

Lars Andreas Lillehaug
Erlend Matre
Vetle Malmin Skretting

Fra regneark til full stack - Utvikling av registreringsystem for Matsentralen

Bacheloroppgave i Dataingeniør
Veileder: Donn Morrison
Mai 2023

Lars Andreas Lillehaug
Erlend Matre
Vetle Malmin Skretting

Fra regneark til full stack - Utvikling av registreringsystem for Matsentralen

Bacheloroppgave i Dataingeniør
Veileder: Donn Morrison
Mai 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Sammendrag

Denne bacheloroppgaven omhandler utvikling av et registreringssystem for Matsentralen. Matsentralen er en organisasjon som har til formål å kutte matsvinn ved å omfordele overskuddsmat fra matbransjen til ideelle organisasjoner som hjelper vanskeligstilte. Den eksisterende løsningen til Matsentralen er flere system av regneark, et for hver av Matsentralens lokasjoner. Disse regnearkene har flere mangler og begrensninger som kompliserer Matsentralen sitt arbeid. Hensikten med prosjektet er å erstatte regnearkene med et full stack system som løser disse utfordringene. Det ble lagt vekt på at systemet skulle være standardisert, skalerbart og brukervennlig.

Det nye systemet ble planlagt med inspirasjon fra regnearkene. Lagringsfunksjonaliteten til regnearkene ble erstattet med tabeller i en relasjonsdatabase. Redigeringsfunksjonaliteten til regnearkene ble erstattet med tabeller og registreringsskjema i brukergrensesnittet. En viktig faktor i det nye systemet var å ha god balanse mellom nødvendig funksjonalitet og god brukeropplevelse.

Under utvikling har gruppen brukt en smidig utviklingsmetode, en tilpasset versjon av Scrum. Metoden bestod av møter ved planlegging i begynnelsen av sprinten, standups i løpet av sprinten og en oppsummering ved slutten av sprinten sammen med produkteier. Dette har gjort at gruppen lettere kunne tilpasse seg og implementere nye krav fra produkteier. Det ble gjennomført brukertester for å teste produktet på sluttbrukerne og få tilbakemeldinger fra dem.

Resultatet er et full stack system bestående av en relasjonsdatabase, REST-API og web-basert brukergrensesnitt. Dataene er standardisert på tvers av Matsentralens lokasjoner, samtidig som hver lokasjon kan tilpasse deler av systemet etter deres arbeidsflyt. Det er implementert autentisering og autorisering for beskyttelse av dataene. Dataene valideres for å redusere feil. Videre arbeid som gjenstår er blant annet å utplassere systemet og overføre tidligere registreringer fra regnearkene til databasen.

Abstract

This bachelor's thesis is about the development of a registration system for Matsentralen. Matsentralen is an organization whose purpose is to cut food waste by redistributing surplus food from the food industry to non-profit organizations that help the disadvantaged. The existing solution for Matsentralen consists of several systems of spreadsheets, one for each of Matsentralen's locations. These spreadsheets have several shortcomings and limitations that hinder Matsentralen. The purpose of the project is to replace the spreadsheets with a full stack system that solves these challenges. Emphasis was placed on the system being standardized, scalable, and user-friendly.

The new system was planned with inspiration from the spreadsheets. The storage functionality of the spreadsheets was replaced with tables in a relational database. The editing functionality of the spreadsheets was replaced with tables and a registration form in the user interface. An important factor in the new system was to have a good balance between the necessary functionality and a good user experience.

During development, the group has used an agile development method, an adapted version of Scrum. The method consisted of meetings during planning at the beginning of the sprint, stand-ups during the sprint, and a summary at the end of the sprint together with the product owner. This has made it easier for the group to adapt and implement new requirements from the product owner. User tests were carried out to test the product on the end users and gather feedback from them.

The result is a full stack system consisting of a relational database, REST API, and web-based user interface. The data is standardized across Matsentralen's locations, while each location can adapt parts of the system according to their workflow. Authentication and authorization have been implemented to protect the data. The data is validated to reduce errors. Further work that remains is, among other things, deploying the solution and transferring previous registrations from the spreadsheets to the database.

Forord

Denne bacheloroppgaven er utarbeidet i forbindelse med Dataingeniør studiet ved Institutt for datateknologi og informatikk hos NTNU Trondheim, og er gjennomført på vegne av Matsentralen Trøndelag. Oppgaven ble valgt på grunn av dens bærekraftige og samfunnsnyttige formål. Det var spennende for oss å utvikle digital infrastruktur som kunne hjelpe folk i sårbare situasjoner over hele Norge.

Produktet er et resultat av smidig utvikling med tre uker lange sprinter. Den første sprinten ble brukt til planlegging og oppfylging av forskjellige krav fra NTNU. For senere sprinter ble kontaktperson involvert i planlegging og fokuset ble satt på selve systemutviklingen. Her ble mye tid brukt på å koordinere med kontaktperson om forskjellige systemkrav. All brukt teknologi var også nytt for oss. Oppgaven har derfor vært svært lærerikt teknisk, samtidig som den ga et sterkt innblikk inn i systemutvikling for ekte kunder.

Vi ønsker å takke veileder Donn Morrison ved NTNU for god veiledning og kontaktperson Jort Reitsma ved Matsentralen Trøndelag for ivrig samarbeid og gode tilbakemeldinger igjennom hele prosjektet.

Trondheim 21. mai 2023



Lars Andreas Lillehaug



Erlend Matre



Vette Malmin Skretting

Oppgavetekst

Ved starten av bachelorprosjektet var oppgavebeskrivelsen gitt som følger:

Matsentralen is a non-profit organisation that collects surplus food from grocery suppliers and donates it to charity organisations who help people in need. We are a national network of eight food banks. Our mission is to reduce food waste and simultaneously help people in need in our society.

A critical task at Matsentralen is the collection of statistics on amounts of incoming and outgoing food donations. This information is used for fair distribution of food, for communication with politicians, European Food banks (FEBA), media, suppliers, charity organisations as well as possible donors. Currently all 8 Norwegian Matsentralen locations use Google Sheets driven by a business intelligence system called Looker Studio (Previously called Data Studio). This system was implemented in 2021 but has not scaled well with the growth of Matsentralen and the increasing amount of food processed.

The aim of this project will be to develop a scalable food registration system that is centralised to all Matsentralen locations and that can be integrated with the current business intelligence front-end. A management interface must be developed that allows Matsentralen to administrate and manage incoming and outgoing food registrations.

Matsentralen works with many volunteers that come and go, and therefore we are looking for an intuitive, reliable, and scalable system that can adapt as our need for data evolves.

If successful, the system can be used by all Matsentralen locations in Norway in the future, helping us do a better job at saving food and helping more people in need.

Etter oppstartsmøte den 20.01.2023 med oppdragsgiver og veileder, ble nøyaktige systemkrav bestemt. Disse kravene står beskrevet i vedlegg C - Visjonsdokument. Fra disse kravene ble brukerhistorier, domenemodell og wireframes laget, og finnes i vedlegg D - Kravdokument.

Innholdsfortegnelse

1	Introduksjon	1
1.1	Bakgrunn	1
1.2	Problemstilling	1
1.3	Struktur	1
2	Matsentralen og relevant teori	2
2.1	Matsentralen	2
2.1.1	Matsentralen bakgrunn	2
2.1.2	Eksisterende løsning	2
2.2	Relasjonsdatabase	3
2.2.1	Relasjonsmodellen	3
2.2.2	Normalisering	4
2.2.3	SQL og transaksjoner	4
2.3	REST API	5
2.4	Autorisering og autentisering	6
2.4.1	OAuth 2.0	6
2.4.2	OpenID Connect 1.0	7
2.5	SPA vs. SSR vs. SSG	7
2.6	Universell utforming	8
2.6.1	WCAG 2.1 Standarden	8
2.7	Smidig systemutvikling	8
2.7.1	Scrum	8
2.7.2	Kanban	9
2.7.3	DevOps	9
2.8	Brukertesting	9
3	Metode	10
3.1	Forskningsmetode	10
3.1.1	Uformell observasjon og brukertesting	10
3.2	Utviklingsmetode	10
3.2.1	Smidig systemutvikling	10
3.2.2	GitLab	10
3.3	Teknologivalg	10
3.3.1	Frontend	10
3.3.2	Backend	11
3.3.3	Database	11
3.3.4	Autentisering	11
3.4	Rollefordeling	12

4	Resultater	13
4.1	Vitenskapelige resultater	13
4.1.1	Database	13
4.1.2	REST-API	14
4.1.3	Brukertest	16
4.1.4	Frontend.....	16
4.2	Ingeniørfaglige resultater	20
4.2.1	Originale funksjonelle resultatmål	20
4.2.2	Ekstra funksjonelle resultatmål etter sprint reviews.....	22
4.2.3	Ikke-funksjonelle egenskaper og andre krav	23
4.2.4	Universell utforming	23
4.3	Administrative resultater	25
4.3.1	Fremdriftsplan	25
4.3.2	Timeregnskap	25
4.3.3	Utviklingsmetodikk.....	26
4.3.4	Utviklingsprosessen	26
5	Diskusjon	29
5.1	Diskusjon av vitenskapelige resultater	29
5.1.1	Iterative systemforbedringer	29
5.1.2	Standardisert, skalerbart og brukervennlig.....	30
5.1.3	Normalisering	31
5.1.4	Brukertester.....	31
5.2	Diskusjon av ingeniørfaglige resultater.....	32
5.2.1	Resultatmål.....	32
5.2.2	Ikke-funksjonelle egenskaper og krav	32
5.2.3	Universell utforming	33
5.3	Diskusjon av administrative resultater	33
5.3.1	Fremdriftsplan	33
5.3.2	Timeregnskap	33
5.3.3	Utviklingsmetodikk.....	33
5.3.4	Utviklingsprosess	34
6	Konklusjon og videre arbeid	35
6.1	Konklusjon	35
6.2	Videre arbeid	35
	Samfunnspåvirkning	37
	Profesjonsetiske problemstillinger	37
	Bærekraftvurderinger	37
	Referanser	39
	Vedlegg	40

Figurer

Figur 1: En lokasjon kan for eksempel bestemme å lagre melk og ost hver for seg.	2
Figur 2: Skjermdump fra en rekke Tine innleveringer.	3
Figur 3: Relasjon eksempel. En bok hører til et forlag [2].	4
Figur 4: Informasjonsflyten i et REST API [5].	6
Figur 5: Abstrakt OAuth protokollflyt.	7
Figur 6: En Scrum prosess [16].	9
Figur 7: Teknologi stack.	11
Figur 8: Utdrag av Innlevering regneark og databasemodell for å vise oversetting til relasjonsdatabase. Leverandørtype, varegruppe og lokasjon (MS) er flyttet ut.	13
Figur 9: Fullstendig databasemodell etter konvertering og normalisering til 3NF.	14
Figur 10: «Tidligere Innleveringer»-siden.	17
Figur 11: «Ny Innlevering»-siden.	18
Figur 12: «Admin»-siden for en admin.	19
Figur 13: «Admin»-siden for en Nasjonal administrator.	20
Figur 14: Timeliste	26
Figur 15: Oversikt over sprintene på GitLab	28
Figur 16: Leverandør-tabell og leverandør regneark.	31

Tabeller

Tabell 1: Originale resultatmål.	22
Tabell 2: Fortløpende resultatmål	23
Tabell 3: WCAG 2.1 Prinsipp 1.....	24
Tabell 4: WCAG 2.1 Prinsipp 2.....	24
Tabell 5: WCAG 2.1 Prinsipp 3.....	25
Tabell 6: WCAG 2.1 Prinsipp 4.....	25

Akronymer

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
CD	Continuous Deployment
CI	Continuous Integration
CRUD	Create, Read, Update, Delete
CSR	Client Side Rendering
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MUI	Material UI
NF	Normal Form
NTNU	Norges teknisk-naturvitenskapelige universitet
ORM	Object-Relational Mapping
REST	Representational State Transfer
SEO	Search Engine Optimization
SPA	Single Page Application
SQL	Structured Query Language
SSG	Static Site Generator
SSR	Server Side Rendered
UI	User Interfaces
URI	Uniform Resource Identifier

1 Introduksjon

1.1 Bakgrunn

Matsentralen Norge har siden 2021 brukt Google Sheets som både registreringsverktøy og database for innleveringer og utleveringer av mat. Denne dataen blir så visualisert i Looker Studio og brukes til kommunikasjon med media, myndighetene og andre interessenter.

Denne løsningen har fungert utmerket så langt. Det er gratis og enkelt å både sette opp og bruke. Dette har vært viktige punkter etter som Matsentralen har begrenset med budsjett og tekniske kunnskaper. Over tid har derimot databasen blitt tregere og forferdelig kompleks, fordi Google Sheets var aldri ment til å brukes som en database. Applikasjonen følger ikke ACID prinsipper, brukere er nødt til å manuelt utvide cellemetoder, med mer. Hvis Matsentralen ønsker å fortsette veksten sin vil ikke det nåværende systemet kunne støtte deres behov for datasikkerhet, effektivitet og skalering.

1.2 Problemstilling

Hensikten med denne oppgaven har vært å utvikle et nytt system for å erstatte den nåværende regneark-løsningen til Matsentralen. Systemet skulle bestå av en database for lagring og et brukergrensesnitt for registrering. Grunnet Matsentralens frivillige arbeidsstyrke ble fokus også satt på at brukergrensesnittet måtte være ekstremt brukervennlig. Problemstillingen ble derfor:

Hvordan transformere flere ulike komplekse system av regneark til et mer standardisert, skalerbart og brukervennlig full stack system?

1.3 Struktur

Kapittel 1 – Introduksjon presenterer Matsentralens bakgrunn og rapportens problemstilling.

Kapittel 2 – Matsentralen og relevant teori inneholder diverse forkunnskaper leseren må kjenne til for å forstå resten av bacheloroppgaven.

Kapittel 3 – Metode forklarer hvordan gruppen arbeidet med oppgaven.

Kapittel 4 – Resultater tar for seg resultatene fra arbeidet og sammenligner de mot resultatmålene som beskrevet i Vedlegg A - forprosjektplanen.

Kapittel 5 – Diskusjon drøfter om resultatene svarte på problemstillingen og hvorfor ting ble som det ble.

Kapittel 6 – Konklusjon oppsummerer arbeidet i sin helhet i forhold til problemstillingen, og tar opp anbefalinger for videre arbeid.

Samfunnspåvirkning drøfter arbeidet i forhold til et helhetlig systemperspektiv. Sluttproduktet settes i et miljømessig, samfunnsmessig og økonomisk perspektiv igjennom bærekraftvurderinger.

2 Matsentralen og relevant teori

Dette kapitlet presenterer teori og forkunnskaper nødvendig for å forstå bacheloroppgaven. Først presenteres en mer detaljert beskrivelse av Matsentralen og deres tidligere regnearkløsning. Videre blir relevante faglige kunnskaper forklart. Til sist tas opp aktuelle termer innen forsknings- og utviklingsmetode.

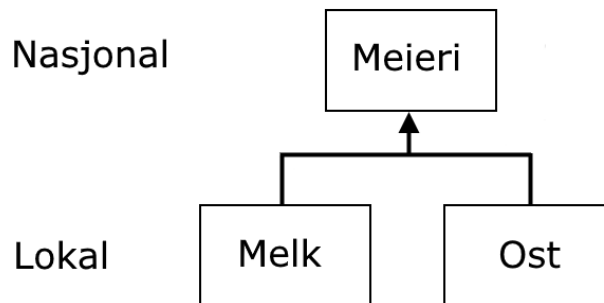
2.1 Matsentralen

2.1.1 Matsentralen bakgrunn

Matsentralen er en organisasjon som har til formål å kutte matsvinn ved å knytte overskuddsmat fra matbransjen med ideelle organisasjoner. I dag består Matsentralen av åtte forskjellige lokasjoner omkring i landet. Hver lokasjon lagrer mat fra innleveringer til de finner en veldedighetsorganisasjon de kan gi den til. Disse leveringene skjer oftest igjennom månedlige og ukentlige avtaler. Riktig registrering av denne maten er svært viktig for Matsentralen. Informasjonen brukes ikke bare til kommunikasjon med interessepartier, men er også nødvendig for ønskelige sertifikater, som Miljøfyrtårn [1].

2.1.2 Eksisterende løsning

Den eksisterende regneark-løsningen består av mange forskjellige ark, og hver lokasjon har sitt eget system av ark. Lokasjonene registrerer mat i ulike kategorier basert på hva deres lokale sjef synes er nyttig å lagre. Disse kategoriene kobles deretter opp til en overkategori som den nasjonale sjefen har bestemt.



Figur 1: En lokasjon kan for eksempel bestemme å lagre melk og ost hver for seg.

De mest relevante arkene for denne bacheloroppgaven er innleveringer, utleveringer, leverandører og organisasjoner. I tillegg til disse arkene tar Matsentralen også i bruk mange ark som de kaller for «spøkelsesark». Disse arkene fungerer på lik måte som relasjoner med fremmed- og primærnøkler i en relasjonsdatabase. Relasjonsdatabaser vil bli forklart nøyere i 2.2 Relasjonsdatabase.

Ved innleveringer blir det registrert hvilken leverandør leveringene kommer ifra, og hvilken varegruppe kategori som varen tilhører. Hvilke verdier som er godtatt i disse feltene bestemmes ut fra de som er definert i spøkelsesarkene, og blir valgt fra en nedtrekksmeny. Nettovekt for innleveringen blir kalkulert ut fra å fylle ut bruttovekt, antall kasser og

kassevekt inn i arket. Kassevekter er meget standardisert for enkelte varer. Nylig ble det også lagt til variabler for om innleveringen er en «ren donasjon», altså mat produsert for å doneres og dermed ikke overskuddsmat. Enda en variabel er for om innleveringen er «primær», som er produkter produsert av Matsentralen selv. Disse produktene er for eksempel potetgull laget av poteter bedømt som ikke salgbare i butikker.

Utleveringer er nesten lik innleveringer. Forskjellen er at det her registreres hvilken organisasjon maten går til, og inn- og utgående kasser for å holde styr på hvilke organisasjoner som låner kasser eller ikke. Det er viktig å holde styr på disse kassene fordi de har en pant. I tillegg er varegrupper erstattet med mer generelle vektgrupper, fordi det er mindre viktig å holde styr på kategorier av utgående mat. Avhengig av lokasjon er vektgrupper kun «Mat» og «Drikke», eller hvordan maten lagres som «Tørr» og «Frys».

Hver lokasjon registrerer inn- og utleveringer i lokasjonens regneark igjennom en bærbar PC. Hver levering registreres med dato. I tillegg kan det legges til et notat til hver registrering, hvis en nøyere beskrivelse av leveringen er ønskelig. Ved innleveringer blir data logget fortløpende ettersom varer kommer ut av varebilen. Det er ingen faste regler på hvordan grupper av leveringer logges, så ofte splittes de litt tilfeldig.

A	B	C	D	E	F	G	I	J
Dato	Leverandør	Varegruppe	Hovedgruppe	Brutto vekt	ser inn	Total vekt	Notat	
28/03/23	Tine Meierier	07 Meieri	Meieri	541.0		541.0 kg	bremyk	
28/03/23	Tine Meierier	07 Meieri	Meieri	177.0		177.0 kg	ridder ost	
28/03/23	Tine Meierier	07 Meieri	Meieri	291.0		291.0 kg		
28/03/23	Tine Meierier	07 Meieri	Meieri	438.0		438.0 kg	bremyk og gulost	
28/03/23	Tine Meierier	07 Meieri	Meieri	541.0		541.0 kg	bremyk	
28/03/23	Tine Meierier	07 Meieri	Meieri	541.0		541.0 kg	bremyk	
28/03/23	Tine Meierier	07 Meieri	Meieri	541.0		541.0 kg	bremyk	
28/03/23	Tine Meierier	07 Meieri	Meieri	394.0		394.0 kg	skinke ost	
28/03/23	Tine Meierier	07 Meieri	Meieri	307.0		307.0 kg	søst	
28/03/23	Tine Meierier	08 Måltider	Div kjøp	202.0		202.0 kg		
28/03/23	Tine Meierier	07 Meieri	Meieri	55.0		55.0 kg		

Figur 2: Skjermdump fra en rekke Tine innleveringer.

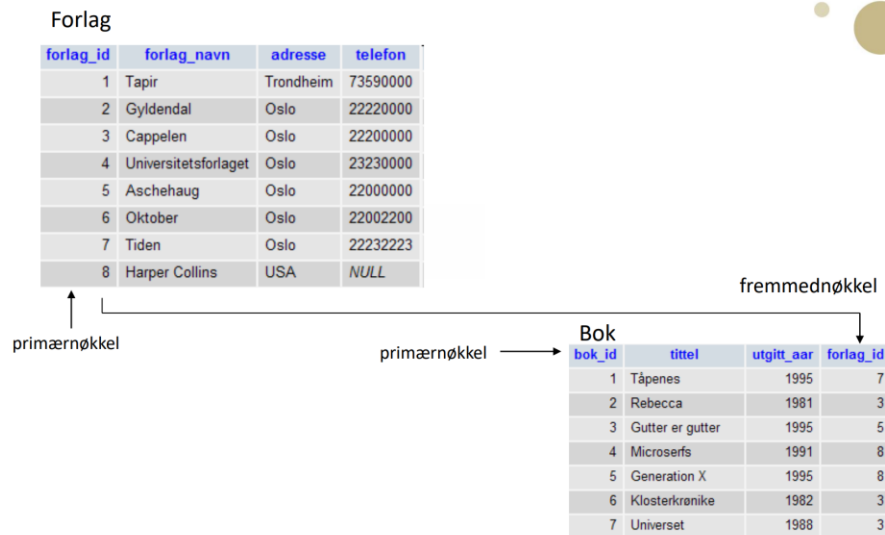
2.2 Relasjonsdatabase

2.2.1 Relasjonsmodellen

Relasjonsdatabaser organiserer data inn i navngitte tabeller, kalt for relasjoner. Hver relasjon har et visst antall kolonner med unike attributtnavn. En tuppel er en sammensatt dataverdi av flere attributter, representert som en rad. En relasjon er en mengde, som vil si at alle tuplene må være unike og ikke være ordnet på noen bestemt måte. Tupler identifiseres ved hjelp av en primærnøkkel. Primærnøkler er en eller flere konstante attributter som unikt identifiserer hver tuppel. Dersom relasjonen mangler passende attributter til å fungere som primærnøkler, kan det brukes en surrogatnøkkel. En surrogatnøkkel er en primærnøkkel laget kun for å være en entydig identifikator for en

tuppl, ofte et tall som teller oppover for hver ny rad som blir lagt til. Surrogatnøkler pleier å brukes i de fleste relasjoner fordi de gjør dem enklere å arbeide med.

Relasjoner bindes sammen med fremmednøkler. En fremmednøkkel er et attributt i en relasjon som er en annen relasjons primærnøkkel. Dette fører til at en tuppl i en relasjon, refererer til en tuppl i en annen relasjon. Disse bindingene kan foregå som en-til-en, en-til-mange eller mange-til-mange forhold. Ved mange-til-mange forhold må det opprettes en koblingstabell, slik at begge relasjonene får et en-til-mange forhold med koblingstabellen.



Figur 3: Relasjon eksempel. En bok hører til et forlag [2].

2.2.2 Normalisering

For å unngå dobbeltlagring og inkonsistenser, normalisere man tabellene. De forskjellige nivåene krever at kriteriene fra alle tidligere nivåene også følges. Høyere normalformer fører vanligvis til flere relasjoner, som senker ytelse. De forskjellige nivåene blir referert til som XNF, der X står for hvilket nivå databasen oppfyller. Første normalform, 1NF, krever at alle relasjoner i databasen har én primærnøkkel og har maksimum én verdi per attributt. 2NF krever at hvis primærnøkkelen er en sammensatt nøkkel, må hver ikke-nøkkelkolonne ikke kunne determineres av deler av primærnøkkelen, kun av hele. 3NF krever at ingen ikke-nøkkelkolonner determineres av andre ikke-nøkkelkolonner. Dette er de vanligste nivåene å tenke over, men det finnes også flere.

2.2.3 SQL og transaksjoner

SQL er et språk for å samhandle med relasjonsdatabaser. Språket bruker uttalelser som INSERT, SELECT, UPDATE og DELETE for å utføre CRUD operasjoner mot databasen. Operasjoner skjer vanligvis igjennom transaksjoner. Transaksjoner har fire egenskaper for å garantere at operasjonene utføres trygt, og kalles for ACID. Dette innebærer at transaksjoner skal være atomiske, føre databasen fra en stabil tilstand til en annen stabil tilstand, isolere dataene som berøres av transaksjonen, og lagre endringer permanent og synliggjøres for andre brukere [3].

2.3 REST API

Et API er et grensesnitt mellom applikasjoner som definerer hvordan applikasjonene skal kommuniserer med hverandre. Ved et REST API sender klienten et metodekall som POST, GET, PUT og DELETE, sammen med en eventuell JSON melding, til en URI hos serveren. Dette kommuniserer et ønske om å utføre SQL operasjoner mot databasen på brukeren sin vegne. Tjeneren sender så en statuskode med et eventuelt JSON formatert resultatet tilbake til klienten.

Som definert av Roy Fielding [4] er et REST API et API som følger de fem påkrevde arkitektoniske betingelsene til REST:

Klient-Tjener

Brukergransesnittets ansvar skal separeres fra datalagringens ansvar slik at begge kan skaleres og utvikles uavhengig av hverandre.

Minneløst

Hver forespørsel fra klienten til tjeneren må inneholde all nødvendig informasjon for å forstå forespørselen. Sesjonstilstanden må derfor lagres hos klienten. Denne betingelsen øker synlighet ettersom man ikke trenger å se lengre enn selve forespørselen ved overvåkning av systemet. Det blir mer robust ettersom feil i en forespørsel ikke får følgefeil. Til slutt blir systemet mer skalerbart ettersom det forenkler ressursbehandling på tjeneren. En av nedsidene er redusert nettverksytelse ettersom det fører til repeterende data som må sendes ved hver forespørsel.

Hurtigbuffer

For å forbedre nettverksytelse skal tjeneren merke responsen dersom den kan lagres i hurtigbuffer hos klienten. Hvis svar kan lagres i hurtigbuffer, trenger klienten kun spørre tjeneren én gang for identiske forespørsler. Dette reduserer antall kall til tjeneren og forbedrer responstiden for klienten. En av nedsidene er redusert robusthet ettersom det øker risikoen for feil data dersom innholdet i hurtigbuffer ikke lenger stemmer overens med responsen forespørselen ville fått fra tjeneren.

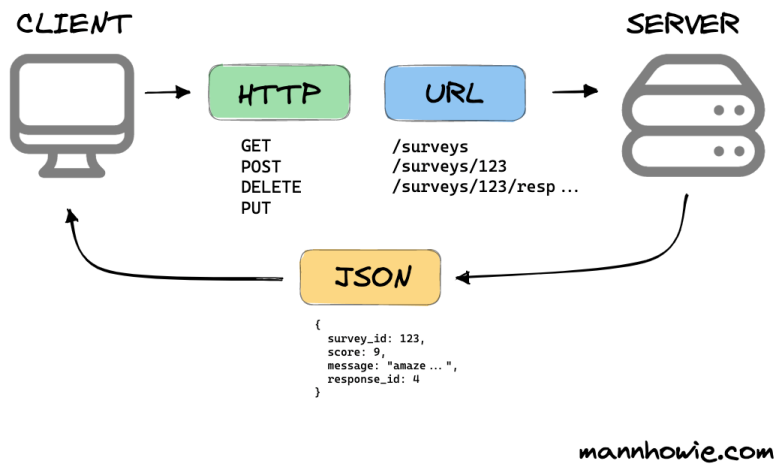
Uniformt grensesnitt

Hovedfokuset til REST er vektleggingen på uniforme grensesnitt mellom komponenter. Dette forenkler hele systemet og interaksjoner mellom komponenter. Implementasjon av komponentene har løs kobling til tjenestene de tilbyr, som muliggjør uavhengig utvikling. Nedsiden er redusert effektivitet ettersom informasjon sendes basert på standardiserte kontrakter i stedet for ut ifra hver applikasjons behov.

Lagdelt system

Systemet deles inn i hierariske lag, slik at hvert lag kun trenger å tenke på laget før og etter seg. Dette forenkler implementasjonen av interaksjonen mellom flere ulike funksjonaliteter som for eksempel lastbalansering, autentisering, autorisering, forretningslogikk og datalagring. En av nedsidene er økt prosesseringstid og dermed redusert ytelse.

WHAT IS A REST API?



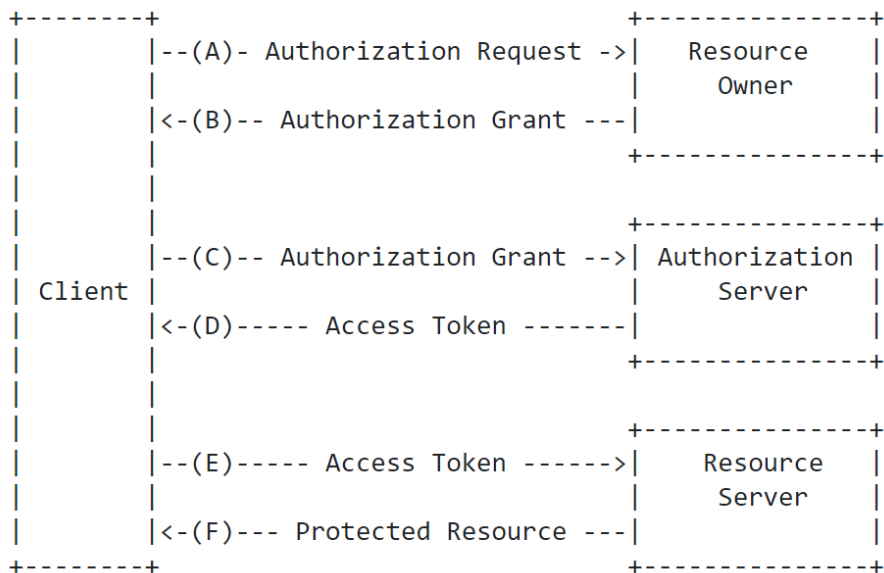
Figur 4: Informasjonsflyten i et REST API [5].

2.4 Autorisering og autentisering

Autorisering er prosessen som handler om å avgjøre om et subjekt skal få tilgang til et objekt. Subjektet er vanligvis en bruker, maskin eller tjeneste, og objektet er vanligvis informasjon eller en tjeneste [6]. Autentisering handler om å bekrefte en påstått identitet [7].

2.4.1 OAuth 2.0

OAuth 2.0 er en autoriserings protokoll som gir tredje-parts applikasjoner begrenset tilgang til en HTTP-tjeneste. Prosessen består av fire aktører; ressurseier, ressurstjener, klient og autoriseringstjener. Klienten ønsker tilgang til ressursene, samtidig som ressurseier ikke vil dele passordet med klienten. Klienten definerer hvilke ressursen den ønsker tilgang til og hvor lenge, og sender en autoriseringsforespørsel. Ressurseier aksepterer klientens vilkår og returner en autoriseringstillatelse. Klienten henter deretter et tilgangstoken fra autoriseringstjeneren som inneholder denne informasjonen. Til slutt kan klienten bruke tilgangstokenet til å aksessere ressursene fra ressurstjeneren frem til utløpstiden. Autoriseringsforespørselen og autoriseringstillatelsen bør helst sendes via autoriseringstjeneren. Ressurstjeneren og autoriserings-tjeneren kan være den samme tjeneren [8].



Figur 5: Abstrakt OAuth protokollflyt.

2.4.2 OpenID Connect 1.0

OpenID Connect 1.0 er en desentralisert autentisering protokoll på toppen av OAuth 2.0 protokollen. OpenID består av tre aktører; klient, sluttbruker og autoriseringstjener. Klienten ønsker å autentisere identiteten til sluttbrukeren uten å lagre brukernavn og passord. Sluttbrukeren autentiseres med en ID-token som inneholder signert informasjon om blant annet sluttbrukeren, klienten og utløpstid. Autentiseringen kan følge en av tre flyter; autorisasjonskode-flyten, implisitt-flyten eller hybrid-flyten [9].

Ved autorisasjonskode-flyten sendes token mellom klienten og autorisasjonsserveren. Klienten får til slutt en autoriseringskode som den kan utveksle med et ID-token for autentisering og tilgangstoken for autorisering.

Ved implisitt-flyten sender sluttbrukeren en forespørsel til autoriseringstjeneren. Autoriseringstjeneren autentiserer brukeren og returnerer en ID-token. Sluttbrukeren kan videre bruke ID-tokenet til å autentisere seg ved en klient. Klienten kan da verifisere signaturen og informasjonen til ID-tokenet for å autentisere sluttbrukeren.

2.5 SPA vs. SSR vs. SSG

Webapplikasjoner deles vanligvis inn i tre følgende typer: Single Page Application (SPA), Server Side Rendered Sites (SSR) og Static Site Generator (SSG). SPA laster ned hele siden på en gang, og har derfor en tregere første-innlastning fordi HTTP ikke er tilgjengelig før all JavaScript er lastet ned. Denne typen webapplikasjon er meget interaktive fordi de kan oppdateres med JavaScript uten å måtte laste ned hele siden på nytt. SSR genererer hele HTML sider på server siden, for å så sende de til klienten. Disse webapplikasjonene er best egnet for offentlige nettsider som trenger svært personalisert innhold og SEO. SSG bygger i stedet statisk sider, som genereres for å så lagres på en server. Slike sider passer best for innhold som ikke endres ofte, som blogger og dokumentasjon [10].

2.6 Universell utforming

Universell utforming er bygd på et prinsipp om at et produkt skal være designet med hensyn på at enhver person kan bruke det på en likeverdig måte. Målet med universell utforming er å hindre diskriminering mot personer med nedsatt funksjonsevne, og heller tilby like muligheter for alle [11]. Innenfor informasjons- og kommunikasjonsteknologi er det i Norge lovpålagt at «alle nettjenester og apper som er rettet mot allmennheten, både i privat og offentlig sektor. Offentlig sektor skal totalt følge 47 av 78 suksesskriterier i WCAG 2.1-standarden.» [12].

2.6.1 WCAG 2.1 Standarden

WCAG 2.1 standarden er et sett med retningslinjer som er knyttet til universell utforming. Standarden er bygget opp på fire hovedprinsipp. De fire prinsippene er «mulig å oppfatte», «mulig å betjene», «forståelig» og «robust». Hvert prinsipp inneholder et sett med retningslinjer som utgjør de grunnleggende målene. Retningslinjene har opptil flere testbare suksesskriterier som gjør det lettere å vurdere om et produkt følger kravene. Suksesskriteriene er delt opp i tre prioriterte nivå: «A», «AA» og «AAA», hvorav «A» er høyest prioritert [13].

2.7 Smidig systemutvikling

Smidig systemutvikling er systemutvikling basert på smidige prinsipper. Disse prinsippene er basert på *The Agile Manifesto* [14], og handler om å ha kunden i fokus, blant annet ved å anerkjenne skiftende krav. Man bør kontinuerlig levere fungerende programvare, der enkelhet skal være i fokus. Motiverte personer fra ulike fagfelt bør jobbe sammen i et støttende miljø og jevnlig kommunisere ansikt-til-ansikt. Gruppen skal være selvorganisert og jevnlig reflektere over effektiviteten og eventuelt justere arbeidsmetodikken.

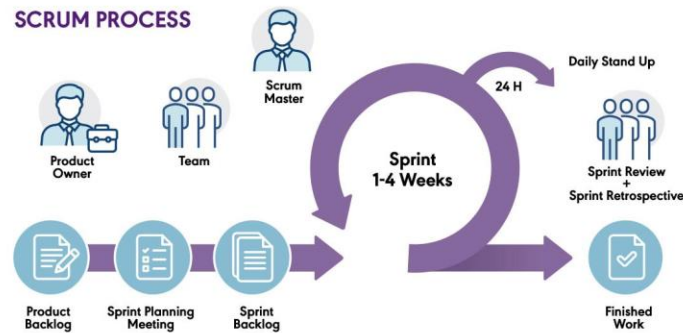
2.7.1 Scrum

Scrum er et rammeverk for utvikling, leveranse og vedlikehold av produkt som følger prinsippene til smidig utvikling. Scrum-teamet består av produkteier, utviklingsteamet og en scrum-master. Produkteier er ansvarlig for å maksimere verdien til produktet og administrere produktbacklogen. Produktbacklogen består av alle oppgaver som må gjøres med tilhørende prioritet og tidsestimat. Utviklingsteamet er en mindre, selvorganisert, tverrfaglig gruppe. Scrum-master er ansvarlig for at scrum-teamet forstår og følger scrum rammeverket [15].

Prosjektet deles inn i tidsbestemte sprints, der hver sprint helst ikke overskrider en måned. Hver sprint starter med en sprintplanlegging. Ved sprintplanleggingen diskuterer utviklingsteamet hvor mye som kan gjøres i løpet av sprinten og hvordan det skal gjennomføres. Dette danner grunnlaget for sprintmålet, et overordnet sluttmaal for sprinten.

Hver dag har utviklingsteamet et 15 minutters møte for å reflektere over gårsdagen og planlegge de neste 24 timene. Medlemmene kan for eksempel svare på de tre spørsmål hver. Hva gjorde jeg i går for å møte sprintmålet? Hva skal jeg gjøre i dag for å møte sprintmålet? Er det noe som hindrer meg i fra å møte sprintmålet?

Ved slutten av sprinten holdes en sprint review og en sprint retrospektiv. Ved sprint review diskuterer scrum-teamet og interessenter hva som har blitt gjort, hvordan det ble gjort og hva som bør gjøres videre. Produktbackloggen oppdateres i henhold til dette. Ved sprint retrospektiv reflekterer scrum-teamet over selve scrum-arbeidsflyten og gjør eventuelle forbedringer før neste sprint.



Figur 6: En Scrum prosess [16].

2.7.2 Kanban

Kanban er et prosessverktøy som er mye brukt i smidig utvikling. Kanban fokuserer på å visualisere arbeidsflyten og å minimere antall oppgaver som pågår samtidig. Prosjektet deles inn i flere mindre oppgaver som skrives på hver sin lapp. Lappene henges på en vegg og delt inn i navngitte kolonner for å visualisere hvor hver oppgave er i arbeidsflyten. Veggene og lappene kan være fysiske eller digitale gjennom et Kanban-programvare. Kolonnene kan for eksempel være backlog, planlagt, pågående, utviklet, testet og ferdig. Det settes en øvre grense for hvor mange oppgaver som kan være i den pågående-kolonnen til enhver tid. Teamet skal fokusere på å fullføre påbegynte oppgaver enn å starte nye [17].

2.7.3 DevOps

DevOps er en metodologi som skal forbedre arbeidsflyten gjennom hele programvareutviklingens livssyklus. DevOps prosessen består av en uendelig loop hvor man planlegger, koder, bygger, tester, publiserer, distribuerer, drifter, overvåker og etter tilbakemelding tilbakestilles loopen til planlegging. I stedet for å vente på at all funksjonalitet skal være ferdig før man publiserer, blir programvaren oppdatert jevnlig med nye funksjonaliteter. Sentralt i DevOps metodologien er Continuous Integration (CI) og Continuous Deployment (CD). Med CI automatiseres prosessen med å jevnlig integrere kodeendringen fra flere utviklere til en felles kodebase. Med CD automatiseres prosessen med å bygge, teste, publisere og distribuere programvaren [18].

2.8 Brukertestning

Brukertestning er et verktøy for å få et innblikk i hvor god brukskvaliteten til et produkt eller prototype er. Opplegget går ut på å gi testbrukere definerte oppgaver å utføre i et kontrollert eller naturlig miljø. Testteamet observerer og stiller spørsmål igjennom opplegget for å samle tilbakemeldinger.

3 Metode

Denne seksjonen er handler om forskings- og utviklingsmetoden brukt for oppgaven. Forskningsmetoden beskriver hvordan gammel kunnskap ble hentet og hvordan ny kunnskap ble skapt. Forståelsen hentet ut fra dette arbeidet er viktig for at produktet skal kunne støttes av kunnskap. Utviklingsmetoden angir teknologivalg, hvordan prosjektet ble utført, og hvorfor det ble utført på denne måten.

3.1 Forskningsmetode

3.1.1 Uformell observasjon og brukertesting

For å få en bedre forståelse om hvordan Matsentralens arbeid foregikk, dro teamet på et besøk til Matsentralen Trøndelag for å observere hvordan de arbeidet i praksis. Ved et senere besøk ble det også utført brukertester. Målet med disse brukertestene var å teste navigasjon, registrerings- og redigeringsfunksjonene med så mange frivillige som mulig, samt administrator-funksjoner med oppdragsgiver. Brukertestene fulgte vedlagt testplan og mal i vedlegg F og G, basert på retningslinjer fra *Praktisk Brukertesting* [19]. Oppgavene under testen ble utformet til å teste de verste tilfellene som applikasjonen er nødt til å håndtere.

3.2 Utviklingsmetode

3.2.1 Smidig systemutvikling

Grunnet at Matsentralens allerede eksisterende system fortsatt var i utvikling ved starten av oppgaven, ble det i starten av prosjektet bestemt at en smidig utviklingsprosess passet oppgaven best. Scrum virket god for problemstillingen på grunn av dens fokus på kommunikasjon med både produkteier og innenfor teamet. Noen av artefaktene ved Scrum er derimot mer tilpasset for større grupper. Siden bachelorgruppen kun besto av tre personer, ble Scrum-master og sprint-retrospektiv kuttet bort. Gruppen tok også inspirasjon fra Kanban-tavler ved å bruke GitLabs innebygde Issue Board for å holde styr på arbeidsoppgaver.

3.2.2 GitLab

GitLab ble benyttet som DevOps-verktøy grunnet NTNUs tilkobling til den. Applikasjonen har vært uvurderlig for samarbeid og versjonskontroll av kildekoden. For å sikre en viss kvalitet på koden, ble det satt et krav om at alle merge-requests måtte bli godkjent av minst et annet gruppemedlem.

3.3 Teknologivalg

3.3.1 Frontend

For valg av frontend-teknologi ble et rammeverk som er lett å opprette og vedlikeholde prioritert. Etter første møte med kontaktperson for Matsentralen ble partene enige i at en SPA-webapplikasjon passet problemstillingen best. Siden registreringsystemet

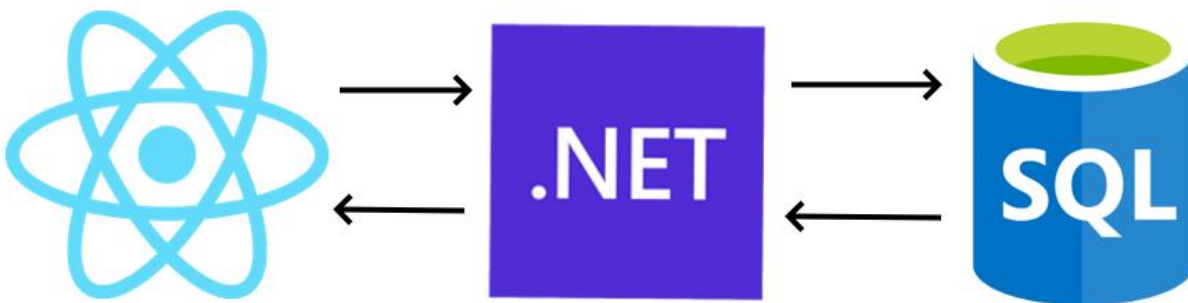
kun ville bli brukt internt og av få brukere om gangen, trengtes det heller ikke en overkomplisert og supereffektiv teknologi stakk. Budsjettet til Matsentralen er også lavt, så popularitet spiller derfor en viktig rolle for å lettere finne utviklere for eventuelt videre arbeid. Gruppen valgte derfor å utføre oppgaven som et React TypeScript prosjekt, satt opp igjennom ViteJS. Prosjektet brukte også MUI som et komponentbibliotek. React er godt egnet for å lage SPA applikasjoner, er svært lett ut av boksen, og har samtidig støtte til mer kompliserte systemer om det en dag skulle trenges.

3.3.2 Backend

Gruppen valgte å bruke .NET som rammeverk for et REST API backend av flere årsaker. For det første er .NET en av de mest stabile og pålitelige plattformene for å utvikle enterprise-applikasjoner, med et omfattende bibliotek av verktøy og funksjoner som støtter utviklerne i å løse komplekse utfordringer. For det andre har .NET et stort og aktivt utviklermiljø, noe som ville gi oss muligheten til å få støtte og råd når det trengtes. Videre har .NET en robust sikkerhetsmodell som kan brukes til å øke beskyttelsen av Matsentralens data. Entity Framework Core ble brukt som et ORM rammeverk for å forenkle samhandling med databasen.

3.3.3 Database

En SQL-basert relasjonsdatabase egner seg godt for den type data Matsentralen registrerer. Hver innlevering og utlevering har fastbestemte felter som skal utfylles og bestemte sammenhenger til leverandører, organisasjoner, varegrupper, vektgrupper, osv. Den eksisterende regneark-løsningen har allerede et regneark for hver enhet, med kolonner for feltene og nedtrekksmenyer for å koble rader i ulike bestemte regneark sammen. Hver nedtrekksmeny ble da en fremmednøkkel, og hvert regneark lot seg naturlig oversette til en SQL-tabell. Noe normalisering måtte derimot fremdeles gjøres for å fjerne unødvendig duplikasjon av data. Transaksjoner ble automatisk tatt hånd om av Entity Framework Core.



Figur 7: Teknologi stack.

3.3.4 Autentisering

For autentisering ble det utnyttet at Matsentralen allerede tar i bruk Google Workspace. Her virket det gunstig å gjemme nettsiden bak en OAuth 2.0 Google-pålogging, som dermed sjekket om Google brukeren er en del av Matsentralen sin Workspace. Dermed brukes et allerede godt integrert system i Matsentralen sitt arbeidsmiljø til autentisering og brukeradministrasjon.

3.4 Rollefordeling

I planleggingsfasen bestemte gruppen seg for hvordan de ikke-kode relaterte rollene skulle fordeles. Disse rollene var dokumentansvarlig, møteansvarlig og kvalitetsansvarlig. Erlend ble valgt som møteansvarlig og hadde ansvar for å skrive agenda, referat og møteinnkallinger. Den som var satt til å være møteleder varierte mellom møtene. Lars var dokumentansvarlig og hadde i oppdrag å samle nødvendige filer og strukturere de i mapper. Målet hans var å gjøre at alt som gruppen trengte for å gjennomføre arbeidsoppgaver var til stede og lett tilgjengelig for medlemmene. Vetle var kvalitetsansvarlig, og skulle kvalitetssikre arbeidet som var gjort og gjennomføre selve innleveringene.

Da arbeidet på selve produktet begynte valgte gruppen å fordele arbeidsområdene i forhold til backend og frontend. Målet var at alle skulle få arbeide med hele produktet, men at fokusområdet var en av de to. Erlend ble satt på backend og Lars på frontend. Vetle ble tildelt en mer fleksibel rolle der han skulle arbeide med den delen av produktet som hadde mest arbeidsoppgaver til enhver periode. Alle hadde i oppgave å dokumentere koden de skrev, lage tester der det passet seg, og kvalitetssikre arbeidet til de andre medlemmene og gi tilbakemeldinger på det.

4 Resultater

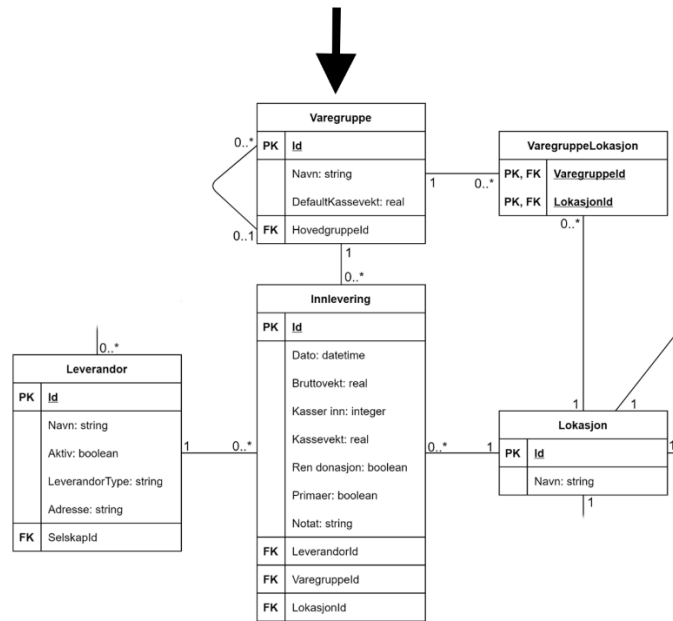
4.1 Vitenskapelige resultater

Lagringsfunksjonaliteten til regnearkene ble erstattet med en relasjonsdatabase og brukergrensesnittet til regnearkene ble erstattet med en nettside. Det ble også laget et REST-API som grensesnitt mellom disse.

4.1.1 Database

Hvert regneark ble oversatt til en tabell med «Id» som primærnøkkel. Det ble definert datatype for hver kolonne, heltall for antall, flyttall for vekt, string for tekst, boolean for ja/nei og datetime for dato. Nedtrekksmenyer som bandt ulike regneark sammen ble erstattet med fremmednøkler. Kolonner i regnearkene som ble automatisk utfylt basert på nedtrekksmenyene ble kuttet ut for å holde tabellene normaliserte.

Dato	Leverandør	leverandør type	Varegruppe	Hovedgruppe	Brutto vekt	Kasser inn	Spesiell/Donasjon	Total vekt	Notat	MS	kassevekt
03/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	1312.0		NEI	1312.0 kg		Trøndelag	1.5 kg
03/01/19	ASKO (NG)		12 Diverse kjøll	Div kjøll	563.0		NEI	563.0 kg		Trøndelag	1.5 kg
04/01/19	NATO		13 Diverse frys	Div frys	3253.0		NEI	3253.0 kg		Trøndelag	1.5 kg
04/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	1022.0		NEI	1022.0 kg		Trøndelag	1.5 kg
07/01/19	Nortura		13 Diverse frys	Div frys	406.0		NEI	406.0 kg		Trøndelag	1.5 kg
08/01/19	Coop Norge AS		13 Diverse frys	Div frys	671.0		NEI	671.0 kg		Trøndelag	1.5 kg
08/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	1091.0		NEI	1091.0 kg		Trøndelag	1.5 kg
08/01/19	Coop Norge AS		11 Diverse tørr	Div tørr	820.0		NEI	820.0 kg		Trøndelag	2.6 kg
10/01/19	Coop Norge AS		01 Grønnsaker	Frukt & Grønt	134.0		NEI	134.0 kg		Trøndelag	1.6 kg
10/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	328.0		NEI	328.0 kg		Trøndelag	1.5 kg
10/01/19	Tine Meierier		12 Diverse kjøll	Div kjøll	686.0		NEI	686.0 kg		Trøndelag	1.5 kg
15/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	1546.0		NEI	1546.0 kg		Trøndelag	1.5 kg
16/01/19	Bama		01 Grønnsaker	Frukt & Grønt	261.0		NEI	261.0 kg		Trøndelag	1.6 kg
17/01/19	Coop Norge AS		01 Grønnsaker	Frukt & Grønt	344.0		NEI	344.0 kg		Trøndelag	1.6 kg
17/01/19	Coop Norge AS		12 Diverse kjøll	Div kjøll	361.0		NEI	361.0 kg		Trøndelag	1.5 kg
17/01/19	Coop Norge AS		11 Diverse tørr	Div tørr	260.0		NEI	260.0 kg		Trøndelag	2.6 kg
17/01/19	Coop Norge AS		11 Diverse tørr	Div tørr	525.0		NEI	525.0 kg		Trøndelag	2.6 kg
18/01/19	Tine Meierier		12 Diverse kjøll	Div kjøll	2886.0		NEI	2886.0 kg		Trøndelag	1.5 kg



Figur 8: Utdrag av Innlevering regneark og databasemodell for å vise oversetting til relasjonsdatabase. Leverandørtype, varegruppe og lokasjon (MS) er flyttet ut.

backend-mappen i Vedlegg H – Kildekode. Klassediagram og oversikt over filstruktur og endepunkt er i Vedlegg E – Systemdokumentasjon.

API-et har et «login»-endepunkt som tar imot en Google Credential JWT token. Tokenet valideres med Googles offentlige nøkkel og dersom den er gyldig, henter den ut e-posten. Brukerinformasjon, som brukerId, lokasjonId og rolle, hentes ut fra databasen basert på e-posten. Brukerinformasjonen brukes til å lage en ny JWT token med en privat symmetrisk nøkkel. Det nye tokenet settes i en Secure-HttpOnly-SameSite cookie i responsen sammen med brukerinformasjonen. Det er også et «logout»-endepunkt for å fjerne tokenet fra cookien. Alle andre endepunkt krever at dette tokenet er til stede og gyldig. Dette vil blant annet innebære korrekt signatur og ikke utgått på dato.

Endepunktene er konfigurert med fem ulike nivå av autoriseringspolicy. Admin refererer her til en lokal administrator, mens super-admin refererer til nasjonal administrator. Det laveste nivået er ingen autorisering, brukeren trenger kun å være innlogget for å få tilgang til ressursen. Endepunkt konfigurert med lokasjon-policy har også lokasjonId som route-parameter. Ved autorisering sjekkes det om lokasjonId i tokenet og lokasjonId i route-parameter matcher eller om rollen i tokenet er super-admin. Admin-policyen sjekker om rollen i tokenet er admin eller super-admin. Super-admin-policyen sjekker om rollen er super-admin. Noen endepunkt er konfigurert med både admin- og lokasjon-policy slik at det sjekkes for om brukeren er administrator ved angitt lokasjon.

Bruker, lokasjon og selskap har endepunkt for å hente alle, registrere ny, oppdatere og slette konfigurert med super-admin-policy.

Leverandør har endepunkt for å hente alle og hente aktive uten autoriseringspolicy, og endepunkt for å registrere ny, oppdatere og slette konfigurert med admin-policy.

Organisasjon har kun et endepunkt for å hente alle uten autoriseringspolicy.

Vektgruppe har et endepunkt for å hente alle med super-admin-policy, et endepunkt for å hente vektgrupper på en lokasjon med lokasjon-policy, og endepunkt for å registrere ny, oppdatere og slette med admin- og lokasjon-policy.

Varegruppe har endepunkt for å hente alle, hente alle som tre-struktur, registrere ny, oppdatere og slette med super-admin-policy. Videre er det endepunkt for å hente varegrupper med lokasjon-policy, og endepunkt for å hente alle på en lokasjon som tre-struktur, og aktivere og deaktivere på en lokasjon med admin- og lokasjon-policy.

Innlevering og utlevering har endepunkt for å hente alle med super-admin-policy. Hente leveringer på en lokasjon i en tidsramme, registrere en liste med nye leveringer, oppdatere og slette gjøres med lokasjon-policy.

Ved registrering og oppdatering av data valideres forespørselen. E-poster blir validert for riktig format og at de er unike. Vekter og antall blir validert for at de ikke er negative. Obligatoriske tekster blir validert for at de ikke er tomme. Datoer blir validert for at de er gyldige datoer etter 1970. Fremmednøkler blir validert for at de eksisterer og eventuelt at lokasjonene matcher. For eksempel valideres det at vektgruppen til en utlevering tilhører samme lokasjon som utleveringen.

4.1.3 Brukertest

Under det andre besøket til Matsentralen ble det tid til to brukertester. For den første testen ble det valgt ut en frivillig arbeider med spesiell lav datakunnskap. Den andre testen ble utført på oppdragsgiver selv. Resultatene i sin helhet finnes i vedlegg G - Brukertester. Hovedkonklusjonen fra testene var at innleveringssidene trengte å bli mer oversiktlige; testtakerne ble bortkommende da for mye informasjon ble satt inn i formen på en gang. Dette løste gruppen ved å sette leverandør-komponentene inn i en trekkspill-komponent hver, slik at brukeren kan gjemme de leverandørene de ikke trenger å samhandle med. Andre funn som uklare inputfelt og forvirrende navngiving ble også håndtert.

4.1.4 Frontend

Det ble utviklet en webapplikasjon med seks sider; «Innlogging», «Tidligere Innleveringer», «Tidligere Utleveringer», «Ny Innlevering», «Ny Utlevering» og «Admin». Kildeteksten er i frontend-mappen i Vedlegg H – Kildetekst. Klassediagram og oversikt over filstruktur er i Vedlegg E – Systemdokumentasjon.

Dersom brukeren ikke er logget inn blir de omdirigert til innloggingssiden. Innloggingssiden har en «Logg på med Google»-knapp knyttet opp til en intern OAuth 2.0 Client ID i Matsentralens Google Cloud organisasjon. Ved gyldig innlogging returneres en Credential JWT token signert med Googles private nøkkel. Denne tokenet sendes videre til innloggings endepunktet i backend som returnerer en informasjonskapsel og brukerinformasjon. Brukerinformasjonen blir lagret lokalt i nettleseren sammen med utløpstiden til informasjonskapselen. Til slutt blir brukeren omdirigert til ønsket side. For å finne ut om brukeren allerede er innlogget sjekkes det om denne brukerinformasjonen er til stede og utløpstiden ikke har gått ut.

«Tidligere Innleveringer»- og «Tidligere Utleveringer»-sidene lar brukeren se, redigere og slette leveranser. Brukeren kan velge start- og sluttdato for leveransene de ønsker å se. Start- og sluttdato er som standard satt fra starten av kalendermåneden til dagens dato. Leveransene er visualisert i en tabell hvor hver rad tilsvarende en leveranse. Kolonnene består av statisk tekst for å vise relevant data. Hvis en leveranse har et notat, kan brukeren holde musen over «Notat»-ikonet for å vise meldingen. De to siste kolonnene har knapper for å henholdsvis redigere og slette raden. Ved sletting av en rad kommer det opp en pop-up for å bekrefte om brukeren ønsker å slette raden. Ved redigering omgjøres den statiske teksten i raden til input-felt med de eksisterende verdiene som standard. Knappene i de to siste kolonnene i raden blir omgjort til «Lagre»- og «Avbryt»-knapper.

Matsentralen		Tidligere Innleveringer										Fra 01.05.2023		Til 12.05.2023	
Dato	Leverandør	Varegruppe	Bruttovekt [kg]	Kasser Inn	Kassevekt [kg]	Nettovekt [kg]	Ren Donasjon	Primær	Notat	Rediger	Slett				
12.5.2023	Grennsaksspesialistene	Frukt og grønt	77.35	4	1.18	72.64	Nei	Nei							
10.5.2023	Juicepresseme	Juice	58.61	2	2.16	54.30	Nei	Nei							
8.5.2023	Trenderbakst	Bakverk	95.02	4	2.23	86.09	Nei	Nei							
7.5.2023	Meiereprodusenten	Meieri	45.45	3	0.38	44.29	Nei	Nei							
7.5.2023	Grennsaksspesialistene	Frukt og grønt	97.50	0	2.70	97.50	Nei	Nei							
7.5.2023	Kjøttgruppen Norge	Kjøtt	18.83	0	2.90	18.83	Nei	Nei							
6.5.2023	Kjøttgruppen Norge	Kjøtt	104.23	4	2.08	95.89	Nei	Nei							
5.5.2023	Kjøttgruppen Norge	Kjøtt	60.03	1	1.60	58.43	Nei	Nei							
5.5.2023	Kjøttgruppen Norge	Kjøtt	76.75	2	0.40	75.95	Nei	Nei							
3.5.2023	Kjøttgruppen Norge	Kjøtt	55.44	1	2.60	52.84	Nei	Nei							

Figur 10: «Tidligere Innleveringer»-siden.

«Ny Innlevering»-siden lar brukeren registrere en leveranse fra ulike leverandører samtidig. I toppen til høyre på siden er et datofelt som automatisk er satt til nåværende dag. Skjemaet er strukturert av flere trekkspillkomponenter som inneholder en leverandør hver. Hver leverandør inneholder videre to tabeller. Den øverste tabellen har kun en rad, og inneholder informasjon om hvilken leverandør leveransen er fra, samt «Ren donasjon» og «Primær» avmerkingsbokser. Deretter er det en tabell med en rad for hver varegruppe fra leverandøren. Når brukeren skriver i den nederste «Varegruppe» raden, legges det til en ny rad slik at brukeren kan fortsette å registrere flere varegrupper dersom det er ønskelig. «Bruttovekt» og «Kasser inn» kolonnene fungerer på samme måte som «Varegruppe» feltene, og summerer i bakgrunnen opp total bruttovekt og totalt antall kasser inn. Totalt antall kasser inn blir ganget med kassevekt og trukket fra total bruttovekt, og summeres til «Nettovekt» kolonnen. Det er også et «Notat»-felt og «Fjern»-knapp koblet til hver Varegruppe. På bunnen av siden er det knapper for å legge til en ny leverandør og registrere leveransen.

Malsentralen
12.05.2023

Ny Innlevering

Ny Innlevering

Ny Utlevering

Varer fra Grønnsaksspesialistene

Leverandør
Ren donasjon
Primer
Fjern

Grønnsaksspesialistene

Varegruppe	Bruttovekt [kg]	Kasser inn	Kassevekt [kg]	Nettovekt [kg]	Notat	Fjern
Frukt og grønt	5,9	2	1,7	13,30		<input type="checkbox"/>
Juice	8,2	2	1,2	16,90		<input type="checkbox"/>
				0,00		

Admin

Varer fra Kjøttgruppen Norge

Leverandør
Ren donasjon
Primer
Fjern

Kjøttgruppen Norge

Varegruppe	Bruttovekt [kg]	Kasser inn	Kassevekt [kg]	Nettovekt [kg]	Notat	Fjern
Kjøtt	56,7	2	2,5	51,70		<input type="checkbox"/>
				0,00		

LEGG TIL LEVERANDØR
REGISTRER INNLEVERINGER

Logg ut

Figur 11: «Ny Innlevering»-siden.

På samme måte kan brukeren på «Ny Utlevering»-siden registrere flere utleveringer med ulike vektgrupper til ulike organisasjoner samtidig. Hver organisasjon har en tabell for hele organisasjonen, med kun en rad og en tabell med en rad for hver vektgruppe. «Bruttovekt»- og «Kasser ut»-kolonnene kan ha flere innskrivingsfelt per celle.

«Admin»-siden er kun tilgjengelig for lokal eller nasjonal administrator. Brukeren får ulik funksjonalitet presentert avhengig av hvilken type administrator brukeren er. Hver funksjon er plassert i en tabell i en trekkspillkomponent. Med unntak av «Varegrupper»-komponenten har alle tabellene knapper for å redigere eller slette en rad, og et skjema i nederste rad for å registrere en ny rad.

Lokal administratorer kan administrere lokale varegrupper, lokale vektgrupper og nasjonale leverandører. I «Varegrupper»-tabellen er nasjonale varegrupper listet opp med avmerkingsbokser for å velge hvilke varegrupper som skal tas i bruk på lokasjonen.

Matsentralen

Ny Innlevering

Ny Utlevering

Tidligere Innleveringer

Tidligere Utleveringer

Admin

Logg ut

Admin

Varegrupper

Navn	Kassevekt	Bruk
Juice	0	<input checked="" type="checkbox"/>
Kjøtt	0	<input checked="" type="checkbox"/>
Frukt og grønt	0	<input checked="" type="checkbox"/>
Meier	0	<input checked="" type="checkbox"/>
Bakverk	0	<input checked="" type="checkbox"/>

Vektgrupper

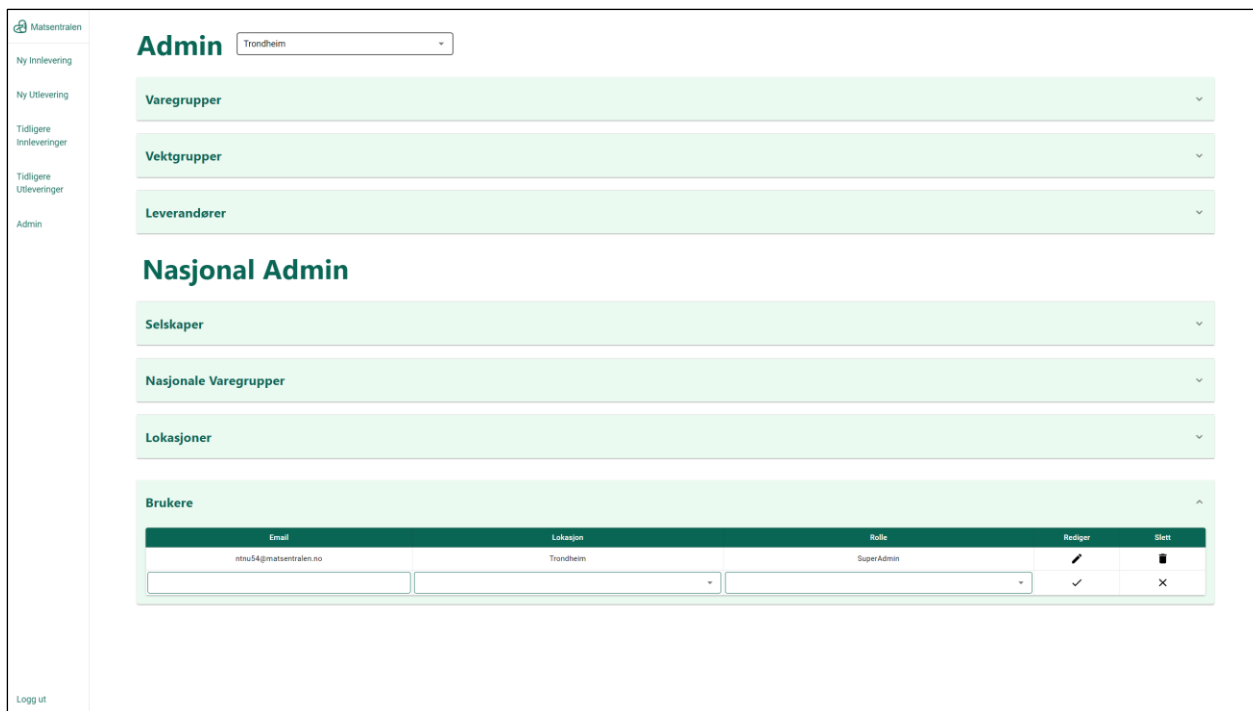
Navn	Kassevekt	Rediger	Slett
Tarr	0	<input type="checkbox"/>	<input type="checkbox"/>
Kjøll	1.4	<input type="checkbox"/>	<input type="checkbox"/>

Leverandører

Navn	Aktiv	Leverandør Type	Adresse	Selskap	Rediger	Slett
Juicepresserne	Ja				<input type="checkbox"/>	<input type="checkbox"/>
Kjøttgruppen Norge	Ja				<input type="checkbox"/>	<input type="checkbox"/>
Grønnsakspesialistene	Ja				<input type="checkbox"/>	<input type="checkbox"/>
Meiereproduzenten	Ja				<input type="checkbox"/>	<input type="checkbox"/>
Trønderbakst	Ja				<input type="checkbox"/>	<input type="checkbox"/>

Figur 12: «Admin»-siden for en admin.

Nasjonal administrator har i tillegg til de lokale administrator funksjonalitetene en nedtrekksmeny for å velge lokasjon, og funksjonalitet for å administrere selskaper, varegrupper, lokasjoner og brukere. Lokasjonen valgt i nedtrekksmenyen vil gjelde alle sidene i applikasjonen, slik at nasjonale administratorer kan administrere innleveringer og utleveringer ved andre lokasjoner. I «Brukere»-tabellen kan nasjonal administrator legge til en ny bruker ved å definere e-post, lokasjon og rolle slik at brukeren blir lagt til i databasen og kan logge inn. Brukerne som blir lagt til her er nødt til å eksistere i Matsentralen sin Google Workspace for å kunne logge inn.



Figur 13: «Admin»-siden for en Nasjonal administrator.

4.2 Ingeniørfaglige resultater

4.2.1 Originale funksjonelle resultatmål

Resultatmål	Oppfylt	Kommentar
Registrere vekt, dato, leverandør og produktkategori ved innlevering.	Ja	Produktkategorier har fått navnet varegruppe, og er det samme som produktgruppe.
Registrere vekt, dato og pick-up organisasjon ved utlevering.	Ja	
Brukere skal kunne redigere og slette tidligere registreringer.	Ja	
Funksjonalitet for å importere data fra den gamle Google Sheets løsningen.	Nei	
Autentisering: Google Workspace integrasjon for å kunne logge inn og ut med Matsentralen e-post.	Ja	Produktet tar i bruk sjekker om e-postadressen som prøver å logge inn er i Matsentralens domene. Selve lagring av rolle og brukere blir gjort i databasen. Etter videre samtaler ble det bestemt å ikke automatisk tildele lokasjon basert på Google Workspace.

Autorisering: Bruker, lokal administrator, nasjonal administrator.	Ja	
Lokal administrator skal kunne legge til nye produktgrupper.	Ja	Produktet har implementert funksjonalitet for at nasjonale administrator kan legge til nye produktgrupper. Lokal administrator kan velge hvilke av disse som brukes ved lokasjonen deres.
Lokal administrator skal kunne legge til nye leverandører.	Ja	
Lokal administrator skal kunne legge til nye organisasjoner.	Nei	Etter videre koordinasjon med Matsentralen igjennom prosjektet ble krav stilt rundt organisasjoner forkastet.
Lokal administrator skal kunne legge til nye vektgrupper.	Ja	
Nasjonal administrator skal ha tilgang til all funksjonalitet som lokal administrator har og kunne bytte mellom lokasjoner.	Ja	
Nasjonal administrator skal kunne legge til nye matbanker.	Ja	
Nasjonal administrator skal kunne legge til nye datapunkter som må registreres ved innlevering eller utlevering.	Nei	Ble ansett som for komplisert for omfanget til oppgaven.
Funksjonalitet for å kunne legge til flere vekter innenfor samme varegruppe på en gang.	Ja	
En bruker har kun tilgang til å registrere ved sin lokasjon.	Ja	Produktet har implementert restriksjoner basert på lokasjonen til brukeren.
Automatisk reduksjon av totalvekt avhengig av antall kasser og kassevekt.	Ja	
Registrere, redigere og slette matsvinn.	Nei	Nedprioritert. Kan enkelt implementeres med eksisterende arkitektur ved videre arbeid.
Slettede registrering skal havne i papirkurv hvor de videre kan slettes for godt eller gjenopprettes.	Nei	Nedprioritert.

En kvittering til pick-up organisasjon.	Nei	Etter videre koordinasjon med Matsentralen igjennom prosjektet ble krav stilt rundt organisasjoner forkastet.
Markere av om en registrering er en vanlig overskuddsdonasjon, eller en "ekte" donasjon av mat som ikke er fra overskudd.	Ja	

Tabell 1: Originale resultatmål.

4.2.2 Ekstra funksjonelle resultatmål etter sprint reviews

Resultatmål	Oppfylt	Kommentar
Det skal være mulig å legge til flere organisasjoner ved registrering av utleveringer.	Ja	
Det skal være mulig å legge til flere leverandører ved registrering av innleveringer.	Ja	
Nasjonal administrator kan legge til varegrupper. Lokal administrator kan velge hvilke av disse som deres lokasjon skal bruke.	Ja	
Varegrupper har muligheten til være en undergruppe av en annen varegruppe. Dette skal være mulig i flere lag.	Ja	
Administrator kan lage forhåndsbestemte avfallshåndtering metoder som brukes under registreringen av matsvinn	Nei	Nedprioritert.
Kunne legge til flere delvekter for varegruppe/vektgruppe som blir summert opp til en felles vekt.	Ja	
Ved visning av tidligere innleveringer/utleveringer, skal en ha muligheten til å kun se registreringer innenfor en gitt start- og sluttdato.	Ja	
Ved visning av tidligere innleveringer, skal en ha mulighet til å se innleveringer knyttet til en gitt leverandør.	Nei	Nedprioritert.

Ved visning av tidligere utleveringer, skal en ha mulighet til å se innleveringer knyttet til en gitt organisasjon	Nei	Nedprioritert.
Det skal være mulig å legge til forhåndsbestemt vekt på kasser basert på hva varegruppe/vektgruppe det er.	Ja	
Ved leveringer skal det være mulig å sette registreringen som primær.	Ja	
Ved utlevering skal det registrere antall kasser som blir levert inn tilbake.	Ja	

Tabell 2: Fortløpende resultatmål

4.2.3 Ikke-funksjonelle egenskaper og andre krav

I sammenheng med planlegging av produktet ble det laget en liste med ikke-funksjonelle egenskaper og andre krav. Et eksempel på dette er at produktet skulle være sikkert å bruke for Matsentralen. Av den grunn bruker produktet autentisering for tilgang til nettside og endepunkter. Noen av de andre tiltakene som ble gjort for å oppfylle de ikke-funksjonelle egenskapene og kravene var å ha god og oversiktlig mappestruktur på prosjektet og god dokumentasjon og struktur på kildekoden, samt at det var minst mulig kobling og mest mulig kohesjon. Sent i utviklingsfasen gjennomgikk produktet også brukertester for å teste brukervennlighet og om produktet hadde logisk struktur og navigasjon. Produktet skulle også følge standardene WCAG 2.1 og Universell Design, resultatene av dette ligger i 4.2.4 Universell utforming.

Det var også noen ikke-funksjonelle egenskaper og krav som ikke ble eller delvis ble oppfylt, som at produktet skulle ha tilstrekkelig med testing. I backend er det testing på kontroller- og serviceklassene, slik at mesteparten av det som er i bruk blir testet. Derimot, på frontend er det ikke implementert noen tester. De to siste punktene som ikke ble oppfylt var at produktet skulle kunnes kobles opp imot Google Looker Studio og at det skulle ha en høy oppetid.

4.2.4 Universell utforming

Gruppen hadde som mål å oppfylle mest mulig av de fire prinsippene i WCAG 2.1. Noen av retningslinjene til prinsippene er ikke relevant til produktet, disse er: «1.2 Alternativer for tidsbasert medier» og «2.2 Nok tid». Her vil bare retningslinjene bli presentert.

Prinsipp 1: Mulig å oppfatte

Definisjon: «Informasjon og brukergrensesnittkomponenter må presenteres for brukere på måter som de kan oppfatte.» [13]

Retningslinje	Definisjon	Implementasjon
1.1 Tekstalternativer	Ikke-tekstlig innhold skal ha tekstalternativer.	De eneste ikke-tekstlige innholdet er inndata som har navn som beskriver formålet.
1.3 Mulig å tilpasse	Innholdet skal kunne presenteres på forskjellige måter uten informasjons- og strukturtap.	Nettsiden vil tilpasse seg skjermstørrelsen. Nettsiden følger også en logisk rekkefølge.
1.4 Mulig å skille fra hverandre	Det skal være enkelt for brukerne å se innholdet.	Nettsiden følger produkteier sitt fargepalett. Det er mulig å tilpasse størrelse på komponenter via innebygde verktøy på enkelte nettlesere.

Tabell 3: WCAG 2.1 Prinsipp 1

Prinsipp 2: Mulig å betjene

Definisjon: «Det må være mulig å betjene brukergrensesnittkomponenter og navigeringsfunksjoner.» [13]

Retningslinje	Definisjon	Implementasjon
2.1 Tilgjengelig med tastatur	Funksjonaliteten skal være tilgjengelig med tastatur.	Det er mulig å navigere nettsiden med tastatur.
2.3 Anfall	Innhold skal ikke forårsake (epileptiske) anfall.	Ingen av komponentene vil blinke mer enn en gang.
2.4 Navigerbar	Mulig for brukeren å navigere, finne innhold og vite hvor de befinner seg.	Nettsiden har en sidemeny, overskrifter og kan navigere seg via peker eller tastatur.
2.5 Inndatametode	Brukeren skal ha flere metoder for inndatametoder enn bare tastatur.	Det er mulig å bruke peker og tastatur som inndatametode.

Tabell 4: WCAG 2.1 Prinsipp 2

Prinsipp 3: Forståelig

Definisjon: «Det må være mulig å forstå informasjon og betjening av brukergrensesnitt.» [13]

Retningslinje	Definisjon	Implementasjon
3.1 Leselig	Innholdet skal være leselig og forståelig.	Nettsiden bruker terminologi fra produkteier.
3.2 Forutsigbar	Nettsiden skal presenteres og fungere på en forutsigbar måte.	Nettsiden er konsekvent i navigering og funksjonalitet. Gjør bare endringer i respons til bruker. Med unntak ved

		registrering vil den navigere til tidligere registreringer.
3.3 Inndatahjelp	Hjelp brukeren med å unngå feil og å rette opp i feil.	Ved sending av inndata vil det bli kontrollert og brukeren vil få beskjed om eventuelle feildata.

Tabell 5: WCAG 2.1 Prinsipp 3

Prinsipp 4: Robust

Definisjon: «Innholdet må være robust nok til at det kan tolkes på en pålitelig måte av brukeragenter, inkludert kompensierende teknologi.» [13]

Retningslinje	Definisjon	Implementasjon
4.1 Kompatibel	Best mulig kompatibilitet med aktuelle og fremtidige brukeragenter, inkludert kompensierende teknologi.	Nettsiden har ingen store kodefeil, har navn og rolle på komponenter. Brukeren får statusmeldinger om viktige endringer.

Tabell 6: WCAG 2.1 Prinsipp 4

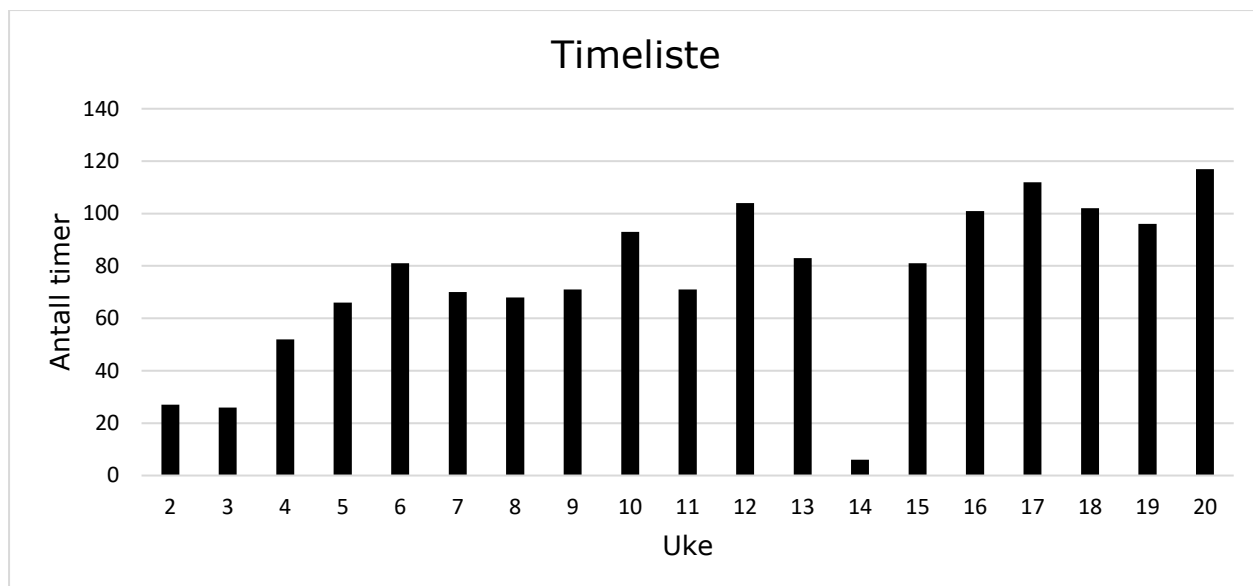
4.3 Administrative resultater

4.3.1 Fremdriftsplan

Ved starten av prosjektet lagde gruppen et GANTT-diagram for å holde styr på viktige milepæler for prosjektet. Planen inkluderte fem sprinter og obligatoriske gjøremål for NTNU. Fremdriftsplanen var en skisse om hvordan arbeidsflyten skulle være, og inneholdt ingen konkrete planer utenom de obligatoriske gjøremålene. De eneste avvikene fra planen var at den ikke inkluderte påsken, og noen ekstra dager ble brukt etter Sprint 5 for å ferdiggjøre produktet. Gruppen bestemte seg ikke for noen obligatoriske arbeidstider.

4.3.2 Timeregnskap

Timeliste med statusrapporter ble ført igjennom hele prosjektet. Nedenfor i Figur 14: Timeliste ligger et søylediagram som visualiserer det samlede tidsforbruket hver uke. Utover prosjektet har den totale arbeidstiden per uke for gruppen vært relativt jevnt, med en betydelig økning mot slutten. Se Vedlegg B – Prosjekthåndbok for GANNT-diagram, statusrapporter og nøyere arbeidsfordeling hver uke.



Figur 14: Timeliste

4.3.3 Utviklingsmetodikk

I dette prosjektet valgte gruppen modifisert Scrum som utviklingsmetodikk. Dette innebar å ha sprint planlegging, sprint review, backlog for oppgaver og gjennomføre standups opptil flere ganger i uken. Sprint planlegging ble holdt i begynnelsen av sprintene, der gruppen diskuterte og lagde oppgaver for neste sprint med utgangspunkt i sprint review. Sprint review ble holdt i slutten av sprintene sammen med kontaktperson fra Matsentralen og eventuelt veileder. Her viste gruppen fremdriften og diskuterte hva som burde prioriteres for neste sprint. For å holde styr på backloggen brukte gruppen GitLab sin innebygde Kanban-tavle, her ble det laget seks kolonner for alle tilstander en oppgave kan være i. De seks tilstandene var åpen, påbegynt, blokkert, feil, klar for gjennomgang og lukket.

Før arbeidet på prosjektet begynte satte gruppen seg veldefinerte prosedyrer som medlemmene skulle tilpasse seg, disse var blant annet hvordan møter skal innkalles både internt og med produkteier eller veileder. Resten av prosedyrene ligger i Vedlegg C - Visjonsdokument.

4.3.4 Utviklingsprosessen

Den første steget i utviklingsprosessen etter dette var et møte med Matsentralen. Målet med møte var å få bedre oversikt over hva de ønsket fra produktet. Som resultat fra møtet ble det bestemt hvordan innkalling til fremtidige møter skulle gjennomføres, første sett med krav til sluttproduktet og at alle parter signerte standardavtalen. Agenda og referat er under møte 1 i Vedlegg B – Prosjekthåndbok. Videre brukte gruppen resterende av tiden før første sprint på å lage en forprosjektplan, visjonsdokument og tidsplan.

Til den første sprinten var målet å lage en wireframe for produktets funksjonalitet og utseende, finne ut hvilke teknologier som var best egnet til problemstillingen og en foreløpig skisse på hvordan database strukturen ville være. Ved slutten av sprinten ble det gjennomført en sprint review med en representant fra produkteieren, der gruppen fikk

tilbakemelding om wireframe og om databasens struktur og innhold. Det ble også holdt et annet møte etter sprint review sammen med en annen gruppe som lager et annet produkt for samme produkteier. Dette var for en potensiell samkjøring av teknologien til de to produktene, men resulterte i at gruppene skulle ta kontakt med hverandre lengre ute i utviklingsprosessen. Agenda og referat er under møte 2 i Vedlegg B – Prosjekthåndbok.

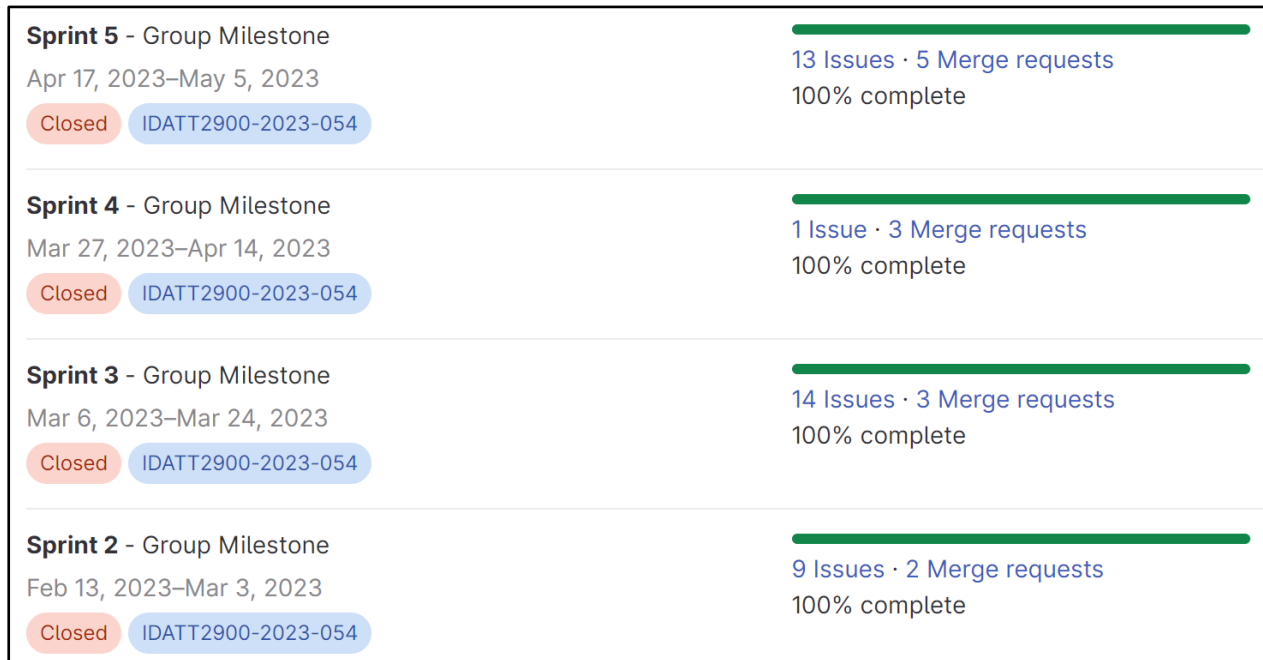
Ved inngangen til den andre sprinten ble det holdt en kort sprint planlegging der gruppen lagde oppgaver som skulle plasseres i backloggen til produktet. Disse oppgavene var knyttet til resultatmålene med høyest prioritet og tilsvarende user stories. Fra den første sprinten var det også nødvendig å få en ny og oppdatert wireframe fra tilbakemeldingene gruppen fikk i første sprint review. Midt i sprinten ble det holdt et kort møte med Matsentralen for å vise frem den oppdaterte wireframen. Møtet resulterte i noen få ønsker til endringer, samt tilgang til wireframe for å vise frem til andre ansatte i Matsentralen. Agenda og referat er under møte 3 i Vedlegg B – Prosjekthåndbok. I slutten av sprinten ble sprint review gjennomført der gruppen ble invitert til Matsentralen sine lokaler for å en bedre forståelse om hvordan arbeidsflyten er under innlevering og utlevering av varer. Ved dette tidspunktet var ikke produktet utviklet langt nok til at det var mulig å få mye konstruktive tilbakemeldinger, av den grunn var fokuset mer på hva som skal gjøres til neste sprint. Gruppen hadde som mål å fokusere på gjør ferdig gamle oppgaver. Agenda og referat er under møte 4 i Vedlegg B – Prosjekthåndbok.

I innledningen til sprint 3 visste gruppen at mye tid kom til å forsvinne til vurderingssituasjoner i et annet emne, samt presentasjon av prosjektet til medstudenter og veiledere. Som resultat ble det da laget mindre oppgaver i backloggen under sprint planlegging, og at fokuset gikk på å fullføre oppgavene som var igjen fra forrige sprint. Ved sprintens slutt var gruppen ferdig med alle resterende oppgaver fra sprint 2, samt at de oppgavene som ble plassert i backloggen for denne sprinten også ble fullført eller flyttet til neste sprint. Til sprint review hadde produktet fått all den kritiske og mest ønskede funksjonaliteten, og var dermed mer konstruktive tilbakemeldinger fra Matsentralen. Til neste sprint ville gruppen øke antall standups, slik at samarbeid og samkjøring blir lettere i denne kritiske slutfasen til prosjektet.

Sprint planlegging til sprint 4 ble holdt i slutten av sprint 3, ettersom uken etter var påske og gruppemedlemmene hadde bestemt for å ikke ha obligatorisk arbeidstid. Under planleggingen ble det bestemt at fokuset skulle være på å implementere de siste funksjonalitetene, og å fullføre resterende av oppgavene i backloggen. Etter påsken var over begynte gruppen å ha flere standups gjennom sprinten. Da slutten av sprinten ankom ble siste sprint review avholdt, her fikk gruppen vist frem et forslag til det endelige produktet. Det ble bestemt at brukertester skulle bli gjennomført i løpet av neste og siste sprint, der noen tilbakemeldinger om ønskede endringer ble gitt.

Sprint 5 var den siste sprinten som var planlagt til utvikling av produktet, av den grunn var resultatet fra sprint planlegging at gruppen skulle bare ferdigstille produktet og ikke legge til noe mer funksjonalitet. Målet var at produktet var klart til brukertesten midt i sprinten, og deretter ville resten av sprinten brukes til å ta i bruk tilbakemeldingene som ble mottatt under testene. Testene ble gjennomført som planlagt ved Matsentralen sine lokaler og gruppen fikk konstruktive tilbakemeldinger fra personer som var kjent med matsentralen sin

vanlige arbeidsflyt. Etter testene ble resten av sprinten brukt til å gi ferdigstille produktet med responsen fra brukertestene.



Figur 15: Oversikt over sprintene på GitLab

5 Diskusjon

5.1 Diskusjon av vitenskapelige resultater

5.1.1 Iterative systemforbedringer

Gjennom utviklingsprosessen ble det gjort flere endringer i systemet etter hvert som gruppen lærte hva som fungerte best og Matsentralen konkretiserte ønskene sine. I ettertid ser gruppen at en rekke endringer og klargjøringer kunne vært unngått om gruppen hadde vært mer aktiv i å ta beslutninger for hvordan systemet skulle fungere. I begynnelsen endte gruppen opp med å implementere mange av kontaktpersons ønsker direkte, uten å diskutere med produkteier om potensielle bedre løsninger som er i henhold til gode standarder. Dette førte da til en rekke overkompliserte design som senere måtte endres.

Utlevering - Vektgruppe

«Utlevering»- og «Vektgruppe»-databasetabellene ble først designet slik at de lignet mer på regnearkene. Regnearkene hadde en kolonne per vektgruppe i utlevering regnearket. Dette kunne ikke direkte oversettes til en relasjonstabell, ettersom en tabell må ha et fast antall kolonner. For å etterligne den eksisterende løsningen mest mulig ble det laget en mange-til-mange relasjon mellom utleveringer og vektgrupper. Koblingstabellen «UtleveringVekt» hadde et attributt for bruttovekt. For å regne ut total bruttovekt for en utlevering måtte man da summere alle bruttovektene i «UtleveringVekt» for en utlevering. Ved å gjøre det på denne måten var hele utleveringen samlet som en rad i «Utlevering»-tabellen.

Gjennom videre samtaler med Matsentralen kom gruppen frem til at denne løsningen var unødvendig komplisert. Matsentralen brydde seg ikke om det var en eller flere rader per fysisk utlevering. Ved innhenting av statistikk i det eksisterende systemet anser de alle utleveringer til en organisasjon på samme dato som en utlevering, selv om det er registrert over flere rader. Gruppen ble derfor enige om å forenkle systemet. «UtleveringVekt»-tabellen ble fjernet og en utlevering rad besto nå heller av en bruttovekt og en vektgruppe. En utlevering bestående av flere vektgrupper registreres nå som en rad per vektgruppe i «Utlevering»-tabellen. Dette er den samme måten som innleveringer blir lagret på.

Varegruppe

Ved første iterasjon ble «Varegruppe»-tabellen designet med en relasjon til en «Hovedgruppe»-tabell slik det var i regnearkene. Etter hvert kom Matsentralen med et ønske om flere nivå av hierarki på varegruppene. Hovedgruppe-tabellen ble da erstattet med en fremmednøkkel fra varegruppe-tabellen til seg selv, slik at varegruppene fikk en hierarkisk tre-struktur.

Bruker

Ved første møte ble partene enige om å bruke Matsentralens eksisterende Google-brukere for innlogging. Brukerne var allerede koblet til Google Workspace hvor man kan lage grupper og administrere brukerne. Tanken var å dele brukerne på Workspace inn i forskjellige grupper basert på lokasjon og roller, for å så sjekke hvilken gruppe brukeren tilhørte i webapplikasjonen. Ved et senere møte ble det klart at Matsentralen ikke ønsket å måtte bruke Google Workspace sitt komplekse brukergrensesnitt til å holde styr på disse

gruppene. Rolle- og lokasjonshåndtering ble derfor i stedet implementert i produktet. Systemet er derimot fremdeles avhengig av Workspace for å sjekke om brukeren eksisterer i Matsentralens domene. Nedsidene i forhold til den originale planen var blant annet økt utviklingstid og tettere kobling mellom brukerne og registreringssystemet.

5.1.2 Standardisert, skalerbart og brukervennlig

Standardisert

Ønsket om å standardisere registreringene kommer fra problemene Matsentralen har hatt med å innhente statistikk fra flere ulike regneark-systemer. Det har foregått svært mye diplomati for å holde hver lokasjon fornøyd med registreringsmetoden, i og med at ingen av lokasjonene har hatt lyst til å endre på sitt eget system. Med det nye systemet er alle leveranser lagret i en felles database og registreres identisk, med unntak av lokasjonsbaserte vektgrupper og varegrupper. Løsningen er dermed tilpasset hver lokasjon på utsiden, men er standardisert på innsiden.

Skalerbart

SQL-databasen kan skales ved å oppgradere størrelsen og/eller datakraften. De fleste skydatabase-tjenestene tilbyr også jevnlig automatisk sikkerhetskopiering. REST-APIet kan skales vertikalt ved å oppgradere datakraften til maskinen og/eller horisontalt ved å kjøre API-et på flere maskiner og introdusere en belastningsfordeler som fordeler forespørslene til de ulike maskinene. Ettersom det er et REST-API kreves det ingen kommunikasjon mellom maskinene. Frontend krever veldig få tjener-ressurser ettersom det er en SPA.

Skalerbarhet handler ikke bare om å håndtere økende mengder data og aktive brukere, men også fremtidige endringer, omstruktureringer og ny funksjonalitet. Med den eksisterende løsningen kan Matsentralens ansatte selv legge til nye regneark, kolonner og koblinger når det er behov for det. Med det nye systemet må de ansette utviklere eller hyre inn et konsulentfirma for å gjøre endringer.

Dersom de for eksempel ønsker å registrere matsvinn og samsvarende avfallshåndtering, må utviklere endre både databasen, REST-APIet og frontend. Det må opprettes en tabell i databasen med attributter og fremmednøkler og eventuelt kjøre en database-migrasjon. I REST-APIet må det opprettes endepunkt med tilsvarende dataoverføringsobjekt, autorisering, validering og testing. Til slutt må det opprettes nye sider i frontend og eventuelt ny funksjonalitet i administrator-panelet.

Det er ingen tvil om at det nye systemet innfører større friksjon for endring sammenlignet med regnearkene. Det kan derfor argumenteres for at det nye systemet er mindre skalerbart, grunnet Matsentralens størrelse og budsjett. Fordelen er når endringene først er implementert vil de være standardisert for alle lokasjonene. I tillegg kan ny funksjonalitet i regnearkene forverre kaoset som allerede plager dem. Det kan derfor også argumenteres for at det nye systemet er mer skalerbart.

Brukervennlig

Brukergrensesnittet ble designet med en tabellstruktur. Tabeller er en strukturert og ryddig måte å presentere data på som ligner på den gamle registreringsmetoden. Tanken bak designet var at det skulle gjøre brukergrensesnittet enkelt å forstå og brukervennlig, selv for de som er vandt med det gamle systemet.

Ettersom de fleste har brukt regneark før, er det eksisterende systemet nokså selvforklarende. På den andre siden er regneark et generelt verktøy som kan brukes til veldig mye. Det nye systemet er spesialdesignet til Matsentralens behov og inneholder ingen overflødig funksjonalitet. I tillegg har systemet validering som reduserer risikoen for brukerfeil.

5.1.3 Normalisering

Ifølge nåværende krav fra Matsentralen er databasen på tredje normalform. Det kan derimot argumenteres for at det her er enda et krav der gruppen skulle ha bestemt mer over produktet. Leverandørtype ble bestemt til å være en fri streng. Bruken av feltet fra regnearket viser derimot at feltet kanskje burde ha blitt dratt ut og lagret i en egen tabell for å minimere overflødighet. Leverandørtype er derimot også ofte søkt etter for å finne leveringer koblet til leverandørtype, så å ofre litt overflødighet her for ytelse er kanskje riktig.

Leverandør		Aktiv	Leverandør	Type		
PK	Id	Yes	3T	HORECA		
		No	Ahead	Produsent		
FK	SelskapId	Yes	Andre	Andre		
		No	Andre	Produsent		
		Yes	ASKO (NG)	Grossist		
		Yes	Bama	Grossist		
		Yes	BaRe (Bama Rema)	Grossist		
		Yes	Blomsterringen	Grossist		
		Yes	Clarion brattøra	HORECA		
		No	Colgate	Produsent		
		Yes	Coop Norge AS	Grossist		
		Yes	Domstein Sjømat	Produsent		
		Yes	Dyrebeskyttelsen	Butikk		
		Yes	Eggen Gardsysteri	Produsent		
		Navn: string Aktiv: boolean LeverandørType: string Adresse: string				

Figur 16: Leverandør-tabell og leverandør regneark.

5.1.4 Brukertester

Brukertestene hadde svært gode resultater, men de hadde fremdeles noen feilkilder som må nevnes. For det første ble det grunnet en svært travel periode hos Matsentralen kun tid til å utføre to brukertester, så systemet ble ikke testet like grundig som originalt planlagt. For det andre førte rekkefølgen av info på oppgavelappene til noe forvirring. Lappene oppførte info i rekkefølgen «vare, leverandør, vekt», i motsetning til «leverandør, vare, vekt», som er rekkefølgen regnearket bruker. Viktig info som «ren donasjon» sto også oppført på slutten av lappene, som ved en ekte levering hadde kommet først. Dette skapte forvirring når data måtte slettes fra formet og legges til på nytt for å registrere lappen riktig. Videre er det også mulig at testtakeren med manglende datakunnskaper ble hjulpet for mye under testen. Testteamet holdt seg for det meste som observatører, men uten å ha blitt trent opp noe som helst i det nye systemet, ble testtakeren til tider forvirret med datamaskinen og måtte dyttes i riktig retning.

5.2 Diskusjon av ingeniørfaglige resultater

5.2.1 Resultatmål

I utviklingsfasen begynte gruppen først med de høyest prioriterte funksjonalitetene. Dette inkluderte registrering av innleveringer og utleveringer. Etter å ha implementert dette fikk gruppen flere ønsker fra produkteier om flere punkter som skulle registreres, blant de var at et notat skulle kunne skrives om registreringen og at flere leverandører eller organisasjoner kunne registreres samtidig. Oppdaterte krav kom kontinuerlig gjennom utviklingsfasen og ble implementert etter de ble mottatt.

Videre arbeidet gruppen mye med de administrative funksjonalitetene i produktet. Blant disse var hva de lokale og nasjonale administratorene skulle ha muligheten til å gjøre, samt hva bare de nasjonale administratorene hadde tilgang til. Også her ble det gitt oppdaterte krav senere ut i utviklingsfasen. Noen av de viktigste endringene i de administrative funksjonalitetene var at nasjonal administrasjon kunne velge hvilken lokasjon de fremstår i og de skulle være ansvarlig for varegrupper. Varegruppene skal ha mulighet til å være registrert i en trestruktur. Disse varegruppene kunne lokale administratorer velge hvilke som deres matbank skulle bruke. Dette ble gjort for å gjøre det lettere å samle informasjon felles for matbankene. Håndtering av organisasjoner ble også bestemt til å bli gjort i et annet produkt, og at dette produktet skulle kobles opp imot en felles database når det var klart. Den største endringen i de administrative funksjonalitetene var hvordan brukere og autentisering ble gjort, det skulle originalt bli gjort via produkteierens Google Workspace. Etter diskusjon ble det endret til at kun verifikasjon av selve e-posten vil bli gjort via Google Workspace, men at lagring av brukerdetaljer vil bli gjort i databasen.

Det første av de gjenstående resultatmålene som ble modifisert eller ikke implementert er at det skulle sendes ut kvittering til organisasjonene ved utlevering som hadde lav prioritet og som produkteier hadde planer om en annen implementasjonsform, og ble derfor ikke gjennomført. Det andre var at slettede registreringer skal havne i en papirkurv, som gruppen og produkteier så på som unødvendig og overflødig. Alle målene som var relatert til avfallshåndtering ble også nedprioritert og ikke implementert siden dette ikke var like viktig som andre funksjonaliteter ifølge produkteieren. Et annet resultatmål som ikke ble implementert var at nasjonal administrator kunne legge til nye datapunkter ved registrering som ble dømt til å være for tidskrevende, ettersom det ville kreve endringer av tabeller i databasen. Det siste resultatmålet som ble etterspurt var om å kunne filtrere visning av tidligere registreringer etter dato og leverandør/organisasjon, men ettersom det var i slutten av utviklingsfasen ble bare dato implementert.

5.2.2 Ikke-funksjonelle egenskaper og krav

De ikke-funksjonelle egenskapene og kravene som ble skrevet i begynnelsen av prosjektet hadde som formål at gruppen hadde kriterier å jobbe etter under alle arbeidsfaser. De fleste av disse var gruppen vellykket i å gjennomføre som nevnt i kapittel 4.2.3 Ikke-funksjonelle egenskaper og krav, blant disse var dokumentasjon og struktur i kildekoden.

I kapittelet var det også nevnt om tre ikke-funksjonelle egenskaper og krav som ikke ble oppfylt eller delvis oppfylt, første av dem var testing. Tester har bare blitt laget for backend i produktet, disse er av typen enhetstester til de forskjellige metodene. Frontend har foreløpig

ingen tester, dette er av den grunn at frontend ikke består av mange metoder, men isteden er det flere sammensatte komponenter. Gruppen bestemte seg at testing av frontend ikke var like relevant til hele produktet var klar til produksjon. Det neste punktet som ikke ble oppfylt var at produktet skulle kobles opp til Google Looker Studio for visuell fremvisning av data, som i etterkant ble funnet ut av at dette er noe produkteier selv må koble seg til. Det siste punktet som ikke var oppfylt var at produktet skulle ha høy oppetid, som ikke har vært mulig å sjekke ettersom det ikke er i produksjon enda.

5.2.3 Universell utforming

Universell utforming var ikke noe gruppen hadde veldig stort fokus på. Produktet ikke er en offentlig tilgjengelig nettside og er bare ment for internt bruk, av den grunn er det ikke lovpålagt at den skal følge 47 av de 78 suksesskriteriene i WCAG 2.1 standarden. [12] Til tross for at det ikke er pålagt at produktet skal følge WCAG 2.1 standarden mener gruppen at det er viktig at den sluttproduktet prøver å følge den. Et annet argument til å følge WCAG 2.1 standarden er at Matsentralen ofte har frivillige som arbeider med registrering av innleveringer og utleveringer, dermed vil det være naturlig at produktet prøver å følge prinsippene i standarden. Til slutt har gruppen prøvd å gjør at produktet følger flest mulig av de viktigste og relevante suksesskriteriene, så lenge det ikke avviker fra ønsker og krav fra produkteier.

5.3 Diskusjon av administrative resultater

5.3.1 Fremdriftsplan

Fremdriftsplanen som gruppa lagde til prosjektet hadde ikke mange detaljer, den inkluderte bare obligatoriske gjøremål og sprintene. Det var heller ingen avvik i fra planen, med unntak av påske som ikke var ført opp. Til tross for at fremdriftsplanen ikke hadde mye innhold eller detaljer, var gruppen positiv til at det var et sted hvor det var enkelt å finne alle viktige datoer og obligatoriske gjøremålene. Det ble også lettere å planlegge fremtidige møter med produkteier og å gi informasjon om hvordan planen var fremover.

5.3.2 Timeregnskap

Gruppen har gjennom hele prosjektet jobbet jevnt og trutt. I begynnelsen var det som forventet mindre arbeid gjort på prosjektet, ettersom det måtte avtales og planlegges med produkteier. Videre ut i prosjektet var det naturligvis en økning i samlet tidsbruk. Det var også litt nedgang rundt uke 8 grunnet eksterne faktorer, som andre vurderingssituasjoner. Totalt brukte gruppen samlet 1427 timer, som var litt under det totale forventede forbruket på 1500 timer, eller 500 timer per gruppemedlem.

5.3.3 Utviklingsmetodikk

Gruppen ville ha en smidig utviklingsmetodikk for å være best mulig egnet til å tilpasse seg etter produkteiers ønsker og krav. Gruppen har hatt erfaring med Scrum før og vet flere av artefaktene i Scrum ville være godt egnet til dette prosjektet. Samtidig er også Scrum mer egnet til en større gruppe, av den grunn valgte gruppen en modifisert versjon av Scrum med bruk av Kanba-tavle. De Scrum artefaktene som gruppen ikke valgte å bruke var sprint

retrospektiv, Scrum master og daglige standups. De ble valgt vekk hovedsakelig på grunn av størrelsen på gruppen, siden disse artefaktene er best egnet til større grupper.

Gruppen ble fornøyd med valget av utviklingsmetodikk. Metodikken gjorde det enkelt å implementere nye krav og ønsker. Kanban-tavlen var også et verktøy som gruppen var tilfreds med, denne gjorde lett å vite hvilke oppgaver som trengtes å gjøre, som var påbegynt og andre tilstander. Gjennom hele prosjektet hadde hele gruppen struktur om hvordan den iterative arbeidsprosessen skulle gjennomføres, som gjorde det enklere for gruppen siden det ikke var like mye tid som måtte gå til å planlegge og samkjøre i hver periode.

5.3.4 Utviklingsprosess

Samarbeidet under utviklingsprosessen har vært bra. Gruppen har vært gode på å kommunisere med hverandre, dette har gjort at det har blitt lettere å gi eller motta hjelp. Alle har bidratt til å gjøre dette produktet en realitet. Under de fem sprintene har det ikke vært noen store konflikter mellom gruppe medlemmer. Det har vært et par diskusjoner om hva som er best å gjøre, men ettersom gruppen består av tre medlemmer har det vært lettere å ta litt vanskelige beslutninger med tanke på at det er mulig å få flertall i en diskusjon.

Dialogen med produkteier har vært produktiv for begge sider, det har vært enkelt for alle parter å få kontakt med hverandre. Dette har gjort det enkelt å tilpasse møter og informasjonsflyten mellom produkteier og gruppen. For hver sprint var det nye og oppdaterte krav fra produkteier. Grappa har vært flinke på å tilpasse seg etter disse ønskene og å implementere det i produktet.

Selve utviklingsprosessen har vært som forventet. Gruppen har holdt seg til rutinene som var satt før arbeidet begynte, som sprint planlegging, standups og sprint review. I Figur 15 er det mulig å se hvordan arbeidet er fordelt gjennom sprintene. Noen av arbeidsoppgavene til sprintene har blitt tildelt feil sprint. Gruppen hadde litt sviktende kommunikasjon om de skulle registreres til når de ble laget eller når de ble gjennomført.

6 Konklusjon og videre arbeid

6.1 Konklusjon

Målet med prosjektet var å transformere Matsentralens ulike system av regneark til et standardisert full stack system som forenkler registreringsprosessen, effektiviserer dataanalysen og muliggjør videre vekst. Det ble utviklet en web-applikasjon og et REST-API med en relasjonsdatabase. Løsningen har oppfylt de viktigste resultatmålene, men systemet er ikke i drift og tidligere data har derfor ikke blitt flyttet fra regnearkene til en database.

Systemet har standardisert dataen på tvers av alle lokasjonene, med unntak av varegrupper og vektgrupper. Varegrupper og vektgrupper ble implementert som en hybrid mellom standardisering og personalisering slik at systemet møter alle lokasjoners behov. Systemet krever mer kompetanse for å vedlikeholde og videreutvikle sammenlignet med regnearkene. Til gjengjeld er systemet mer skalerbart og robust. Brukergrensesnittet er spesialdesignet til å passe Matsentralens behov, og tar inspirasjon fra tabellstrukturen til regnearkene i det eksisterende systemet. Tabellstrukturen er en ryddig måte å presentere dataen og gjør overgangen mellom systemene enklere for brukerne.

Gruppen valgte en smidig utviklings metode, der mange av artefaktene fra Scrum ble benyttet. Etter møte med Matsentralen ble det laget resultat- og effektmål, samt noen ikke-funksjonelle egenskaper og krav. I utviklingsprosessen fikk gruppen visst frem hva som hadde blitt gjort, og fikk tilbakemeldinger om det og om hva som burde gjøres videre. Som resultat av tilbakemeldingene har gruppen tilpasset produktet og noen av de originale målene. Enkelte mål har blitt nedprioritert, mens andre har oppstått med høy prioritet. Gruppen har hatt et godt samarbeid gjennom hele prosjektet og har fått uvurderlig erfaring om både teknologiene som ble brukt og kommunikasjon med en ekte klient.

6.2 Videre arbeid

Det er flere ting som kan gjøres videre med produktet. Flere av resultatmålene med lavere prioritet ble ikke implementert til fordel for resultatmål med høyere prioritet. Matsentralen ønsker å kunne registrere matsvinn og tilsvarende avfallshåndtering. Ved sletting av leveranser må brukeren nå bekrefte handlingen før leveransen blir slettet, men det kan likevel være ønskelig å implementere en papirkurv som hindrer uønsket permanent sletting av data. Ved utlevering til en organisasjon er det ønskelig at det automatisk sendes en kvittering på e-post til organisasjonen. Ved visning av tidligere leveranser er det nå mulig å filtrere på dato, men det er også ønskelig å kunne filtrere på leverandør/organisasjon. Det må implementeres et «Er du sikker» pop-up ved forlatelse av registreringssidene. Ved å sette formkomponentene under trekkspillkomponenter mot slutten av prosjektet, mangler det også tilbakemeldinger om skjerm status mens komponentene er lukket.

Det mest nærliggende arbeidet som gjenstår, er å ta systemet i bruk og drifte løsningen. I tillegg må tidligere data migreres fra regnearkene til databasen og Looker Studio kobles opp til databasen og konfigureres deretter. Videre vil det også være nødvendig å skrive flere testing for å gjøre produktet mer stabilt, dette vil inkludere diverse typer testing i frontend

og ende-til-ende testing i backend. Det siste av det mest nærliggende arbeidet vil da være å prøve å finne eventuelle feil i programmet og gjøre det helt klart til drift.

Etter hvert som Matsentralen fortsetter å utvikle seg vil nok også deres krav og behov endres. For å maksimere systemets effekt bør systemet kontinuerlig vedlikeholdes og utvikles i takt med Matsentralens ønsker.

Samfunnspåvirkning

Profesjonsetiske problemstillinger

Som utviklere av et registreringssystem brukt av både frivillige og ansatte har vurderingen av stabilitet, sikkerhet og tilgjengelighet vært en sentral del av arbeidet. En av utfordringene med utviklingen av produkter som skal bli brukt av både ansatte og frivillige er gjøre det enkelt for alle å bruke, samt som beholder den nødvendige kompleksiteten for å fungere. Videre kan sikkerhet være et dilemma når frivillige skal ha tilgang til produktet, det vil da være nødvendig å sette inn de korrekte sikkerhetsbegrensningene. Til slutt er stabilitet en viktig faktor i et registreringssystem, slik at innholdet er tilgjengelig og ikke går tapt.

Hensikten for denne oppgaven har vært å effektivere og forenkle registreringsprosessen til et registreringssystem. Drøfting av dette har blitt gjort gjennom universell utforming, resultatmål og mer i diskusjonsdelen av rapporten.

I produkter som kan bli brukt av flere aktører, som ansatte og frivillige, er det nødvendig at brukeropplevelsen er god og tilpasset alle. Samtidig er det også viktig at produktet beholder de kritiske funksjonalitetene som får produktet til å være effektivt, uansett om dette fører til økt kompleksitet. Dersom det er dårlig balanse mellom brukervennlighet og kritisk funksjonalitet vil det føre til at blant annet at effektiviteten til produktet ikke oppnår målene, at brukere ikke vil bruke det nye systemet og at frivillige er mindre motivert til å jobbe.

Et registreringssystem innebærer innsamling av potensiell sensitiv data. Det er viktig å sikre at disse dataene blir håndtert på en forsvarlig måte og at det er klare retningslinjer for tilgang av informasjonen. Prinsippet om det minste privilegium krever at enhver aktør kun har tilgang til den informasjonen og de ressursene som er nødvendige. Sikkerhet må implementeres i alle deler av systemet, fra datalagring til brukergrensesnitt. Dersom det skulle være sikkerhetshull i en del av systemet kan det i verste fall føre til datalekkasje av all informasjon eller andre dataangrep.

Bedrifter er avhengige av at de digitale verktøyene de bruker fungerer i bedriftens daglige drift. En sentral del av systemutvikling er dermed å maksimere oppetid og minimere feil. Man bør ha en plan for hvordan man skal oppnå ønsket oppetid og en plan for hva man skal gjøre ved eventuell nedetid. Dersom det skulle være lengre perioder med nedetid uten noe gunstig erstatning kan det ha alvorlige konsekvenser for bedriften. Feil i systemet kan føre til feil i innsamling, lagring og kommunikasjon av data, som igjen kan ha alvorlige konsekvenser for bedriften.

Bærekraftvurderinger

Hvert år kastes det mer enn 450 000 tonn spiselig mat i Norge. Samtidig anslår FN at rundt 10% av befolkningen i Norge lever under den relative fattigdomsgrensen. Matsentralen hjelper aktører i matbransjen med å redusere sitt matsvinn og samtidig tar et samfunnsansvar som bidrar til 7 av FNs 17 bærekraftsmål [20]. Registreringssystemet kan hjelpe Matsentralen å effektivisere og forenkle prosessene sine og muliggjør videre vekst for å redusere matsvinnet enda kraftigere og hjelpe enda flere.

Omtrent 85 kg spiselig mat per innbygger kastes i Norge hvert år. Kategoriene det kastes mest av er ferske bakervarer, frukt og grønt og måltidsrester. 48% av matsvinnet stammer fra husholdninger, videre kommer 43% fra matindustrien, dagligvarehandelen og jordbruk. I 2022 delte Matsentralen ut 5508 tonn mat til ideelle organisasjoner rundt i landet. Dette tilsvarer omtrent 11 millioner måltider. Det er også kun omtrent 1,2% av alt matsvinn i Norge. FNs bærekraftsmål 12.3 er å halvere matsvinnet målt i kilo per innbygger i dagligvarehandelen og hos forbrukerleddet innen 2030. Matsvinn er en stor del av klima- og miljøproblemene ettersom matproduksjon krever land, vann, energi og arbeidskraft. I tillegg er det store klimagassutslipp fra produksjon, transport og pakking av maten, til avfallshåndtering av emballasjen.

Forskjellene i det norske samfunnet øker og rammer spesielt utsatte grupper som blant annet flyktninger, rusavhengige, minstepensjonister, langtidsledige, enslige forsørgere, uføretrygde, hjemløse, innvandrere og tidligere innsatte. Over 115 000 barn i Norge vokser opp i familier med lav inntekt og nesten 4 av 10 innvandrerbarn lever med vedvarende lav inntekt. Mange risikerer å falle utenfor både økonomisk og sosialt. Matsentralen samarbeider med 537 mottakerorganisasjoner som i tillegg til å omfordele overskuddsmaten, ofte også tilbyr tjenester som leksehjelp, veiledning til utdanning og jobb, språkopplæring, rehabilitering for avhengighet, arbeidstrening, utdeling av klær, veiledning/hjelp fra sykepleier, hjelp med oppfølging NAV, psykolog, juridisk rådgivning og økonomisk rådgivning.

Registreringssystemet kan hjelpe matbransjen, ideelle organisasjoner, enkeltpersoner, miljøet og klimaet indirekte gjennom å være et sentralt verktøy for Matsentralens daglige drift.

Referanser

- [1] «Stiftelsen Miljøfyrtårn», 20. januar 2016. <https://www.miljofyrtarn.no/> (åpnet 19. mai 2023).
- [2] T. Mallaug og K. T. Hansen, «IDATT2103 Databaser (2021 HØST) - Relasjonsmodellen». NTNU.
- [3] K. T. Hansen og T. Mallaug, *Databaser*, 2. utg. Oslo: Gyldendal Akademisk og Stiftelsen TISIP, 2019.
- [4] R. T. Fielding, «Architectural Styles and the Design of Network-based Software Architectures», 2000. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (åpnet 2. mai 2023).
- [5] «REST APIs Explained - 4 Components». <https://mannhowie.com/rest-api> (åpnet 20. mai 2023).
- [6] T. H. Nätt, «autorisering – informasjonssikkerhet», *Store norske leksikon*. 26. januar 2023. Åpnet: 1. mai 2023. [Online]. Tilgjengelig på: https://snl.no/autorisering_-_informasjonssikkerhet
- [7] T. H. Nätt og S. J. Knapkog, «autentisering», *Store norske leksikon*. 26. januar 2023. Åpnet: 1. mai 2023. [Online]. Tilgjengelig på: <https://snl.no/autentisering>
- [8] D. Hardt, «The OAuth 2.0 Authorization Framework», Internet Engineering Task Force, Request for Comments RFC 6749, okt. 2012. doi: 10.17487/RFC6749.
- [9] «Final: OpenID Connect Core 1.0 incorporating errata set 1». https://openid.net/specs/openid-connect-core-1_0.html (åpnet 1. mai 2023).
- [10] «What is the difference between SPAs, SSGs, and SSR?», *Hygraph*, 16. juni 2021. <https://hygraph.com/blog/difference-spa-ssg-ssr> (åpnet 20. mai 2023).
- [11] Inger Marie Lid, «universell utforming», *Store Norske Leksion*, 27. desember 2021. https://snl.no/universell_utforming (åpnet 18. mai 2023).
- [12] «Universell utforming», *Regjeringen*. <https://nettsteder.regjeringen.no/standarddesign/universell-utforming/> (åpnet 16. mai 2023).
- [13] «Oppbygging av WCAG 2.1», *uutilsynet*. <https://www.uutilsynet.no/wcag-standarden/oppbygging-av-wcag-21/139> (åpnet 15. mai 2023).
- [14] «Principles behind the Agile Manifesto». <http://agilemanifesto.org/principles.html> (åpnet 4. mai 2023).
- [15] K. Schwaber og J. Sutherland, «The Scrum Guide», nov. 2017. [Online]. Tilgjengelig på: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- [16] pmadmin, «What is Scrum? | The Agile Journey with PM-Partners», *PM Partners*, 22. juni 2021. <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/> (åpnet 20. mai 2023).
- [17] H. Kniberg og M. Skarin, *Kanban and Scrum: making the most of both*. i InfoQ enterprise software development series. s. l.: C4Media, 2010.
- [18] G. Kim, J. Humble, P. Debois, J. Willis, og N. Forsgren, *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution, 2021.
- [19] E. Toftøy-Andersen og J. G. Wold, *Praktisk brukertesting*. Oslo: Cappelen Damm akademisk, 2011.
- [20] «Matsentralen Norge - Food Banks Norway». <https://www.matsentralen.no> (åpnet 18. mai 2023).

Vedlegg

Vedlegg A - Forprosjektplan

Vedlegg B - Prosjekthåndbok

Vedlegg C - Visjonsdokument

Vedlegg D - Kravdokumentasjon

Vedlegg E - Systemdokumentasjon

Vedlegg F - Testplan

Vedlegg G - Brukertester

Vedlegg H - Kildekode

