskelig å bruke
2 Testbruker 2	Greot og oversiktlig. Ligner på andre Microsoft produkter, derfor lett å bruke. Innlogging og å legge til bruker var veldig lett.
3 Testbruker 3	Innlogging var bra, liker ikke å bli sendt til ny side. Navn på hovedmeny var vanskelig å forstå. Trodde grupper kun var for å opprette grupper og at godkjenning var for å legge til og fjerne brukere.
4 Testbruker 4	Synes systemet er helt greit. Likte at det var få menyvalg og at brukernavnet sto øverst.
5 Testbruker 5	Innlogging fungerte bra. Forsto ikke at hovedmenyen i MyApps var en meny. Syntes captcha testen var grei, selv om ikke egentlig er fan.
6 Testbruker 6	Innlogging var enkel og bra. Uvant navigasjon i MyApps. Skjønte ikke hvor det skulle jobbes, men når ho skjønte det var den lett å bruke.
7 Testbruker 7	Skjønte ikke at menyen var en meny, men når skjønte det var MyApps ryddig og oversiktlig. Synes det var bra med pop up skjema på sletting. Det var knotete å finne hvor man kunne endre passord
8	Forbedringsforslag?
1 Testbruker 1	Savner bekreftelse på at ting er blitt gjort. Ønsker seg bedre måter å filtrere grupper på.
2 Testbruker 2	Ønsker glemt passord link der man velger bruker.
3 Testbruker 3	Ikke ha autetiseringsadata for seg selv, men sammen med de andre dataene.
4 Testbruker 4	Forbedre designet, synes det ikke var så pent. Skulle gjerne sett logoen bedre
5 Testbruker 5	Skulle gjerne hatt en tilbakeknapp. Ønsker tilbakemelding om hvilken bruker som har blitt slettet. Vil ha farger på menyen, slik at det synes bedre at det er menyvalg.

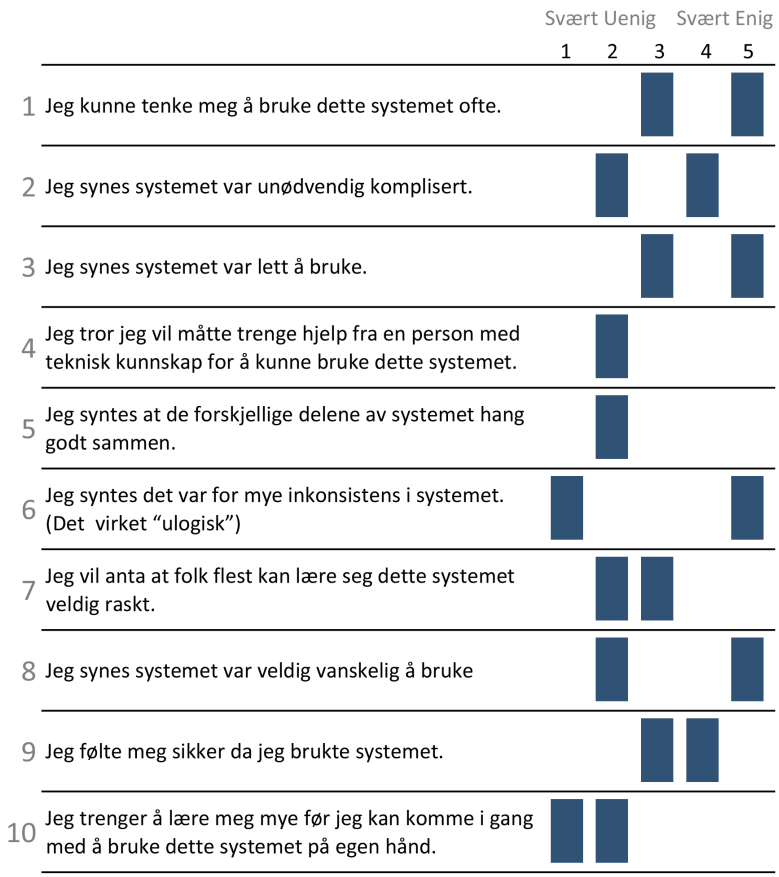
C.2.2 Observasjoner fra brukertesting av AAD portalen

	Norkart ID
1	Lag en ny gruppe for bedriften, kall gruppen HiG. Gi HiG tilgang til applikasjonen Norkart Manual.
1 Tesbruker 8	Lagde gruppe uten problemer Så ikke pilen tibake, skjønte ikke hvordan han skulle komme tilbake. Går til applikasjoner, så til configure, tilabke til applikasjoner og users and groups under applikasjoner, velger HiG. Brukte en del forsøk. Synes GUI er uovsersiktlig. Glemte bunn og toppmeny kontekst Sitat: "Tilbakepilen er så stor at den ikke synes" "Jeg ville tippet at jeg må på applikasjoner eller configure, men jeg fant det ikke der"
2	Opprett en ny bruker med dataene Brukernavn: perper Navn: Per Persen Jobbtelefon: 33 44 33 44 Mobiltelefon: 99 88 99 77 Sett autentiseringsdata: Ekstern Telefon: 99 88 99 77 Ekstern epost: per@persen.no Gi Per rollen som User Admin
1 Tesbruker 8	Ga rolle under oppretting av ny bruker, klarte dette lett. Vanskelig å finne ut hvor persondata skulle settes. Øsnker flere checkbokser i stedet for tekstfelter. Forrvirrende at ting heter det samme, men ligger flere steder. Sitat: "" det skulle vært en checkbox ved siden av tlf og epost om man ville bruke disse som autentiseringsdata "Jeg la ikke merke til underoverskriftene her, så jeg skjønte ikke hva alle tekstboksene var til"
3	Endre slik at brukere kun kan autentisere seg med ekstern e-post for å resette sitt passord.
2 Testbruker 9	Gikk til riktig sted, men så ikke at det var riktig. Sjekket alle tabene under user, skjønner at det ikke ligger her. går tilbake til configure og klarer det.
4	Gjør alle brukere i AAD til Premium brukere
2 Testbruker 9	Usikker på hvor han skal begynne å lete. Tenkte han skulle velge all users og at det der var et valg om å sette alle brukere til premium, det var det ikke. Finner etter litt tid gruppen All users og klarer oppgaven.
5	Bruk urlen "https:/keikoinnovation.com" og registrer den nye web applikasjonen i Azure AD med navnet Keiko. Vis testteamet hvor applikasjons id til den nyregistrerte applikasjonen ligger og hvordan du oppretter en klient secret for 1 år.
2 Testbruker 9	Klarer raskt å registrere applikasjon. Leter en stund etter id, prøver tre valg før configure. Klarte å lage client secret etter hint om nøkkel.
6	Naviger til applikasjonen Norkart Manual og gi den skrive og lese rettigheter til Graph Api. Graph Api er kalt Windows Azure Active Directory
2 Testbruker 9	Klarer oppgaven fort
7	Hva synes du om systemet?
1 Tesbruker 8	Synes det var rart at det var to nivåer på venstre siden med lite informasjon, men veldig mye informasjon der man jobber. Det var et trangt GUI og vanskelig å navigere både oppe og nede. Starten er ryddig, men det er vanskelig å finne veien videre.
2 Testbruker 9	Veldig enkelt og fint. Lett navigasjonsflyt. Ikke vant til at bunnmargen har så mye funksjonalitet, men det gjør ikke noe.
8	Forbedringsforslag?
1 Tesbruker 8	Workflowen var for delt opp og ikk tilpasset arbeidsoppgavene, dette burde forbedres

C.2.3 SUS resultat for MyApps og innlogging



C.2.4 SUS resultat for AAD portalen



D Krav til nettløsninger i WCAG 2.0

Under er disse kravene gjennomført med utgangspunkt i Azure AD [43]. Kravene satt til om språk, 3.1.1 og 3.1.2, kunne ikke testes da tilgang til Azure AD kildekoden på nettsiden ikke var mulig.

- ☒ 1.1.1 Ikke-tekstlig innhold
- ☒ 1.2.1 Bare lyd og bare video (forhåndsinnspilt)
- ☒ 1.2.2 Teksting (forhåndsinnspilt)
- ☒ 1.3.1 Informasjon og relasjoner
- ☒ 1.3.2 Meningsfylt rekkefølge
- ☒ 1.3.3 Sensoriske egenskaper
- ☒ 1.4.1 Bruk av farge
- ☒ 1.4.2 Styring av lyd
- ☐ 1.4.3 Kontrast (minimum)
- ☒ 1.4.4 Endring av tekststørrelse
- ☒ 1.4.5 Bilder av tekst
- ☒ 2.1.1 Tastatur
- ☒ 2.1.2 Ingen tastaturfelle
- ☒ 2.2.1 Justerbar hastighet
- ☒ 2.2.2 Pause, stopp, skjul
- ☒ 2.3.1 Terskelverdi på maksimalt tre glimt
- ☒ 2.4.1 Hoppe over blokker
- ☒ 2.4.2 Sidetitler
- ☒ 2.4.3 Fokusrekkefølge
- ☒ 2.4.4 Formål med lenke (i kontekst)
- ☒ 2.4.5 Flere måter
- ☒ 2.4.6 Overskrifter og ledetekster
- ☒ 2.4.7 Synlig fokus
- ⊡ 3.1.1 Språk på siden
- ⊡ 3.1.2 Språk på deler av innhold
- ☒ 3.2.1 Fokus
- ☒ 3.2.2 Inndata
- ☒ 3.2.3 Konsekvent navigering
- ☒ 3.2.4 Konsekvent identifikasjon
- ☒ 3.3.1 Identifikasjon av feil
- ☒ 3.3.2 Ledetekster eller instruksjoner
- ☒ 3.3.3 Forslag ved feil
- ☒ 3.3.4 Forhindring av feil (juridiske feil, økonomiske feil, datafeil)
- ☒ 4.1.1 Parsing (oppdeling)
- ☒ 4.1.2 Navn, rolle, verdi

E Besluttningslogg

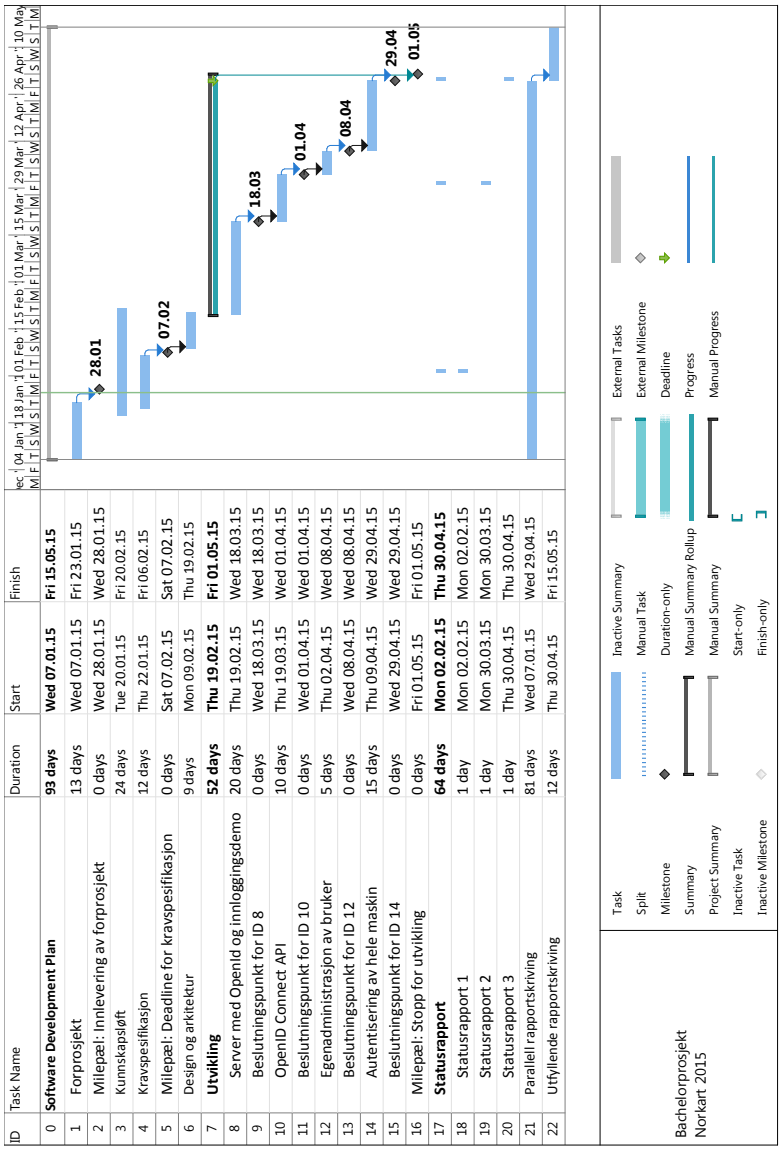
Større beslutninger som er tatt underveis i prosjektet.

Hva	**Hvem**	**Når**	**Hvordan**
Bestemt å gå dypere inn på IdentityServer3 med Azure AD som katalogtjeneste.	Prosjektgruppen og Oppdragsgiver	9/1-2015	Enstemig på planleggingsmøte
Godkjent utkast til kravspesifikasjon.	Prosjektgruppen og Oppdragsgiver	4/2-2015	Enstemig
Bestemt å se nærmere på om Azure AD kan dekke behovet for hele tjenesten.	Prosjektgruppen og Oppdragsgiver	2/3-2015	Enstemig
Bestemt å fokusere på anvendelse av Azure AD videre i prosjektet.	Prosjektgruppen og Oppdragsgiver	5/3-2015	Enstemig
Bestemt at prosjektgruppen ikke skal se på det juridiske spørsmålet rundt lagring av persondata utenfor Norge.	Prosjektgruppen og Oppdragsgiver	5/3-2015	Enstemig
Bestemt at punktet om krav til lokal lagring av brukerdatabase bortfaller	Prosjektgruppen og Oppdragsgiver	9/3-2015	Enstemig
Avgjort at demoapplikasjoner for test av autentiseringstjenesten også skal gjøres på Android applikasjon. Andorid applikasjonen skal kun fokusere på innlogging.	Prosjektgruppen	9/3-2015	Enstemig
Avgjort at prosjektgruppen vil anbefale bruk av Azure AD gitt at en brukeradministrasjons-portal blir laget i tillegg, for å dekke alle krav i kravspesifikasjon.	Prosjektgruppen i gruppemøte	30 April 2015	Enstemig

F Arbeidsplan og Tid

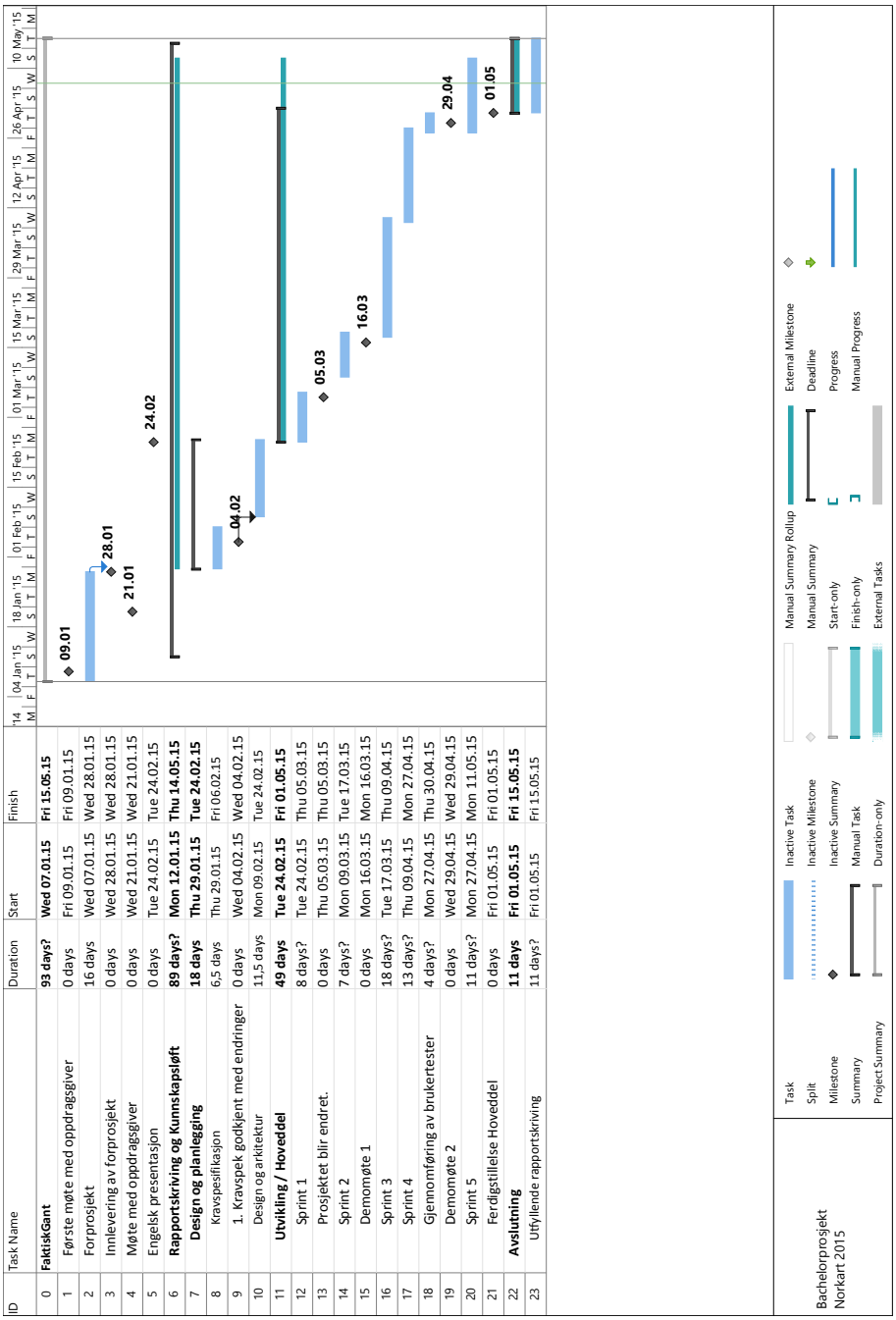
Først arbeidsplan i form av planlagt Gantt. Derretter porsjektforløpet slik det endte opp med å bli. Til slutt, arbeidstid for hver enkelt gruppemedlem.

F.1 Arbeidsplan



ID	Task Name	Duration	Start	Finish
0	**Software Development Plan**	**93 days**	**Wed 07.01.15**	**Fri 15.05.15**
1	Forprosjekt	13 days	Wed 07.01.15	Fri 23.01.15
2	Milepæl: Innlevering av forprosjekt	0 days	Wed 28.01.15	Wed 28.01.15
3	Kunnskapsløft	24 days	Tue 20.01.15	Fri 20.02.15
4	Kravspesifikasjon	12 days	Thu 22.01.15	Fri 06.02.15
5	Milepæl: Deadline for kravspesifikasjon	0 days	Sat 07.02.15	Sat 07.02.15
6	Design og arkitektur	9 days	Mon 09.02.15	Thu 19.02.15
7	**Utvikling**	**52 days**	**Thu 19.02.15**	**Fri 01.05.15**
8	Server med OpenId og innloggingsdemo	20 days	Thu 19.02.15	Wed 18.03.15
9	Beslutningspunkt for ID 8	0 days	Wed 18.03.15	Wed 18.03.15
10	OpenID Connect API	10 days	Thu 19.03.15	Wed 01.04.15
11	Beslutningspunkt for ID 10	0 days	Wed 01.04.15	Wed 01.04.15
12	Egenadministrasjon av bruker	5 days	Thu 02.04.15	Wed 08.04.15
13	Beslutningspunkt for ID 12	0 days	Wed 08.04.15	Wed 08.04.15
14	Autentisering av hele maskin	15 days	Thu 09.04.15	Wed 29.04.15
15	Beslutningspunkt for ID 14	0 days	Wed 29.04.15	Wed 29.04.15
16	Milepæl: Stopp for utvikling	0 days	Fri 01.05.15	Fri 01.05.15
17	**Statusrapport**	**64 days**	**Mon 02.02.15**	**Thu 30.04.15**
18	Statusrapport 1	1 day	Mon 02.02.15	Mon 02.02.15
19	Statusrapport 2	1 day	Mon 30.03.15	Mon 30.03.15
20	Statusrapport 3	1 day	Thu 30.04.15	Thu 30.04.15
21	Parallell rapportskriving	81 days	Wed 07.01.15	Wed 29.04.15
22	Utfyllende rapportskriving	12 days	Thu 30.04.15	Fri 15.05.15

F.2 Prosjektløp



F.3 Arbeidstid

For hver enkelt gruppe medlem.

F.3.1 Samlet tidsoversikt

Samlet tidsoversikt for hele gruppen gjennom prosjektperioden. Inndeling etter uker.

F.3.2 Arbeidstid for Alf

Detailed report

toggl

2015-01-05 - 2015-05-15
Total 423 h 45 min

Alf Hammerseth selected as users
Bachelor selected as projects

Date	Description	Duration	User
01-08	**Bachelor møte** Hig / Norkart - Bachelor	**1:45:00** 13:15-15:00	Alf Hammerseth
01-08	**Møte med gruppa for å diskutere leder** Hig / Norkart - Bachelor	**1:00:00** 15:00-16:00	Alf Hammerseth
01-09	**Møte med NorKart lillehammer** Hig / Norkart - Bachelor	**1:30:00** 12:00-13:30	Alf Hammerseth
01-11	**Lese bacheloroppgaver** Hig / Norkart - Bachelor - [mobile]	**0:30:00** 18:15-18:45	Alf Hammerseth
01-12	**Administrative bachelor tasks** Hig / Norkart - Bachelor	**4:30:00** 09:30-14:00	Alf Hammerseth
01-13	**Group rules** Hig / Norkart - Bachelor	**2:00:00** 11:00-13:00	Alf Hammerseth
01-14	**Ser over reglene som er satt og begynner på risikoanalysen** Hig / Norkart - Bachelor	**2:00:00** 09:40-11:40	Alf Hammerseth
01-16	**Risikoanalysen ferdigstilles til presentasjon på mandag. Ser over tidligere arbeid** Hig / Norkart - Bachelor	**0:30:00** 12:00-12:30	Alf Hammerseth
01-19	**Ferdigstilte risikoanalysen, dokumentasjon, kildekode og konfigurasjonsstyring** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
01-20	**Gjennomgang av dokumentet så langt og lagt inn mitt egetarbeid i LaTeX. Polerte rutiner og regler.** Hig / Norkart - Bachelor	**5:30:00** 08:00-13:30	Alf Hammerseth
01-21	**Jobba med forprosjektet, reiste å møtte Norkart og fikk innføring i Azure miljøet vi skal bruke. Fikk også opprettet kontoer innen de forskjellige verktøyene vi skal ha i deploy miljøet.** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
01-22	**Korrekturlest forprosjektet. Ferdigstilt som første utkast til å sende til veildere. Plan er å få tilbakemelding innen innleveringsfristen som er 28.jan og revidere etter hva vi får fra veileder.** Hig / Norkart - Bachelor	**4:00:00** 13:00-17:00	Alf Hammerseth
01-23	**Ferdigstilt første utkast av forprosjektet og sendt det til veiledere for tilbakemelding** Hig / Norkart - Bachelor	**3:00:00** 07:00-10:00	Alf Hammerseth
01-26	**Lest om OpenID Connect og Oath 2** Hig / Norkart - Bachelor	**3:30:00** 08:30-12:00	Alf Hammerseth
01-27	**Litteraturstudie mot OpenID Connect** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Alf Hammerseth
01-28	**Lest om C# og gjort tutorials** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Alf Hammerseth

01-29	**Definerte kravspekk** Hig / Norkart - Bachelor	**2:00:00** 09:00-11:00	Alf Hammerseth
01-29	**Jobbet med litteratur og så på kravspekken.** Hig / Norkart - Bachelor	**2:00:00** 14:00-16:00	Alf Hammerseth
02-02	**Jobbet med mine punkter i kravspekken.** Hig / Norkart - Bachelor	**5:00:00** 09:00-14:00	Alf Hammerseth
02-03	**Jobbet med gruppen og gikk igjennom alle kravspek punktene og utarbeidet alle use casene ilag** Hig / Norkart - Bachelor	**11:30:00** 08:30-20:00	Alf Hammerseth
02-09	**Leste på software architecture og authentication oppsett** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
02-11	**Research for server architecture og lese om server oppsett for websider** Hig / Norkart - Bachelor	**3:00:00** 10:00-13:00	Alf Hammerseth
02-13	**Laga draft av server architecture** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
02-15	**Lest og sett over skissen min** Hig / Norkart - Bachelor	**2:00:00** 10:00-12:00	Alf Hammerseth
02-16	**Leste om Gluu, Katana, OWIN og så over skissen en siste gang** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
02-17	**MOte med gruppen der vi gikk igjennom hvordan vi ligger an til A fA ferdig design og arkitektur.** Hig / Norkart - Bachelor	**3:00:00** 08:00-11:00	Alf Hammerseth
02-18	**Jobbet med extended use case** Hig / Norkart - Bachelor	**3:00:00** 10:00-13:00	Alf Hammerseth
02-19	**Jobbet enda mer med extended use case. Ble nesten ferdig med dem** Hig / Norkart - Bachelor	**4:30:00** 10:00-14:30	Alf Hammerseth
02-20	**Versjonkontroll rutine utarbeidet og jobbet videre med extended use casene** Hig / Norkart - Bachelor	**2:00:00** 08:00-10:00	Alf Hammerseth
02-21	**Gruppemøte på skype for å samkjøre gruppa. Snakka om hva vi hadde gjort angående design og arkitektur** Hig / Norkart - Bachelor	**1:30:00** 12:00-13:30	Alf Hammerseth
02-23	**Gjennomgang av presentasjon og leste endringer i kravspekken** Hig / Norkart - Bachelor	**1:00:00** 23:00-00:00	Alf Hammerseth
02-24	**Satt opp PBI for logg inn og satt opp priortering** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Alf Hammerseth
02-25	**Jobbet med første MVP, PBI priotering, finpuss på design og arkitektur. Startet med research og testing av OpenID Connect** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Alf Hammerseth
02-26	**Leste igjennom foreløpig rapport og researcha OpenID Connect** Hig / Norkart - Bachelor	**3:00:00** 12:00-15:00	Alf Hammerseth
02-27	**Reasearch OpenID Connect, sette opp egen løsning** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
02-28	**Jobbet med OpenID Connect, første task i TFS** Hig / Norkart - Bachelor	**2:00:00** 18:00-20:00	Alf Hammerseth

03-01	**Jobbet med OpenID Connect, første task i TFS, men lengre, kom til legge inn claims og har problemer.** Hig / Norkart - Bachelor	**4:00:00** 11:30-15:30	Alf Hammerseth
03-02	**Sette opp MVP og tildeling av oppgaver** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
03-03	**Jobbet med MVP og mine oppgaver** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
03-04	**Innsett at vi har misforstått litt så måtte jobbe mye med å research av Azure, ASP.NET Identity, ADFS, OpenID, Identityserver3** Hig / Norkart - Bachelor	**4:00:00** 08:00-12:00	Alf Hammerseth
03-05	**Jobbet med tasks og fått webapp 1 til å fungere med vår Azure.** Hig / Norkart - Bachelor	**5:00:00** 10:30-15:30	Alf Hammerseth
03-09	**Satt opp prioriteringsliste over hva som skal gjøres til mandag, 16/3 og delegert oppgaver. Jobbet også med å få på plass litt tanker om hvordan vi nå skal gå fram med tanke på den store endringer i fokus i bacheloren.** Hig / Norkart - Bachelor	**8:00:00** 08:30-16:30	Alf Hammerseth
03-10	**Azure og OpenID Connect utkast** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
03-11	**SAML, OAuth, SSO og møte med gruppa** Hig / Norkart - Bachelor	**5:00:00** 10:30-15:30	Alf Hammerseth
03-12	**Hva er Azure dokumentasjon, første utkast** Hig / Norkart - Bachelor	**2:00:00** 11:00-13:00	Alf Hammerseth
03-13	**Hva lagre hvor i Azure og Azure AD** Hig / Norkart - Bachelor	**2:00:00** 11:00-13:00	Alf Hammerseth
03-14	**Jobbet med å få oversikt over oppgaven og så på strukturen til Per Christian** Hig / Norkart - Bachelor	**2:00:00** 12:00-14:00	Alf Hammerseth
03-16	**Jobbet med å redefinere oppgaven og klargjorde informasjon til demo 1 for Norkart.** Hig / Norkart - Bachelor	**9:00:00** 08:00-17:00	Alf Hammerseth
03-17	**Leste om database import i Azure** Hig / Norkart - Bachelor	**4:00:00** 11:00-15:00	Alf Hammerseth
03-18	**Ferdigstilt nye oppgaver og PBI'er med tasks. Sammkjøring av gruppa angående hva vi fikk som tilbakemelding på mandag som var og hvordan vi skal bevege oss framover** Hig / Norkart - Bachelor	**5:00:00** 07:30-12:30	Alf Hammerseth
03-19	**SAML inn i sharelatex. jobbe litt mer med det i morgen. Visdomstann skal man ikke spøke med, litt smerte på g her.** Hig / Norkart - Bachelor	**2:00:00** 14:00-16:00	Alf Hammerseth
03-20	**OIDC, SSO og SAML inn i sharelatex** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Alf Hammerseth
03-23	**OAuth og ferdigstilte andre biter i teorien** Hig / Norkart - Bachelor	**4:00:00** 08:00-12:00	Alf Hammerseth
03-24	**Gruppemøte om framgang i rapportskrivigen og bacheloren generelt** Hig / Norkart - Bachelor	**1:00:00** 08:00-09:00	Alf Hammerseth
03-25	**Startet på WS-federation. Ferdigstilte Azure AD. Gjorde ferdig AD** Hig / Norkart - Bachelor	**5:00:00** 08:00-13:00	Alf Hammerseth

03-26	**WS-Federation, Statusrapport 3 og så igjennom feedback fra Frode H.** Hig / Norkart - Bachelor	**3:30:00** 10:00-13:30	Alf Hammerseth
04-08	**Forberedelser til møtene i sprint 3.** Hig / Norkart - Bachelor	**1:30:00** 16:00-17:30	Alf Hammerseth
04-09	**Retrospektivt møte og endringsmøte for sprint 3. gjennomførte også planleggingsmøte for sprint 4 og startet sprint 4. Hadde et mindre Google Hangout møte med Einar og Håkon for å få litt tilbakemeldinger om det skulle være noe usikkerheter** Hig / Norkart - Bachelor	**6:30:00** 09:00-15:30	Alf Hammerseth
04-10	**Korrekturlest kap. 3** Hig / Norkart - Bachelor	**2:00:00** 09:30-11:30	Alf Hammerseth
04-13	**Møtte gruppa på biblioteket og leste om grupper og roller. Jobbet også med single og multi tennant stoff.** Hig / Norkart - Bachelor	**3:30:00** 08:00-11:30	Alf Hammerseth
04-14	**Korrekturlest kap. 1. Jobbet mye med kapittel 5 og lest mye om grupper og roller, singel og multi tenant i AAD. Skrev utkast til grupper og roller for gjennomgang i morra.** Hig / Norkart - Bachelor	**8:00:00** 07:30-15:30	Alf Hammerseth
04-15	**Leste igjennom grupper og roller. Ferdigstilte grupper og roller. Tok tak i masse og enkelt registrering av brukere. Mye lesing om hvilke muligheter det er. Skrev et utkast til enkelt og masse registrering av brukere og kommer til å revidere dette i morgen.** Hig / Norkart - Bachelor	**8:00:00** 07:30-15:30	Alf Hammerseth
04-16	**Ferdigstilte masse og enkelt registrering. Skrev forslag til lisensieringsmodell. skrev Katana og OWIN forslag.** Hig / Norkart - Bachelor	**8:00:00** 07:30-15:30	Alf Hammerseth
04-17	**Skisserte opp forslag til konklusjonstruktur. skrev mer på OWIN. Fikk feedback fra PC om lisensiering og ferdgistilte delkapitlet.** Hig / Norkart - Bachelor	**1:30:00** 08:30-10:00	Alf Hammerseth
04-20	**Så over konklusjonstruktur. Lagde utkast for App proxy. utkast for egenadm og brukeradm.** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
04-21	**Reviderte SAML og litt Oauth. Korrekturleste kap. 5. Planla litt uka. Endelig utkast til struktur på konklusjonskappittelet** Hig / Norkart - Bachelor	**4:30:00** 10:00-14:30	Alf Hammerseth
04-22	**Begynte på testing delen av rapporten. Gikk igjennom operasjonelle krav og testa litt. Se over det i morra og sjekk med gruppa klokka 10.** Hig / Norkart - Bachelor	**6:00:00** 08:00-14:00	Alf Hammerseth
04-23	**Jobba med testing og gjennomførte WCAG med standpunkt mot Azure AD** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth
04-24	**Forberedelse til rapport gjennomgang, leste på rapporten** Hig / Norkart - Bachelor	**2:00:00** 13:00-15:00	Alf Hammerseth
04-25	**Gruppegjennomgang av rapporten.** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Alf Hammerseth
04-26	**La inn korrekturlesing og endringer jeg hadde etter gjennomgang av rapporten.** Hig / Norkart - Bachelor	**1:00:00** 14:00-15:00	Alf Hammerseth
04-27	**Ferdigstilte utkast til WCAG, lagde vedlegg, leste igjennom tutorials og jobbet mye med gruppen. Reviderte kravspek ilag med gruppa** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Alf Hammerseth

04-28	**Reviderte struktur i konklusjon etter møte med Frode og Eigel. Fiksa testing, og korrekturleste 2 ganger igjennom gammal kravspek og design** Hig / Norkart - Bachelor	**8:00:00** 08:30-16:30	Alf Hammerseth
04-29	**Jobbet med testingskapitlet og planlagg testing av norkart ansatte, gjennomførte dette og gjorde klar for testing av andre medstudenter den 29.04. Skrev forslag til statusrapport 4** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Alf Hammerseth
04-30	**Gjennomført sluttbruker testing og skrevet forslag under funksjonell testing i kap for testing.** Hig / Norkart - Bachelor	**6:00:00** 08:00-14:00	Alf Hammerseth
05-01	**Jobbet med testingkapitlet. Gjennomførte funksjonelle tester og jobbet med gruppen. Jobbet så med de ulike revideringstaskene i ulike kapitler.** Hig / Norkart - Bachelor	**7:00:00** 10:00-17:00	Alf Hammerseth
05-02	**Revidering av kap. 3 og kap. 5. også jobbet med gruppen om diskusjon og drøfting i slutten av test kapittel.** Hig / Norkart - Bachelor	**6:30:00** 10:00-16:30	Alf Hammerseth
05-03	**Siste revideringene i kap. 5 og ferdigstilte utkast til MyApps i kap. 3** Hig / Norkart - Bachelor	**1:00:00** 08:30-09:30	Alf Hammerseth
05-04	**Korrekturleste igjennom kapittel 1-5 og korrigerte feil + fikk innholdsoversikt.** Hig / Norkart - Bachelor	**9:00:00** 08:30-17:30	Alf Hammerseth
05-05	**Leste igjennom resten av rapporten** Hig / Norkart - Bachelor	**6:00:00** 08:00-14:00	Alf Hammerseth
05-06	**Gikk igjennom avslutningskapittel med gruppa og skrev utkast** Hig / Norkart - Bachelor	**9:00:00** 08:00-17:00	Alf Hammerseth
05-07	**Reviderte avslutningskapittel, jobbet med revidere i ulike deler av rapporten** Hig / Norkart - Bachelor	**6:00:00** 08:00-14:00	Alf Hammerseth
05-08	**La inn endringer fra feebacken til Anna og Synne. Total reviderte kapittel 4. Identifiserte også tasker for å få en oversikt over hva som gjenstår i bacheloren** Hig / Norkart - Bachelor	**7:00:00** 10:00-17:00	Alf Hammerseth
05-09	**Leste igjennom rapport til Synne og Anna. Gjorde litt mindre oppgaver, rettet OAuth til OAuth 2.0** Hig / Norkart - Bachelor	**4:00:00** 16:00-20:00	Alf Hammerseth
05-10	**Korrekturleste gammel krav spek og identityserver3. Utarbeidet kode forslag om PowerShell og Graph API.** Hig / Norkart - Bachelor	**5:00:00** 10:00-15:00	Alf Hammerseth
05-11	**Høytlesning av kapittel 1 og definerte mye om estikk til rapport. Fullførte operasjonelle krav, WCAG revidering.** Hig / Norkart - Bachelor	**10:30:00** 08:00-18:30	Alf Hammerseth
05-12	**Høytlesning av rapporten. kom til kapittel 5.** Hig / Norkart - Bachelor	**10:00:00** 08:00-18:00	Alf Hammerseth
05-13	**Ferdig å lese igjennom rapporten høyt og gjorde noen små tasker** Hig / Norkart - Bachelor	**11:30:00** 08:00-19:30	Alf Hammerseth
05-14	**Ferdigstille rapporten og levere inn.** Hig / Norkart - Bachelor	**3:00:00** 09:00-12:00	Alf Hammerseth

Created with toggl.com

F.3.3 Arbeidstid for Ida

Detailed report

toggl

2015-01-05 - 2015-05-15
Total 461 h 00 min

Idagranholt selected as users
Bachelor selected as projects

Date	Description	Duration	User
01-07	**Første gruppemøte, forberedte spørsmål til veileder. Møte med veileder. Fokus på oppgave og kravspek.** Hig / Norkart - Bachelor	**3:00:00** 08:30-11:30	Idagranholt
01-08	**Lynkurs i prosjkthåndtering. Gruppemøte, bestemte verktøy og valgte gruppeleder** Hig / Norkart - Bachelor	**2:30:00** 13:00-15:30	Idagranholt
01-09	**Møte med oppdragsgiver, drøftet oppgavebeskrivelsen** Hig / Norkart - Bachelor	**4:30:00** 11:00-15:30	Idagranholt
01-10	**Oppdatering av OneNote** Hig / Norkart - Bachelor	**0:30:00** 20:30-21:00	Idagranholt
01-11	**Latex testing, lagd mal for prosjektplan** Hig / Norkart - Bachelor	**3:30:00** 17:00-20:30	Idagranholt
01-12	**Arbeidsmøte, Gikk gjennom og fordelte oppgaver for forprosjekt. Bestemte need to have og nice to have. Reasearch av utviklingsmetoder. Lest bacheloroppgave fra i fjord** Hig / Norkart - Bachelor	**6:30:00** 09:30-16:00	Idagranholt
01-13	**Research Lean startup, Møte med veileder, språk, loggbok, refleksjonsnotat, forprosjekt. Gruppemøte, loggbok, risikoanalyse, statusmøter** Hig / Norkart - Bachelor	**3:30:00** 13:00-16:30	Idagranholt
01-14	**Research Lean Startup og andre modeller** Hig / Norkart - Bachelor	**1:30:00** 13:00-14:30	Idagranholt
01-15	**Prosjektplan og latex** Hig / Norkart - Bachelor	**2:00:00** 10:00-12:00	Idagranholt
01-19	**Arbeidsmøte, Gikk gjennom utviklingsmetode, hovedinndeling, risikoanalyse, fordelte oppgaver. Skrevet hovedinndeling, møter og utviklingsmetode** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Idagranholt
01-20	**Lagd gant diagram og lest gjennom forprosjekt. Gruppegjennomgang av Gant og forprosjekt, oppretting av VM** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Idagranholt
01-21	**Skrevet om hovedinndeling til forprosjekt, Møte med Norkart, need to have og nice to have, gant skjema, utvkilingsmetode, innføring i miljø** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Idagranholt
01-22	**Fin/korreksjonslesing av forprojekt individuelt, så i gruppe. Endring på hovedinndeling og gantt, oppprettelse av enkel nettside** Hig / Norkart - Bachelor	**6:30:00** 10:00-16:30	Idagranholt
01-23	**gjennomgang av forprojekt** Hig / Norkart - Bachelor	**1:30:00** 08:00-09:30	Idagranholt
01-26	**Research på sikkrehetsmekanismer/rutiner rundt innlogging, OpenIDConnect, OAuth og Azure AD** Hig / Norkart - Bachelor	**4:00:00** 13:00-17:00	Idagranholt

01-27	**Møte med veileder om forprosjekt, Jobbet med forprosjekt og oppsett på kravspek**	**7:00:00**	Idagranholt
	Hig / Norkart - Bachelor	12:00-19:00	
01-28	**Research OpenID Connect, levere forprosjekt og prosjektavtale**	**5:00:00**	Idagranholt
	Hig / Norkart - Bachelor	11:00-16:00	
01-29	**Arbeidsmøte, oppsett på kravspek, overordnet use case diagram, fordelte operasjonelle krav, skre statusrapport**	**4:00:00**	Idagranholt
	Hig / Norkart - Bachelor	10:00-14:00	
02-02	**Operasjonelle krav, Tankekart rundt use case, levere statusrapport,**	**5:00:00**	Idagranholt
	Hig / Norkart - Bachelor	12:00-17:00	
02-03	**Gått gjennom operasjonelle krav, utarbeidet user stories, prioritert user stories, veiledermøte**	**13:00:00**	Idagranholt
	Hig / Norkart - Bachelor	09:00-22:00	
02-04	**Gejnnomgang av user stories, videre planlegging av design og arkitekturfasen, møte med oppdragsgiver om kravspesifikasjonen**	**7:00:00**	Idagranholt
	Hig / Norkart - Bachelor	08:00-15:00	
02-05	**Research systemarkitektur, endring på utviklingsmodell, lagd møteplan for oppdragsgiver**	**3:30:00**	Idagranholt
	Hig / Norkart - Bachelor	13:00-16:30	
02-07	**Webside**	**2:00:00**	Idagranholt
	Hig / Norkart - Bachelor	17:30-19:30	
02-09	**Webside**	**2:00:00**	Idagranholt
	Hig / Norkart - Bachelor	17:30-19:30	
02-10	**Webside**	**6:30:00**	Idagranholt
	Hig / Norkart - Bachelor	12:00-18:30	
02-11	**Logo for Norkart ID, design på webside, deploye webside**	**3:00:00**	Idagranholt
	Hig / Norkart - Bachelor	17:00-20:00	
02-12	**Forslag til oppsett av arkitektur, research på 4+1 modellen og hva de forskjellige viewsene inneholder**	**2:00:00**	Idagranholt
	Hig / Norkart - Bachelor	13:30-15:30	
02-16	**Lagd development og deployment modell, sekvensdiagram for logg inn, lest om OpenID Connect arkitektur, tatt MVC med Azure tutorial**	**10:00:00**	Idagranholt
	Hig / Norkart - Bachelor	09:00-19:00	
02-17	**Gruppemøte, gått gjennom arkitektur og lagd videre plan for arkitektur, Møte med veileder, tilbakemelding på kravspesifikasjonen, Lest på Lag Arkitektur og MVC**	**8:30:00**	Idagranholt
	Hig / Norkart - Bachelor	09:00-17:30	
02-18	**Jobbet med klassediagram**	**6:00:00**	Idagranholt
	Hig / Norkart - Bachelor	09:30-15:30	
02-19	**Jobbet med PerChristian angpende arkitektur og begynt på komponent diagram**	**2:00:00**	Idagranholt
	Hig / Norkart - Bachelor	10:00-12:00	
02-20	**Ferdiggjort komponent diagram og endret på klassediagram**	**3:00:00**	Idagranholt
	Hig / Norkart - Bachelor	19:30-22:30	
02-21	**Gruppemøte for å gå gjennom arkitektur, egenarbeid med å skrive forklaring på logisk view, komponentdiagram, klassediagram, og deployment view**	**3:00:00**	Idagranholt
	Hig / Norkart - Bachelor	12:00-15:00	
02-23	**Lagd til use case for pototype og PBIer, lagt til dette og high level use case i latex, lagt til og renskrevet utviklingsmetode kapittel i latex**	**8:00:00**	Idagranholt
	Hig / Norkart - Bachelor	11:30-19:30	

02-24	**Gruppemøte, planleggingsmøte del 1 for sprint 1, valgte pbi for sprint, lagde tasks, veiledermøte, presentasjon av forprosjekt, tilbakemelding arkitektur** Hig / Norkart - Bachelor	**7:00:00** 08:00-15:00	Idagranholt
02-25	**Planleggingsmøte del2, prioritere tasks, Fant første MVP, satte sprint til 2 uker, Individuelt arbeid med arkitektur og design, (1t), research på OpenID Connect og IdentyServer3** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Idagranholt
02-26	**Klargjort TFS, Begynt på IdentityServer3 tutorial** Hig / Norkart - Bachelor	**6:30:00** 10:00-16:30	Idagranholt
03-02	**Gruppemøte, gjennomgang av Identity tutorial, forberede møte med Norkart, Møte med Norkart, research på Identity server** Hig / Norkart - Bachelor	**6:00:00** 10:30-16:30	Idagranholt
03-03	**Gruppemøte, oppsett av Azure AD, research av Azure AD vs IdentityServer3** Hig / Norkart - Bachelor	**5:00:00** 10:00-15:00	Idagranholt
03-04	**Gruppemøte, Avtalt hva vi skal gjøre videre mtp Azure AD, research av Azure AD, demo** Hig / Norkart - Bachelor	**10:00:00** 08:30-18:30	Idagranholt
03-05	**Gruppepøte om hvordan vi utføre oppgaven etter endringene, Møte med oppdragsgiver om det samme** Hig / Norkart - Bachelor	**4:30:00** 12:00-16:30	Idagranholt
03-09	**Gruppemøte planleggingav sprint 2 og fordeling av oppgaver, jeg begynte på proof of concept demo applikasjoner** Hig / Norkart - Bachelor	**8:00:00** 08:30-16:30	Idagranholt
03-10	**Veiledermøte, presenterte status og fikk tips til rapportskriving** Hig / Norkart - Bachelor	**1:30:00** 13:00-14:30	Idagranholt
03-11	**Lynkurs 2, rapportskriving** Hig / Norkart - Bachelor	**1:00:00** 12:00-13:00	Idagranholt
03-12	**Jobbet med demo applikasjoner, testet single sign on og litt single sign out** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Idagranholt
03-16	**Statusmøte med gruppa, forberede til demomøte, Demomøte 1 med Norkart. Viste frem presentasjon rundt Azure og Azure AD pluss demo av logg inn** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Idagranholt
03-17	**Testet forskjellige innloggingsscenarier på PoC appene, funnet feil ved Authorize pga SSL, Satt opp forlsag på endelig levering** Hig / Norkart - Bachelor	**6:00:00** 14:00-20:00	Idagranholt
03-18	**Endring og retrospektivt møte for spirnt 2, Planleggingsmøte for sprint 3, Diskutert hva som skal leveres i endelig levering til Norkart, funnet ut at teori, valg og en del implementasjon skal gjøres ferdig i sprint 3, Individuelt arbeid med å researche testmetoder og sette opp tester for logg inn, logg ut og resett passord** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Idagranholt
03-20	**Skrevet om hvordan lege til web apps i Azure AD og hvordan implemenetere Azure AD i en web app** Hig / Norkart - Bachelor	**2:00:00** 10:00-12:00	Idagranholt
03-23	**Opprettet webside for Norkart manual, begynt på manual fro registrering av applikasjoner i Azure AD** Hig / Norkart - Bachelor	**4:30:00** 12:30-17:00	Idagranholt
03-24	**Statusmøte** Hig / Norkart - Bachelor	**2:00:00** 08:00-10:00	Idagranholt

03-24	**Skrevet om legg til appplikasjon, ferdiggjort legg til tutorial, sett på autentisering med Azure AD** Hig / Norkart - Bachelor	**2:30:00** 15:00-17:30	Idagranholt
03-28	**Manual fo implementasjon Av AAD i web app, reseach av Organizational Acoount autentisering** Hig / Norkart - Bachelor	**4:30:00** 08:30-13:00	Idagranholt
03-29	**Skrevet om implementering av Azure i web app** Hig / Norkart - Bachelor	**1:30:00** 08:30-10:00	Idagranholt
03-30	**Ferdiggjort implementering av webapp i rapporten** Hig / Norkart - Bachelor	**3:30:00** 09:30-13:00	Idagranholt
04-08	**Skrevet om endre passord konfigueering og glemt passord** Hig / Norkart - Bachelor	**6:00:00** 09:00-15:00	Idagranholt
04-09	**Endrings og retrospektivt, fant måter for å motivere gruppa, Planleggingsmøte for sprint 4,** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Idagranholt
04-13	**Reset glemt passord, implementering av web app, tutorial web app** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Idagranholt
04-14	**Revidering av kravspesifikasjon og use case, erevidering av kapittel 5, veiledermøte** Hig / Norkart - Bachelor	**7:00:00** 10:00-17:00	Idagranholt
04-15	**High level use case beskrivelse, registrering av applikasjoner i AAD** Hig / Norkart - Bachelor	**6:30:00** 09:30-16:00	Idagranholt
04-16	**Skrevet om registrering av applikasjoner i AAD** Hig / Norkart - Bachelor	**2:30:00** 10:00-12:30	Idagranholt
04-17	**Registrere applikasjoner, Endre passord, Glemt passord, Konfigurering av passord, korrekturlese kap 5** Hig / Norkart - Bachelor	**2:30:00** 15:00-17:30	Idagranholt
04-19	**Owin, Katana, registrere applikasjoner. korrekturlesing av kap 1 og 2** Hig / Norkart - Bachelor	**4:00:00** 10:30-14:30	Idagranholt
04-20	**Statusmøte med gruppa. Skrevet om håndtering av applikasjoner og AAD** Hig / Norkart - Bachelor	**5:00:00** 08:00-13:00	Idagranholt
04-21	**Klargjort tutorial for web, Strukturert og forklart test kapittelet, sett på brukeranalyse og brukertesting** Hig / Norkart - Bachelor	**8:00:00** 09:00-17:00	Idagranholt
04-22	**Research og notater rundt brukertesting og brukervennlighetsanalyse** Hig / Norkart - Bachelor	**6:00:00** 10:00-16:00	Idagranholt
04-23	**Omskriving og gjennomlesing av kapittel 1, gruppemøte** Hig / Norkart - Bachelor	**3:00:00** 10:30-13:30	Idagranholt
04-24	**Gjennomlesing av rapport** Hig / Norkart - Bachelor	**2:00:00** 16:00-18:00	Idagranholt
04-25	**Gruppegjennomgang av rapport** Hig / Norkart - Bachelor	**4:00:00** 12:00-16:00	Idagranholt
04-27	**Reviderte kravspek, gikk gjennom roller og funksjoner, videre planlegging av testing** Hig / Norkart - Bachelor	**7:00:00** 09:00-16:00	Idagranholt

04-28	**Planlegging av brukertester** Hig / Norkart - Bachelor	**7:00:00** 08:30-15:30	Idagranholt
04-29	**Klargjøring av brukertester, brukertester utført ho Snorkart, Statusmøte hos Norkart** Hig / Norkart - Bachelor	**7:00:00** 09:00-16:00	Idagranholt
04-30	**Brukertester på sluttbruker og lokal admin, skrevet inn data fra brukertester** Hig / Norkart - Bachelor	**8:30:00** 08:30-17:00	Idagranholt
05-01	**Skrevet om brukertesting og resulatt av brukertesting** Hig / Norkart - Bachelor	**7:00:00** 12:30-19:30	Idagranholt
05-02	**Gruppemøte, gått gjenom testkapittel** Hig / Norkart - Bachelor	**3:00:00** 12:30-15:30	Idagranholt
05-03	**Skrevet avslutning på kapittel 6** Hig / Norkart - Bachelor	**3:00:00** 18:30-21:30	Idagranholt
05-04	**Skrevet om kravspek, Revidert kap 4, Lagt til testvedlegg** Hig / Norkart - Bachelor	**9:00:00** 10:00-19:00	Idagranholt
05-05	**Omstrukturert kap 7, skrivefeil kap 7, veiledermøte** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Idagranholt
05-06	**Omstrukturering av kap 10, skrevet om kritikk av oppgaven, rettet halve kap 7, lest annen rapport** Hig / Norkart - Bachelor	**8:00:00** 10:30-18:30	Idagranholt
05-07	**Skrive på avslutingskapittelet, få tilbakemelding på rapport** Hig / Norkart - Bachelor	**9:00:00** 08:00-17:00	Idagranholt
05-08	**Gikk gjennom tilbakemelding fra veiledere. Begynte å revidere test kapitlet.** Hig / Norkart - Bachelor	**6:00:00** 11:00-17:00	Idagranholt
05-09	**Revidereing av funksjonelle tester** Hig / Norkart - Bachelor	**3:00:00** 16:30-19:30	Idagranholt
05-11	**Revidere testkapitlet, høytlesing av bacheor** Hig / Norkart - Bachelor	**9:30:00** 09:00-18:30	Idagranholt
05-12	**Høytlesning av rapporten, kap 2 - 5** Hig / Norkart - Bachelor	**10:00:00** 08:00-18:00	Idagranholt
05-13	**Høytlesning av rapporten, pynting** Hig / Norkart - Bachelor	**11:30:00** 08:00-19:30	Idagranholt
05-14	**Ferdiggjøring** Hig / Norkart - Bachelor	**2:00:00** 09:00-11:00	Idagranholt

Created with toggl.com

F.3.4 Arbeidstid for Per Christian

Detailed report

toggl

2015-01-05 - 2015-05-15
Total 434 h 45 min

Per Christian Kofstad selected as users
Bachelor selected as projects

Date	Description	Duration	User
01-07	**Teambuilding, Lunsj, diskusjoner om forventninger, avklaringer og bli kjent.** Hig / Norkart - Bachelor	**5:00:00** 09:00-14:00	Per Christian Kofstad
01-08	**Lynkurs Prosjektstyring, høyskolen v/Tom Røise** Hig / Norkart - Bachelor	**2:00:00** 13:00-15:00	Per Christian Kofstad
01-09	**Møte Norkoart m/kjøring - Introduksjon, spesifisering og avgrensning av oppgaven** Hig / Norkart - Bachelor	**4:30:00** 11:00-15:30	Per Christian Kofstad
01-10	**Individuell jobb, klargjøring av verktøy, oppsett av OneNote med innhold vi allerede hadde.** Hig / Norkart - Bachelor	**1:30:00** 12:30-14:00	Per Christian Kofstad
01-12	**Arbeidsmøte, kommende ukes arbeidsmøte, og diskusjon av oppgaveavgrensning** Hig / Norkart - Bachelor	**4:30:00** 09:30-14:00	Per Christian Kofstad
01-19	**Arbeidsmøte - Gikk igjennom forrige ukes arbeid, la opp jobb for kommende uke** Hig / Norkart - Bachelor	**5:00:00** 08:00-13:00	Per Christian Kofstad
01-19	**Individuell jobb, mål, oppgavebeskrivelse og avgrensing til forprosjekt** Hig / Norkart - Bachelor	**3:00:00** 13:00-16:00	Per Christian Kofstad
01-20	**Idividuell jobb, jobbet med omfang og oppgavebeskrivelse i forprosjekt** Hig / Norkart - Bachelor	**5:15:00** 08:00-13:15	Per Christian Kofstad
01-20	**Veildermøte** Hig / Norkart - Bachelor	**0:30:00** 13:30-14:00	Per Christian Kofstad
01-20	**Arbeidsmøte - Legge plan etter veiledermøte** Hig / Norkart - Bachelor	**2:00:00** 14:00-16:00	Per Christian Kofstad
01-21	**Individuell jobb, pussing på forprosjekt** Hig / Norkart - Bachelor	**3:00:00** 08:00-11:00	Per Christian Kofstad
01-21	**Møte m/Norkart + kjøring til Lillehammer - Oppsett av verktøy og bekreftelse av oppgaveforståelse** Hig / Norkart - Bachelor	**5:00:00** 11:00-16:00	Per Christian Kofstad
01-22	**Individuell jobb, pussing på forprosjekt etter tilbakemeldinger fra Norkart** Hig / Norkart - Bachelor	**0:30:00** 10:30-11:00	Per Christian Kofstad
01-22	**Individuell jobb, forprosjekt pussing** Hig / Norkart - Bachelor	**1:00:00** 12:00-13:00	Per Christian Kofstad
01-22	**Arbeidsmøte gjennomgang forprosjektet** Hig / Norkart - Bachelor - [mobile]	**1:30:00** 13:30-15:00	Per Christian Kofstad
01-23	**Arbeidsmøte, gjennomlesning forprosjekt** Hig / Norkart - Bachelor	**1:15:00** 08:00-09:15	Per Christian Kofstad

01-27	**Individuell jobb, gjennomlesing av jobb og referater**	**1:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	18:00-19:30	
01-28	**Individuell jobb, lese på teori, jobbe med teorien**	**4:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	10:00-14:00	
01-29	**Arbeidsmøte, kravspesifikajon**	**3:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-11:00	
01-30	**Individuell jobb, kravspesifikasjon, oppdelt seksjon**	**1:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-09:30	
02-02	**Individuell jobb, kravspesifikasjon, oppdelt seksjon**	**1:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	16:30-17:30	
02-03	**Arbeidsdag med gruppen, kravspesifikasjon, felles middag, veiledermøte, user stories**	**14:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-22:00	
02-04	**Arbeidsdag med gruppen, puss på kravspesifikasjon, møte med oppdragsgiver**	**7:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-15:00	
02-11	**individuell jobb, utkast systemdesign**	**8:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-16:00	
02-16	**individuell jobb, testing av visualStudio Toturials for C# og azure**	**6:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	09:30-16:00	
02-17	**Arbeidsmøte og individuelt arbeid, research av 4+1 process view**	**3:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-11:00	
02-17	**Veildermøte**	**0:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	14:00-14:30	
02-18	**Individuell jobb, Process view**	**3:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	12:00-15:00	
02-19	**Individuelt og jobbing med Ida, desing og kravspesifikasjon**	**5:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-13:30	
02-21	**Gruppemøte, designarbeid**	**1:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	12:00-13:30	
02-21	**Individell jobb, designarbeid, forberedelse av utkast til innsending**	**1:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	16:00-17:30	
02-22	**Individuell jobb, redigering utkast design**	**1:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	10:00-11:00	
02-23	**Individuell jobb, utkast presentasjon**	**2:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-10:00	
02-23	**Individuell Jobb, research**	**2:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	14:00-16:00	
02-24	**Gruppearbeid, engelsk presentasjon #1 MVP**	**8:30:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-16:30	
02-25	**Gruppearbeid, #MVP1 møte**	**2:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-10:00	
02-26	**Individuell Jobb, rapportformattering og pussing**	**5:00:00**	Per Christian Kofstad
	Hig / Norkart - Bachelor	08:00-13:00	

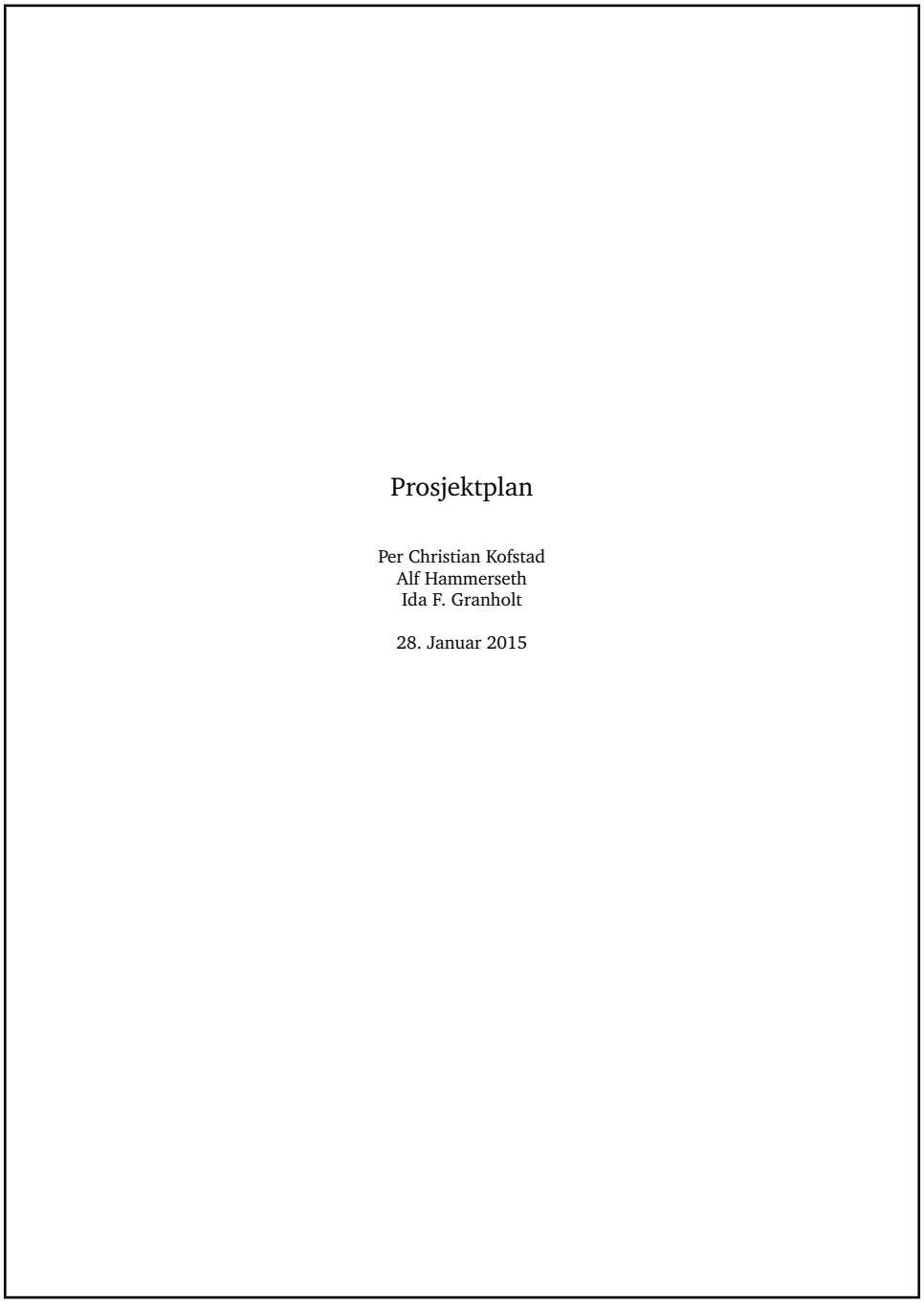
02-28	**Individuell jobb, tutorial** Hig / Norkart - Bachelor	**3:00:00** 09:00-12:00	Per Christian Kofstad
03-02	**Gruppe jobb, tutorial, møte med Norkart** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Per Christian Kofstad
03-03	**Gruppearbeid, research azure ad og openID Connect** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Per Christian Kofstad
03-04	**Gruppearbeid, research azure ad og openID Connect** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
03-04	**Individuell jobb, midlertidig rapport til Norkart** Hig / Norkart - Bachelor	**3:00:00** 16:30-19:30	Per Christian Kofstad
03-04	**Individuell jobb, midlertidig rapport til Norkart** Hig / Norkart - Bachelor	**1:00:00** 21:00-22:00	Per Christian Kofstad
03-05	**Gruppearbeid, norkartmøte og research** Hig / Norkart - Bachelor	**8:00:00** 08:30-16:30	Per Christian Kofstad
03-09	**Gruppearbeid og individuelt arbeid - MVP2 definering, oppgavestruktur og research av lisensiering** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Per Christian Kofstad
03-10	**Individuelt arbeid, og veiledermøte - Jobbet med rapportstruktur og rapportskrivingsprosesser** Hig / Norkart - Bachelor	**7:00:00** 08:00-15:00	Per Christian Kofstad
03-11	**Individuelt arbeid, research achure ad b2c** Hig / Norkart - Bachelor	**3:00:00** 08:00-11:00	Per Christian Kofstad
03-11	**Lynkurs 2** Hig / Norkart - Bachelor	**1:00:00** 12:00-13:00	Per Christian Kofstad
03-12	**Omvisning, Skolebesøk** Hig / Norkart - Bachelor	**2:00:00** 11:00-13:00	Per Christian Kofstad
03-16	**Gruppearbeid og demomøte 1** Hig / Norkart - Bachelor - [mobile]	**8:30:00** 08:00-16:30	Per Christian Kofstad
03-17	**Individuelt arbeid, rapport** Hig / Norkart - Bachelor	**6:30:00** 08:00-14:30	Per Christian Kofstad
03-18	**Gruppemøter avslutte sprint2 oppstart sprint3** Hig / Norkart - Bachelor	**5:00:00** 08:00-13:00	Per Christian Kofstad
03-18	**Individuelt arbeid, single and multi tenant reserach** Hig / Norkart - Bachelor	**2:30:00** 14:00-16:30	Per Christian Kofstad
03-19	**Individuelt arbeid, rapport, single-mulit tenant og implementasjonsaspekter ved implementasjon** Hig / Norkart - Bachelor	**2:30:00** 08:00-10:30	Per Christian Kofstad
03-19	**Omvisning, Skolebesøk** Hig / Norkart - Bachelor	**2:00:00** 11:00-13:00	Per Christian Kofstad
03-23	**Individuelt arbeid. rapport skriving, valg av løsning** Hig / Norkart - Bachelor	**6:00:00** 08:00-14:00	Per Christian Kofstad
03-24	**Gruppemøte, statusmøte** Hig / Norkart - Bachelor	**2:00:00** 08:00-10:00	Per Christian Kofstad
03-26	**Individuelt arbeid. rapport skriving, graph api** Hig / Norkart - Bachelor	**1:00:00** 12:00-13:00	Per Christian Kofstad

04-04	**Individual work, report, status report, commenting on other groupmemebers work** Hig / Norkart - Bachelor	**4:00:00** 13:30-17:30	Per Christian Kofstad
04-08	**Individual work, report chapter 1 and editing** Hig / Norkart - Bachelor	**5:30:00** 09:00-14:30	Per Christian Kofstad
04-09	**Gruppemøte og planlegging** Hig / Norkart - Bachelor	**5:00:00** 10:30-15:30	Per Christian Kofstad
04-11	**Individuelt arbeid, android demo app** Hig / Norkart - Bachelor	**5:00:00** 10:00-15:00	Per Christian Kofstad
04-13	**individuelt arbeid, rapport, kapittel 1** Hig / Norkart - Bachelor	**5:30:00** 08:00-13:30	Per Christian Kofstad
04-13	**Individuelt arbeid, android demo app** Hig / Norkart - Bachelor	**1:00:00** 20:30-21:30	Per Christian Kofstad
04-14	**Individuelt arbeid og veiledermøte** Hig / Norkart - Bachelor	**7:00:00** 08:00-15:00	Per Christian Kofstad
04-15	**Individuelt arbeid, rapport, toturial, struktur, implementasjon** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
04-16	**Individuelt arbeid, rapport, kapittel 5** Hig / Norkart - Bachelor	**3:45:00** 08:00-11:45	Per Christian Kofstad
04-17	**Individuelt arbeid, rapport, generell azure ad konfigurasjon** Hig / Norkart - Bachelor	**1:30:00** 08:00-09:30	Per Christian Kofstad
04-20	**Individuelt arbeid, rapport, implemntasjon-innloggingsmekanismer** Hig / Norkart - Bachelor	**6:30:00** 09:00-15:30	Per Christian Kofstad
04-20	**Individuelt arbeid, rapport, implemntasjon-innloggingsmekanismer** Hig / Norkart - Bachelor	**2:00:00** 19:00-21:00	Per Christian Kofstad
04-21	**Individuelt arbeid, rapport, kap 3** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
04-22	**Individuelt arbeid, rapport, kap 3** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Per Christian Kofstad
04-23	**Individuelt arbeid, rapport, kap 1, møte og gjennomlesning** Hig / Norkart - Bachelor	**3:00:00** 08:00-11:00	Per Christian Kofstad
04-23	**Individuelt arbeid, rapport, gjennomlesning** Hig / Norkart - Bachelor	**1:00:00** 12:00-13:00	Per Christian Kofstad
04-24	**Individuelt arbeid, rapport, gjennomlesning** Hig / Norkart - Bachelor	**1:00:00** 08:00-09:00	Per Christian Kofstad
04-24	**Individuelt arbeid, rapport, gjennomlesning** Hig / Norkart - Bachelor	**1:00:00** 14:30-15:30	Per Christian Kofstad
04-25	**Gruppe arbeid, rapportgjennomgang** Hig / Norkart - Bachelor	**4:30:00** 11:30-16:00	Per Christian Kofstad
04-27	**Individuelt arbeid, rapport retting** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
04-27	**Individuelt arbeid, rapport kap 1** Hig / Norkart - Bachelor	**1:30:00** 18:30-20:00	Per Christian Kofstad

04-28	**Individuelt arbeid, rapportskriving, kap1, testing** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Per Christian Kofstad
04-29	**Individuelt arbeid, rapportskriving, og gruppearbeid brukertesting** Hig / Norkart - Bachelor	**8:00:00** 08:00-16:00	Per Christian Kofstad
04-29	**Individuelt arbeid, gjennomgang av todo, og nye figurer og beskrivelser i innloggingsmekanismer** Hig / Norkart - Bachelor	**2:30:00** 18:00-20:30	Per Christian Kofstad
04-30	**Gruppe arbeid, brukertesting** Hig / Norkart - Bachelor	**4:00:00** 08:00-12:00	Per Christian Kofstad
05-01	**Individuelt arbeid, funksjonstester og avslutningskapittel** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Per Christian Kofstad
05-02	**Individuelt arbeid, ws-federation, avslutningskapittel, gruppediskusjon testing** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
05-04	**Individuelt arbeid, avslutningskapittel** Hig / Norkart - Bachelor	**6:00:00** 09:00-15:00	Per Christian Kofstad
05-04	**Individuelt arbeid, revidering kap 3** Hig / Norkart - Bachelor	**2:00:00** 18:00-20:00	Per Christian Kofstad
05-05	**Individuelt arbeid, revidering og avslutningskapittel** Hig / Norkart - Bachelor	**8:30:00** 08:00-16:30	Per Christian Kofstad
05-05	**Individuelt arbeid, gjennomlesning samarbeidsgruppe** Hig / Norkart - Bachelor	**2:00:00** 20:30-22:30	Per Christian Kofstad
05-06	**Individuelt arbeid og gruppearbeid, avsluttende kapittel og vedlegg** Hig / Norkart - Bachelor	**7:30:00** 08:00-15:30	Per Christian Kofstad
05-07	**Individuelt arbeid, avsluttningskapittel** Hig / Norkart - Bachelor	**4:00:00** 08:00-12:00	Per Christian Kofstad
05-07	**Individuelt arbeid, veiledermøte, vedlegg, forrord, abstract** Hig / Norkart - Bachelor	**4:00:00** 14:00-18:00	Per Christian Kofstad
05-08	**Individuelt arbeid, abstrakt, forrord, vedlegg** Hig / Norkart - Bachelor	**1:30:00** 10:30-12:00	Per Christian Kofstad
05-08	**Individuelt arbeid, diverse redigering av rapporten** Hig / Norkart - Bachelor	**4:30:00** 13:00-17:30	Per Christian Kofstad
05-09	**Individuelt arbeid, diverse redigering av rapporten** Hig / Norkart - Bachelor	**6:00:00** 09:00-15:00	Per Christian Kofstad
05-10	**Gruppemøte** Hig / Norkart - Bachelor	**2:30:00** 11:30-14:00	Per Christian Kofstad
05-11	**Gruppemøte Høytlesning av rapport** Hig / Norkart - Bachelor	**10:30:00** 08:00-18:30	Per Christian Kofstad
05-12	**Gruppemøte Høytlesning av rapport** Hig / Norkart - Bachelor	**10:00:00** 08:00-18:00	Per Christian Kofstad
05-13	**Gruppemøte Høytlesning av rapport** Hig / Norkart - Bachelor	**11:30:00** 08:00-19:30	Per Christian Kofstad
05-14	**Siste puss før innlevering.** Hig / Norkart - Bachelor	**3:00:00** 08:00-11:00	Per Christian Kofstad

G Forprosjekt

Forprosjektet er inkludert i sin helhet og ligger side for side.



Prosjektplan

Per Christian Kofstad
Alf Hammerseth
Ida F. Granholt

28. Januar 2015

1 Mål og rammer

1.1 Bakgrunn

Norkart har i dag flere ulike tjenester og programvarepakker de tilbyr sine kunder. Nesten samtlige av løsningene deres krever autentisering på brukernivå ved daglig bruk. Administrasjon av tilgangsstyring tar i dag mye tid og er tungvint. Norkart ser at de bør over på en mer sentralisert løsning. Norkart ønsker noe som er enklere å administrere og enklere for brukere å forholde seg til, i tillegg ønsker de å se på mulighetene for å øke sikkerheten ytterligere rundt sine applikasjoner. En bruker og et passord for alle løsninger. Norkart ønsker å se på muligheten for å beholde autentiseringen på tvers av programvare og på tvers av sesjoner. Norkart ønsker å gjøre noe med dette på sikt, men ser på dette prosjektet som en potensiell kickstart for å komme igang, kartlegge behovene og få et innblikk i kompleksiteten. De ønsker seg et konsept de kaller NorkartID. En brukerkonto som følger brukeren og som kan brukes på alle tjenester Norkart tilbyr såfremt brukeren har tilgang og riktige rolletildelinger. Norkart utvikler, drifter og bruker i dag nesten utelukkende Microsoft teknologi. Det er da naturlig at utviklingen gjøres på Microsoft produkter rettet mot Microsoft produkter.

1.2 Prosjektmål

Prosjektmålene er sammensatt av resultatmål og effektmål, i tillegg har vi definert noen læringsmål.Prosjektmålene har bachelorgruppen definert selv, med utgangspunkt i problemstilling og oppgaven.

1.2.1 Resultatmål

Resultatmål er det som konkret skal foreligge når prosjektet er ferdig. Det prosjektet har ført til, og som kan overleveres som status og produkt for oppdragsgiver.

- Kartlegge teknologier og muligheter innenfor definerte rammer gitt av Norkart for å løse prosjektoppgaven.
- Introdusere Norkart for gitte teknologier i forhold til bruk for NorkartID: Azure, Azure AD og OpenID Connect.
- Utarbeide designskisser for NorkartID til Norkart
- Utvikle prototype av NorkartID med tilhørende elementer.
- Kickstarte utviklingsprosjektet NorkartID for Norkart.
- Tilrettelegge for økt sikkerhet i forbindelse med brukerhåndtering og tilgangsstyring av tjenestene til Norkart.

1.2.2 Effektmål

Effektmål er det oppdragsgiver ønsker å oppnå med prosjektet, etter at prosjektet er ferdig, er implementert og har fått kjørt seg inn. Dette er resultater man ikke nødvendigvis ser umiddelbart etter prosjektslutt, men som vil bli synlige etter at brukerne og systemet er innkjørt.

- Tilrettelegge for å redusere total tid brukt på innlogging på Norkart-tjenster for bruker
- Tilrettelegge for forenklet innlogging til tjenester for bruker ved å kun ha et brukernavn og passord på alle tjenester.
- Tilrettelegge for redusere tid brukt på administrasjon av brukere ved bruk av Azure AD, samt å kun ha en brukerdatabase for alle tjenester.
- Tilrettelegge for økt sikkerhet rundt brukerhåndtering av tjenester levert av Norkart.

1.2.3 Læringsmål

Læringsmål er ønsket oppnådd kompetanse etter endt prosjekt.

- Lære om informasjonsikkerhet relativt til autentiseringstjenester, innlogging og utfordringer rundt brukeradministrasjon.
- Bygge kunnskap og erfaringer ved bruk av Azure, Azure AD (Active Directory), C# og Team Foundation Server(TFS) .
- Bygge kunnskap og erfaringer om løpende distribusjon av kode underveis i et utviklingsprosjekt.
- Bygge kunnskap og erfaringer i bruk av vitenskapelig metode for utvikling av system og forskning på teknologi.
- Lære å utvikle designskisser av prototype
- Bygge erfaringer med å jobbe på tvers av linjer og på tvers av tidligere erfaringer innenfor et sammensatt prosjekt.

1.3 Rammer

I utførelsen av dette prosjektet er det satt noen rammer og krav fra Norkart og Høgskolen i Gjøvik (HiG).

Rammene Norkart har definert:

- Utvikle i Azure.
- Bruke ASP.NET MVC som utviklingsstruktur.
- Benytte OpenID Connect og OAuth 2.0 som autentiseringsteknologier i NorkartID.
- Anvende TFS som verktøy for versjonskontroll.
- Møte hos Norkart ved hver milepæl.

HiG sine rammer:

- Vi har fått utgitt veiledere fra HiG.
- Bachelorprosjektet må tilsvare 20 studiepoeng, dette tilsvarer 30 timer i uken.
- Bachelor er satt til å fullføres fra 7.januar til 15.mai.
- Det må leveres tre statusrapporter til veiledere i prosjektperioden.
- Forprosjekt leveres innen 28.januar.
- Hovedrapporten leveres innen 15.mai.

2 Omfang

Oppgaven gitt fra Norkart er spesifisert ned til teknologivalg, hva som skal implementeres, hva som skal støttes og på hvilken plattform det skal utvikles. Hvor mye bachelorgruppen ser for oss å rekke i løpet av prosjektperioden er opp til oss å definere. Både Norkart og bachelorgruppen ønsker å definere noen tydelige minimumskrav og forventninger til hva som skal lages. Om gruppen ser at det blir tid til å bygge mer på løsningen enn hva som er definert så står vi fritt til å gjøre dette.

2.1 Fagområde

Fagområdet for oppgaven defineres som informasjonssikkerhet relatert til autentisering og autentiseringsmekanismer. Med dette mener vi ulike rutiner og tiltak som er satt ved innlogging, fysisk sikkerhet, passordhåndtering og brukermentalitet. Altså, sjekke at entiteten [1] gir fra seg en identitet som er faktisk tilhører entiteten.

2.2 Avgrensing

Vi skal tilpasse og konfigurere en autentiseringsløsning for Norkart. For å gjøre dette må vi sette oss inn i en rekke teknologier. Vi må som et minimum se nærmere på OpenID Connect, med OAuth 2.0, samt en rekke prinsipper og mekanismer for autentisering.

Prototypen av autentiseringsløsningen skal bruke OpenID Connect protokollen med OAuth 2.0 og SingleSignOn. Løsningen skal utvikles i og integreres i et Microsoft Azure miljø. Prototypen skal skrives i C# og avgrenses til å fokusere på autentisering framfor autorisering i første omgang. Bachelorgruppen skal ikke definere sikkerhetsregime i forhold til rolletildelinger innenfor systemet, men kan, dersom det blir tid, tilrettelegge for tildeling av roller innenfor systemet. Bacheloroppgaven skal initiellt fokusere på selve serverkonfigurasjonen med tilhørende webgrensesnitt for testing av autentiseringsfunksjonalitet.

2.3 Oppgavebeskrivelse

Oppgaven går ut på å designe, spesifisere og utvikle rammene for en ny autentiseringsløsning for Norkart. Dette skal gjøres med utgangspunkt i eksisterende biblioteker. Løsningen skal på sikt bli til NorkartID og skal være et SingleSignOn brukerhåndteringssystem som tillater brukere å ha et brukernavn og passord for alle tjenestene og programmene levert av Norkart. Som en del av prosjektet skal vi lage en prototype som et proof of concept innenfor de spesifiserte teknologiene som er gitt av Norkart. Løsningen skal på sikt driftes på dedikerte servere i driftsmiljøet til Norkart. Den skal kunne brukes uavhengig av plattform, skjermstørrelse og programvare, dette vil gjøre at sluttproduktet krever støtte for standardprotokoller for autentisering. Det er et mål for bachelorgruppen å gjøre oppgaven så god at Norkart kan bruke dette som utgangspunkt når de videreutvikler NorkartID.

[1] Med entitet mener vi her sluttbruker, applikasjoner eller maskiner

For å gi oss selv et leveransekrav samt gi oss noe å strekke oss etter om vi skulle rekke mer, har vi satt opp en veiledende prioriteringsliste. Denne kan endres underveis i prosjektet. Initiellt er listen prioritert etter rekkefølge. Listen er også delt i to, hvor første del er elementer vi skal ha med, og del to er elementer vi tar med om vi får tid til mer.

Skal:

- Konfigurasjon av server med OpenID Connect implementasjon.
- Webside for testing av OpenID Connect på en testwebside
- Autentisering av brukere med sikkerhetsfunksjonalitet ift. påloggingsbeskyttelse
- OpenID Connect API[2]
- Webside for testing av OpenID Connect API
- Webside for testing av OpenID Connect med egenadministrasjon av bruker.
- Autentisering av nettleser/hele maskin/klient

Ved tid:

- Test av API mot annen teknologi enn webservere
- Backend for administrasjon av brukere
- Webside for administrasjon av brukere og tilgang av brukerdatabasen
- API for egenadministrasjon av bruker

Det er flere ting vi kunne ført på listen. Dersom ting skulle vise seg å ta kort tid, er både vi og oppdragsgiver åpne for å kunne legge på mer elementer og funksjonalitet.

[2]API er et akronym som står for Application Programming Interface, på norsk applikasjonsprogrammeringsgrensesnitt. Det fungerer som et grensesnitt i en programvare slik at deler av denne kan kjøres (kalles) fra en annen programvare.

3 Prosjektorganisering

3.1 Ansvarsforhold og roller

Gruppeleder er Alf Magnus Kittang Hammerseth og hans rolle vil være å holde oversikt over gruppen samt se til at det som skal gjøres blir gjort. Ved konflikter og uenighet vil veileder bli kalt inn for å ta avgjørelsen sammen med gruppen. Ida F. Granholt er lokal informasjonsforvalter og håndterer informasjonsflyten innad i gruppen, både mot arbeidsgiver og høgskolen. Per Christian Kofstad fungerer som møteleder ved gruppemøter og er teknologiansvarlig. Han passer på at vi forholder oss til den satte agendaen og har stått for opplæring av de andre medlemmene innenfor de ulike administrative verktøyene vi bruker.

Overordnet ansvar for at punktene ovenfor blir overholdt faller på prosjektlederen. Ved eventuelle behov for signaturer kommer prosjektleder til å signere for gruppen og stå som ansvarlig.

3.1.1 Øvrige roller

Oppdragsgiver
Navn: Håkon Sagehaug, NorKart, Scrum Master
Epost: hakon.sagehaug@norkart.no

Navn: Einar Tomter, NorKart, Product Owner
Epost: einar.tomter@norkart.no

Veiledere
Navn: Frode Haug
Epost: frode.haug@hig.no

Navn: Eigil Obrestad
Epoest: Eigilo@hig.no

3.2 Regler og rutiner i gruppa

Grupperegler er viktig å utvikle tidlig slik at vi alle har samme grunnlag når gruppene beveger seg sammen mot prosjektmålene. Dette er regler vi følger fortløpende og er viktig å ha i bakhånd om det skulle skje noe med gruppen.

3.2.1 Regler

Hvert gruppemedlem har sagt seg enig i disse reglene:

1. Møte til angitt arbeidtid og jobbe i snitt minst 30 timer ukentlig.
2. Melde i fra om eventuelle frafall god tid i forveien rundt avtalte tidsfrister og lignende
3. Opparbeide og sette seg inn i nok litteratur til å gjennomføre oppgaven.
4. Holde arbeidsloggen og andre arbeidsverktøy oppdatert til en hver tid slik at framdrift blir dokumentert slik som det skal.

Om et medlem i gruppen ikke skulle følge disse reglene vil det settes inn sanksjoner mot vedrørende medlem. Det påpekes i første runde at om det ikke gjøres tilstrekkelig innsats, vil man få en sjanse til for å innhente seg. Skulle dette ikke innfris taes det kontakt med veileder angående hva som skal gjøres med vedkommende da alle parter involvert. Det vil si gruppen og medlemmet det gjelder møter opp og tar den vanskelig avgjørelse på hva som skal skje fremover. I tillegg ser vi for oss at veileder er med på å avgjøre hvordan avskjedigelsen skal gjennomføres.

Fellesansvar for gruppen er:

1. Tidsfrister blir respektert
2. At tidsloggen og møter blir dokumentert og gjennomgått ved neste samling for å finne ut status for framdrift.
3. Ved eventuelle kostnader blir disse fordelt på alle medlemmer i gruppen, om ikke annet blir avtalt.

3.2.2 Rutiner

Definerte rutiner vi skal bruker i dette prosjektet:

1. Bruk av versjonskontroll, og kun deploye kode som er gjennomgått
2. Implementering av nye funksjoner skal skje i branches[1] slik at kun fungerende og testet funksjonalitet blir lagt inn i master-branchen[2].
3. Følge C# kode standarder
4. Kommentere og dokumentere kode underveis

[1] En branch er et objekt under revisjon. Om du brancher ut fra dette lager du en kopi. Dette muliggjør at du kan gjøre endringer på begge objektene parallelt og samkjøre når en av dem er ferdigstilt.
[2] Hovedobjektet i oppgaven.

4 Planlegging, oppfølging og rapportering

4.1 Hovedinndeling av prosjektet

For å få en oversikt over prosjektet har vi delt det inn i tre hoveddeler.

- Kunnskapsløft som resulterer i informasjon til oppdragsgiver om de forskjellige teknologiene brukt under utvikling av prototypen.
- Design og utvikling av en prototype av NorkartID
- Utarbeiding av rapport som inneholder dokumentasjon av prosessen, utviklingen og resultatene av prosjektet.

Ettersom oppdraget skal ferdiggjøres på kun noen måneder har vi valgt å bruke en smidig utviklingsmodell. Med en smidig utviklingsmodell bruker vi iterasjoner for å hele tiden kunne endre på prototypen uten at det får for store konsekvenser. Fordi utviklingstiden for oppdraget er relativt kort anbefalte oppdragsgiver oss å bruke elementer fra Lean Startup. Vi kommer derfor til å bruke Minimum Viable Product (MVP)[1] og iterasjoner fra Lean Startup. Med Lean Startup kan vi sørge for at vi tidlig og ofte har en fungerende prototype som kan leveres til oppdragsgiver. Vi kommer i tillegg til å bruke elementer fra Scrum siden alle i gruppen er godt kjent med Scrum og oppdragsgiver bruker modellen mye i deres utviklingsprosjekter. I hovedsak kommer vi til å bruke rollene, møter og product backlog fra Scrum.

4.2 Plan for statusmøter og beslutningspunkter

Gruppen kommer til å ha møte med veiledere en gang i uken, så fremt det er nødvendig. Vi kommer også til å benytte oss av møtene som hører til Scrum. I begynnelsen av hver iterasjon vil vi ha et planleggingsmøte hvor vi bestemmer hvilke oppgaver som skal gjøres i oppkommende sprint. Disse oppgavene må til sammen utgjøre en MVP. I enden av hver iterasjon vil vi holde demomøter for oppdragsgiver og veileder hvor det gis feedback på prototypen. Etter demomøte vil det bli holdt et vurderingsmøte innad i gruppen hvor det bestemmes om produktet må endres i neste iterasjon eller om nye funksjoner kan legges til. Vi kommer også til å ha et retrospektivt møte hvor vi går gjennom utviklingsprosessen som har vært og gjør eventuelle endringer til neste iterasjon.

Hver mandag vil gruppen møtes for å ha statusmøte for å få et overblikk over hvordan vi ligger an i prosjektet. Vi kommer også til å benytte oss av Scrums daglige møter slik at alle i gruppen vet hvordan alle ligger an og om noen trenger hjelp.

[1] En versjon av produktet som dekker kundens behov og går gjennom en Lean Startup itearsjon med minst mulig bruk av ressurser og tid

5 Organisering og kvalitetssikring

5.1 Dokumentasjon, standardbruk og kildekode

Høgskolen i Gjøvik har gitt ut en LaTeX mal for bruk til større oppgaver. Denne malen tar vi i bruk for å sikre at vi møter de krav som stilles til format og presentasjon. Relatert med standardbruk ser vi for oss å forholde oss til de internasjonale retningslinjene og standardene for C#. Disse vil bli brukt for å veilede oss når vi tar i bruk programmeringsspråket. Kommentering underveis samt gode funksjonsnavn er hvordan vi planlegger å dokumentere koden vi skriver. Norkart har vi gitt oss tilgang til hver vår virtuell datamaskin som er satt opp med muligheter til å jobbe sammen mot en felles nettside. Denne siden kommer til å representere prototypen vi kommer til å utvikle. Vi bruker Microsoft Visual Studio som utviklingsmiljø. Det er satt opp slik at når vi er klare for å bygge og teste, blir dette dyttet inn på nettsiden og dermed kan alle i gruppen se endringene. Vi skal kommentere koden underveis som den produseres. I tillegg skal funksjonsnavn tydelig beskrive hva en funksjon gjør. Ved å etterstrebe dette vil vi få oppnå en viss likhet i koden som blir produsert.

5.2 Konfigurasjonsstyring

5.2.1 Verktøy

Under finner du en liste med oversikt over hvilke verktøy vi bruker i dette prosjektet:

- ShareLatex
- Google Disk
- Toggl
- Trello
- Microsoft OneNote
- Microsoft Visual Studio

Utvikling og testing skjer igjennom Windows Azure .

5.2.2 Versjonskontroll

Norkart bruker TFS for kildekode håndtering av de ulike versjonene som vi kommer til å opparbeide oss når vi jobber mot å utvikle det vi skal. Google Disk har sin egen versjonkontroll innebygd så her trengte vi ikke lete etter noe annet alternativ. Innen ShareLaTeX er det også innebygd versjonkontroll som vi tar nytte av når vi produserer rapporten. Vi ser for oss at vi også setter opp egne repositories når vi går innom de ulike teknologiene for eksperimenteringsformål.

5.2.3 Teknologier

Her er det en liste av språk og teknologier vi har tenkt å bruke:

- JavaScript
- Azure AD
- C#
- HTML5
- PHP
- OpenID Connect
- OAuth 2.0

5.3 Risikoanalyse

For å undersøke om det verdt å gjennomføre et prosjekt utførte vi en risikoanalyse. Ved å gjennomføre en risikoanalyse identifiseres de ulike risikoene i et prosjekt.

		Konsekvens			
		Lav	Medium	Høy	Veldig Høy
Sannsynlighet	Veldig høy	8	10	15	25
	Høy	6	8	12	20
	Medium	4	6	8	15
	Lav	2	3	4	10

Sannsynlighet	Hyppighet	Gradering
veldig høy	daglig	5
høy	ukentlig	3
medium	månedlig	2
lav	semester/årlig	1

Konsekvens	Arbeidsmengde	Gradering
veldig høy	Kan ikke fortsette	5
høy	Vil ikke komme i mål	4
medium	Mye merarbeid	3
lav	Merarbeid	2
Ingen	Svært lite merarbeid	1

Figur 1: Gradering og vurderingsgrunnlag.

Ovenfor, se figur 1 , ser du hvilken skala vi brukte og hva vi vurderte etter. Under, se figur 2, ser du de ulike risikoene vi har funnet.

Nr	Risiko	Sannsynlighet	Konsekvens	Risiko	Begrunnelse
1	Kort uforutsett fravær, f.eks sykdom	Medium	Lav	2	Reduksjon i produksjon og mer arbeid på de andre gruppemedlemmene.
2	Lengre uforutsett fravær	Lav	Høy	4	Langtidssykdom, familiære situasjoner og større uhell.
3	Frafall av mentor	Lav	Høy	4	Vanskeligere å få innblikk i hva som ønskes og hvilke kompetanse vi bør anskaffe.
4	Frafall av veileder	Lav	Medium	3	Finnes andre veiledere
5	Overstige estimert tid i framdriftsplan	Medium	Medium	8	Vanskeligere å si seg ferdig med ulike deler av prosjektet og ting må bli gjort utenom planlagt tid.
6	Ufullstendig prototype/demo til overlevering	Lav	Høy	4	Motivasjon blir lavere og NorKart ikke får infridd forventninger.
7	Trøbbel med versjonskontrollverktøy	Lav	Høy	4	Katastrofalt da her vil det ligge informasjon og kode.
8	Frafall av gruppemedlem	Lav	Høy	4	Krise for bacheloroppgaven som helhet og det blir vanskelig å hente seg fra noe slikt.
9	Korrupt data	Medium	Veldig Høy	15	Det som skal overleveres er webside så dette er helt avgjørene at det ikke skjer.
10	Liten effektivitet/Lav motivasjon i gruppen	Medium	Medium	12	Reduksjon i framgang og kvaliteten vil blir lavere
11	Problemer med utviklingsmiljøet	Medium	Høy	8	Får ikke produsert og kodet i Azure miljøet fra NorKart
12	Tap av data	Høy	Veldig Høy	20	Data tap er katastrofalt for progresjon i gruppearbeidet og for framgang
13	Endring av teknologi	Lav	Veldig Høy	5	Ikke totalt krise men en del kan risikeres å måtte skrives om.
14	Norkart er ikke tilgjengelig etter fullført iterasjon	Høy	Medium	9	Får ikke validert om det vi har gjort er tilfredstillende og må dermed vente.
15	Internett ikke tilgjengelig over lengre perioder	Medium	Veldig Høy	15	Får ikke jobbet ettersom all vår jobbing på prosjektet krever at vi jobber på systemer som er koblet til nettet

Figur 2: Risikoer.

Vi bestemte oss for å sette en nedre risikopoenggrense til 9 og har dermed følgende liste med tiltak for å være forberedt om noen av dem inntreffer. Se figur 3.

Nummer	Tiltak	Risiko
5	Daglig sjekk av framdriftsplan for å se at vi er i rute og opparbeidet tilfredstillende timeantall	8
9	Backup ukentlig og etter større arbeidsøkter for å sikre at tap av informasjon ikke forekommer	15
10	Ha statusmøter ofte samt rapportere til andre gruppemedlemmer om framgang og eventuelle hindringer som har oppstått ved tildelt oppgave. Utføre teambuilding øvelser for å heve motivasjon.	12
11	Lokal backup mot alternative systemer og opprette dialog med oppdragsgiver/mentor	8
12	Implementer systemer for å ha både fjern backup og lokal backup	20
14	Samle opp ulike iterasjoner for å presentere etter hverandre for å få anerkjent om at prosjektet beveger seg i en god retning	9
15	Flytte til enten Norkart sine lokaler, eventuelt sjekke om noen av gruppemedlemene har internett	15

Figur 3: Tiltak.

6 Plan for gjennomføring

For å få en oversikt over estimert tid på inndelingen av prosjektet har vi opprettet et Gantt skjema, se figur 4. Hele prosjektet varer over 4 måneder, fra 7. januar til 15. mai(ID 0). Oppgaven fra Norkart er estimert til å ferdigstilles på tre og en halv måned.

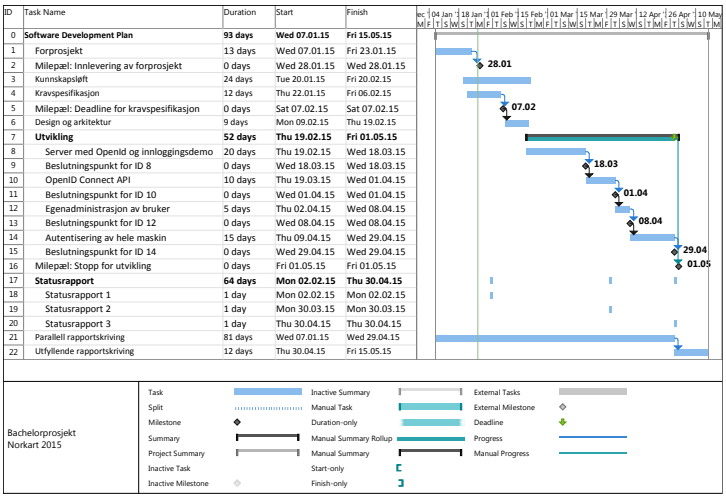
Forprosjekt er det første som skal gjøres og estimeres til å bli ferdig på to og en halv uke(ID 1). Innlevering av forprosjekt er satt til 28.januar(ID 2). Det er estimert to uker til utforming av kravspesifikasjon(ID 4) som skal være ferdig den 7.februar(ID 5). En og en halv uke er satt av til å planlegge design og arkitektur for utvikling av prototypen(ID 6). Parallelt med dette vil alle medlemmene i gruppa gjøre et kunnskapsløft innenfor visse teknologier(ID 3).

Utviklingstid av selve prototypen er satt til 50 dager(ID 7). Utviklingstiden vil bestå av flere iterasjoner. Ettersom vi baserer iterasjonene i utviklingen på bygge MVP'er vil det ikke være mulig å estimere hvor lang tid hver iterasjon vil ta eller hvor mange iterasjoner som vil forekomme i løpet av utviklingstiden. Siden vi ikke kan estimere iterasjoner har vi delt opp utviklingstiden i fire faser:

- Fase 1: Server med OpenID og innloggingsdemo(ID 8)
- Fase 2: OpenId Connect API(ID 10)
- Fase 3: Egenadministrasjon av bruker(ID 12)
- Fase 4: Autentisering av hele maskin(ID 14)

I slutten av disse fasene er det beslutningspunkt(ID 9, 11, 13 og 15) slik at fasene må være ferdig til dette. Det kan være flere iterasjoner innen hver fase. Etter hver iterasjon vil det være beslutningspunkter for hva som skjer i neste iterasjon og hvor lang tid den skal ta. Milepæl for all utvikling er satt til 29. april(ID 16), da skal prototypen av NorkartID være klar for overlevering til Norkart.

Dokumentasjon til rapporten føres parallelt med løsning av oppgaven fra Norkart (ID 21). Etter endt utvikling er det estimert to uker til utfyllende rapportskriving hvor rapporten vil finskrives (ID 22). I tilleg vil det bli levert statusrapporter(ID 17) til veiledere tre ganger i løpet av prosjektet.

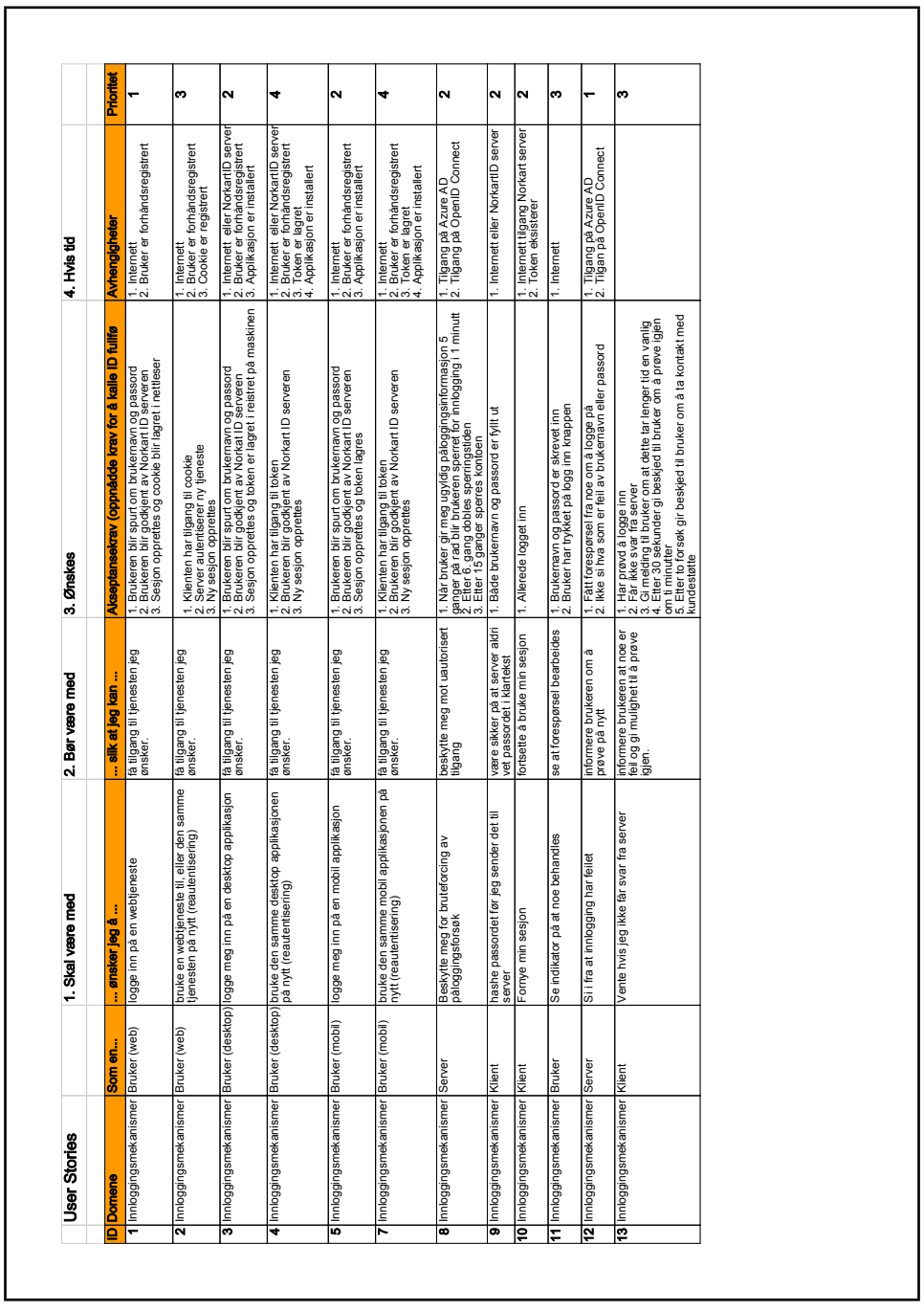


ID	Task Name	Duration	Start	Finish
0	**Software Development Plan**	**93 days**	**Wed 07.01.15**	**Fri 15.05.15**
1	Forprosjekt	13 days	Wed 07.01.15	Fri 23.01.15
2	Milepæl: Innlevering av forprosjekt	0 days	Wed 28.01.15	Wed 28.01.15
3	Kunnskapsløft	24 days	Tue 20.01.15	Fri 20.02.15
4	Kravspesifikasjon	12 days	Thu 22.01.15	Fri 06.02.15
5	Milepæl: Deadline for kravspesifikasjon	0 days	Sat 07.02.15	Sat 07.02.15
6	Design og arkitektur	9 days	Mon 09.02.15	Thu 19.02.15
7	**Utvikling**	**52 days**	**Thu 19.02.15**	**Fri 01.05.15**
8	Server med OpenId og innloggingsdemo	20 days	Thu 19.02.15	Wed 18.03.15
9	Beslutningspunkt for ID 8	0 days	Wed 18.03.15	Wed 18.03.15
10	OpenID Connect API	10 days	Thu 19.03.15	Wed 01.04.15
11	Beslutningspunkt for ID 10	0 days	Wed 01.04.15	Wed 01.04.15
12	Egenadministrasjon av bruker	5 days	Thu 02.04.15	Wed 08.04.15
13	Beslutningspunkt for ID 12	0 days	Wed 08.04.15	Wed 08.04.15
14	Autentisering av hele maskin	15 days	Thu 09.04.15	Wed 29.04.15
15	Beslutningspunkt for ID 14	0 days	Wed 29.04.15	Wed 29.04.15
16	Milepæl: Stopp for utvikling	0 days	Fri 01.05.15	Fri 01.05.15
17	**Statusrapport**	**64 days**	**Mon 02.02.15**	**Thu 30.04.15**
18	Statusrapport 1	1 day	Mon 02.02.15	Mon 02.02.15
19	Statusrapport 2	1 day	Mon 30.03.15	Mon 30.03.15
20	Statusrapport 3	1 day	Thu 30.04.15	Thu 30.04.15
21	Parallell rapportskriving	81 days	Wed 07.01.15	Wed 29.04.15
22	Utfyllende rapportskriving	12 days	Thu 30.04.15	Fri 15.05.15

Figur 4: Gantt skjema.

H User Stories for Norkart ID

User stories utarbeidet til kravspesifikasjon for IdentityServer3.



User Stories		1. Skal være med		2. Bør være med	3. Ønskes	4. Hvis tid	
ID	Domene	Som en...	... ønsker jeg å ...	... slik at jeg kan ...	Akseptansekrav (oppnådde krav for å kalle ID fullfø	Avhengigheter	Prioritet
1	Innloggingsmekanismer	Bruker (web)	logge inn på en webtjeneste	få tilgang til tjenesten jeg ønsker.	1. Brukeren blir spurt om brukernavn og passord 2. Brukeren blir godkjent av NorkartID serveren 3. Sesjon opprettes og cookie blir lagret i nettleser	1. Internett 2. Bruker er forhåndsregistrert	1
2	Innloggingsmekanismer	Bruker (web)	bruke en webtjeneste til, eller den samme tjenesten på nytt (reautentisering)	få tilgang til tjenesten jeg ønsker.	1. Klienten har tilgang til cookie 2. Server autentiserer ny tjeneste 3. Ny sesjon opprettes	1. Internett 2. Bruker er forhåndsregistrert 3. Cookie er registrert	3
3	Innloggingsmekanismer	Bruker (desktop)	logge meg inn på en desktop applikasjon	få tilgang til tjenesten jeg ønsker.	1. Brukeren blir spurt om brukernavn og passord 2. Brukeren blir godkjent av NorkartID serveren 3. Sesjon opprettes og token er lagret i nettstet på maskinen	1. Internett eller NorkartID server 2. Bruker er forhåndsregistrert 3. Applikasjon er installert	2
4	Innloggingsmekanismer	Bruker (desktop)	bruke den samme desktop applikasjonen på nytt (reautentisering)	få tilgang til tjenesten jeg ønsker.	1. Klienten har tilgang til token 2. Brukeren blir godkjent av Norkart ID serveren 3. Ny sesjon opprettes	1. Internett eller NorkartID server 2. Bruker er forhåndsregistrert 3. Token er lagret 4. Applikasjon er installert	4
5	Innloggingsmekanismer	Bruker (mobil)	logge meg inn på en mobil applikasjon	få tilgang til tjenesten jeg ønsker.	1. Brukeren blir spurt om brukernavn og passord 2. Brukeren blir godkjent av NorkartID serveren 3. Sesjon opprettes og token lagres	1. Internett 2. Bruker er forhåndsregistrert 3. Applikasjon er installert	2
7	Innloggingsmekanismer	Bruker (mobil)	bruke den samme mobil applikasjonen på nytt (reautentisering)	få tilgang til tjenesten jeg ønsker.	1. Klienten har tilgang til token 2. Brukeren blir godkjent av Norkart ID serveren 3. Ny sesjon opprettes	1. Internett 2. Bruker er forhåndsregistrert 3. Token er lagret 4. Applikasjon er installert	4
8	Innloggingsmekanismer	Server	Beskytte meg for bruteforcing av påloggingsforsøk	beskytte meg mot uautorisert tilgang	1. Når bruker gir meg ugyldig påloggingsinformasjon 5 ganger på rad blir brukeren sperret for innlogging i 1 minutt 2. Etter 6 gang dobles sperringstiden 3. Etter 15 ganger sperres kontoen	1. Tilgang på Azure AD 2. Tilgang på OpenID Connect	2
9	Innloggingsmekanismer	Klient	hashe passordet før jeg sender det til server	være sikker på at server aldri vet passordet i klartekst	1. Både brukernavn og passord er fylt ut	1. Internett eller NorkartID server	2
10	Innloggingsmekanismer	Klient	Fornye min sesjon	fortsette å bruke min sesjon	1. Allerede logget inn	1. Internett tilgang Norkart server 2. Token eksisterer	2
11	Innloggingsmekanismer	Bruker	Se indikator på at noe behandles	se at forespørsel bearbeides	1. Brukernavn og passord er skrevet inn 2. Bruker har trykket på logg inn knappen	1. Internett	3
12	Innloggingsmekanismer	Server	Si i fra at innlogging har feilet	informere brukeren om å prøve på nytt	1. Fått forespørsel fra noe om å logge på 2. Ikke si hva som er feil av brukernavn eller passord	1. Tilgang på Azure AD 2. Tilgan på OpenID Connect	1
13	Innloggingsmekanismer	Klient	Vente hvis jeg ikke får svar fra server	informere brukeren at noe er feil og gi mulighet til å prøve igjen.	1. Har prøvd å logge inn 2. Får ikke svar fra server 3. Gi melding til bruker om at dette tar lenger tid en vanlig 4. Etter 30 sekunder gi beskjed til bruker om å prøve igjen om ti minutter 5. Etter to forsøk gir beskjed til bruker om å ta kontakt med kundestøtte		3

14	Innloggingsmekanismer	Applikasjon	verifisere bruker gjennom API kall	være sikker på at bruker har lov til å bruke meg	1. Applikasjonen er skrevet med innlogging innebygd 2. Bruker skriver brukernavn og passord 3. Applikasjonen sender forespørsel til server	1. Tilgang til NorkartID 2. Bruker er forhåndsregistrert	**1**
15	Innloggingsmekanismer	Applikasjon	logge bruker ut av NorkartID gjennom API kall	være sikker på at brukeren er logget ut fra meg	1. Allerede logget inn 2. Trykt på logg ut	1. Tilgang på NorkartID 2. Bruker er innlogget	**1**
16	Innloggingsmekanismer	Applikasjon	fornye NorkartID sesjonen gjennom API kall	fortsette å gi bruker tilgang til meg	1. Allerede logget inn	1. NorkartID tilgang	**2**
17	Innloggingsmekanismer	Bruker	Logge meg ut av NorkartID	være sikker på at min data er beskyttet	1. Allerede innlogget 2. Trykt på logg ut	1. Internett	**1**
18	Innloggingsmekanismer	Server	Terminere sesjoner som har vært inaktive i mer enn seks timer ut av NorkartID	hjelpe brukere å sikre egne data og tjenester	1. Allerede innlogget 2. Innaktiv i mer enn seks timer		**4**
19	Innloggingsmekanismer	Bruker	å resette passord	få tilgang	1. Bruker finnes 2. Bruker trykker på glemt passord	1. Tilgang til mail server 2. Internett	**2**
20	Egenadministrasjon	Bruker	kunne endre passord, fullt navn og epost adresse(brukerid)	administrere brukerprofilen min selv	1. Allerede innlogget		**3**
21	Egenadministrasjon	Server	sende bruker mail for endring av passord	hjelpe bruker å resette passord så smidig og sikkert som mulig	1. Bruker finnes 2. Bruker har gitt nytt passord	1. Tilgang Azure AD 2. Tilgang til mail server	**3**
22	Egenadministrasjon	Applikasjon	sette i gang glemt passord funksjonalitet	hjelpe bruker å logge inn på meg	1. Bruker kjenner brukerID 2. Funksjonaliteten er implementert i applikasjonen	1. Tilgang til NorkartID 2. Bruker er forhåndsregistrert	**3**
23	Egenadministrasjon	Klient	informere bruker om hvor godt passordet er mens bruker oppretter nytt passord	legge passord inn i Azure AD med trygghet i at passordkravene er tilfredsstilt	1. Bruker oppretter passord 2. Passord skal være minimum 8 bokstaver langt 3. Passord skal minimum inneholde en stor, en liten og et tall.	1. Tilgang til Azure AD	**3**

I Prosjektavtale

HØGSKOLEN I GJØVIK

PROSJEKTAVTALE

mellom Høgskolen i Gjøvik (HiG) (utdanningsinstitusjon),

Norkart AS

(oppdragsgiver), og

Ida Fretang Granholdt
Per Christian Kofstad
Alf Hammerseth

(student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra Januar 2015 til Mai 2015.

   Studentene skal i denne perioden følge en oppsatt fremdriftsplan der HiG yter veiledning. Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra HiG å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
   - Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon/fax, reiser og nødvendig overnatting på steder langt fra HiG. Studentene dekker utgifter for trykking og ferdigstillelse av den skriftlige besvarelsen vedrørende prosjektet.
   - Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.

3. HiG står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av faglærer/veileder og sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.

4. Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode, disketter, taper mv. som inngår som del av eller vedlegg til besvarelsen, gis det en kopi av til HiG, som vederlagsfritt kan benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av HiG til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved HiG og/eller studenter har interesser.

   Besvarelser med karakter C eller bedre registreres og plasseres i skolens bibliotek. Det legges også ut en elektronisk prosjektbesvarelse uten vedlegg på bibliotekets del av skolens internett-sider. Dette avhenger av at studentene skriver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig i papir og nettutgave (jfr. Lov om opphavsrett). Oppdragsgiver og veileder godtar slik

offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og dekan om de i løpet av prosjektet endrer syn på slik offentliggjøring.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.

6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.

7. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i Fronter. I tillegg leveres et eksemplar til oppdragsgiver.

8. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av HiG er det dekan/prodekan som godkjenner avtalen.

9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og HiG som nærmere regulerer forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene.

   Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale, skjer dette uten HiG som partner.

10. Når HiG også opptrer som oppdragsgiver trer HiG inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.

11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene i mellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.

12. Deltakende personer ved prosjektgjennomføringen:

HiGs veileder (navn): Frode Haug

Oppdragsgivers kontaktperson (navn): Grete Rudi

Student(er) (signatur): Per Kristian Kapstad dato 12/5-15

Gda F. Granholt dato 12.5-2015

Alf M. K. Hammerseth dato 12/5-205

_____ dato _____

Oppdragsgiver (signatur): Grete Rudi dato 23/1-15

IMT Dekan/prodekan (signatur): _____ dato _____

Versjon Januar 2011hb