

Henna Fredrika Björkskog  
Truls Henrik Jakobsen

# Videreutvikling av webapplikasjon for undervisningsopplegg

Bacheloroppgave i Informasjonsbehandling  
Veileder: Atle Olsø  
Mai 2022



Henna Fredrika Björkskog  
Truls Henrik Jakobsen

# Videreutvikling av webapplikasjon for undervisningsopplegg

Bacheloroppgave i Informasjonsbehandling  
Veileder: Atle Olsø  
Mai 2022

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



Kunnskap for en bedre verden



---

## Forord

Denne rapporten er skrevet i forbindelse med prosjektoppgaven ”Videreutvikling av webapplikasjon” som utføres som bacheloroppgave i studieprogrammet ”Bachelor i Informasjonsbehandling” ved Institutt for datateknologi og informatikk, NTNU, våren 2022.

Prosjektet baserer seg på et tidligere prosjekt som ble utført som bacheloroppgave av en gruppe studenter ved NTNU våren 2021 og er en videreutvikling av dette. For å skille mellom de to prosjektene har vi i rapportene valgt å kalle den tidligere prosjektoppgaven ”eksisterende applikasjon”, og vårt prosjekt blir kalt ”ny applikasjon eller ”videreutviklet applikasjon”. NTNU/AIT ved Majid Rouhani er oppdragsgiver, og produktet utvikles med tanke på muligheter for senere videreutvikling og utvidelse av applikasjonen.

Studieprogrammet er nettbasert, og bacheloroppgaven er utført som et heldigitalt prosjekt. Grupped medlemmene har samarbeidet digitalt fra henholdsvis Oslo og Trondheim. Alle møter mellom grupped medlemmer, oppdragsgiver og veileder vært digitale. Dette har fungert godt takket være digitale samhandlingsplattformer som Discord, Google Drive og Zoom. I løpet av prosjektet har vi fått god innsikt i hva som kreves for et godt teamarbeid, og har lært mye om hvordan gjennomføre et produktutviklingsprosjekt fra start til ende. Dette er kunnskaper vi vil ta med oss videre etter endt studie.

Gruppen ønsker å takke veileder Atle Olsø for god hjelp, støtte og tilbakemeldinger underveis i prosjektet. Vi ønsker også å takke Majid Rouhani for gode svar på spørsmål som har dukket opp under utviklingen av produktet. En stor takk rettes også til de lærere ved Ryenberget skole som har bidratt med brukertesting og tilbakemeldinger til produktet.

  
Henna Fredrika Björnskog

  
Truls Henrik Jakobsen

Oslo / Trondheim  
20. mai 2022

---

## Oppgavetekst

Hensikten med oppgaven er å videreutvikle en allerede påbegynt webapplikasjon. Oppdragsgiver ga ved starten av prosjektet et inntrykk av at den eksisterende applikasjonen ikke svarer til de krav og ønsker som ble stilt da det opprinnelige prosjektet ble utført våren 2021. Det var derfor et ønske om at vi i vår bacheloroppgave skulle arbeide videre med applikasjonen, basert på den eksisterende applikasjonen og på bakgrunn av samtaler med oppdragsgiver. I oppstartsmøtet mellom gruppe medlemmene, veileder og oppdragsgiver ble bakgrunnen til prosjektet forklart, og krav og mål til applikasjonen ble illustrert.

Webapplikasjonen er tenkt å være et ledd i et større produkt, og det er i utviklingsprosessen tatt hensyn til en mulig senere videreutvikling av applikasjonen. Webapplikasjonen skal i første omgang fungere som en type oppslagsverk for undervisningsopplegg som blir laget i forbindelse med fagene *IT6203 - Grunnleggende programmering for lærere* og *IT6204 – Anvendt programmering for lærere* ved NTNU, som ble lansert for lærere i den norske skolen som skal undervise i programmering men som mangler den kunnskap som trengs for dette. Målet med applikasjonen er at disse undervisningsoppleggene skal kunne lastes opp og gjøres tilgjengelig på internett, slik at andre lærere også kan benytte seg av disse.

---

## Sammendrag

Som en konsekvens av at programmering blir lagt inn i eksisterende fag i den norske grunnskolen, må lærere lære seg grunnleggende programmering. Dette kan være utfordrende både med tanke på mangel på tid, mangel på gode læringsressurser og manglende motivasjon.

NTNU har lansert to fag hvor grunnskolelærere kan lære programmering i regi av NTNU. I løpet av fagene vil lærerne lage enkle undervisningsopplegg som kan brukes i undervisning i grunnskolen. Nå ønsker NTNU å kunne legge ut disse undervisningsoppleggene online slik at andre lærere også kan benytte seg av disse oppleggene. Våren 2021 gjorde en gruppe studenter ved NTNU, som sin bacheloroppgave, et forsøk på å utvikle en webapplikasjon som kan tilby dette. Resultatet av den oppgaven var mangelfull, og applikasjonen møtte ikke de krav som ble stilt av oppdragsgiver.

Vi fikk derfor i oppdrag at som vår bacheloroppgave gjøre et nytt forsøk og videreutvikle den eksisterende applikasjonen. Vår problemstilling fokuserer på brukervennlighet, og lyder: **”Kan man lage en webapplikasjon som lærere i den norske grunnskolen ønsker å bruke når de skal søke etter og dele undervisningsopplegg for programmering i grunnskolen?”** med det utvidede spørsmålet *”Hvilken funksjonalitet må være med for at brukeropplevelsen skal være høy nok til at lærerne opplever bruken av webapplikasjonen som **verdt det?**”*.

For å svare på problemstillingen valgte vi å inkludere brukertesting som et viktig moment i vår prosess. I tillegg til å videreutvikle webapplikasjonen og gjøre den mer brukervennlig og legge til manglende funksjonalitet, rekrutterte vi også testpersoner for å sjekke applikasjonens brukervennlighet og brukskvalitet. Tilbakemeldingen vi fikk på videreutviklingen av applikasjonen var positiv, og samtlige testpersoner sa at de ville tatt i bruk applikasjonen dersom den ble lansert. På spørsmålet om hva som må til av funksjonalitet for at applikasjonen skal være *verdt* å bruke, var det viktigste svaret at det handler mer om kvaliteten på de undervisningsopplegg som lastes opp, enn om applikasjonen i seg selv.

Arbeidet med bacheloroppgaven har vist at dette produktet fyller en funksjon. Applikasjonen er langt på vei godkjent av brukerne allerede, og vi har i vårt arbeid sørget for å legge et godt grunnlag for en videreutvikling av applikasjonen. Et viktig innspill fra testpersonene er dog at det er viktig å sikre en god kvalitet på oppleggene som lastes opp også, for uten en god kvalitet på oppleggene mangler applikasjonen validitet.

---

## Abstract

As a consequence of programming being added to the syllabus of subjects in Norwegian primary and secondary schools, teachers have to learn how to do programming. This can be challenging for them considering their already pressed schedules, a lack of proper learning resources, and a middling amount of motivation.

NTNU has created two subjects where primary and secondary school teachers can learn the basics of programming. Throughout the subjects the teachers will create teaching plans featuring programming that can be used in regular classes in school. Now NTNU wants to publish these teaching plans online such that other teachers can also reap the benefits of their creation. In the spring of 2021 a group of students at NTNU made an attempt, as their bachelors thesis, of developing a web application that could offer this to NTNU. The result of that was deficient, and did not meet the specifications set by the client.

Our thesis is thusly another attempt at fulfilling NTNU's goal by continuing the development already done. Our research question has ease of use as a main focus and reads as follows: **"Is it possible to make a web application that teachers *want* to use when they are looking to find or share teaching plans for programming in primary and secondary school?"** with the extension *"What functionality has to be included for the user experience of the web application to be good enough for the teacher to feel that using it is **worth it**?"*.

To answer this research question we chose to include user testing as an important part of our process. In addition to further developing the existing application and making it more user friendly and adding functionality, we also recruited test users to properly check the usability and ease of use. The feedback we got from the testers was positive, and several of them said they would already use this application if it was available to the public. Regarding the question of what functionality has to present for it to be *worth* using, the most important answer we got was that the quality of the teaching plans really matters more than the quality of the application itself.

The work we have done with this thesis has shown that the product is fulfilling its purpose. The application is already being accepted by users, and we have laid the foundations for ease of further development. An important message we have gotten from several users is that we need a process for ensuring the quality of the teaching plans as well. If the application is good but the teaching plans are not, then the application is not really serving its purpose.



---

## Forkortelser og akronymer

**UDIR:** Utdanningsdirektoratet

**IKT:** Informasjons- og kommunikasjonsteknologi

**UX:** User experience (brukeropplevelse)

**UI:** User interface (brukergrensesnitt)

**CI:** Continuous integration (kontinuerlig integrasjon)

**SNL:** Store Norske Leksikon

**SSB:** Statistisk sentralbyrå

**uutilsynet:** Tilsynet for universell utforming av IKT

**GDPR:** General Data Protection Regulation

---

## Viktige begreper

**Smidig utvikling:** En utviklingsmetode hvor fokus er å ha tett samarbeid med kunden underveis i prosjektet, og ha hyppige del-leveringer.

**Kanban:** En metode for smidig utvikling som baserer seg på et tavleoppsett hvor arbeidsoppgaver er visualisert som kort. Arbeidsoppgavene i en Kanban-tavle kan ha ulik tilstand, f.eks. "Todo/Backlog", "Doing" og "Done". Syklustiden i Kanban-tavlen er basert på hvor lang tid det tar å flytte en oppgave fra "Todo", via "Doing", til "Done" (Atlassian 2022c).

**Scrum:** En smidig utviklingsmetode hvor produktutviklingsarbeidet blir delt opp i små bolker, kalt sprinter. Syklustiden i en Scrum-basert metode er tiden mellom to sprinter, som ofte er på rundt to uker, eller i et mer overordnet perspektiv på f.eks. et halvt år (Högstrand 2022).

**Programmering:** Skrive en rekke instruksjoner som en datamaskin (PC, mobil, vaskemaskin eller andre elektroniske apparater) skal utføre.

**Kunnskapsløftet:** Høsten 2020 ble det innført et nytt læreplanverk som gjelder for grunnsopplæringen, dvs. grunnskolen og videregående skoler (UDIR 2022c).

**Kompetansemål:** Definerede mål i læreplanene for den norske skolen (SNL 2022d).

**Digitalisering:** å legge tilrette for generering og håndtering/utnyttelse av digital informasjon ved hjelp av informasjonsteknologi (SNL 2022a).

**Grunnopplæringen:** Det 13-årige opplæringsløpet i Norge som omfatter grunnskole i 10 år, og videregående skole i 3 år (SNL 2022b).

**Grunnskole:** Den obligatoriske 10-årige skolen for barn og unge mellom 6 og 16 år i Norge (SNL 2022c).

**Webapplikasjon:** En dynamisk nettside som er utviklet for å gi en form for merverdi til brukeren (Wikstrøm 2022).

**Brukervennlighet:** *"Et system skal være lett å lære, effektivt, lett å huske, håndtere feil på en god måte, samt at det skal være behagelig å bruke"* - Are Halland (Ryvarden 2022).

**Blackboard:** En nettbasert læringsplattform som brukes ved NTNU.

**Refaktorisering:** Omstrukturering av programkode, hvor hensikten er å forbedre kvaliteten på koden.

---

# Innhold

<b>Forord</b>	<b>i</b>
<b>Oppgavetekst</b>	<b>ii</b>
<b>Sammendrag</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Forkortelser og akronymer</b>	<b>v</b>
<b>Viktige begreper</b>	<b>vi</b>
<b>1 Introduksjon og relevans</b>	<b>1</b>
1.1 Bakgrunn . . . . .	1
1.2 Problemstilling . . . . .	1
1.2.1 Problemet . . . . .	1
1.2.2 Formulering av problemstillingen . . . . .	1
1.3 Rapportens struktur . . . . .	2
<b>2 Teori</b>	<b>3</b>
2.1 Brukervennlighet . . . . .	3
2.1.1 Brukergrensesnitt - UI . . . . .	3
2.1.2 Brukeropplevelse - UX . . . . .	4
2.1.3 Brukertestning . . . . .	4
2.2 Universell utforming . . . . .	5
2.3 Programmering i skolen . . . . .	5
2.4 Systemutvikling . . . . .	6
2.4.1 Smidig utvikling . . . . .	6
2.4.2 Versjonskontroll . . . . .	7
2.5 Teknologi . . . . .	8
2.5.1 Relasjonsdatabaser . . . . .	8
2.5.2 Webapplikasjon . . . . .	9
2.5.3 Klient-server . . . . .	9
2.6 Personvern og informasjonssikkerhet . . . . .	9
<b>3 Valg av teknologi og metode</b>	<b>11</b>
3.1 Teknologi . . . . .	11

---

3.1.1	React	11
3.1.2	Typescript	11
3.1.3	React Bootstrap	11
3.1.4	MySQL	11
3.1.5	Sequelize	11
3.1.6	Visual Studio Code	11
3.1.7	GitHub	12
3.1.8	L <sup>A</sup> T <sub>E</sub> X	12
3.2	Metode	12
3.2.1	Administrative verktøy	12
3.2.2	Fordeling av arbeidsoppgaver	13
<b>4</b>	<b>Resultater</b>	<b>14</b>
4.1	Vitenskapelige resultater	14
4.2	Ingeniørfaglige resultater	24
4.2.1	Funksjonelle krav	25
4.2.2	Ikke-funksjonelle krav	25
4.2.3	Brukertesting	25
4.3	Administrative resultater	27
4.3.1	Prosjektplan	27
4.3.2	Timeregnskap	28
4.3.3	Produktutviklingsprosessen	29
<b>5</b>	<b>Diskusjon</b>	<b>30</b>
5.1	Drøfting av vitenskapelige resultater	30
5.2	Drøfting av ingeniørfaglige resultater	30
5.2.1	Funksjonelle krav	30
5.2.2	Ikke-funksjonelle krav	31
5.2.3	Brukertesting	31
5.3	Drøfting av administrative resultater	32
5.3.1	Prosjektplan	32
5.3.2	Timeregnskap	33
5.3.3	Produktutviklingsprosessen	33
<b>6</b>	<b>Konklusjon og videre arbeid</b>	<b>35</b>
6.1	Konklusjon	35

---

---

6.2 Videre arbeid . . . . .	35
<b>Bibliografi</b>	<b>37</b>
<b>Vedlegg</b>	<b>39</b>
A Visjonsdokument . . . . .	40
B Kravdokumentasjon . . . . .	50
C Systemdokumentasjon . . . . .	66
D Testplan . . . . .	81
D.1 Testplan . . . . .	81
D.2 Oppgaver . . . . .	81
D.3 Spørsmål . . . . .	82
E Respons på skjema fra brukertesting . . . . .	84

---

# 1 Introduksjon og relevans

## 1.1 Bakgrunn

Høsten 2020 ble det tatt i bruk nye læreplaner i den norske grunnskolen. De nye læreplanene baserer seg på Kunnskapsløftet 2020, som er et nytt læreplansverk for den norske grunnopplæringen (UDIR 2022c). Hensikten med de nye læreplanene er å gi elevene en relevant og fremtidsrettet læring. Som en del av det nye læreplansverket ble det besluttet at programmering skulle tas inn i den norske grunnopplæringen. Det var en stund ikke avklart om programmering skulle tas inn som et eget IKT-fag eller om det skulle legges inn i eksisterende fag. Etter hvert kom man frem til beslutningen at programmering skulle legges inn i eksisterende fag. I første omgang gjelder dette i matematikk, men dette vil etter hvert bli utvidet til å gjelde også andre fag (UDIR 2022a).

## 1.2 Problemstilling

### 1.2.1 Problemet

Det er noen utfordringer knyttet til å ta inn programmering i et eksisterende fag. For det første må det frigjøres plass for programmering i timeplanene, og det kan innebære at andre viktige temaer får mindre plass. Dersom det blir for mange temaer i kompetansekravene i læreplanen innebærer det at det blir lite tid for lærerne til å undervise i alle temaer. Dette kan i sin tur føre til at det blir mindre tid for elevene å lære de ulike temaene.

For det andre blir lærerne pålagt å undervise elevene sine i programmering, selv om de kanskje selv ikke har noen erfaring med programmering. Førsteamanuensis Odd Tore Kaufmann ved Høgskolen i Østfold sier i en artikkel på forskning.no: *”Mange lærere har mangelfull kompetanse innen programmering. De har lite kjennskap til hvordan programmering og matematikk kan integreres”* (Johansen 2022).

Vi kan altså se en utfordring i kombinasjonen mindre tid til rådighet, og et helt nytt tema som skal læres. Dette gjør at lærerne blir satt i en vanskelig situasjon. De er pålagt å lære programmering, og på veldig kort tid. De får heller ikke ekstra tid til å lære seg programmering for å kunne lære det bort til elevene, men må selv sette av tid til dette i tillegg til alle andre arbeidsoppgaver de har. En konsekvens av dette kan være at lærerne hverken har tid eller motivasjon til å lære programmering.

NTNU har lansert to fag for å hjelpe lærerne med å lære grunnleggende programmering: *IT6203 - Grunnleggende programmering for lærere* (NTNU 2022a) og *IT6204 - Anvendt programmering for lærere* (NTNU 2022b). I løpet av kursene vil lærerne lage enkle undervisningsopplegg i programmering som de kan bruke i sin undervisning i grunnskolen. Disse undervisningsoppleggene blir per nå lastet opp på Blackboard. Oppdragsgiver NTNU/AIT, ved Majid Rouhani som er fagansvarlig for disse programmeringsfagene, ønsker nå at disse undervisningsoppleggene skal kunne lastes opp fritt tilgjengelige på internett slik at andre lærere også skal kunne ha nytte av dem selv om de ikke tar programmeringsfagene ved NTNU.

Oppgaven til studentene som utviklet den eksisterende applikasjonen var å lage en webapplikasjon som tilbyr denne muligheten. Men som nevnt i forordet svarte ikke resultatet av den oppgaven til de krav som oppdragsgiver hadde stilt. Vår oppgave er derfor å videreutvikle applikasjonen slik at den bedre svarer til kravene.

### 1.2.2 Formulering av problemstillingen

Problemstillingen til den eksisterende applikasjonen var å ”finne en måte å søke i et stort antall undervisningsopplegg”. Vi ønsker å nærme oss problemet fra et litt annet utgangspunkt og fokusere på brukervennligheten til og brukeropplevelsen av en slik applikasjon. Vår grunnleggende tanke er at en applikasjon kun har den verdien som brukerne gir den. Dersom en applikasjon, uansett hvor godt den er utviklet, ikke fyller en funksjon eller på en tilfredsstillende måte løser et problem

---

som brukeren har, vil den ikke bli brukt. Det samme gjelder dersom applikasjonen er vanskelig å bruke, lite intuitiv eller tar for lang tid å bruke. Ettersom lærere i den norske skolen mangler tid og kanskje også motivasjon til å lære programmering, ønsker vi å sørge for at det produkt vi utvikler er så brukervennlig som mulig slik at terskelen for å bruke produktet er så lav som mulig. Denne tanken har vært vårt utgangspunkt i formuleringen av problemstillingen. Vi ønsker å undersøke hva som skal til for at denne applikasjonen faktisk ville bli brukt, slik at et godt grunnlag blir lagt for en senere videreutvikling og utvidelse av applikasjonen.

Formuleringen av vår problemstilling er som følger:

**Kan man lage en webapplikasjon som lærere i den norske grunnskolen *ønsker* å bruke når de skal søke etter og dele undervisningsopplegg for programmering i grunnskolen?**

(i) Hvilken funksjonalitet må være med for at brukeropplevelsen skal være høy nok til at lærerne opplever bruken av webapplikasjonen som *verdt det*?

### 1.3 Rapportens struktur

Rapporten starter med en teoridel som blir lagt som vitenskapelig grunnlag for arbeidet med prosjektet. Deretter følger en gjennomgang av teknologi- og metodevalg. Etter dette blir resultatene av prosjektet presentert. Vi deler her inn i vitenskapelige, ingeniørfaglige og administrative resultater. Deretter følger en diskusjon som analyserer resultatene. Til slutt kommer en konklusjon av rapporten og en redegjørelse for videre arbeid som kan brukes som grunnlag ved en senere videreutvikling av applikasjonen.

---

## 2 Teori

I denne delen av hovedrapporten kommer vi til å gå gjennom de relevante vitenskapelige teoriene rundt prosjektet og produktet som utvikles. Hensikten er å muliggjøre en dypere forståelse av problemstillingen og produktutviklingsprosessen. Vi ønsker å legge et godt grunnlag for gjennomgangen av resultatet av utviklingsprosessen og diskusjonen som følger.

### 2.1 Brukervennlighet

Brukervennlighet, også kjent som brukskvalitet, defineres av hvordan applikasjonen oppfattes av brukeren. Vi ser ofte at begrepene brukervennlighet/brukskvalitet og brukeropplevelse brukes synonymt. I denne rapporten skiller vi mellom disse to, ved å definere brukervennlighet som et begrep som favner større enn det brukeropplevelse gjør. Vi vil i det følgende forklare brukervennlighet, for så å gå over til å beskrive brukergrensesnitt og brukeropplevelse.

Vi nevnte tidligere et sitat fra Are Halland som poengterer at et system som er brukervennlig er (Ryvarden 2022):

- Lett å lære.
- Effektivt.
- Lett å huske.
- Skal håndtere feil på en god måte.
- Skal være behagelig å bruke.

Et produkt bør altså ikke hemme brukeren, men bør heller være til hjelp for brukeren slik at denne på en så smidig måte som mulig kan utføre oppgavene sine.

Halland beskriver videre noen aspekter som bør være tilstede ved utviklingen av et produkt for å sikre god brukskvalitet. Produktet bør, i tillegg til å være brukervennlig, være nyttig og verdifullt for brukeren. Dersom ingen har bruk for et produkt, så har det ingen verdi og ingen nytte. Dette samsvarer med vår grunnleggende tanke om at et produkt kun har den verdi som brukeren gir den. Produktet bør i tillegg være tiltalende, lett å forstå og håndtere, samt tilgjengelig. Designvalg som er tiltalende for brukeren vil gi produktet en større sjanse for å bli tatt i bruk. Det er også viktig at brukeren kan forstå produktet og enkelt finne det man trenger når man trenger det. Tilgjengelighet sikrer at også personer med nedsatt funksjonsevne kan nyttiggjøre seg produktet. Til sist bør produktet også være troverdig.

Basert på denne definisjonen kan vi forstå at brukervennlighet eller brukskvalitet til en stor del er en subjektiv opplevelse av et system. Kravene for hva som bør inngå for at et system skal kunne kalles brukervennlig kan sies å være objektive, men hvordan en bruker oppfatter et system er subjektivt for nettopp den brukeren eller den brukergruppen. Det er derfor ikke alltid enkelt å avgjøre hvorvidt et system er brukervennlig eller ikke. Man bør derfor benytte seg av brukertesting for å finne ut om brukerne opplever produktet som brukervennlig eller ikke.

#### 2.1.1 Brukergrensesnitt - UI

Brukergrensesnitt (User interface: UI) er enkelt forklart det visuelle designet til en applikasjon. Det er dette brukerne ser når de samhandler med produktet. Hartson og Pyla beskriver det som ”en av portalene gjennom hvilken brukerne interagerer med produktet” (sitat oversatt fra engelsk, fra Hartson and Pyla 2018).



---

### 2.1.2 Brukeropplevelse - UX

Brukeropplevelse (User experience: UX) kan på sin side beskrives som interaksjonsdesignet til produktet (Hartson and Pyla 2018). Mens brukergrensesnittet utelukkende gjelder den visuelle fremtoningen til produktet, handler brukeropplevelse om produktets funksjonelle egenskaper og hvordan brukerne opplever bruken av produktet. Brukeropplevelse omfatter aspekter som brukers interaksjon med produktet, brukers totale opplevelse av produktet, hvordan brukeren opplever produktet på et indre plan; samt konteksten rundt bruken av produktet.

### 2.1.3 Brukertesting

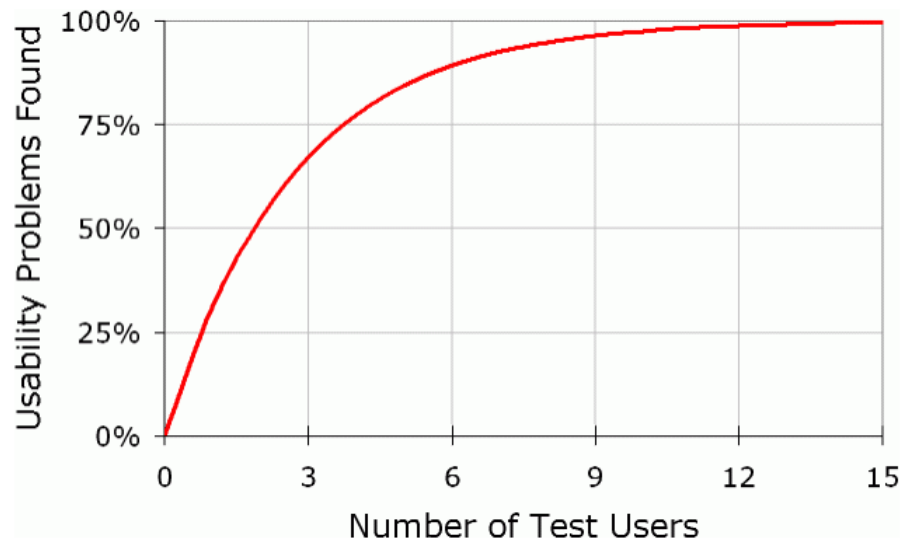
Brukertesting er, som nevnt, et viktig verktøy for å finne ut om et produkt er brukervennlig eller ikke. Det er flere måter å utføre brukertesting på, men felles for dem er at de fokuserer på målgruppen for produktet. Ved å velge ut et antall testpersoner som kan representere resten av målgruppen, kan man få målbare testdata som svarer på spørsmålet om brukskvalitet. Ved å benytte forhåndsbestemte krav for når produktet oppleves som brukervennlig nok kan man på bakgrunn av brukertesting finne ut om produktet når målene når det gjelder brukervennlighet. Statistisk sentralbyrå poengterer i sin håndbok om praktisk brukertesting at ”overordnede krav til brukervennlighet bør defineres allerede i kravspesifikasjonen” (SSB 2022).

Ved brukertesting lar man et antall testpersoner utføre et bestemt sett oppgaver ved hjelp av et bestemt produkt. Hensikten er å finne ut av hvor godt testpersonene kan utføre oppgavene med hjelp av produktet. Ofte kan det være hensiktsmessig å utføre testing ved flere ulike anledninger i utviklingsprosessen. Dette er særlig nyttig ved smidige utviklingsprosesser hvor produktet ofte justeres underveis i prosessen, samt ved store utviklingsprosesser. Brukervennlighetstesting faller inn under ”brukersentrert produktutvikling” (SSB 2022).

SSB beskriver prosessen for brukertesting på følgende måte: Brukertesting av et system starter oftest med at det blir gjennomført en intern testing. Dette kan enten utføres av utviklerne av produktet, eller andre personer som inngår i teamet eller de som senere skal utføre brukertesting. Her går man gjennom eventuelle kravspesifikasjoner og sjekker om produktet møter de krav som på forhånd er stilt. Dette kan være både produktspesifikke krav, men også universelle krav som generelle retningslinjer for nettsider eller universell utforming. Man setter ofte opp en testplan som beskriver testteam, testpersoner, oppgaver som skal utføres i testen og annen informasjon rundt brukertesting. Deretter rekrutterer man testpersoner og gjennomfører brukertesting. Man henter ofte inn informasjon i form av video- og/eller lydfiler, notater fra testteamet eller spørreskjemaer som blir utfylt av testpersonene etter gjennomføringen av brukertesting. Deretter skal denne informasjonen og andre funn fra brukertesting analyseres. Basert på denne analysen kan man gjøre justeringer på produktet eller komme med forslag på forbedringer.

Hvor mange testpersoner som bør rekrutteres for brukertesting avhenger av hvilken type produkt som testes og hvilken del av produktet som skal testes. Dersom målet er å sjekke hvor godt produktet tåler belastning over tid eller stresstesting over et kortere tidsrom, kan det være hensiktsmessig å rekruttere en stor gruppe testpersoner. Hvis man ønsker å teste navigering av en nettside, kan det hende at det holder med en representativ men liten gruppe testpersoner.

Ofte er det en balansegang når det kommer til å ta avgjørelser om hvor store ressurser som skal brukes for brukertesting, kontra hvor mye testdata man kan få ut av testene. Jakob Nielsen (Nielsen 2022) argumenterer for at det ofte holder med opp til fem testpersoner. Han påstår at mindre tester med få personer, men som utføres oftere i løpet av utviklingsprosessen, er mer effektivt enn å utføre en stor test helt på slutten av prosjektet. Han har fremstilt en graf hvor kurven viser at innsikten man kan få fra brukertesting varierer basert på hvor mange testbrukere man tar med (se figur 1). Ved null testbrukere er innsikten også null. Men allerede ved å ta inn én testperson kan man få omtrent en tredel av testdataen man trenger. Ved tre testpersoner får man omtrent to tredeler innsikt. Nielsen viser på at kurven rettes ut ved omtrent fem testpersoner, og at innsikten ikke øker like raskt etter det. Innsikten man får ved seks testpersoner og ni testpersoner er omtrent på samme nivå.



Figur 1: Kurve som viser at innsikt fra brukertesting varierer basert på antall testbrukere. Figur hentet fra Nielsen 2022.

## 2.2 Universell utforming

I tillegg til de subjektive brukeropplevelsene til et produkt, stilles det også noen spesifikke funksjonskrav til produktet. Hensikten med disse kravene er å sikre at de fleste personer, uavhengig av funksjonsevne, skal kunne bruke produktet. Disse kravene går under termen universell utforming. Fra 1. januar 2021 er det et krav at alle IKT-systemer i Norge skal være universelt utformet (Digitaliseringsdirektoratet 2022). For nettsider og webapplikasjoner gjelder mer spesifikt ”Retningslinjer for tilgjengelig webinnhold” eller Web Content Accessibility Guidelines (WCAG), som baserer seg på fire hovedprinsipper (uutilsynet 2022):

1. Produktet må være mulig å oppfatte. Dette gjelder uavhengig av hvordan brukeren får med seg innhold, vare seg det er med syn, lyd eller berøring.
2. Produktet må være mulig å betjene. Det må være mulig for brukeren å navigere i produktet, enten det er med stemmekommando, med skjermforstørrelser eller uten bruk av mus.
3. Informasjonen i produktet må være forståelig. Det må være mulig for brukeren å forstå hvordan produktet skal brukes, og instruksjonene må bli presentert på en slik måte at brukeren kan ta til seg informasjonen.
4. Produktet må være robust. Det bør fungere med de fleste nettlesere, operativsystem og de vanligste hjelpemidlene på markedet. Utviklerne av produktet bør også ta høyde for fremtidig utvikling.

I praksis innebærer universell utforming at utviklere må tenke over fargebruk når det gjelder kontraster og fargekombinasjoner. Det bør også være et hierarki når det gjelder bruk av overskrifter i applikasjonen. Overskrifter må ha riktig størrelse i forhold til hverandre, f.eks. slik at overskrifter på samme nivå må ha samme størrelse, for at skjermlesere skal kunne tolke innholdet på siden på riktig måte. Det bør også være lagt inn alternativ tekst til alle bilder og figurer slik at brukere som ikke kan se, likevel kan høre skjermleseren beskrive hva bildet inneholder.

## 2.3 Programmering i skolen

Som nevnt i introduksjonen ble det i 2020 tatt i bruk nye læreplaner i den norske skolen. Som et resultat av de nye læreplanene ble programmering tatt inn som tema i læreplanene. Per dags dato

---

er det matematikkfaget dette gjelder, men det vil etter hvert utvides til flere fag (UDIR 2022a). Dette kan sees i kompetansemålene for 5. klasse matematikk. Elevene skal etter 5. trinn kunne ”lage og programmere algoritmer med bruk av variabler, vilkår og løkker” (UDIR 2022b).

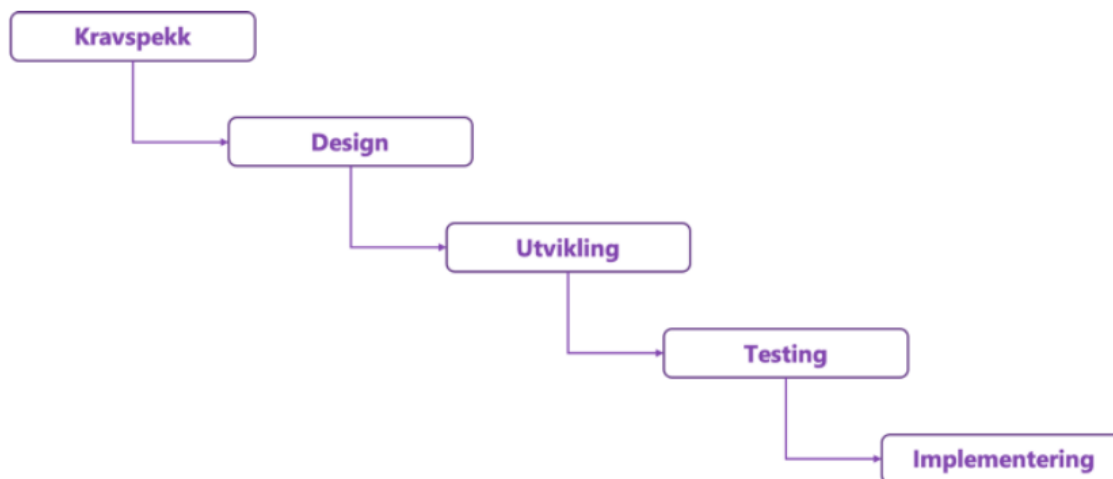
Beslutningen om å ta inn programmering i grunnskolen kom som en følge av en diskusjon som har pågått de siste årene. Det har vært mange som har ment at det å lære programmering også innebærer en økt evne til kreativitet og logisk tenkning (Husabø 2022). Men noen har også argumentert for at programmeringsferdigheter fører med seg lite annet enn ferdigheter i programmering og ikke bør tas inn i eksisterende fag hvor det skyver andre viktige temaer til side (Dahl et al. 2022).

EU lanserte i 2010 en ”Digital agenda for Europa” (Europaparlamentet 2022). Der ble det poengtert at behovet for en befolkning med grunnleggende kunnskaper i IKT blir alt større i årene som kommer. Digitalisering av samfunnet regnes som en av de største årsakene til at behovet for IKT-kompetanse øker. Et av målene i den digitale agendaen er at 80 prosent av den europeiske befolkningen bør ha grunnleggende digitale kunnskaper innen år 2030. For å nå dette målet anbefales det å implementere IKT i skolen. Den digitale agendaen ble i år 2021 utvidet med en tiårsplan, en ”Digital kompass”, som enda tydeligere vektlegger behovet for digital kompetanse i befolkningen. Blant de nordiske landene har, i tillegg til Norge, Sverige og Finland lagt inn programmering i grunnskolen (Bocconi et al. 2018).

## 2.4 Systemutvikling

### 2.4.1 Smidig utvikling

Metoden for smidig utvikling ble utviklet som et alternativ til den tradisjonelle fossefallsmetoden. Fossefallsmetoden er en lineær utviklingsmetode hvor oppgaver blir utført sekvensielt og man ikke går videre til neste fase før man er ferdig med foregående fase (Ullfoss 2022). Metoden kalles fossefallsmetode eller vannfallsmetode fordi den visuelt ligner mye på et fossefall (se figur). I fossefallsmetoden går man normalt sett ikke tilbake til en fase når den er fullført. Kravspesifikasjon og designvalg skjer oftest sammen med kunden helt i starten av prosjektet, og når det er vedtatt har man kanskje ikke kontakt med kunden før endelig levering av produktet.

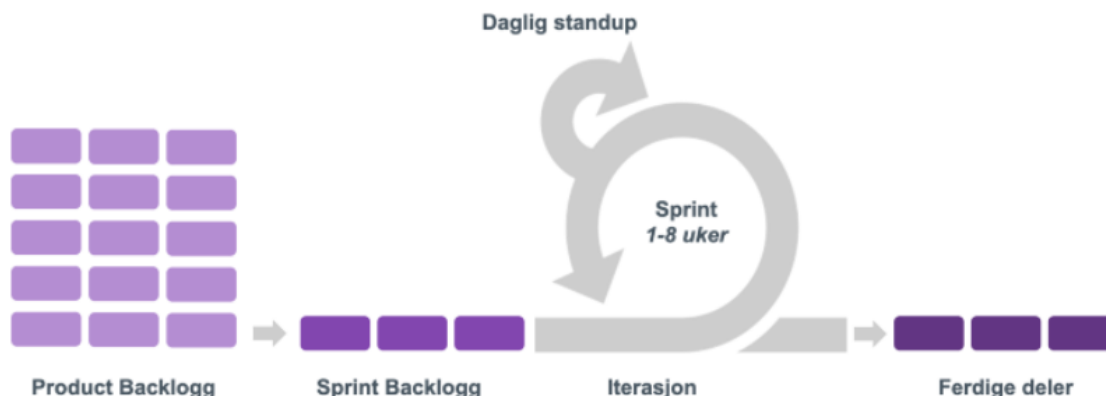


Figur 2: Visualisering av fossefallsmetoden. Bilde hentet fra Ullfoss 2022

Med smidig utviklingsmetode ønsket man å få en mer smidig tilnærming til utviklingens ulike faser. Man så at det noen ganger oppstår behov for å kunne gå tilbake til en tidligere fase og gjøre forandringer, enten basert på problemer som har oppstått i en senere fase og krever endring av design eller kravspesifikasjoner, eller på bakgrunn av tilbakemelding fra kunder. Smidig utvikling er lagt opp som en iterativ metode hvor fokus er å ha små og hyppige del-leveringer underveis i

---

prosessen. Man har også tett samarbeid med kunden gjennom hele prosessen, slik at man kan ta med innspill og tilbakemeldinger fra kunden og justere produktet underveis. Dette gjør at man kan få et produkt som ved endelig levering bedre svarer til kundens behov, ettersom kunden har vært delaktig gjennom hele utviklingsprosessen.



Figur 3: Visualisering av smidig utviklingsmetode. Bilde hentet fra Ullfoss 2022

Når man snakker om smidig utvikling og hyppige del-leveringer er det ofte Scrum (Atlassian 2022d) som arbeidsmetode man refererer til. Scrum er teambasert, og man deler inn arbeidet i korte sprinter hvor et definert sett oppgaver skal utføres. Oppgavene hentes fra en backlog hvor alle oppgaver som skal utføres gjennom hele prosjektet ligger. Man har oftest en daglig standup hvor alle teammedlemmene redegjør for status på oppgavene sine. I slutten av hver sprint har man en gjennomgang av hvilke oppgaver som ble utført i løpet av sprinten, og hvilke oppgaver som eventuelt må videreføres til neste sprint eller gå tilbake til backlogen.

Kanban er en annen arbeidsmetode som eksisterer innenfor smidig utvikling (Atlassian 2022b). Når man bruker Kanban-metoden benytter man seg oftest av en tavle, enten fysisk eller digital, hvor alle arbeidsoppgaver til et prosjekt blir lagt inn som ulike kort. Ved å legge inn arbeidsoppgavene på tavlen får alle teammedlemmene en visuell oversikt over hva som skal gjøres. Oppgavene blir på samme måte som ved Scrum lagt inn i en backlog. Forskjellen her er at oppgavene ikke blir fordelt i sprinter, men man deler heller inn Kanban-tavlen i ulike deler som forteller noe om status for enhver oppgave. En vanlig inndeling er "Todo", "Doing" og "Done". Teammedlemmene velger oppgavekort fra backlogen, og flytter så kortet over til "Doing" og etter hvert til "Done". På den måten kan alle andre i temaet følge med på utviklingsprosessen. Ideen med Kanban er at hvert teammedlem ikke skal ha for mange oppgaver på "Doing" til enhver tid, slik at man raskt kan gjøre ferdig oppgavene man har valgt.

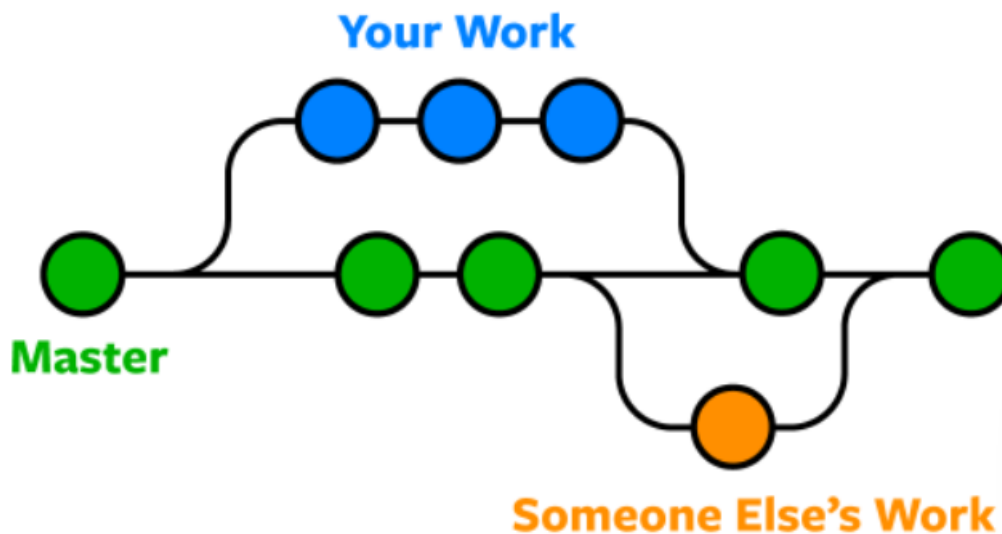
Agile er et begrep man ofte hører nevnt når man snakker om smidig utvikling. Begrepet stammer fra 2001 og en gruppe mennesker som samlet seg for å diskutere fremtiden til systemutvikling (Atlassian 2022a). Denne gruppen satte sammen et "Agile manifesto" som besto av 12 prinsipper. Disse prinsippene nevner blant annet "Mennesker og interaksjoner før prosesser og verktøy", "kundesamarbeid før kontraktssamtaler" og "Respons på forandring før etterlevelse av planer". Poenget med Agile manifesto var å legge vekt på at smidig utvikling handler om å raskt kunne respondere på behov for forandringer, å fokusere på menneskelig samhandling fremfor å strikt følge en fastsatt plan og å ikke være for opptatt av dokumentasjon men heller fokusere på å levere fungerende programvare. Gruppen mente at fremtiden for systemutvikling er avhengig av at utviklingen er smidig.

#### 2.4.2 Versjonskontroll

Versjonskontroll er et viktig element ved systemutvikling. Dette gjelder særlig hvis det er flere personer som utvikler sammen. Versjonskontroll er, enkelt forklart, en måte å kunne spore endringer i kildekode. Git er den vanligste metoden for versjonskontroll (NobleDesktop 2022). Bildet i figuren

---

under beskriver hvordan versjonskontroll fungerer.



Figur 4: Versjonskontroll. Bilde hentet fra NobleDesktop 2022

Når man arbeidet sammen på et utviklingsprosjekt har man ofte en hovedversjon av kildekoden, her kalt master. Den ligger oftest på en skybasert lagringsplass for kildekode som for eksempel GitHub. Når utviklerne skal legge til ny kode lagrer de en lokal kopi av hovedkoden, en lokal branch. Etter arbeidet i den lokale branchen må denne innlemmes i master-koden igjen. Versjonskontroll sporer alle endringer i både master og i de lokale branchene. Hvis det er noe i koden som går galt etter en oppdatering kan man enkelt rulle tilbake til en tidligere versjon av koden. Versjonskontroll sikrer også at alle kan få en oppdatert sisteversjon av koden, slik at det er enklere å flette ulike lokale brancher sammen med hovedkoden igjen.

## 2.5 Teknologi

### 2.5.1 Relasjonsdatabaser

Relasjonsdatabaser er databaser som bygger på relasjonsmodellen. Ganske enkelt så består de av tabeller, også kalt relasjoner, som kan henvise til hverandre ved hjelp av nøkler (IBM 2022). Hver rad i en tabell tilsvare ett innslag av data og tabellen kan ha flere kolonner så vi kan ha mye data på hver rad. Vi kan for eksempel ha en tabell med **personer** og kolonnene våre kan være navn, adresse, telefonnummer, og fødselsdato. Da vil hver rad i denne tabellen med personer tilsvare én person.

Hver rad må også ha en unik nøkkel vi kan identifisere den etter. Dette er fordi vi ønsker å alltid ha muligheten til å finne nøyaktig den raden vi er ute etter hvis vi vil modifisere den eller slette den. Dette kan vi oppnå med å ha én kolonne vi alltid vet er unik for hver rad (for personer kunne dette f.eks. vært personnummer), eller vi kan se på flere kolonner samtidig for å oppnå noe som er unikt (fornavn + etternavn + adresse). Problemet med det siste alternativet er at vi aldri kan være 100% sikker på at det vil gi oss noe unikt. På grunn av dette gir man ofte hver tabell en spesiell kolonne man kan kalle *id*. Dette kan være helt ubetydelige tall eller tekst som databasen finner på, men så lenge de er unike så er det nyttig for oss.

Det med nøkler kan virke litt styr, men det gir en utrolig stor gevinst! Nå som vi alltid kan identifisere riktig rad kan diverse rader henvise til andre rader i helt andre tabeller. Da sier man at det er en relasjon mellom disse tabellene, derav ordet relasjonsdatabaser. Da kan vi tenke oss enda en tabell for **hus** som kan ha kolonnene *id*, *adresse*, *farge*, *etasjer*, *soverom*, *bad*. Istedenfor at personer da har en kolonne med adresse, så kan de istedenfor bare henvise til id'en til huset sitt! Hvis vi da skal hente ut informasjon om en person så kan vi samtidig hente ut informasjon

---

om huset deres, og hvis vi vil vite hvem som bor i et gitt hus så trenger vi bare se hvilke personer som henviser til det huset. Men hvordan henter vi egentlig ut denne informasjonen fra databasen?

Her kommer Structured Query Language (SQL) inn! SQL er et språk som kan brukes for å interagere med databasen. Noe som er bra med det er at språket nesten leses som engelsk. Det lar deg be om data fra spesifikke tabeller, sette inn data, modifisere eksisterende rader, og slette data.

### 2.5.2 Webapplikasjon

En webapplikasjon er en modernisering av applikasjoner som lar deg levere fulle applikasjoner til brukere i nettleseren (Wikstrøm 2022). Man slipper altså å laste ned og installere programvaren før man kan benytte den, slik som man måtte før i tiden. Veldig mange av nettsidene vi bruker i dag er webapplikasjoner, f.eks. YouTube, Google Drive, og Twitter er alle webapplikasjoner.

### 2.5.3 Klient-server

Klient-server eller klient-tjener er en IT-arkitektur som karakteriseres ved at en klient forespør tjenester eller ressurser fra en eller flere servere for å gjøre noe videre med ressursene lokalt (Geeks-ForGeeks 2022). Ofte har man også ting man gjerne vil holde privat eller kun gi ut delvis til offentligheten, så man gjør det arbeidet på en tjener og gir kun ut det man vil til klienter som spør. Dette kan for eksempel være API-nøkler eller all brukerdata man har, hvor man kun vil gi hver bruker tilgang til sin egen data.

En gevinst ved klient-server arkitekturen er at man kan fordele arbeid mellom din server og på brukeren sin nettleter (klienten). F.eks. hvis du vil filtrere et utvalg data så trenger du ikke gjøre filtreringen på serveren, du trenger bare hente all relevant data så kan klienten ta seg av filtreringen.

## 2.6 Personvern og informasjonssikkerhet

I 2018 ble det i Europa innført en ny lov om personopplysninger. Denne går under navnet General Data Protection Regulation, GDPR. I Norge ble en ny personvernforordning utarbeidet på bakgrunn av dette (Regjeringen 2022). Hensikten med GDPR er å styrke personvernet i Europa. Dette gjelder særlig med tanke på den alt større mengde digital data som blir generert. Noen av de viktigste kravene GDPR stiller er at forbrukere skal ha kjennskap til hvor, hvordan og hvorfor personlig informasjon om dem blir lagret. Det innebærer at alle som oppbevarer slik data må kunne svare på disse spørsmålene. I tillegg må de kunne vise fram en gyldig grunn til hvorfor informasjonen blir lagret. I det øyeblikk informasjonen ikke lenger trengs, skal den slettes. Forbrukere skal gi sitt samtykke før informasjonen skal lagres, og de har "retten til å bli glemt". Dette betyr at de har rett til å få all digital informasjon om dem selv slettet dersom de spør om det.

Personvern handler altså om hvordan personlig informasjon om personer blir vernet fra å misbrukes. Slik informasjon kan misbrukes på mange ulike måter. En aktuell problemstilling er i dag identitetstyveri. ID-tyveri defineres som det som skjer når noen "skaffer seg, besitter, overfører, benytter eller fremstår som rette innehaver av et identifikasjonsbevis eller personopplysningene til en person for å begå økonomisk svindel, bedrageri eller annen kriminalitet" (Datatilsynet 2022). Dette kan anses å være en av de mest ødeleggende formene for misbruk av personlig informasjon. Men det er også andre og mindre destruktive måter som informasjon kan misbrukes. Man kan kanskje gå så langt som å si at å dele en kundes kontaktinformasjon med en ekstern tredjepart, uten at kunden har gitt tillatelse til det, kan anses å være misbruk. Det er ikke sikkert at den eksterne tredjeparten har planlagt å bruke denne informasjonen til noe ødeleggende, men det er likevel et tillitsbrudd ovenfor kunden. Det er derfor viktig å ha gode personvernsrutiner på plass for å sikre at slike ting ikke skjer.

Når man skal vurdere risikoer og konsekvenser av feilhåndtering av personlig informasjon i arbeidet med informasjonssikkerhet, kan man bruke en risikomatrix. Risikomatriksen består av en tabell

som sier noe om sannsynlighet for at en hendelse kan skje, og konsekvensen som oppstår dersom hendelsen skjer. I HMS-portalen til Universitetet i Bergen kan vi finne et godt eksempel på hvordan en risikomatrix kan settes opp (UIB 2022):

		Konsekvens				
		A Ufarlig	B En viss fare	C Farlig	D Kritisk	E Svært kritisk
Sannsynlighet	5 Svært sannsynlig					
	4 Meget sannsynlig					
	3 Sannsynlig					
	2 Mindre sannsynlig					
	1 Lite sannsynlig					

Figur 5: Eksempel på risikomatrix. Bilde hentet fra UIB 2022

I figuren ser vi en todimensjonal tabell med "Sannsynlighet" på en akse og "Konsekvens" på den andre aksene. Skalaen på sannsynlighetsaksen går fra "lite sannsynlig" til "svært sannsynlig". Skalaen på konsekvensaksen strekker seg fra "ufarlig" til "svært kritisk". Ved å lage en slik matrix og plassere inn potensielle hendelser i tabellen, kan man finne ut hvor alvorlig risiko den aktuelle hendelsen har. Ulike hendelser vil ha ulik stor sannsynlighet for å skje, og ulik stor konsekvens dersom den skulle skje. På denne måten kan vi enkelt sette opp en risikoanalyse, noe som er svært nyttig i arbeid med informasjonssikkerhet.

---

## 3 Valg av teknologi og metode

### 3.1 Teknologi

#### 3.1.1 React

Den eksisterende frontend-applikasjonen er utviklet i React.js. Gruppen så det derfor som logisk å fortsette i samme bane. React er et JavaScript rammeverk for å bygge opp brukergrensesnitt gjennom å bruke UI komponenter (Meta Platforms 2022). Man definerer altså mindre komponenter og snekrer dem sammen til hele sider.

#### 3.1.2 Typescript

For å få statisk typesjekkning valgte vi å benytte oss av TypeScript til dette prosjektet. Da får vi muligheten til å se hvilke typer argumenter funksjoner forventer å få inn, og vi kan definere våre egne typer og bruke disse i hele applikasjonen. Da blir ofte intensjonene bak en funksjon eller en kodesnutt mye klarere for de som kommer etter deg som skriver det (Microsoft 2022b).

#### 3.1.3 React Bootstrap

React Bootstrap ble brukt som rammeverk. React Bootstrap er et CSS rammeverk for å lage responsiv layout (Kopecky 2022) på nettsider uten å måtte bruke mye tid på å kode CSS selv. React Bootstrap er en parallell versjon Bootstrap som er skrevet om for å benytte mange features som React stiller med. Forrige gruppe hadde allerede tatt dette i bruk og vi så ingen grunn til å vike fra dette da alt som sparer tid i et slikt prosjekt som dette er velkomment.

#### 3.1.4 MySQL

MySQL er et databaseadministrasjonssystem som kan kjøre på de aller fleste vanlige operativsystem (Oracle 2022). Vi valgte å bruke dette ettersom NTNU har tilgjengelige servere for studenter hvor de kan sette opp egne databaseinstanser. Det å sette opp ting der er en prosess vi har vært gjennom mange ganger før, og med tanke på tid og at begge utviklere da kan dele på samme instans var det bare gode incentiver til å gå videre med dette slik som forrige gruppe.

#### 3.1.5 Sequelize

Sequelize er et Typescript og Node.js ORM-verktøy for SQL-baserte databaseadministrasjonssystem (Sequelize-Contributors. 2022). ORM eller Object-Relational Mapping er en teknikk for å konvertere data fra relasjonelle databaser til noe som kan brukes i objekt-orienterte programmeringsspråk. I vårt tilfelle kunne vi bruke dette for å automatisere oppsett av en helt ny database og gjøre dataen vår lettere å jobbe med i TypeScript. Altså kan man nå gi litt informasjon om hvordan man kan koble seg til databasen man ønsker å bruke, så vil Sequelize koble seg til og opprette alle tabellene man trenger og ha mulighet til å sette inn mock-data.

#### 3.1.6 Visual Studio Code

Visual Studio Code (VS Code) er en kildekode-editor laget av Microsoft med mange hjelpeverktøy for utviklere (Microsoft 2022a). Akkurat hvilken editor hver enkelt av oss brukte har ikke så stor innvirkning på resultatet så det er lov å velge og vrake, men vi begge endte opp med å bruke VS Code.



---

### 3.1.7 GitHub

GitHub er en plattform som tilbyr distribuert versjonskontroll basert på *git* og hosting av kode (GitHub 2022). Man kan lage *repositories*, eller oppbevaringssteder for kildekode hvor samarbeidspartnere på prosjektet kan sende sin endringer til. Endringene blir da lagret og alle som har en lokal kopi av prosjektet kan laste ned endringene og få dem gjenspeilet hos seg. Det har selvsagt mange flere funksjoner som kontinuerlig integrasjon og pull requests (PR). Når noen sender en PR til et prosjekt så sender de en samling med endringer de har gjort og spør om du som eier av prosjektet har lyst å ta inn disse endringene. Det var noe vi hadde lyst å bruke aktivt i prosjektet her så GitHub var et naturlig valg. Det finnes selvfølgelig flere alternativer som GitLab, men vi landet hos GitHub pga. personlig preferanse.

### 3.1.8 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X er et typesettingssystem for dokumentproduksjon. Det baserer seg på at forfatterne av et dokument kun skal trenge å skrive og ikke tenke så veldig mye på å få dokumentet til å se fint ut. Det eneste man selv gjør er å dele opp dokumentet i logiske seksjoner også fyller man inn med eget innhold, så tar L<sup>A</sup>T<sub>E</sub>X seg av alt med layout og oppsett (LaTeX-Project 2022). På grunn av dette har ofte dokumenter laget med L<sup>A</sup>T<sub>E</sub>X et karakteristisk utseende og er lette å gjenkjenne.

## 3.2 Metode

### 3.2.1 Administrative verktøy

Arbeidet med dette prosjektet har til sin helhet foregått digitalt. Gruppemedlemmene er lokalisert i forskjellige deler av landet, og det var derfor viktig å finne gode digitale samhandlingsplattformer for å sikre et effektivt og strømlinjeformet samarbeid. Studieprogrammet som denne bacheloroppgaven er en del av er et fullt ut digitalt studieprogram. Vi har derfor fått god øving i digital samhandling helt fra starten av studieprogrammet. Det har vært til stor hjelp i arbeidet med dette prosjektet.

Vi valgte å benytte oss av Google Drive (Disk) for lagring av filer og relevant materiale for prosjektet. Timelister, møtereferater, veiledningsdokumenter, eksempeloppgaver og maler har blitt lagret på Google Drive. Dette har sikret at begge gruppemedlemmene har hatt tilgang til relevant informasjon når det har vært behov. All kildekode har, som nevnt, blitt lagret på GitHub. Dette har sikret en versjonskontroll og at man i etterkant kan følge utviklingsprosessen slik den har foregått i løpet av prosjektet.

For rapportskrivning valgte vi å benytte oss av Overleaf, som er en nettbasert L<sup>A</sup>T<sub>E</sub>X-editor (Overleaf 2022). I Overleaf ligger alle filer lagret online, og ved å dele skrivetilgang til filene har begge gruppemedlemmene hatt lik tilgang til alle rapportene. Overleaf gir også mulighet for samskriving og tekstkommentarer. Dette har vært en stor fordel i arbeidet med rapportskrivning ettersom gruppemedlemmene har kunnet arbeide sammen og samtidig på en rapport, selv om den geografiske avstanden har vært stor.

Discord, som er en online kommunikasjonsplattform i hovedsak rettet mot gaming-fellesskap (Discord 2022), ble benyttet for møter og samtaler mellom gruppemedlemmene. Der hadde vi ukentlige ”voice calls”, og benyttet også chat-funksjonen for å dele ideer og linker og kommunisere med hverandre mellom de ukentlige møtene. Det ble også satt opp en forbindelse med GitHub i Discord, slik at hver gang det ble gjort en oppdatering i GitHub, ble det lagt inn en notifikasjon i Discord. Dette gjorde samarbeidet oversiktlig, og det var enkelt å følge med på når det ble lagt til noe i GitHub som trengte en godkjenning fra gruppemedlemmene.

For møter med veileder benyttet vi oss av videosamtaler via Google Meet. Dette fungerte utmerket ettersom vi enkelt kunne sende ut møteinvitasjon via Gmail med link til møtet i Google Meet. I de tilfeller vi hadde behov for å snakke med oppdragsgiver valgte vi å bruke Zoom.

Vi valgte å bruke Kanban-metoden i dette prosjektet. Gruppestørrelsen er så liten at en Scrum/sprint-

---

metode virket litt for omfattende, og det tavle-baserte oppgavesystemet til Kanban ga mer mening. I stedet for å sette opp delmål og fordele oppgaver ut fra sprinter, valgte vi å legge inn alle oppgaver på en tavle. Derfra kunne hvert gruppemedlem velge oppgaver å utføre. Vi hadde likevel ukentlige møter hvor vi snakket kort om hva vi gjort i løpet av uken, og hva vi planla for den påfølgende uken.

Trello ble brukt som plattform for gruppens Kanban-tavle. Her ble alle arbeidsoppgaver lagt inn som kort, og etterhvert som oppgaver ble utført, flyttet kortene seg fra "Backlog", til "Todo", "Doing" og til sist til "Done". Trello gjorde det mulig å holde oversikt over hvilke oppgaver hver enkelt gruppemedlem utførte til enhver tid, og det var særlig viktig med tanke på å dokumentere fremgangen i prosjektet. TeamGantt ble brukt i starten av prosjektet for å sette opp en prosjektplan i form av et Gantt-diagram.

### **3.2.2 Fordeling av arbeidsoppgaver**

Gruppemedlemmene har ved tidligere anledninger i studiet samarbeidet i gruppeprosjekter. Vi har derfor god erfaring med å jobbe sammen, og kjenner til hver av gruppemedlemmenes styrker. Dette valgte vi å utnytte i arbeidet med bacheloroppgaven. Gruppen valgte å dele hovedansvaret for prosjektet på følgende måte: Truls har hatt hovedansvar for utviklingsprosessen, mens Henna bidrar med utviklingsoppgaver. Henna har hatt hovedansvar for rapporter og dokumentasjon av prosjektet: å klargjøre maler, hente inn materiale og å ferdigstille rapporter i tråd med tidsfristene som gruppen satte internt. Truls bidrar med skriving av innhold i rapportene. Fordelingen av ansvar har vist seg å være meget effektivt. Gruppemedlemmene har kunnet dra nytte av sine styrker, samtidig som begge har bidratt i prosjektet som helhet. Denne fordelingen av arbeidsoppgaver har også sikret en god kvalitet på alle områder i prosjektet.

---

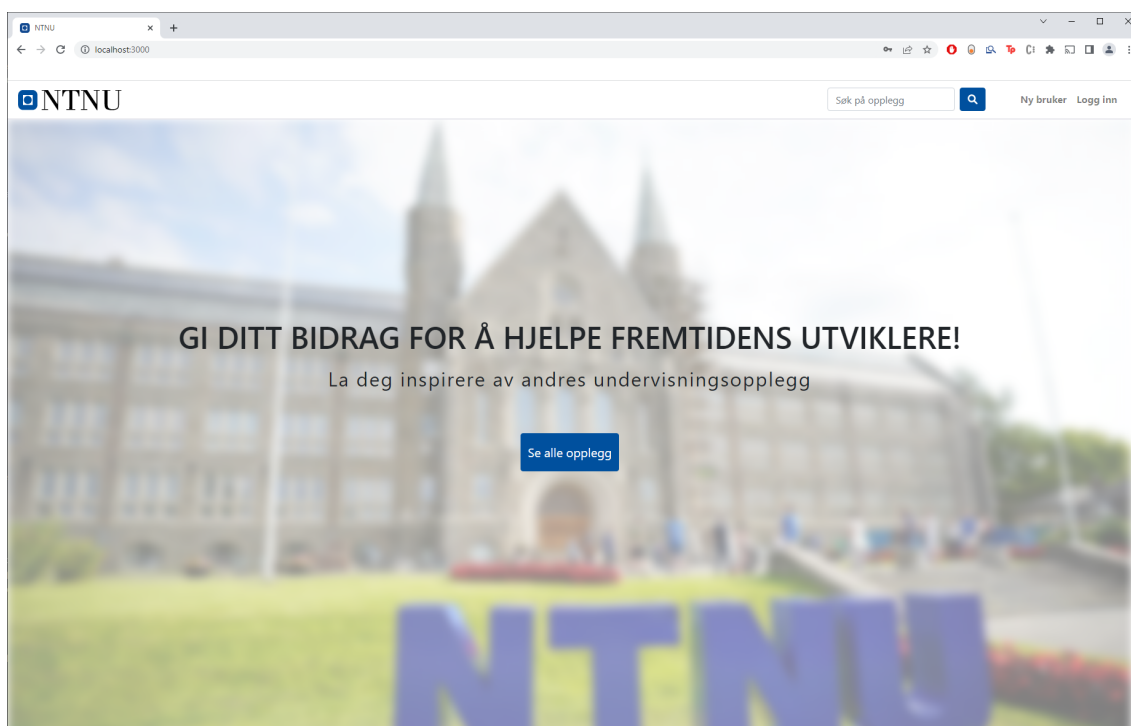
## 4 Resultater

Denne delen av rapporten tar for seg resultatene av arbeidet med produktutviklingsprosjektet. Bakgrunnen for presentasjonen av resultatene er, i tillegg til applikasjonen og kildekoden, visjonsdokumentet, kravdokumentasjonen og systemdokumentasjonen som blir levert som vedlegg i denne rapporten (vedlegg A-C), samt prosjekthåndboken som blir levert som vedlegg i en separat zip-fil. For presentasjon av resultater fra brukertesting blir det henvist til testplanen og responsen på skjemaet fra brukertesting. Disse er vedlagt i denne rapporten som vedlegg D og E.

### 4.1 Vitenskapelige resultater

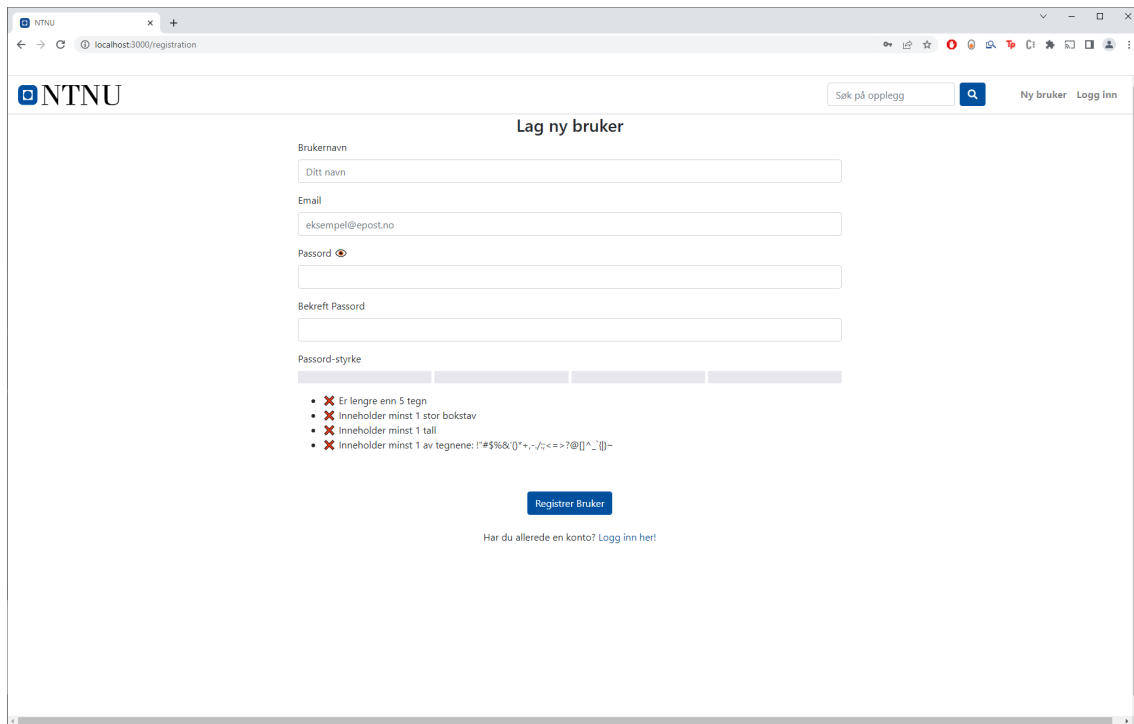
Produktet vi har utviklet er en lett webapplikasjon som lar brukere opprette nye og lese eksisterende artikler. Disse artiklene er selvsagt tiltenkt at de skal være undervisningsopplegg om programmering, men kan i teorien være hva som helst.

Vi har gått for samme fargepalett som på NTNU sin hjemmeside så det første som møter brukeren skal være så kjent som mulig.



Figur 6: Hovedsiden når man ikke er logget inn

Her er hovedsiden når man ikke har logget inn. Man blir invitert til å se på alle undervisningsoppleggene som er tilgjengelig, men man har også muligheten til å søke eller logge seg inn med en bruker.



Figur 7: Registrering av ny bruker


På siden hvor man kan lage seg en ny bruker blir man presentert med et skjema hvor man kan fylle ut informasjon om brukernavn og emailen man vil bruke. Her benytter vi validering som er innebygget i HTML for å sjekke at det f.eks. er en gyldig email som brukeren skriver inn. Vi har også validering på passordet og et par krav til hva det må inneholde.

---

## Lag ny bruker

Brukernavn





Email

Passord 

Bekreft Passord

Passord-styrke



-  Er lengre enn 5 tegn
-  Inneholder minst 1 stor bokstav
-  Inneholder minst 1 tall
-  Inneholder minst 1 av tegnene: !"#\$%&'()\*+,-./:;<=>?@[^\_`{|}~

[Registrer Bruker](#)

Har du allerede en konto? [Logg inn her!](#)

Figur 8: Registrering med dårlig passord

Her kan du se at passordet som brukeren forsøker å sette bare oppfyller kravene om å være lengre enn 5 tegn og det å inneholde minst 1 tall. Om vi prøver å registrere oss nå så vil vi få en feilmelding.

---

## Lag ny bruker


Passordet er for svakt

Brukernavn

NyBruker

Email

ny@bruker.no

Passord 





.....

Bekreft Passord

.....

Passord-styrke



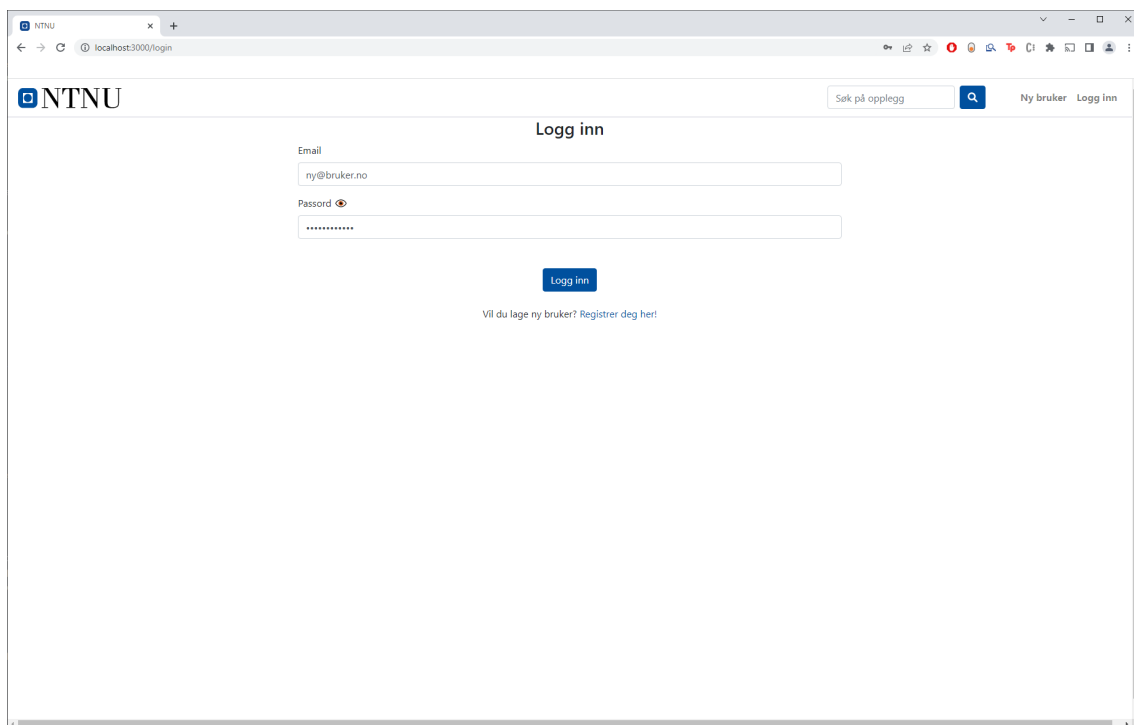
-  Er lengre enn 5 tegn
-  Inneholder minst 1 stor bokstav
-  Inneholder minst 1 tall
-  Inneholder minst 1 av tegnene: !"#\$%&'()\*+,-./:;<=>?@[^\_`{|}~

Registrer Bruker

Har du allerede en konto? [Logg inn her!](#)

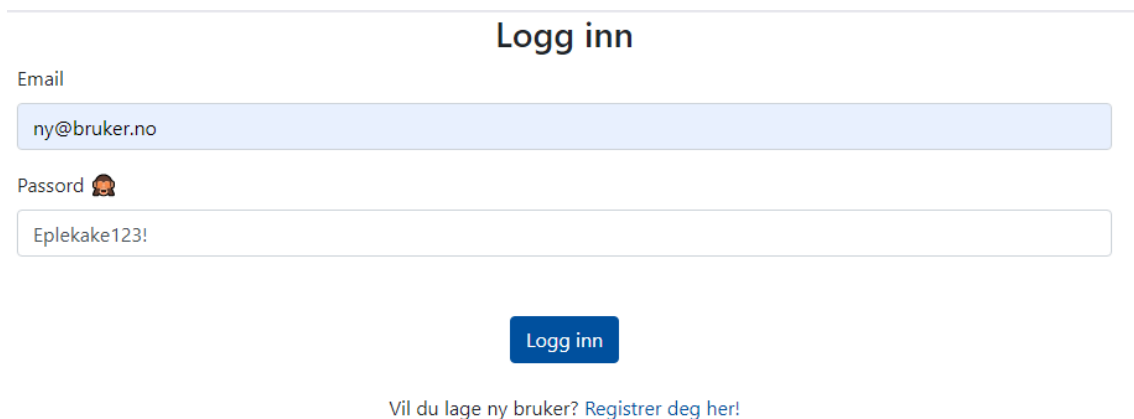
Figur 9: Feilmelding for dårlig passord

Om vi skriver inn et bedre passord som oppfyller alle kravene vil vi få lov å sende inn skjemaet og man blir sendt videre til innloggingssiden.



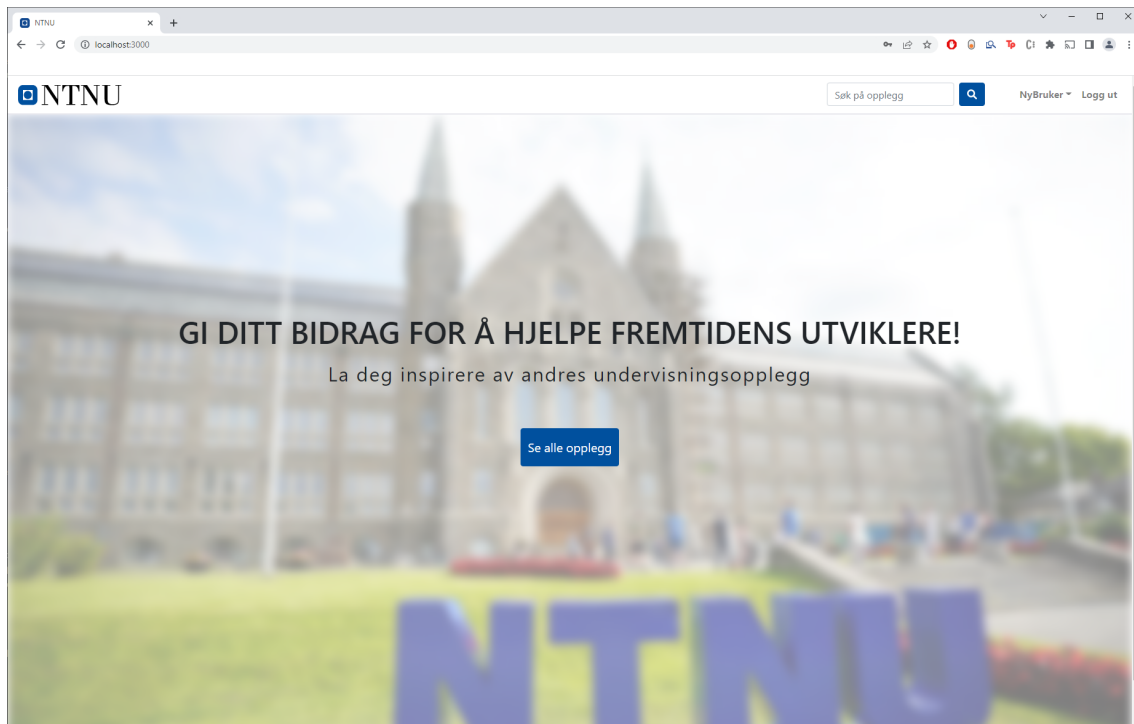
Figur 10: Innloggingssiden

Her på innloggingssiden blir man bedt om å skrive inn email og passord. Dersom man skriver inn feil passord eller feil epost vil man få en nøytral tilbakemelding som sier at minst én av tingene er feil, men bekrefter ikke hvilken av dem som er det av sikkerhetssårsaker.



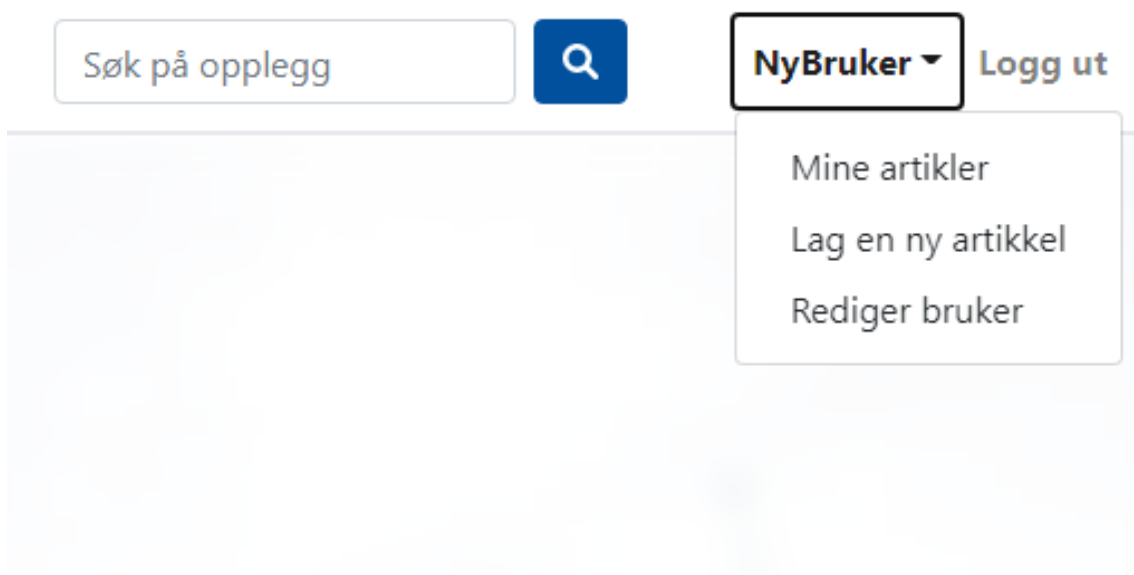
Figur 11: Mulighet for å vise passord

Man har også mulighet for å se passordet man har skrevet inn dersom man er usikker ved å flytte musepekeren over øyet ved siden av *Passord*. Da blir øyet om til en ape og man kan se passordet i klartekst. Når man flytter vekk musepekeren vil passordet igjen bli skjult og øyet kommer tilbake.



Figur 12: Hovedsiden når man er logget inn

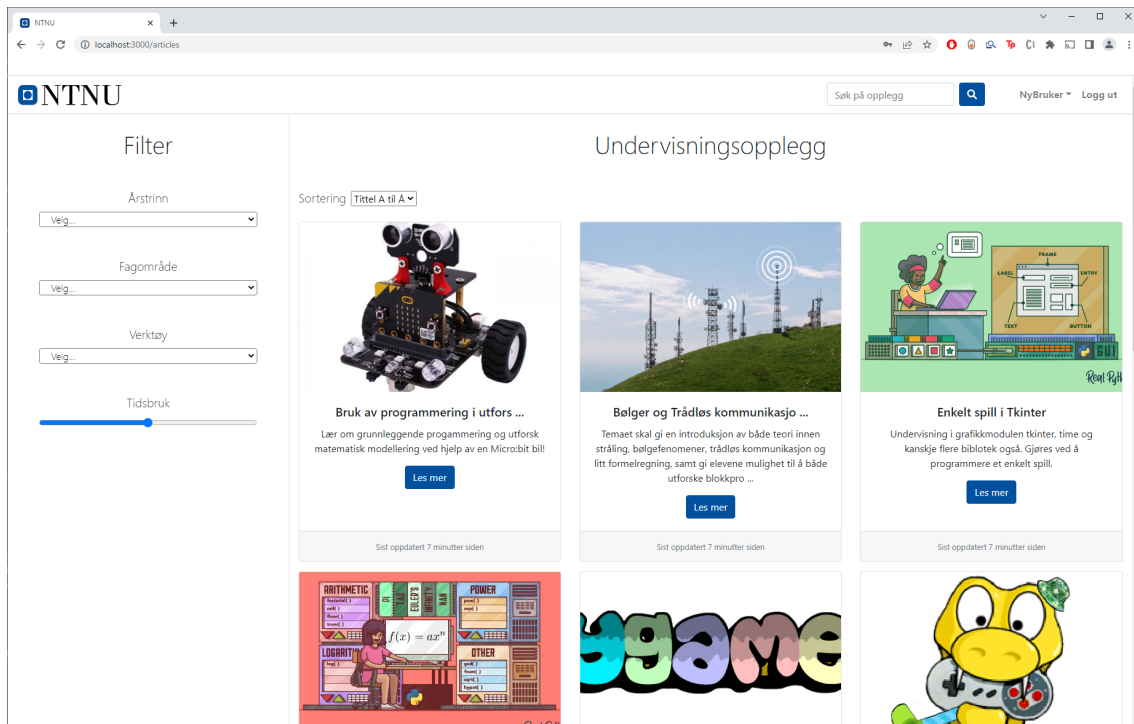
Når man så har logget inn med en bruker kan man se at brukernavnet vises oppe til høyre og man har muligheten til å logge ut igjen.



Figur 13: Valgmeny for innloggede brukere

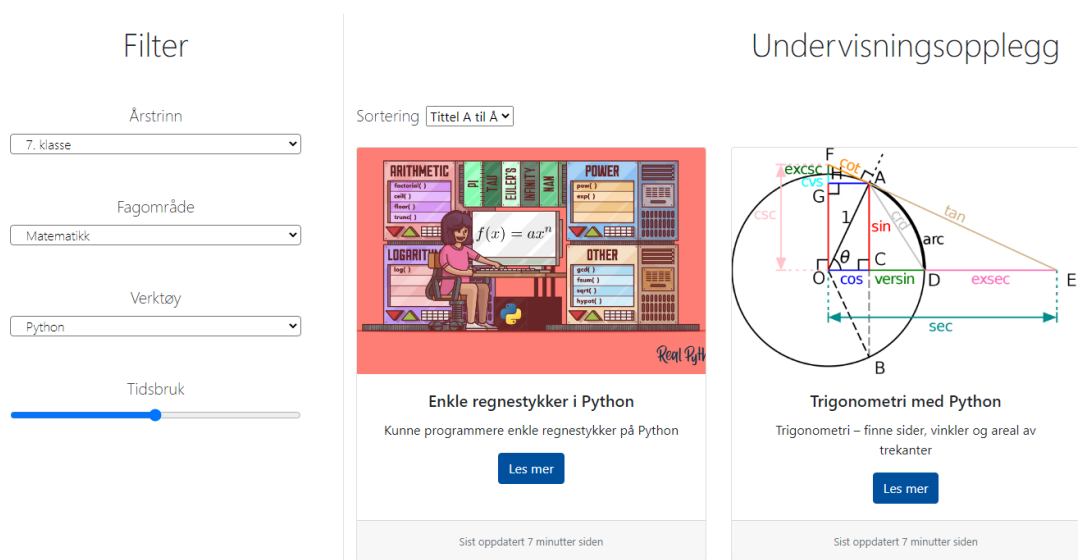
Hvis man klikker på brukernavnet sitt vil man få en dropdown-meny hvor man kan se alle sine egne artikler, lage en ny artikkel, eller redigere brukeren sin.





Figur 14: Oversikt over alle artikler i applikasjonen

I oversikten over alle artikler kan man se hver enkelt artikkel som er lagt ut i applikasjonen. Man har diverse muligheter for sortering, filtering, og søk som man kan se avbildet under.

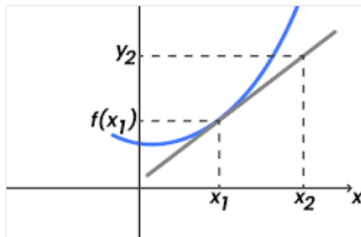


Figur 15: Muligheter for å filtrere artikler på tags

Når man filtrerer på tags så får du kun se artikler som har de relevante tags som er valgt.

## Undervisningsopplegg

Sortering Tittel Å til A

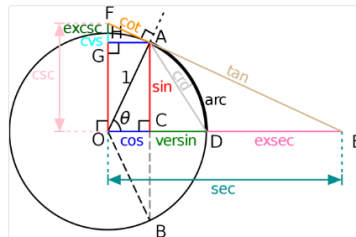


### Utforskning av derivasjon ved n ...

I undervisningsopplegget skal elevene utforske derivasjon ved å bli kjent med ulike metoder for å finne den deriverte. Programmeringsdelen av dette blir å bli introdusert for ...

[Les mer](#)

Sist oppdatert 8 minutter siden



### Trigonometri med Python

Trigonometri – finne sider, vinkler og areal av trekanter

[Les mer](#)

Sist oppdatert 8 minutter siden



### Titring og derivasjon

Elevene skal i kjemi bli kjent med titrering som metode, både manuell og ved hjelp av datalogger. De skal se hvordan vi kan bestemme ekvivalenspunktet ved å bruke den derivert ...

[Les mer](#)

Sist oppdatert 8 minutter siden

Figur 16: Muligheter for å sortere artikler

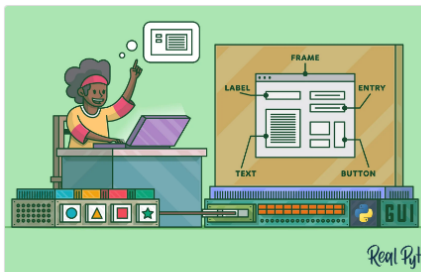
Man kan også sortere på tittel a til å, tittel å til a, og publiseringsdato i stigende eller synkende rekkefølge.

spill



## Undervisningsopplegg

Sortering Tittel Å til A



### Enkelt spill i Tkinter

Undervisning i grafikkmodulen tkinter, time og kanskje flere bibliotek også. Gjøres ved å programmere et enkelt spill.

[Les mer](#)

Sist oppdatert 8 minutter siden

# game

### Lag et eget spill i PyGame!

Eleven får fungerende kode, og blir bedt om å endre på koden i spillet slik at spillet blir "deres eget". Spillet skal ikke ligne grafisk på spillet de fikk av meg.

[Les mer](#)

Sist oppdatert 8 minutter siden



### Lage et spill

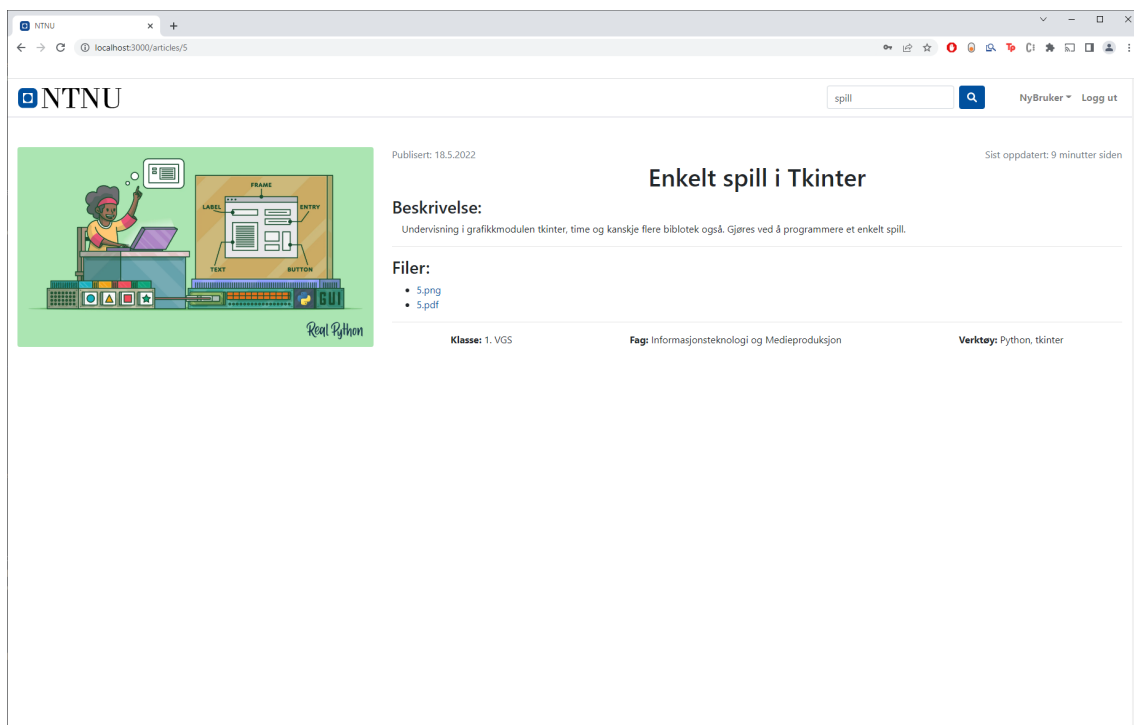
Elevene skal lage et spill der skal kunne bevege seg fram. På veien vil den møte hin kommer im

[Les mer](#)

Sist oppdatert 8 r

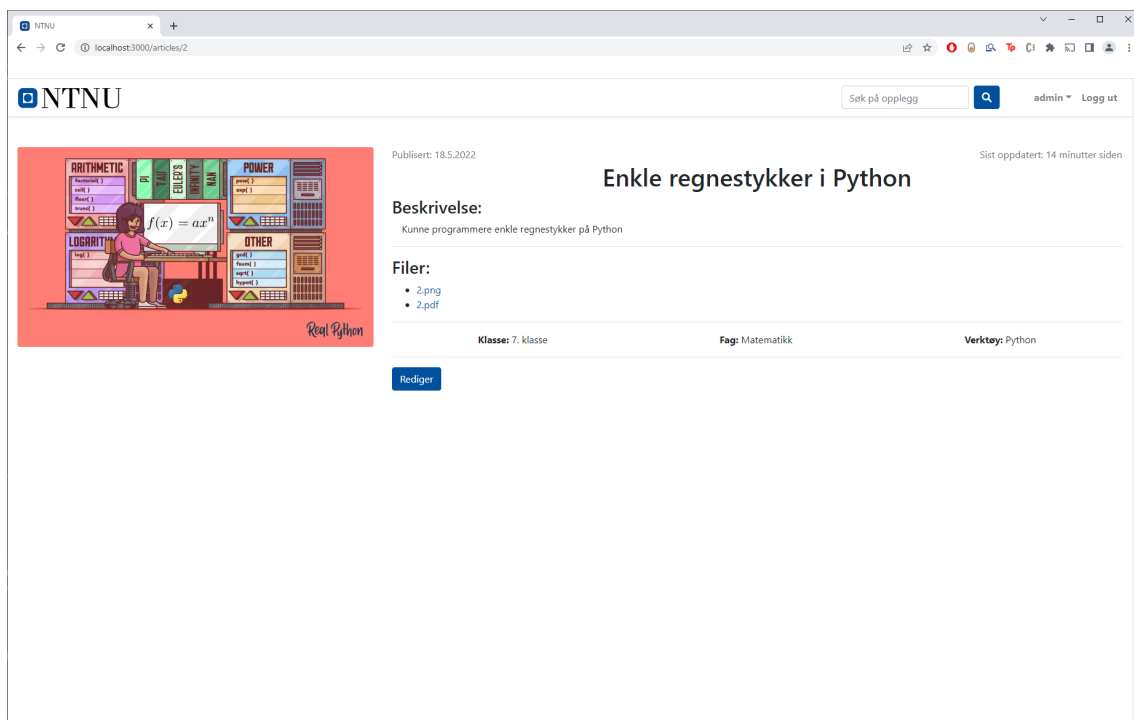
Figur 17: Muligheter for å søke etter artikler

Det man skriver i søkefeltet blir matchet mot artikler sine titler og beskrivelser, og alt som inneholder søketeksten vil bli vist.



Figur 18: Detaljer for en enkelt artikkel

Når man klikker seg inn på en artikkel man er interessert i får man se all informasjonen knyttet til den presentert og man kan laste ned eventuelle vedlegg som er knyttet til artikkelen. Stort sett ligger mesteparten av undervisningsoppleggene som vedlegg foreløpig.



Figur 19: Detaljer for en enkelt artikkel man er skaper av

Dersom man er eier av artikkelen man har klikket seg inn på vil man også få opp en redigeringsknapp.

The screenshot shows a web browser window with the URL localhost:3000/articles/2/edit. The page title is "Rediger undervisningsopplegg". The form contains the following fields and options:

- Tittel:** Enkle regnestykker i Python
- Beskrivelse:** Kunne programmere enkle regnestykker på Python
- Klassestrinn:** Radio buttons for 1. klasse, 2. klasse, 3. klasse, and 4. klasse.
- Fag:** Radio buttons for Matematikk (checked), Matematikk P, Matematikk R, and Matematikk T.
- Verktøy:** Radio buttons for Python (checked), skinter, Javascript, and C#.
- Last opp thumbnail:** Velg fil | 2.png
- Last opp filer:** Velg filer | 2.pdf

Buttons at the bottom: Tilbake, Lagre.

Figur 20: Redigering av en artikkel

Når man trykker på redigeringsknappen vil man få opp et forhåndsutfylt skjema hvor man fritt kan redigere informasjonen til artikkelen.

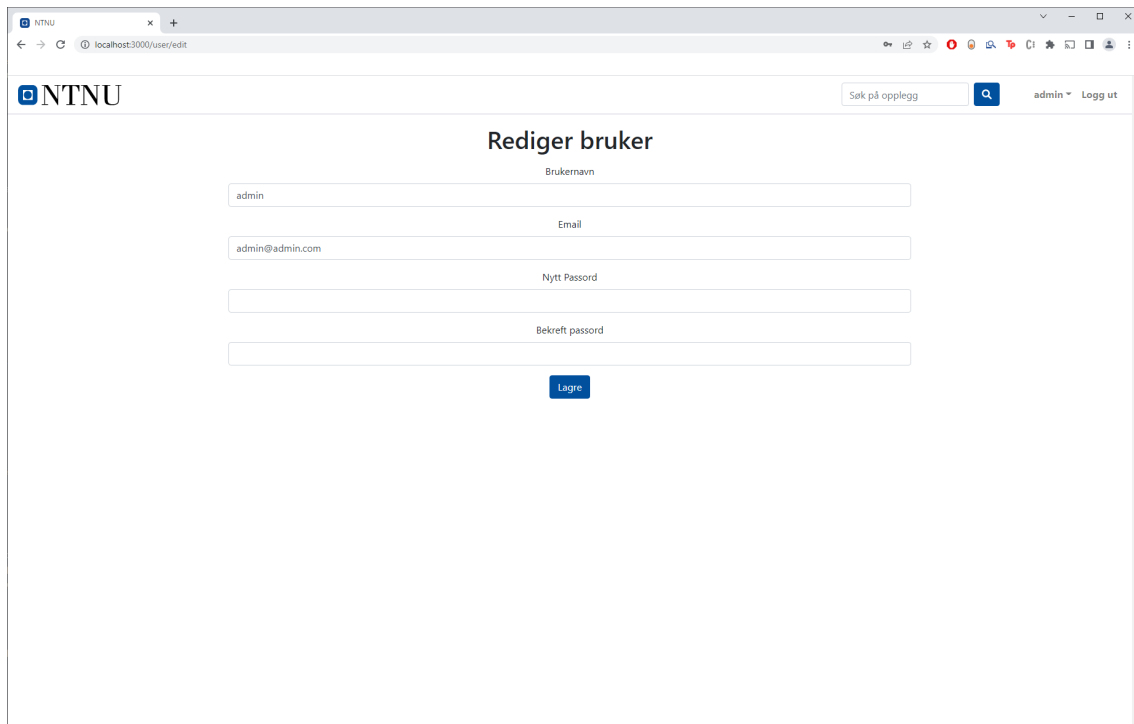
The screenshot shows a web browser window with the URL localhost:3000/articles/new. The page title is "Nytt undervisningsopplegg". The form contains the following fields and options:

- Tittel:** (empty)
- Beskrivelse:** (empty)
- Klassestrinn:** Radio buttons for 1. klasse, 2. klasse, 3. klasse, and 4. klasse.
- Fag:** Radio buttons for Matematikk, Matematikk P, Matematikk R, and Matematikk T.
- Verktøy:** Radio buttons for Python, skinter, Javascript, and C#.
- Last opp thumbnail:** Velg fil | Ingen fil valgt
- Last opp filer:** Velg filer | Ingen fil valgt

Buttons at the bottom: Tilbake, Lagre.

Figur 21: Opprettelse av en ny artikkel

Dette skjemaet er identisk med skjemaet for å lage nye artikler.



Figur 22: Brukerredigering

Til slutt så har man også mulighet til å endre epost som brukeren din er knyttet til, og man kan endre passord og brukernavn.

## 4.2 Ingeniørfaglige resultater

Kravene for applikasjonen ble ved start delt opp i funksjonelle og ikke-funksjonelle krav, disse finner man i kravdokumentasjonen (Vedlegg B). I de følgende tabellene går vi gjennom disse kravene og noterer status for hvert av dem. Hver status vil begynne med ett av tre symboler. ”✓” betyr at kravet ble fullført, ”✗” betyr at kravet ikke ble fullført, og ”⚡” betyr at kravet ble sløffet underveis. Det at et krav blir sløffet betyr ikke nødvendigvis at funksjonaliteten ikke lenger er ønsket, men at det ble valgt å nedprioriteres for å spare tid eller for muligheten å fokusere på andre krav.

#### 4.2.1 Funksjonelle krav

Funksjonelt krav	Status
Opplasting av undervisningsopplegg	✓ - Brukere kan laste opp undervisningsopplegg med vedlegg når de er logget inn med en brukerkonto.
Undervisningsopplegg skal ikke publiseres frem til en administrator har godkjent dem	⊗ - Dette kravet valgte vi å sløyfe for å få nok tid til grunnfunksjonaliteten. Derfor har administratorrolle ingen effekt i dag og alle undervisningsopplegg er synlige umiddelbart.
Brukere skal kunne se sine egne undervisningsopplegg og se om de er publisert eller ikke	✓ og ⊗ - Brukere kan få en oversikt over alle sine egne undervisningsopplegg, men siden alle vil være offentlige kan de ikke se status for publisert/ikke-publisert.
Administrator skal kunne se alle undervisningsopplegg som ikke har blitt offentliggjort ennå i en liste.	⊗ - Administrator-funksjoner ble sløyfet.
Administrator skal fra denne listen kunne godkjenne eller avslå undervisningsopplegg.	⊗ - Administrator-funksjoner ble sløyfet.
Brukere skal kunne se en liste over alle publiserte undervisningsopplegg og søke/filtrere denne listen.	✓ - Brukere kan søke etter undervisningsopplegg, filtrere etter kategorier, og sortere etter tittel og publiseringsdato i stigende og synkende rekkefølge.

#### 4.2.2 Ikke-funksjonelle krav

Ikke-funksjonelt krav	Status
Produktet må ha god søkbarhet. Det skal være enkelt å sortere og filtrere undervisningsopplegg slik at lærerne kan finne et godt egne opplegg å bruke i undervisningen.	✓ - Det er mulig å søke basert på tittel og beskrivelse av undervisningsopplegg. Filtringen gjør det også mulig å finne opplegg til spesifikke trinn, fag, eller som bruker spesifikke verktøy.
Godt dokumentert kildekode med gode rutiner for å unngå teknisk gjeld og bra muligheter for enkel utvidelse.	✓ - Systemdokumentasjonen (Vedlegg C) dokumenterer applikasjonen, og arkitekturen er valgt for å fordele ansvaret for ulike typer operasjoner på ulike typer filer.
REST-standarder følges (navngivning av ruter, ordentlig bruk av verb, osv.)	✓ - Alle ressurser i applikasjonen har en egen rute og HTTP verbene til de ulike ressursene gjør det som forventes.
Applikasjonen skal kreve minimalt med oppsett hos de som ønsker å bruke dem (dvs. ikke stort mer enn å kjøre 'npm install' og 'npm run start')	✓ - Frontend kan installeres og kjøres svært enkelt, og backend krever minimalt med ekstra oppsett, stort sett bare SQL-server som må settes opp.

#### 4.2.3 Brukertesting

Da utviklingen av applikasjonen kom til et punkt hvor det meste av funksjonalitet var på plass og fungerte, satte vi opp brukertesting. Hovedfokuset med brukertesting var å finne ut hvor brukervennlig applikasjonen oppleves av personer i målgruppen. Vi ønsket også å få mer informasjon direkte fra målgruppen om deres behov med hensyn til vår problemstilling. Dette for å kunne se om vårt produkt fyller den funksjon som er tenkt. Vi laget en testplan som inneholdt informasjon om bakgrunn, hensikt og mål med testene, samt oppgaver og spørsmål som testpersonene ble bedt om å utføre og svare på. Oppgavene ble laget basert på kravene som ble stilt til applikasjonen i kravdokumentasjonen. Se vedlegg D for informasjonen og en detaljert oversikt over oppgavene som testbrukerne ble bedt om å utføre, og vedlegg E for en detaljert beskrivelse av responsen fra

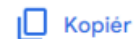
---

spørsmålene som ble stilt.

Vi kontaktet Ryenberget skole i Oslo, hvor tre lærere sa seg villige til å stille opp som testbrukere. Ettersom vår databaseløsning på tidspunktet for brukertesting krevde innlogging på NTNUs nettverk, valgte vi å utføre brukertesting på Ryenberget skole med medbrakt laptop med VPN-tilgang til nettverket. Henna utførte brukertesting alene ettersom Truls var i Trondheim. Spørsmålene ble stilt før og etter utførelsen av oppgavene. Det ble laget et spørreskjema i Google Skjemaer med alle spørsmålene, og testbrukerne fikk tilsendt lenken til spørreskjemaet per epost.

Testpersonene klarte å navigere i applikasjonen og utførte uten hjelp de oppgavene som ble gitt. På spørsmålet om noen av dem hadde brukt en slik applikasjon tidligere var svaret nei. To av testpersonene har erfaring med å undervise i programmering i grunnskolen. Den tredje testbrukeren skal på et tidspunkt undervise i programmering men har ikke gjort det enda. Samtlige testpersoner var positive til å ta i bruk dette produktet dersom de hadde trengt å finne et undervisningsopplegg for programmering.

#### 4. Har du noen gang undervist i programmering?

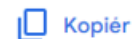


3 svar



Figur 23: Andelen testbrukere som har undervist i programmering. Skjerm bilde fra vedlegg E.

#### 10. Dersom du hadde vært på leting etter et undervisningsopplegg for programmering, ville du brukt dette produktet?



3 svar

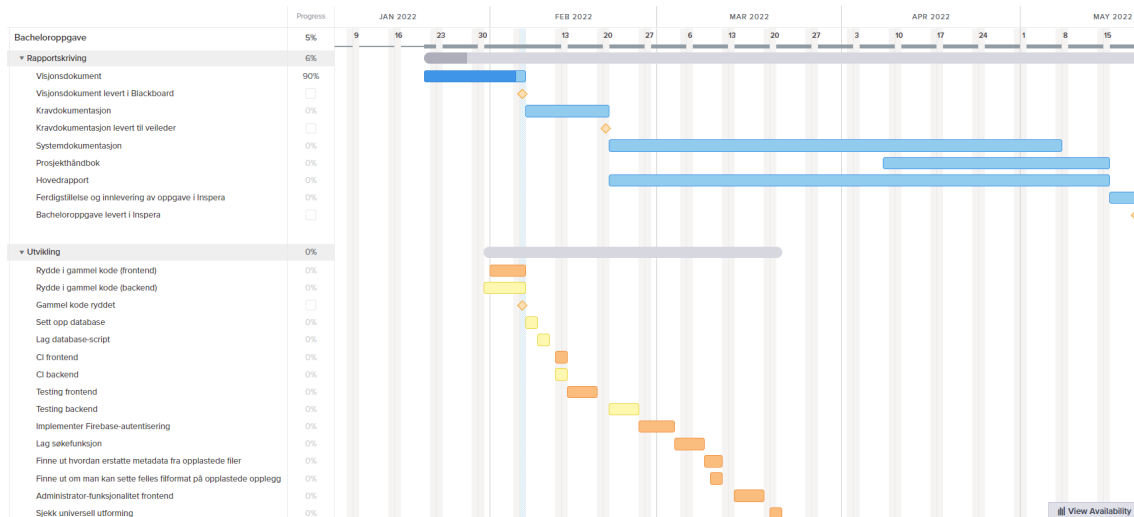


Figur 24: Testbrukerne er positive til applikasjonen. Skjerm bilde fra vedlegg E.

## 4.3 Administrative resultater

### 4.3.1 Prosjektplan

Det ble opprettet et Gantt-diagram som fungerte som en initiell prosjekt- eller fremdriftsplan i starten av prosjektet. Her ble det estimert hva som skulle gjøres og hvor lang tid de ulike oppgavene ville ta. Vi delte opp oppgavene i to kategorier: "Rapportskriving" og "Utvikling", og delte innledningsvis opp utviklingsoppgavene i "Frontend" og "Backend".

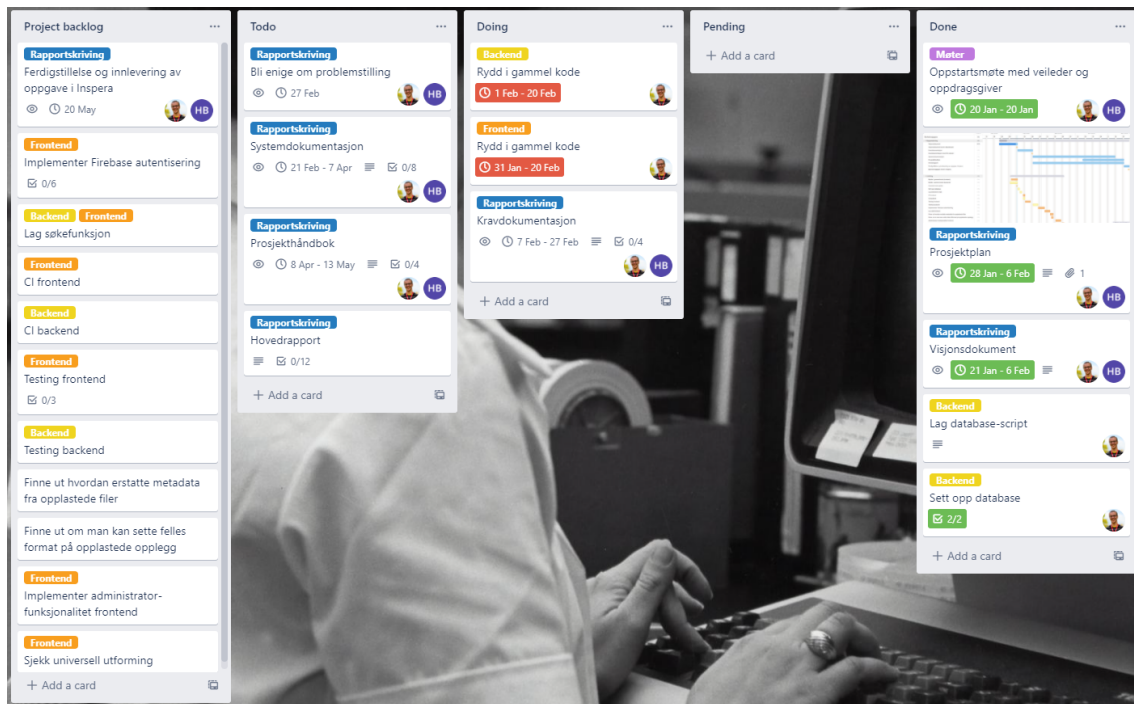


Figur 25: Prosjektplan i form av Gantt-diagram

I kategorien "Rapportskriving" la vi inn alle rapporter som skulle leveres i prosjektet og fordelte disse utover i kalenderen basert på vår estimerte tidsbruk. Vi satte også inn to milepæler underveis - ett for når det ferdigskrevne visjonsdokumentet skulle leveres i Blackboard, og ett for når vi planla å presentere en første versjon av kravdokumentasjonen for veileder. Systemdokumentasjonen og hovedrapporten ble estimert til å starte etter ferdigstillelse av kravdokumentasjonen, sammen med videreutviklingen av produktet.

I kategorien "Utvikling" ble det lagt inn oppgaver som ble laget basert på oppstartsmøtet med oppdragsgiver og gjennomgang av dokumentasjonen til den eksisterende oppgaven. Disse oppgavene ble justert underveis i prosjektet, og det ble lagt til nye oppgaver. Gantt-diagrammet ble likevel ikke oppdatert underveis i prosjektet ettersom vi tok i bruk en Trello-tavle for å ha oversikt over fremdriften i prosessen og oppgavene i prosjektet.





Figur 26: Eksempel fra Trello-tavlen, fra 21.02.2022

I Trello-tavlen ble oppgaver lagt til som kort, og de ble lagt under en spesifikk kategori basert på status på oppgaven. I eksempelet fra Trello-tavlen kan vi se et øyeblikksbilde fra prosessen, hvor blant annet visjonsdokumentet er ferdigstilt og arbeidet som akkurat i dette øyeblikket utføres er refaktorisering av eksisterende kode, samt skrivning av kravdokumentasjon. Se prosjekthåndboken og ukerapportene der for en nærmere beskrivelse av aktiviteter og hendelser underveis i prosessen.

#### 4.3.2 Timeregnskap

For å holde oversikt over tidsbruk i løpet av prosjektet førte vi timeregnskap med hjelp av en mal som også oppsummerte det totale antallet timer hver person brukte. Vi skrev også en timerapport hver uke, som kort forklarte hva hvert av gruppemedlemmene hadde arbeidet med i uken som var. Timelistene og ukerapportene blir i sin helhet presentert i prosjekthåndboken. I starten av prosjektet satte vi opp et estimat på hvor mange timer hver person totalt ville bruke. Dette estimatet var satt til 400 timer per person, ettersom begge gruppemedlemmene jobber ved siden av studiet og vi ikke trodde vi ville klare å nå de normerte 500 timene for et prosjekt av denne størrelsen. Det totale antallet timer endte opp med å stemme ganske godt med det vi i starten estimerte. Truls, som måtte bruke en del tid på å fikse problemer som oppstod i forbindelse med utviklingen av produktet, endte opp med å bruke noen flere timer enn estimert.

### Oppsummering av timelister i prosjekt nr: 13

Ukenr	Henna	Truls	Sum timer pr uke
2	14.0	6.0	20.0
3	19.5	13.5	33.0
4	24.5	15.5	40.0
5	29.0	25.0	54.0
6	12.0	10.0	22.0
7	22.0	8.0	30.0
8	20.0	21.0	41.0
9	0.0	2.0	2.0
10	26.0	11.0	37.0
11	23.0	9.5	32.5
12	19.0	17.5	36.5
13	12.5	21.5	34.0
14	22.5	3.5	26.0
15	0.0	40.0	40.0
16	20.0	34.0	54.0
17	23.0	35.0	58.0
18	30.0	71.0	101.0
19	40.0	37.0	77.0
20	46.0	41.0	87.0
<b>Sum antall timer pr person/totalt</b>	<b>403.0</b>	<b>422.0</b>	<b>825.0</b>

Figur 27: Oppsummering av totalt antall timer per grupped medlem per 18.05.2022

#### 4.3.3 Produktutviklingsprosessen

Den geografiske avstanden umuliggjorde fysiske møter mellom grupped medlemmene eller med veileder og/eller oppdragsgiver. Etersom dette studieprogrammet i sin helhet er et nettbasert studie var ikke dette noe som var ukjent eller uvant for grupped medlemmene. Samtidig anser vi det verdt å nevne i forbindelse med presentasjon av de administrative resultatene av prosjektet, da det satte noen spesielle begrensninger til arbeidet og arbeidsformene, f.eks. ved gjennomføring av brukertesting.

Arbeidsformen vi valgte for dette prosjektet er, som tidligere nevnt, basert på Kanban-metoden. Vi vurderte dette som en egnet form grunnet gruppestørrelsen. Dette innebærer at vi ikke hadde noen tydelige delmål som vi arbeidet mot, slik som det hadde vært hvis vi brukte en sprint-basert metode. Produktutviklingsprosessen foregikk med en løpende fremdrift, hvor oppgaver ble valgt og utført fra uke til uke. Vi hadde ukentlige møter innad i gruppen, og møte med veileder omtrent annenhver uke. Vi hadde også et ekstra møte med oppdragsgiver litt utover i semesteret for å få svar på noen spørsmål om produktet som dukket opp underveis i prosessen.

---

## 5 Diskusjon

### 5.1 Drøfting av vitenskapelige resultater

Designet av applikasjonen er sterkt inspirert av NTNU sin egen hjemmeside. Vi bruker de samme fargene som primærfarge, sekundærfarge, og bakgrunn samt. samme font. Dette var retningen vi ønsket å gå helt fra start og var det vi brukte som basis når vi lagde wireframes. Disse wireframesene finnes i visjonsdokumentet (Vedlegg A).

Vi har forsøkt å lage et design som tar hensyn til universell utforming og benytter oss aktivt av semantisk HTML. Det vil si at vi bruker deskriptive tags istedenfor f.eks. bare en *div*, selv om funksjonaliteten til de ulike taggene er identiske. Når man gjør dette får skjermlesere og diverse hjelpeverktøy mer informasjon om applikasjonen og kan guide brukeren mer enn ellers (W3Schools 2022).

Vi har klart å oppfylle nesten alt vi ønsket i samsvar med wireframesene våre sett bortifra den funksjonalitet som har blitt sløffet (f.eks. innlogging med Google-konto).

Brukerne som testet applikasjonen ga inntrykk av at applikasjonen var lett å navigere da de kom seg dit de ønsket uten hjelp, men vi fikk tilbakemelding om at det å legge inn sitt eget undervisningsopplegg var litt tungvint å komme seg frem til da man måtte innom en unødvendig meny. Etter den tilbakemeldingen fjernet vi den menyen og la istedet til en nedtrekksmeny under brukernavnet til kontoen som er logget inn hvor vi lå alle handlinger man kan gjøre som er relevant til en konto. Det er da å se sine egne undervisningsopplegg, lage et nytt undervisningsopplegg, eller å redigere sin egen brukerkonto.

Gjennom utviklingen av applikasjonen var det situasjoner hvor vi ønsket at vi hadde satt opp og aktivt brukte programvaretester på en bedre måte. Det var et par ganger vi opplevde at en endring i én del av applikasjonen påvirket en annen del negativt til tross for at disse virket urelaterte.

### 5.2 Drøfting av ingeniørfaglige resultater

#### 5.2.1 Funksjonelle krav

Noe vi burde ha gjort mer i starten av prosjektet er å ytterligere konkretisere de funksjonelle kravene. Sann som kravene er skrevet nå er de utrolig sammensatte og det kan være vanskelig å destillere ut nøyaktig hva det er hvert krav spør etter. Ta for eksempel *"Brukere skal kunne se en liste over alle publiserte undervisningsopplegg og søke/filtrere denne listen."* Her vet vi at brukere skal kunne få opp en liste over alle publiserte undervisningsopplegg, dette kunne vært to-tre krav i seg selv. Så skal denne listen kunne søkes i og filtreres, men hva vil det si? Hva skal man kunne filtrere på? Hvordan skal man søke? Skal vi ha et søkefelt, hvor skal det være, osv.

Denne mangelen på spesifisitet har ført til at når man skulle jobbe med oppgaver måtte man gjøre større antagelser rundt hva det er kravene spør etter, og funksjonalitet kan ha blitt tapt på den måten.

Underveis med arbeidet så vi at vi ikke kom til å få tid til å lage all funksjonaliteten vi ønsket i starten. Vi gjorde da beslutningen om å sløyfe all administratorfunksjonalitet og fokuserte istedenfor på å lage en bra prototype for funksjonene en normal sluttbruker vil benytte seg av. Dette tenkte vi var et smart valg ettersom vi ønsket å gjøre brukertester senere. Dette valget innebærer dog at et par av resultatmålene som vi satte opp i visjonsdokumentet ikke er nådd. Et mål var at produktet skulle være klart til å tas i bruk etter levering av bacheloroppgaven, men det vil nok kreves mer arbeid før applikasjonen er klar for å brukes. I tillegg ønsket vi at 90% av kravene satt til produktet skulle være implementert, men det klarte vi ikke å nå med tanke på sløyfingen av administratorfunksjonaliteten.

---

### 5.2.2 Ikke-funksjonelle krav

Når det kommer til ikke-funksjonelle krav mener vi egentlig at vi oppfyller alle kravene vi satt oss selv. Det er gjort en innsats for å forenkle prosessen rundt oppsett av applikasjonen og for at det å utvide produktet for de som kommer etter oss skal være så lett som mulig. Dette var ett av resultatmålene i visjonsdokumentet, og det kan vi derfor anse for å være nådd.

Også når det gjelder de ikke-funksjonelle kravene kunne vi ha tjent godt på å gjøre kravene enda mer spesifikke og delt dem opp. Når krav er ikke-funksjonelle kan dette være litt vanskeligere å få til, men sånn som kravene er i dag er det ikke lett å se f.eks. hva ”Godt dokumentert kildekode” skal bety.

### 5.2.3 Brukertesting

Brukervennlighet har vært en viktig del av vårt arbeid med dette prosjektet. Det var derfor helt essensielt at vi fikk hentet inn brukertesting som et element i prosessen. Vi ønsket å få testpersoner som var i målgruppen for produktet for å få et mest mulig virkelighetstro testresultat. Det var derfor gledelig å få positiv tilbakemelding fra Ryenberget skole om muligheter for brukertesting. Det eneste man kanskje kunne justert når det gjelder vår brukertesting, er antallet testpersoner som gjennomførte brukertesting. Det ble i teoridelen av denne rapporten nevnt at et optimalt antall testbrukere er fem personer, men at man ved bruk av tre testbrukere likevel får opp til to tredeler av de innspill man trenger. Det hadde selvfølgelig vært en fordel med fem testpersoner, men selv om vi bare fikk rekruttert tre testbrukere opplever vi likevel at vi har fått en viktig indikasjon på produktets verdi for brukerne. Ved en videreutvikling av applikasjonen anbefales at man prøver å få tak i fem testpersoner, og gjerne i en variert aldersgruppe. De testpersoner vi brukte var alle rundt 30 år. Det ville vært interessant å se testresultatene fra testbrukere som er litt eldre og kanskje har mindre erfaring med bruk av teknologi.

Testgjennomførelsen viser at produktet er på riktig vei. Testbrukerne klarte å navigere i applikasjonen uten problemer, og virket positive til en slik applikasjon. De kom også med positive tilbakemeldinger angående design og funksjonalitet. Det virker som at brukeropplevelsen av applikasjonen vurderes som høy. Ved testgjennomførelsen fikk vi også en god samtale med lærerne om problemstillingen. Det viste seg at det var mye nyttig og relevant informasjon og erfaring som de hadde å dele med oss. Det meste av det som ble nevnt i den samtalen kommer også frem i responsen fra spørreskjemaet (vedlegg E). Det ble i den samtalen, og i responsen fra spørreskjemaet, bekreftet at vår initiale tanke om at lærerne ikke har tid, kompetanse eller er så veldig interessert i å lære programmering viser seg å stemme godt med virkeligheten. Lærerne nevnte at all hjelp de kan få, er veldig velkomment.

Et interessant innspill som en av testbrukerne kom med var følgende: *”Programmet blir ikke bedre enn oppleggene som ligger der. At dette brukes kommer an på hvordan ressursene oppleves, når brukervennligheten er på plass”* (se figur).

---

Har du andre innspill/forslag til forbedring?

3 svar

Som tidligere nevnt: Det hadde vært fint med en side inne på applikasjonen som kunne vært "programmering for dummies".

Ellers syns jeg det virka som en god side. Veldig fint å vite at undervisningen er lagd spesielt til barne- og ungdomsskolen, og at de er jobbet godt med av studenter som studerer programmering.

Håper på at plattformen blir tilgjengelig så fort som mulig!

Programmet blir ikke bedre enn oppleggene som ligger der. At dette vil brukes kommer an på hvordan ressursene oppleves, når brukervennligheten er på plass.

Figur 28: Sitat fra testbruker. Skjerm bilde fra vedlegg E.

Dette er et oppsiktsvekkende innspill, fordi det løfter frem et viktig aspekt ved applikasjoner av denne typen. Testpersonen peker på at det viktigste med en applikasjon som tilbyr programmeringsopplegg er kvaliteten på undervisningsoppleggene som blir lagret der. Dette er interessant fordi det stiller litt andre krav til applikasjonen enn det som vi gikk ut fra. Det handler altså ikke bare om applikasjonen slik som den er, men også om kvaliteten og tilgjengeligheten på det som blir lastet opp der. Et eksempel som man i denne sammenhengen kan sammenligne med, er strømmetjenester. Det finnes en del ulike strømmetjenester, og alle har de sitt spesifikke innhold. Dersom det er en spesiell serie eller film man ønsker å se, benytter man seg av den strømmetjenesten som tilbyr akkurat denne. Hvis strømmetjenesten ikke har noe annet du ønsker å se enn akkurat den spesielle serien, så er veien kort til å avslutte abonnementet til fordel for en annen strømmetjeneste med et bedre innhold. Vi ser her at det ikke er strømmetjenesten som sådan som tilbyr verdi til brukeren, men innholdet de tilbyr. Det må også sies å være tilfelle for den applikasjonen vi har utviklet. Det er viktig at man tar en gjennomgang av oppleggene som allerede eksisterer, og gjerne tar utgangspunkt i en analyse av hva brukerne opplever som et godt undervisningsopplegg og hvordan et slikt ser ut. Dette er noe som må bli vektlagt ved en videreutvikling av applikasjonen.

Effektmålene vi satte opp i visjonsdokumentet i starten av prosjektet handlet om at vi ønsket å bidra til å forenkle arbeidet for lærere som skal undervise i programmering å finne gode undervisningsopplegg og å gi de et nettsted som kan fungere som et slags oppslagsverk for undervisningsopplegg. Resultatene av både applikasjon og brukertesting viser oss at disse målene på mange måter kan sies å være, om ikke helt, så i hvert fall delvis nådd. Applikasjonen krever noe merarbeid for å være det som lærerne trenger, men den er på god vei.

### 5.3 Drøfting av administrative resultater

Her vil vi presentere og diskutere årsakene til at det oppsto eventuelle avvik fra de planer vi opprinnelig satte for prosjektutviklingsprosessen.

#### 5.3.1 Prosjektplan

I Gantt-diagrammet, som vi opprettet i starten av prosjektet, hadde vi estimert en tidsplan for de ulike delene av prosjektet. Det viste seg likevel at det var noen av tidsfristene som vi ikke klarte å overholde. Hvis man ser på Gantt-diagrammet og tidsplanen vi satte for kravdokumentasjonen, så kan man se at vi planla å være ferdige med kravdokumentasjonen rundt 20. februar. Vi hadde dog ikke en ferdig versjon av kravdokumentasjonen før 13. mars. En viktig årsak til dette kan være at vi ikke satte opp en faktisk og endelig frist for oss selv. I stedet snakket vi litt løst om at

---

vi ønsket å være ferdig med kravdokumentasjonen til det tidspunktet. I retrospekt kan vi nok si at en strengere tidsplan for oss selv hadde vært en fordel i denne sammenheng.

Et annet eksempel hvor vi ikke klarte å holde vår egen tidsplan, var ved refaktorisering av eksisterende kildekode. Det tok vesentlig mye lenger tid å fullføre enn vi i utgangspunktet hadde planlagt. Men samtidig må det sies at ingen av oss kunne se for oss hvor stor jobb refaktoreringen faktisk skulle komme til å bli. Den eksisterende kildekoden var ikke kjørbær, så det krevde en stor innsats fra Truls for å i det hele tatt få applikasjonen til å kjøre. Dette viser at det er aspekter ved gjennomføring av et systemutviklingsprosjekt som ikke alltid kan planlegges og utfordringer som er vanskelig å forutse.

### 5.3.2 Timeregnskap

Vårt opprinnelige estimat for hvor mange timer hvert gruppelem totalt ville bruke på prosjektet viste seg å stemme ganske godt. Ettersom begge jobber ved siden av studiet var vi klar over at vi ikke kunne bruke det antall timer som er normalt i en oppgave av denne størrelsen. Vi valgte derfor å sette 400 timer som estimat i stedet for de normerte 500 timene, fordi det var et mer realistisk mål for oss og fordi vi anså at vi likevel ville ha god nok tid for alle deler av prosjektet.

Når vi ser på oppsummeringen av den totale tidsbruken per person, kan vi se at timene som logget per uke er jevnt over færre enn de hadde vært hvis vi kunne bruke all tid på dette prosjektet. De siste ukene i prosjektet tok vi fri fra jobb for å jobbe med prosjektet. Derfor kan man se at antall timer per uke går opp de siste ukene. Dette viser at vi, selv om vi ikke hadde kapasitet til å bruke det antall timer vi i utgangspunktet kunne ønske, likevel hadde en god innsikt i arbeidskapasiteten og tidstilgjengeligheten til de enkelte gruppelemmene.

### 5.3.3 Produktutviklingsprosessen

Som tidligere nevnt er den geografiske avstanden mellom gruppelemmene relativt stor. Det har vært både fordeler og ulemper knyttet til dette. En stor fordel er nok at vi har kunnet samarbeide effektivt hjemmefra over nett, i stedet for å bruke tid på å reise hjemmefra for å møtes. Vi har også kunnet være fleksible på når på dagen vi møtes, da det eneste som trengs er at vi har tid og har tilgang til Discord. En ulempe med å være på ulike steder i landet viste seg særlig når det gjaldt brukertesting. Det hadde vært naturlig å gjennomføre brukertesting sammen, men ettersom den ene av oss er i Oslo og den andre i Trondheim, kunne bare den ene av oss delta. Men i det store hele har det fungert utmerket å samarbeide digitalt med dette prosjektet. Digitale samhandlingsplattformer er tilgjengelige, og arbeidet med dette prosjektet har gitt oss en god og nyttig erfaring i digital samhandling.

Ettersom vi brukte Kanban som utviklingsmetode, og vi ikke hadde sprinter hvor det kom tydelig frem hvilke oppgaver som skulle utføres og hvor vi stod i prosessen ved ethvert delmål, har det til tider vært litt utfordrende å dokumentere utviklingsprosessen. Dette gjelder særlig hvis man ønsker å få en rask oversikt over kjerneoppgaver i prosessen, slik som ferdigstillelse av rapporter eller tidspunkt for når applikasjonen er ferdig for brukertesting. Vi brukte Trello-tavle for å holde oversikt over fremdriften i prosessen, men tavlen forandret seg ofte fra dag til dag. Vi tok skjermbildet av Trello-tavlen ved jevne mellomrom, men det er likevel ting som skjedde i Trello-tavlen som ikke er dokumentert. For å dokumentere alt, hadde vi vært nødt til å ta skjermbilde av Trello-tavlen hver dag. Dette ble ikke gjort, og det kan kanskje anses som en svakhet i dokumentasjonen av utviklingsprosessen. I retrospekt kunne vi kanskje tatt inn noen elementer fra Scrum, slik som flere delmål å arbeide frem mot, og kombinert Scrum og Kanban slik at prosessen enklere kunne dokumenteres.

Gruppelemmene har, som nevnt, tidligere erfaring med å jobbe sammen i team. Vi trengte derfor ikke å bruke tid på å finne ut hvordan vi best bør fordele oppgavene. Tildeling av oppgaver har skjedd naturlig, og det har til og med vært slik at vi til tider automatisk har tatt oppgaver som vi vet passer oss best, uten at vi har blitt enige om det på forhånd. Dette har gjort at arbeidet med prosjektet har gått smidig, og samarbeidet har fungert meget godt. I tillegg kan vi tydelig se

---

at gruppemedlemmenes styrker utfyller hverandre, slik at vi har kunnet sikre en god kvalitet på alle områder i produktutviklingsprosjektet.

I starten av prosjektet satte vi opp prosessmål som vi ønsket å nå i løpet av arbeidet med denne bacheloroppgaven. Prosessmålene står beskrevet i visjonsdokumentet som er vedlagt i denne rapporten. Vi ønsket i prosessmålene at dette prosjektet skulle gi oss økt kompetanse i å gjennomføre et større systemutviklingsprosjekt, økte samarbeids- og kommunikasjonsevner samt økt erfaring av selvstendig planlegging. Arbeidet med dette prosjektet har fra det ståstedet vært fremgangsrikt. Vi har fått økt kunnskap og erfaring med produktutviklingsprosjekter, og har fått god øving i kommunikasjon og samarbeid. I løpet av prosjektet har vi hjulpet hverandre underveis dersom det har vært behov, og vi har stadig jobbet mot samme mål. En fordel i arbeidet har vært at vi fra starten har vært veldig enige i vår forståelse av problemstillingen. Det har hjulpet oss i arbeidet med å prøve å finne frem til en løsning.

---

## 6 Konklusjon og videre arbeid

### 6.1 Konklusjon

Vi startet arbeidet med dette prosjektet med kunnskapen om at det fra tidligere var gjennomført et forsøk på å lage en applikasjon, og at vårt bidrag vil være en videreutvikling av den eksisterende applikasjonen. Vi visste også at den eksisterende applikasjonen ikke møtte de krav som ble stilt fra oppdragsgivers side. Det var på mange måter et spennende prosjekt å starte på. Det at det allerede eksisterte en versjon av applikasjonen gjorde at vi hadde noe å gå ut fra når vi skulle sette i gang med en videreutvikling. Men samtidig viste det seg også at det krevde en del arbeid med refaktorisering av eksisterende kode for å få en kjørende applikasjon å videreutvikle.

Vårt utgangspunkt for arbeidet med problemstillingen var et litt annet enn det som den eksisterende applikasjonen bygde på. Vårt ønske var å lage en applikasjon hvor fokuset er på brukervennlighet. Vi gikk ut fra at lærere i den norske skolen ikke har tid eller motivasjon til å lære programmering, og vi ønsket å tilby et produkt som kan gi lærerne et lavterskelprodukt som kan brukes som redskap for å finne eksisterende undervisningsopplegg. Vår problemstilling **”Kan man lage en webapplikasjon som lærere i den norske grunnskolen ønsker å bruke når de skal søke etter og dele undervisningsopplegg for programmering i grunnskolen?”** og det utvidede spørsmålet: *”Hvilken funksjonalitet må være med for at brukeropplevelsen skal være høy nok til at lærerne opplever bruken av webapplikasjonen som **verdt det?**”* guidet oss inn i et arbeid hvor fokus var på intuitiv layout og en funksjonalitet som gir mening for brukerne.

Vårt fokus på brukervennlighet innebar at vi ønsket å benytte oss av brukertesting for å sjekke om vi nådde våre mål om brukervennlighet. Vi fikk i brukertesting gode tilbakemeldinger på produktet og en indikasjon på at produktet har en rolle å spille. På spørsmålet som problemstillingen kommer med, om hvorvidt det er mulig å lage et produkt som lærerne *ønsker* å bruke når de skal søke etter undervisningsopplegg, tror vi at svaret er ”ja”. Det er mulig å lage et slikt produkt, og basert på brukertesting virker det også som at produktet som vi har videreutviklet er ønsket. Det er utfordrende for lærere å lære programmering på den tid de har til rådighet, og det virker ikke som at de er så veldig motiverte til å lære det. Derfor gir det mening å si at produktet fyller en funksjon.

Når det gjelder spørsmålet om hvilken funksjonalitet som må være på plass for at applikasjonen skal være *verdt* å bruke, er svaret litt mer diffust. Tilbakemeldingen på produktet, slik produktet fremstår etter videreutviklingen, var positiv. Samtlige testbrukere meldte at de ville brukt produktet dersom det hadde vært tilgjengelig. Men samtidig kom også innspillet om at ”produktet er bare så godt som kvaliteten på undervisningsoppleggene som blir lastet opp der”. Eller hvis man sier det på en litt annen måte: så lenge undervisningsoppleggene er gode, så har det mindre å si hvordan applikasjonen ser ut. Selvsagt bør applikasjonen fortsatt være brukervennlig, men det som har størst betydning for om produktet vil bli tatt i bruk eller ikke, er hvordan undervisningsoppleggene er fremstilt. Dette fører oss inn på samtalen om videre arbeid og en eventuell videreutvikling av applikasjonen.

### 6.2 Videre arbeid

For å få applikasjonen til å bli så god den kan være må det gjøres et arbeid med å utvikle en felles modell for hvordan undervisningsoppleggene skal presenteres. Vi startet med mange av de samme antagelsene som gruppen før oss hadde gjort og utvidet på dem, men måten undervisningsoppleggene fremstår i den videreutviklede applikasjon er trolig ikke godt nok for å gi utbytte av undervisningsoppleggene. Derfor mener vi at man burde vurdere å endre på oppgaveordlyden som blir gitt ut til lærerne som tar fagene ved NTNU når de skal lage undervisningsopplegg, for å få oppleggene til å ha et mer samlet format.

I det vi har sett av eksisterende oppgaver, altså eksisterende undervisningsopplegg produsert i regi av NTNU, så er det et veldig stort sprik i hvordan lærerne velger å presentere opplegget. Noen har formatert opplegget som et eget dokument og gjort det mer presentabelt og tilgjengelig for andre å



---

forstå. Andre har selve opplegget nedgravd dypt inne i en rapporten som leveres i faget. Kanskje det ville være mulig å vektlegge tilgjengeligheten av undervisningsoppleggene som blir produsert i faget uten at det tar vekk ifra hva det egentlige målet til faget er?

Generelt oppfatter vi denne applikasjonen som en med mye potensiale så lenge man har muligheten til å forsøke litt frem og tilbake med flere brukertester, spørreunder, og man kan samarbeide med erfarne sluttbrukere om hvordan man skal representere et undervisningsopplegg på best mulig måte for brukere som har dårlig tid og mulig lav motivasjon.

Når det gjelder det tekniske så er det mye forbedringspotensiale i applikasjonen, det presenterer vi her i denne tabellen:

Hva	Beskrivelse
Administratorer kan legge til moderatorer	Administratorer bør ha mulighet for å gjøre andre brukere til moderatorer
Administratorer kan godkjenne eller avslå opplegg	Når et undervisningsopplegg blir publisert så kan man la være å vise det offentlig til alle sammen frem til en administrator har godkjent det. Man kan da ha en kø hvor opplegg som ikke har blitt behandlet ennå kan ligge. Hvis de blir avslått kan de fjernes fra køen.
Kommentarfelt	Det er godt mulig brukere kommer til å ønske å legge igjen kommentarer eller tilbakemeldinger på undervisningsopplegg. Da kunne det vært ypperlig med en form for kommentarfelt.
Visningstall	Hvis man lagrer antall visninger på hvert undervisningsopplegg kan man begynne å gjøre antagelser om og sortere på popularitet. Dette kan være med på å sikre at de beste oppleggene får god synlighet.
Likes	Litt det samme som med visningstall og kommentarfelt her. Hvis brukere kan gi en tilbakemelding på om de synes opplegget er bra eller ikke kan man vise en samlet score opplegget har fått og sortere etter denne.
SSL-sertifikater	Nå i dag genererer vi selv-signerte sertifikater til backend og nettlesere klager på dette. Om man får skaffet ordentlig signerte sertifikater fra en utsteder hadde mange nettlesere oppført seg bedre. Men dette kan muligens være utfordrene for lokal utvikling.

---

## Bibliografi

- Atlassian (2022a). *Is the Agile Manifesto still a thing?* URL: <https://www.atlassian.com/agile/manifesto> (visited on 18th May 2022).
- (2022b). *What is a kanban board?* URL: <https://www.atlassian.com/agile/kanban/boards> (visited on 18th May 2022).
- (2022c). *What is kanban?* URL: <https://www.atlassian.com/agile/kanban> (visited on 2nd May 2022).
- (2022d). *What is Scrum?* URL: <https://www.atlassian.com/agile/scrum> (visited on 18th May 2022).
- Bocconi, Stefania, Augusto Chiocciariello and Jeffrey Earp (2018). ‘The Nordic approach to introducing Computational Thinking and programming in compulsory education’. In: *Report prepared for the Nordic@ BETT2018 Steering Group*, pp. 397–400.
- Dahl, G., K. Ranestad and A. Hole (2022). *Programmering rammer dybdel ring i matematikk*. URL: <https://www.aftenposten.no/meninger/kronikk/i/E0zga/programmering-rammer-dybdelaering-i-matematikk-geir-dahl-kristian-ranestad-og-arne-hole> (visited on 3rd May 2022).
- Datatilsynet (2022). *ID-tyveri*. URL: <https://www.datatilsynet.no/personvern-pa-ulike-omrader/internett-og-apper/id-tyveri/> (visited on 18th May 2022).
- Digitaliseringsdirektoratet (2022). *Referansekatalogen for IT-standardar. Universell utforming av ikt*. URL: <https://www.digdir.no/standarder/universell-utforming-av-ikt/1499> (visited on 12th May 2022).
- Discord (2022). *IMAGINE A PLACE...* URL: <https://discord.com/> (visited on 17th May 2022).
- Europaparlamentet (2022). *Digital Agenda for Europe*. URL: <https://www.europarl.europa.eu/factsheets/en/sheet/64/digital-agenda-for-europe> (visited on 3rd May 2022).
- GeeksForGeeks (2022). *Client-Server Model*. URL: <https://www.geeksforgeeks.org/client-server-model/> (visited on 17th May 2022).
- GitHub, Inc. (2022). *Where the world builds software*. URL: <https://github.com/> (visited on 17th May 2022).
- Hartson, Rex and Pardha S Pyla (2018). *The UX book: Agile UX design for a quality user experience*. Morgan Kaufmann.
- H gstrand, J. (2022). *Scrum - et enkelt rammeverk for komplekst arbeid*. URL: <https://www.prosjektbloggen.no/scrum-et-enkelt-rammeverk-for-komplekst-arbeid> (visited on 16th May 2022).
- Husab , I. (2022). *B r kidsa kode p  skulen?* URL: <https://forskning.no/partner-skole-og-utdanning-vestlandsforskning/bor-kidsa-kode-pa-skulen/449820> (visited on 3rd May 2022).
- IBM (2022). *Relational Databases Explained*. URL: <https://www.ibm.com/cloud/learn/relational-databases> (visited on 17th May 2022).
- Johansen, A-K. (2022). *Programmering vil bli en utfordring for l rere*. URL: <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-l rere/1711838> (visited on 5th May 2022).
- Kopecky, C (2022). *React Bootstrap tutorial: Upgrade React apps with a CSS framework*. URL: <https://www.educative.io/blog/react-bootstrap-tutorial> (visited on 6th May 2022).
- LaTeX-Project, The (2022). *LaTeX - A document preparation system*. URL: <https://www.latex-project.org/> (visited on 17th May 2022).
- Meta Platforms, Inc. (2022). *React, A JavaScript library for building user interfaces*. URL: <https://reactjs.org/> (visited on 17th May 2022).
- Microsoft (2022a). *Code editing. Redefined*. URL: <https://code.visualstudio.com/> (visited on 17th May 2022).
- (2022b). *TypeScript is JavaScript with syntax for types*. URL: <https://www.typescriptlang.org/> (visited on 17th May 2022).
- Nielsen, J. (2022). *Why You Only Need to Test with 5 Users*. URL: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (visited on 13th May 2022).
- NobleDesktop (2022). *What is Git and Why Should You Use it?* URL: <https://www.nobledesktop.com/learn/git/what-is-git> (visited on 18th May 2022).
- NTNU (2022a). *IT6203 - Grunnleggende programmering for l rere*. URL: <https://www.ntnu.no/studier/emner/IT6203#tab=omEmnet> (visited on 5th May 2022).
- (2022b). *IT6204 - Anvendt programmering for l rere*. URL: <https://www.ntnu.no/studier/emner/IT6204#tab=omEmnet> (visited on 5th May 2022).

- 
- Oracle (2022). *MySQL*. URL: <https://www.mysql.com/> (visited on 17th May 2022).
- Overleaf (2022). *Latex, Evolved*. URL: <https://www.overleaf.com> (visited on 6th May 2022).
- Regjeringen (2022). *Ny personopplysningslov*. URL: <https://www.regjeringen.no/no/tema/statlig-forvaltning/personvern/ny-personopplysningslov/id2340094/> (visited on 18th May 2022).
- Ryvarden, E. (2022). *IT-bransjen roter med brukervennlighet*. URL: <https://www.digi.no/artikler/it-bransjen-roter-med-brukervennlighet/315833> (visited on 5th May 2022).
- Sequelize-Contributors. (2022). *Sequelize*. URL: <https://sequelize.org/> (visited on 17th May 2022).
- SNL (2022a). *Digitalisering*. URL: <https://snl.no/digitalisering> (visited on 3rd May 2022).
- (2022b). *Grunnopplæring*. URL: <https://snl.no/grunnoppl%C3%A6ringen> (visited on 5th May 2022).
- (2022c). *Grunnskolen*. URL: <https://snl.no/grunnskolen> (visited on 5th May 2022).
- (2022d). *Kompetansemål*. URL: <https://snl.no/Kompetansem%C3%A5l> (visited on 3rd May 2022).
- SSB (2022). *Praktisk brukertesting*. URL: [https://www.ssb.no/a/histstat/ssh/ssh\\_87.pdf](https://www.ssb.no/a/histstat/ssh/ssh_87.pdf) (visited on 12th May 2022).
- UDIR (2022a). *Hvorfor har vi fått nye læreplaner?* URL: <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvorfor-nye-lareplaner/> (visited on 3rd May 2022).
- (2022b). *Kompetansemål og vurdering - Kompetansemål etter 5. trinn*. URL: <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv19?lang=nob> (visited on 3rd May 2022).
- (2022c). *Om overordnet del*. URL: <https://www.udir.no/lk20/overordnet-del/om-overordnet-del/> (visited on 2nd May 2022).
- UIB, Universitetet i Bergen (2022). *Risikomatrise*. URL: <https://www.uib.no/hms-portalen/137767/risikomatrise> (visited on 18th May 2022).
- Ullfoss, S. (2022). *Smidig vs. tradisjonell prosjektmetodikk - hva er best?* URL: <https://sprint.no/artikler/smidig-vs-tradisjonell-prosjektmetodikk-hva-er-best> (visited on 18th May 2022).
- uutilsynet (2022). *WCAG sortert etter prinsipp*. URL: <https://www.uutilsynet.no/wcag-standarden/wcag-sortert-etter-prinsipp/713> (visited on 12th May 2022).
- W3Schools (2022). *HTML Accessibility*. URL: [https://www.w3schools.com/html/html\\_accessibility.asp](https://www.w3schools.com/html/html_accessibility.asp) (visited on 17th May 2022).
- Wikstrøm, M. M. (2022). *Hva er en webapplikasjon*. URL: <https://inereo.no/blogg/kort-om-webapplikasjoner/> (visited on 5th May 2022).

## Vedlegg

Her følger vedleggene våre:

## A Visjonsdokument



INSTITUTT FOR DATATEKNOLOGI OG INFORMATIKK

INFT2900 - BACHELOROPPGAVE I INFORMASJONSBEHANDLING

---

# Videreutvikling av webapplikasjon for undervisningsopplegg

---

VISJONSDOKUMENT

*Forfattere:*  
Henna Fredrika Björkskog  
Truls Henrik Jakobsen

Versjon 1.0

## Innhold

<b>1 Introduksjon</b>	<b>2</b>
1.1 Hensikt . . . . .	2
1.2 Omfang . . . . .	2
<b>2 Sammendrag problem og produkt</b>	<b>2</b>
2.1 Problemsammendrag . . . . .	2
2.2 Produktsammendrag . . . . .	3
2.3 Prosjekt mål . . . . .	3
2.3.1 Effektmål . . . . .	3
2.3.2 Resultatmål . . . . .	3
2.3.3 Prosessmål . . . . .	4
<b>3 Overordnet beskrivelse av interessenter og brukere</b>	<b>4</b>
3.1 Oppsummering av interessenter . . . . .	4
3.2 Oppsummering brukere . . . . .	4
3.3 Brukermiljøet . . . . .	4
3.4 Sammendrag av brukernes behov . . . . .	5
3.5 Alternativer til vårt produkt . . . . .	7
<b>4 Produktoversikt</b>	<b>7</b>
4.1 Produktets rolle i brukermiljøet . . . . .	7
4.2 Forutsetninger og avhengigheter . . . . .	8
<b>5 Produktets funksjonelle egenskaper</b>	<b>8</b>
<b>6 Ikke-funksjonelle egenskaper og andre krav</b>	<b>8</b>

## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
6 februar 2022	0.1	Første utkast av visjonsdokumentet	Henna F. Björkskog & Truls H. Jakobsen
17 mai 2022	1.0	Justert på bakgrunn av tilbakemelding fra veileder	Henna F. Björkskog

# 1 Introduksjon

## 1.1 Hensikt

Dette visjonsdokumentet er utarbeidet i forbindelse med prosjektet "Videreutvikling av webapplikasjon for undervisningsopplegg" som gjennomføres som bacheloroppgave i informasjonsbehandling ved NTNU våren 2022. Hensikten med dokumentet er å tydeliggjøre visjonen som ligger bak prosjektet og applikasjonen som utvikles. Dokumentet tar for seg de problemstillingene og behovene som utgjør bakgrunnen for prosjektet og som det utviklede produktet vil være med på å løse.

Opgaven er å videreutvikle et produkt som ble utviklet i forbindelse med en tidligere bacheloroppgave, utført av en gruppe studenter ved NTNU våren 2021. Målet er å ferdigstille produktet i sin helhet slik at det skal kunne tas i bruk etter levering av bacheloroppgaven, samt gjøre justeringer og legge inn manglende funksjonalitet for å forbedre produktet.

## 1.2 Omfang

Visjonsdokumentet dekker målene og kravene som er satt til prosjektet. I dokumentet presenteres problem og produkt og det blir også gitt en beskrivelse av interessenter og brukere og de behovene disse har. Dokumentet legger også frem produktets rolle og nevner de funksjonelle og ikke-funksjonelle kravene til produktet.

# 2 Sammendrag problem og produkt

## 2.1 Problemsammendrag

<b>Problem med</b>	Programmering blir en del av utdanningsplanen for barn. Kompetansen for å undervise i dette er ikke nok utbredt, og undervisningsoppleggene for det er ikke lett tilgjengelig gitt at de finnes. Eksisterende undervisningsopplegg inneholder ikke programmering og må derfor erstattes.
<b>Dette berører</b>	lærere, elever, foreldre, skoleadministratorer.
<b>Som resultatet av dette</b>	vil mange lærere måtte anskaffe seg kompetanse innenfor programmering og lage nye undervisningsopplegg, uten at det skal påvirke pågående undervisning. Lærerne vil da sannsynligvis måtte gå utover sin ordinære timeplan.
<b>En vellykket løsning vil</b>	gi lærere en solid mulighet til å finne og dele undervisningsopplegg som inneholder programmering. Dette vil hjelpe lærere spare tid med utvikling av undervisningsopplegg, hvor de kanskje istedet kan bruke den tiden på å øke sin egen kompetanse som vil resultere i undervisning av høyere kvalitet for elevene.



## 2.2 Produktsammendrag

<b>For</b>	NTNU
<b>som</b>	har behov for et online ”oppslagsverk” hvor lærere kan finne og dele undervisningsopplegg som inneholder programmering. NTNU har et årslangt kurs for lærere som ønsker å tilegne seg programmeringskompetanse hvor de i løpet av kurset produserer et undervisningsopplegg innenfor sitt fagområde.
<b>Produktet <i>Prosjekt Programmeringsreform</i></b>	er en webapplikasjon
<b>som</b>	gir lærerne akkurat denne muligheten til å søke gjennom hundrevis av høykvalitets undervisningsopplegg slik at de istedet kan bruke tid på å heve sin egen kompetanse eller planlegge undervisning for å kunne bedre tilbudet for elevene.
<b>I motsetning til</b>	at lærere må produsere alt materiale på egen hånd eller søke vilkårlig rundt på nettet etter ideer
<b>har vårt produkt</b>	samlet undervisningsopplegg produsert i regi av NTNU på et effektivt og søkbart format slik at lærere kan finne nøyaktig hva de leter etter uten mye bry.

## 2.3 Prosjektmål

På bakgrunn av problem- og produktsammendragene kan vi nå se på de ulike målene for prosjektet. Disse deles inn i effektmål, resultatmål og prosessmål. Effektmålene handler om de langsiktige målene som det er ønskelig at produktet har, samt hvordan produktet kan bidra og påvirke i et lengre perspektiv. Resultatmålene tar for seg de mer konkrete og målbare resultatene etter at prosjektet er gjennomført. Prosessmålene retter seg mot gjennomføringen av prosjektet som en prosess, og viser på de forventninger som er knyttet til arbeidet med prosjektet.

### 2.3.1 Effektmål

- Bidra til å forenkle arbeidet for lærere i den norske skolen som skal lære programmering for i sin tur å kunne lære barn å programmere.
- Gjøre det enklere og raskere for lærere å finne gode undervisningsopplegg for programmering i skolen.
- Gi lærerne et nettsted som kan fungere som et kompendium eller oppslagsverk for undervisningsopplegg, hvor de kan vite at oppleggene har god kvalitet og er rettet mot den norske skolen.

### 2.3.2 Resultatmål

- Videreutvikle produktet slik at det er klart til å tas i bruk etter levering av bacheloroppgaven 20. mai 2022.
- Implementere 90 prosent av kravene som er satt til det ønskede produktet.
- Åpne opp for muligheten for en ytterligere videreutvikling av produktet ved en senere anledning.

### 2.3.3 Prosessmål

- Få økt kompetanse i gjennomføring av et større systemutviklingsprosjekt.
- Tillegne seg gode samarbeids- og kommunikasjonsevner og økte ferdigheter i teamarbeid.
- Øke erfaringen med selvstendig planlegging, og lære hvordan god prosjektgjennomføring kan bidra til å oppnå ønskede mål.

## 3 Overordnet beskrivelse av interessenter og brukere

### 3.1 Oppsummering av interessenter

Interessent	Utdypende beskrivelse	Rolle under utvikling
Lærere i den norske skolen	Sluttbruker	Har laget undervisningsopplegg som kan legges inn i databasen for testing og utvikling av produktet. Kan kanskje også bidra med testing underveis.
Oppdragsgiver	Produkteier	Har rollen som kunde. Kommer med krav og ønsker til utviklingsteamet underveis i prosessen.
Veileder	Faglærer	Bidrar med støtte og hjelp underveis i prosessen.
Prosjektgruppen	Utviklere	Utvikler produktet.

### 3.2 Oppsummering brukere

Bruker	Utdypende beskrivelse	Rolle under utvikling	Representert av
Lærere i den norske skolen	Bruker av systemet og de som lager/laster opp undervisningsopplegg i applikasjonen.	Testing	Utviklere og eventuelt eksterne personer.
Administrator for systemet	Bruker som har administratorrolle for systemet og som skal godkjenne undervisningsopplegg før de blir gjort tilgjengelig for nedlasting.	Testing	Utviklere og eventuelt eksterne personer.

### 3.3 Brukermiljøet

Produktet utvikles for lærere i den norske skolen som skal undervise i programmering. Applikasjonen skal kunne brukes både i forbindelse med opplasting av undervisningsmateriale i NTNUs programmeringsfag for lærere og som oppslagsverk for lærerne når de på egen hånd skal lage/finne undervisningsmateriale i programmering for barn i skolen. Lærerne vil kunne lage sin egen bruker i applikasjonen. Produktet vil være tilgjengelig via nettleser, og lærerne vil derfor kunne bruke produktet uten å være koblet til NTNUs nettverk eller pålogget med NTNU-bruker.

### 3.4 Sammendrag av brukernes behov

Behov	Prioritet	Dagens løsning	Foreslått løsning	Vedrører
Laste opp undervisningsopplegg	Høy	Eksisterende løsninger som har undervisningsopplegg i programmering på nett har gjerne bare ett eller to opplegg som de har kurert selv, noe som stopper andre lærere fra å legge til direkte.	Vi vil starte med å publisere alle undervisningsopplegg produsert i regi av NTNU for de lærere som ønsker det, og deretter gjøre det tilgjengelig for hvem som helst å laste opp sitt eget undervisningsopplegg.	Lærere som laster opp (produsenter), og lærere som laster ned (konsumere).
Legge inn tagger for søkbarhet	Høy	Man kan gjerne sortere på årstrinn eller vanskelighetsgrad og programmeringsspråk	Vi ønsker også at alt skal være søkbart på dette, men også at man som opplaster kan legge til flere tags som konsumere kan søke etter.	Hovedsakelig konsumere, vil forbedre fremkomligheten i produktet.
Søke blant opplastede undervisningsopplegg	Høy	Samme som forrige punkt, så har de fleste løsningene søkefunksjonalitet for årstrinn og programmeringsspråk	Vi ønsker å ha en bredere søkefunksjonalitet som lar deg søke på fagområde, tekst fra opplegget, og brukerdefinerte tags.	Hovedsakelig konsumere, vil forbedre fremkomligheten i produktet.
Fortsetter på neste side				

– fortsettelse fra forrige side

<b>Behov</b>	<b>Prioritet</b>	<b>Dagens løsning</b>	<b>Foreslått løsning</b>	<b>Vedrører</b>
Laste ned undervisningsopplegg	Høy	En del eksisterende løsninger tilbyr planer for større undervisningsopplegg, men de baserer seg ofte på at på at programmering skal være et eget fag man setter av tid til og ikke så mye integrert i de andre fagene.	Vi vil gjerne få mer moderne undervisningsopplegg som er planlagt for at de skal kunne brukes i ordinær undervisning i andre fag, istedenfor at programmering skal være et eget spesialfag. At brukere lett skal kunne laste opplegg er da utrolig viktig.	Konsumere
Lage bruker	Middels	Noen av de eksisterende løsningene har funksjonalitet for brukere, men det er da som oftest for at du som en elev som skal utføre oppgaver har tilgang til oppgavene og kan holde rede på hvor langt du har kommet.	Vi ønsker at lærere skal kunne logge inn for å publisere innhold, men også at man skal kunne gi vurderinger og kommentere på innhold samt. lagre favoritter for å komme tilbake til dem senere.	Opplastere og konsumere vil få mer kontroll over innholdet dersom de har kontoer
Logge inn	Middels	Følger naturlig av funksjonen med å kunne lage bruker.	Vi ønsker å inkludere integrasjoner med andre påloggingstjenester som Google og FEIDE.	Alle brukere
Logge ut	Middels	Naturlig følge	Naturlig følge	Alle brukere
Slette bruker	Middels	Dette er kravpålagt at det skal finnes dersom man har noen form for brukersystem	Vi ønsker også å ha dette, med funksjonalitet som lar det bevare innhold du har publisert gitt at det anonymiseres.	Alle brukere

Fortsetter på neste side

– fortsettelse fra forrige side

Behov	Prioritet	Dagens løsning	Foreslått løsning	Vedrører
Godkjenne undervisningsopplegg før det blir tilgjengelig for nedlasting (administrator)	Lav	Stort sett på eksisterende løsninger er det kun administratorer som kan legge til innhold. Hvis man ønsker å bidra med noe selv må man sende inn en epost og vente på svar før noe kanskje blir lagt til	Vi ønsker at hvem som helst skal kunne laste opp innhold og at dette skal være tilgjengelig så fort som mulig. Vi vurderer å ha en kø som opplastet innhold havner i og må godkjenner fra før det publiseres, men omfanget kan bli svært stort her. Et alternativ er å ha gode modereringsverktøy som at brukere kan flagge dårlig innhold, administratorer kan slette innhold og fjerne brukere.	Alle brukere, spesielt administratorer

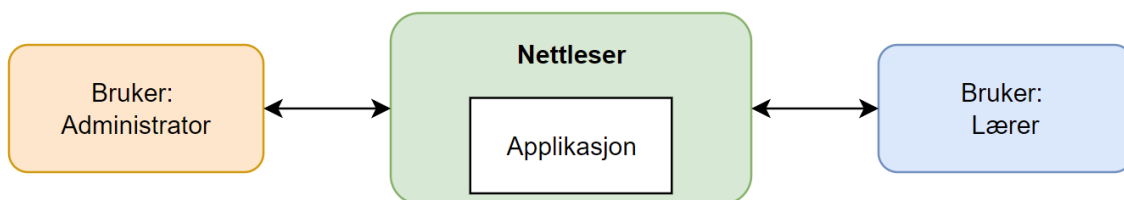
### 3.5 Alternativer til vårt produkt

Det finnes allerede en del undervisningsopplegg for programmering på internett. Ved et enkelt Google-søk kan man få opp flere slike. Det eksisterer også nettsider som har samlet flere undervisningsopplegg på samme sted, f.eks. Lær Kidsa Koding (<https://www.kidsakoder.no/>). Learnlink (<https://www.learnlink.no/>) tilbyr programmeringskurs for barn mot betaling. I tillegg finnes Minecraft Education Edition (<https://education.minecraft.net/en-us>) som kan brukes slik at barn selv kan lære programmering via Minecraft. For å bruke Minecraft Education Edition trenger man en Microsoft-skolekonto.

## 4 Produktoversikt

### 4.1 Produktets rolle i brukermiljøet

Figur 1 viser brukermiljøet og relasjonen mellom produktet og brukerne. Applikasjonen kjører i en nettleser. To typer brukere har tilgang til applikasjonen: den som administrerer applikasjonen, og de lærere som skal ta i bruk applikasjonen. Lærerne laster opp sitt undervisningsopplegg i applikasjonen for godkjenning, og administrator godkjenner oppleggene og gjør de tilgjengelige for nedlasting.



Figur 1: Produktets rolle i brukermiljøet

## 4.2 Forutsetninger og avhengigheter

Applikasjonen forutsetter at brukeren har tilgang til en datamaskin med internett-tilgang og en epostadresse.

## 5 Produktets funksjonelle egenskaper

For at produktet skal kunne imøtegå behovene som brukerne har, er det en del funksjonelle egenskaper som er helt essensielle. Disse egenskapene vil variere litt basert på hvilken type bruker det er. Listen som følger inneholder noen av de funksjonelle krav som må være implementert i det ferdige produktet. Disse vil også bli gjennomgått nærmere i kravdokumentasjonen hvor også brukerhistorier presenteres.

- Opprette bruker (administrator og lærer)
- Logge inn (administrator og lærer)
- Logge ut (administrator og lærer)
- Slette bruker (administrator og lærer)
- Laste opp undervisningsopplegg (lærer)
- Legge til tagger for å forbedre systemets søkbarhet (lærer)
- Godkjenne undervisningsopplegg (administrator)
- Laste ned undervisningsopplegg (lærer)
- Se liste over registrerte brukere (administrator)

## 6 Ikke-funksjonelle egenskaper og andre krav

Det er viktig at applikasjonen følger kravene om universell utforming. Dette vil bli lagt vekt på i arbeidet med prosjektet. I tillegg bør produktet være brukervennlig og enkelt å ta i bruk. Det er også krav på et visst nivå av informasjonssikkerhet til systemet. Selv om det ikke vil bli lagret sensitiv informasjon i systemet, er det likevel avgjørende at informasjon om brukerne ikke kommer på avveie. Dette stiller krav til håndtering av metadata og påloggingsrutiner. Det vil også være av stor betydning at kvaliteten på undervisningsoppleggene er på et godt nivå, med tanke på kvalitetssikring av applikasjonen.

## B Kravdokumentasjon



INSTITUTT FOR DATATEKNOLOGI OG INFORMATIKK

INFT2900 - BACHELOROPPGAVE I INFORMASJONSBEHANDLING

---

# Videreutvikling av webapplikasjon for undervisningsopplegg

---

KRAVDOKUMENTASJON

*Forfattere:*  
Henna Fredrika Björkskog  
Truls Henrik Jakobsen

Versjon 1.0

## Innhold

<b>1 Innledning</b>	<b>2</b>
<b>2 Brukerhistorier og akseptansekriterier</b>	<b>2</b>
<b>3 Domenemodell</b>	<b>5</b>
<b>4 Funksjonelle og ikke-funksjonelle krav</b>	<b>6</b>
4.1 Funksjonelle krav . . . . .	6
4.2 Ikke-funksjonelle krav . . . . .	6
4.3 Videre utvidelser . . . . .	7
<b>5 Prototyper</b>	<b>7</b>
5.1 Wireframes . . . . .	7



## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
13 mars 2022	0.1	Første utkast av kravdokumentasjonen	Henna F. Björkskog & Truls H. Jakobsen
17 mai 2022	1.0	Justert basert på tilbakemelding fra veileder	Henna F. Björkskog

## 1 Innledning

Denne kravdokumentasjonen er skrevet i forbindelse med prosjektet "Videreutvikling av webapplikasjon for undervisningsopplegg" som utføres som bacheloroppgave ved NTNU, våren 2022. Dokumentasjonen tar for seg en detaljert beskrivelse av kravene som ligger til grunn for applikasjonen som utvikles. I dokumentasjonen får vi presentert brukerhistorier og akseptansekriterier. Disse vil legges til grunn for testekriteriene i forbindelse med utvikling av applikasjonen. Kravdokumentasjonen vil også legge fram en prototype for applikasjonen, som vil presenteres ved hjelp av wireframes.

## 2 Brukerhistorier og akseptansekriterier

Brukerhistorie 1
Som bruker Ønsker jeg å se oversikten over opplastede undervisningsopplegg
Akseptansekriterier:
<b>Scenario 1:</b> se oversikt over alle opplastede undervisningsopplegg Gitt at jeg er på forsiden av applikasjonen Og jeg ikke er pålogget Når jeg trykker på "Utforsk" Så skal jeg bli sendt til siden for oversikt over alle opplastede undervisningsopplegg Og jeg skal se en liste over alle opplastede undervisningsopplegg
<b>Scenario 2:</b> filtrere blant opplastede undervisningsopplegg basert på tagg Gitt at jeg er på siden for oversikt over alle opplastede undervisningsopplegg Når jeg velger fagområdet "matematikk" i filterlisten Så skal kun de undervisningsoppleggene som har taggen "matematikk" vises
<b>Scenario 3:</b> tilbakemelding ved tomt filterresultat Gitt at jeg er på siden for oversikt over alle opplastede undervisningsopplegg Og ingen av de opplastede undervisningsoppleggene har taggen "kjemi" Når jeg velger fagområdet "kjemi" i filterlisten Så skal jeg få beskjed om "Ingen søkeresultater. Prøv med et annet filter."

## Brukerhistorie 2

Som bruker  
Ønsker jeg å laste opp undervisningsopplegg

## Akseptansekriterier:

**Scenario 1:** bli rutet til side for registrering av nytt opplegg  
Gitt at jeg er pålogget som bruker  
Når jeg velger "Last opp nytt undervisningsopplegg" i dropdownmenyen  
Så skal jeg bli videresendt til siden for å lage et nytt undervisningsopplegg

**Scenario 2:** kunne velge fil for opplasting  
Gitt at jeg er på siden for å lage nytt undervisningsopplegg  
Når jeg trykker på "Velg fil"  
Så skal jeg få opp en Filutforsker-boks hvor jeg kan velge en fil fra min datamaskin

**Scenario 3:** legge inn tagger for å øke søkbarheten i applikasjonen  
Gitt at jeg er på siden for å lage et nytt undervisningsopplegg  
Når jeg trykker i checkboxen hvor det står "matematikk"  
Så skal checkboxen krysses av

**Scenario 4:** laste opp undervisningsopplegg  
Gitt at jeg har lagt inn informasjon i alle tekstfelter og checkboxene  
Og valgt en fil  
Når jeg trykker "Last opp"  
Så skal jeg bli sendt til listen over mine undervisningsopplegg  
Og få en beskjed om at "Din opplasting venter på godkjenning"

**Scenario 5:** tilbakemelding om manglende informasjon  
Gitt at jeg er på siden for å lage et nytt undervisningsopplegg  
Og jeg ikke har lagt inn informasjon i alle tekstfelter og checkboxer  
Når jeg trykker på "Last opp"  
Så skal jeg få feilmelding om "Manglende informasjon. Du må legge inn informasjon i alle tekstfelter og checkboxer"

## Brukerhistorie 3

Som bruker  
Ønsker jeg å søke blant eksisterende undervisningsopplegg

## Akseptansekriterier:

**Scenario 1:** søke blant alle undervisningsopplegg ved hjelp av søkefelt  
Gitt at jeg er på siden for oversikt over opplastede undervisningsopplegg  
Når jeg skriver "matematikk" i søkefeltet  
Så skal alle undervisningsopplegg som inneholder ordet "matematikk" i tittelen vises  
Og jeg skal få vite hvor mange søkeresultater som ble funnet

**Scenario 2:** få tilbakemelding om tomt søk  
Gitt at jeg er på siden for oversikten over opplastede undervisningsopplegg  
Og jeg skriver "dajrklga" i søkefeltet  
Så skal jeg få beskjed om "Ingen resultater funnet. Prøv med et annet søkeord."

## Brukerhistorie 4

Som bruker  
Ønsker jeg å logge inn og logge ut av applikasjonen

## Akseptansekriterier:

**Scenario 1:** logge inn med Google  
Gitt at jeg er på siden for innlogging  
Når jeg trykker på "Logg inn med Google"  
Så skal jeg få opp en popup som ber meg fylle ut påloggingsinformasjon for min Google-konto

**Scenario 2:** logge inn med applikasjonsbruker og passord  
Gitt at jeg er på siden for innlogging  
Og jeg har laget en bruker for applikasjonen  
Når jeg fyller ut brukernavn og passord  
Og trykker på Logg inn  
Så skal jeg bli logget inn i applikasjonen

**Scenario 3:** logge ut av applikasjonen  
Gitt at jeg er pålogget i applikasjonen  
Når jeg trykker på "Logg ut"  
Så skal jeg bli logget ut  
Og bli sendt til forsiden av applikasjonen

## Brukerhistorie 5

Som administrator av applikasjonen  
Ønsker jeg å godkjenne undervisningsopplegg  
Slik at de blir gjort tilgjengelige for nedlasting

## Akseptansekriterier:

**Scenario 1:** se liste over ikke godkjente undervisningsopplegg Gitt at jeg er pålogget som administrator  
Når jeg trykker på "Undervisningsopplegg til godkjenning"  
Så skal jeg få opp en liste over undervisningsopplegg som venter på godkjenning

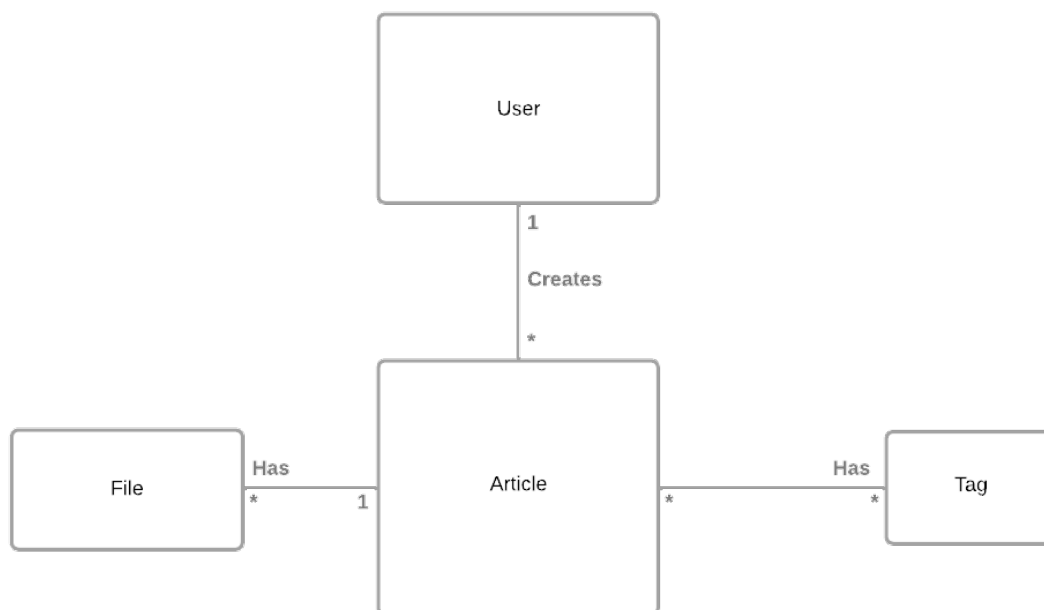
**Scenario 2:** godkjenne undervisningsopplegg  
Gitt at jeg er administrator for applikasjonen  
Og det er et opplegg i listen over undervisningsopplegg som venter på godkjenning  
Når jeg trykker på "Godkjenn undervisningsopplegg"  
Så skal opplegget gå fra status "venter på godkjenning" til "godkjent" Og vises i listen over alle opplastede undervisningsopplegg

Brukerhistorie 6
Som bruker Ønsker jeg å laste ned undervisningsopplegg
Akseptansekriterier:
<b>Scenario 1:</b> laste ned undervisningsopplegg Gitt at jeg er på siden som viser informasjon om et spesifikt undervisningopplegg Og jeg velger "Last ned" Så skal filene til undervisningsopplegget lastes ned til min lokale datamaskin

### 3 Domenemodell

Her har vi et enkelt konseptuelt overblikk over hvordan data i applikasjonen vil henge sammen. Brukere lager artikler som vil være knyttet opp mot deres konto, de kan lage så mange artikler de bare måtte ønske.

Hver artikkel får så bilder og filer knyttet til seg av brukeren samt. ulike tagger som kan brukes for å kjapt beskrive hva artikkelen handler om. Disse taggene kommer da i typene: Fagområde, tema, verktøy, og årstrinn.



Figur 1: Domenemodell for applikasjonen

## 4 Funksjonelle og ikke-funksjonelle krav

### 4.1 Funksjonelle krav

Konkrete funksjonelle krav til applikasjonen ble presentert i akseptansekriteriene i brukerhistoriene. Det ble også lagt frem en liste på funksjonelle egenskaper til produktet i visjonsdokumentet. I tillegg er det noen punkter det er hensiktsmessig å nevne når det gjelder funksjonelle krav.

Universell utforming er fra 1. januar 2021 et lovpålagt krav til alle IKT-løsninger i Norge (jfr. <https://www.digdir.no/felleslosninger/universell-utforming-av-ikt/1499> og <https://www.digdir.no/universell-utforming/universell-utforming-av-ikt/874>). Det vil derfor være et viktig krav til denne applikasjonen. Universell utforming betyr at de aller fleste skal kunne bruke applikasjonen på en tilfredsstillende måte, uavhengig av funksjonsevne. Dette omfatter aspekter som fargebruk, kontraster, tilgjengelighet på alternativ tekst, robusthet i applikasjonen når det gjelder tilgjengelighet i ulike nettlesere, samt at informasjonen skal være forståelig for brukerne. Det må også være mulig å betjene produktet uavhengig av måten brukeren navigerer i produktet. Dette gjelder f.eks. uten bruk av mus, med stemmekommando eller med skjermforstørrelser. WCAG-retningslinjene bør følges ved utviklingen av produktet, og nettsidene til ”Tilsynet for universell utforming av IKT” kan med fordel brukes.

For bruk av applikasjonen ved AIT/NTNU gjelder følgende funksjonelle krav, som tidligere nevnt i akseptansekriteriene:

- Brukere skal kunne laste opp undervisningsopplegg.
- Et undervisningsopplegg skal ikke vises offentlig før en administrator har godkjent det.
- Brukere skal kunne se alle sine egne undervisningsopplegg og se status om de er publisert eller ei.
- Administratorer skal kunne se alle undervisningsopplegg som ikke har blitt offentliggjort ennå i en kø/liste. Fra denne listen skal administratorer kunne godkjenne eller avslå undervisningsopplegg.
- Man skal kunne se en liste av alle undervisningsopplegg og søke/filtrere gjennom disse på alle attributter.

### 4.2 Ikke-funksjonelle krav

Et av de viktigste ikke-funksjonelle kravene som stilles til denne applikasjonen er brukervennlighet. Brukerne av dette produktet vil være lærere som kanskje egentlig ikke har tid eller overskudd til å lære programmering, men som er nødt til å lære det. Det har derfor vært et mål å gjøre applikasjonen så intuitiv som mulig, slik at brukerne kan fokusere på å lære programmering, ikke på å lære å bruke applikasjonen. Brukervennlighet er dog et vanskelig krav å måle. Det er derfor helt essensielt at det blir utført brukertesting av produktet, slik at man kan få en indikasjon på brukervennligheten til produktet.

Brukervennlighet forstås i dette prosjektet som følger:

- Produktet må ha god søkbarhet. Det skal være enkelt å sortere og filtrere undervisningsopplegg slik at lærerne kan finne et godt egnet opplegg å bruke i undervisningen i grunnskolen.
- Produktet må være pålitelig og robust. Det skal fungere som det skal, når det skal. Det er viktig å begrense nedetiden så mye som mulig. For å gjøre produktet robust i forhold til bruken av det, bør man minimere risikoen for brukerfeil. Dette kan gjøres ved å eliminere mulighetene for feilbruk, f.eks. ved å skjule knapper som ikke er i bruk og ved å gi gode instruksjoner og tilbakemeldinger til brukeren. Det kan f.eks. legges inn små hint ved opprettelse av undervisningsopplegg, som forteller brukeren at ”jo mer detaljert informasjon

du legger inn om undervisningsopplegget, jo lettere vil det være for dine kollegaer å finne og bruke ditt opplegg”.

Et annet viktig ikke-funksjonelt krav for applikasjonen er at det følger personopplysningsloven og forskriftene til den. Dette handler om brukerens rett til å bli glemt, rett til å vite hva som lagres og hvordan det lagres. Det bør bli lagt inn en side som forteller om hvordan personvernet til brukerne blir ivaretatt. Dette vil ikke bli gjort i denne versjonen av applikasjonen, men vil bli nevnt under videre utvidelser av applikasjonen.

- Godt dokumentert kildekode med gode rutiner for å unngå teknisk gjeld og bra muligheter for enkel utvidelse.
- REST-standarder følges (navngivning av ruter, ordentlig bruk av verb, osv.)
- Applikasjonen skal kreve minimalt med oppsett hos de som ønsker å bruke dem (dvs. ikke stort mer enn å kjøre ‘npm start‘)

### 4.3 Videre utvidelser

Når det gjelder videre utvidelser av applikasjonen, så er det noen punkter som vi har vurdert til å gå utenfor scopet for denne oppgaven. Dette er funksjoner som kan være nyttige å implementere i en senere utvidelse av applikasjonen.

- Legge opp til mulighet for innlogging med FEIDE. Applikasjonen vil eies av AIT ved NTNU, og brukerne som skal laste opp undervisningsmateriale vil i første omgang være studenter med FEIDE-brukere.
- Åpne opp for kommentarer på undervisningsopplegg.
- Ratingsystem for opplegg. I den anledning vil man også kunne sette opp sortering av opplegg basert på rating.
- Legge inn en side for personvern, hvor brukeren får informasjon om hvordan personvern blir ivaretatt.
- Firebase Cloud Storage for lagring av data. Per nå lagres data i local storage, men for å sikre skalerbarhet i applikasjonen vil det være hensiktsmessig å flytte lagring over på sky. Firebase brukes allerede for autentisering, så det ville være logisk å ta i bruk Firebase Cloud Storage i tillegg.

For å gjøre det enklere å administrere applikasjonen og brukertyper kan det også være nyttig å legge inn følgende funksjonalitet i en videreutvikling av applikasjonen:

- En administrator skal kunne gjøre andre brukere til administratorer. Dette bør kunne gjøres i frontend.
- Sette inn flere roller/brukertyper. Ett eksempel som kan være nyttig er å legge inn brukertypen ”moderator”, som har færre rettigheter enn administrator men som kan godkjenne og avslå opplegg.

## 5 Prototyper

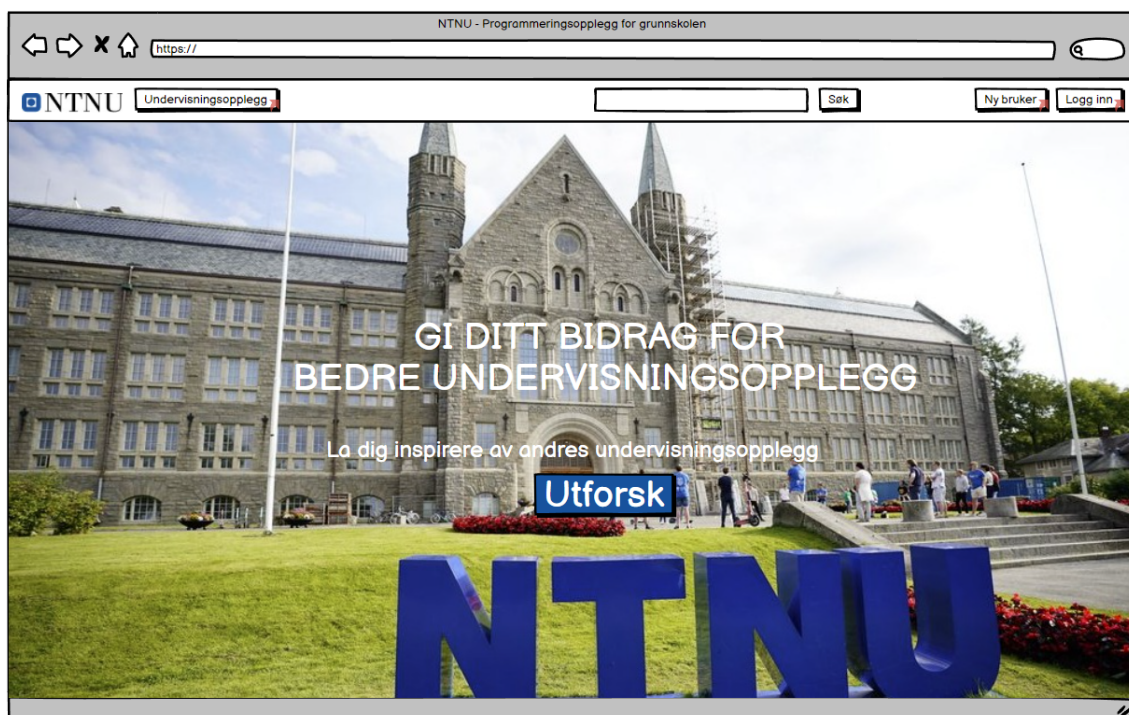
### 5.1 Wireframes

Forsiden til den eksisterende applikasjonen vil til store deler beholdes som den er. Figur 2 viser den eksisterende applikasjonen mens figur 3 viser wireframe for endringene som skal gjøres. Bakgrunnsfargen i header vil bli endret til hvit, og NTNU logo vil bli lagt inn i venstre kant. Dette for

å gjøre applikasjonen mer lik NTNU sin nåværende nettside. Det blir også satt inn en knapp "Undervisningsopplegg" ved siden av NTNU logo, som erstatter de eksisterende dropdown-knappene i den eksisterende applikasjonen. Ved å trykke på "Undervisningsopplegg" kommer man til en oversikt over alle undervisningsopplegg.



Figur 2: Eksisterende applikasjon - forside

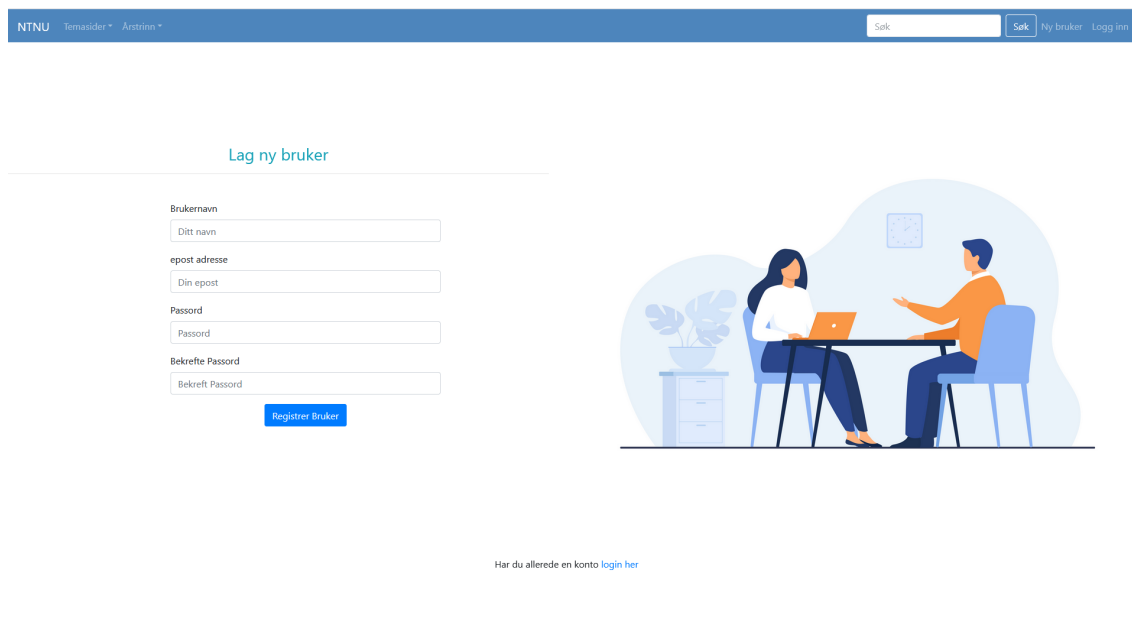


Figur 3: Wireframe - Ny forside

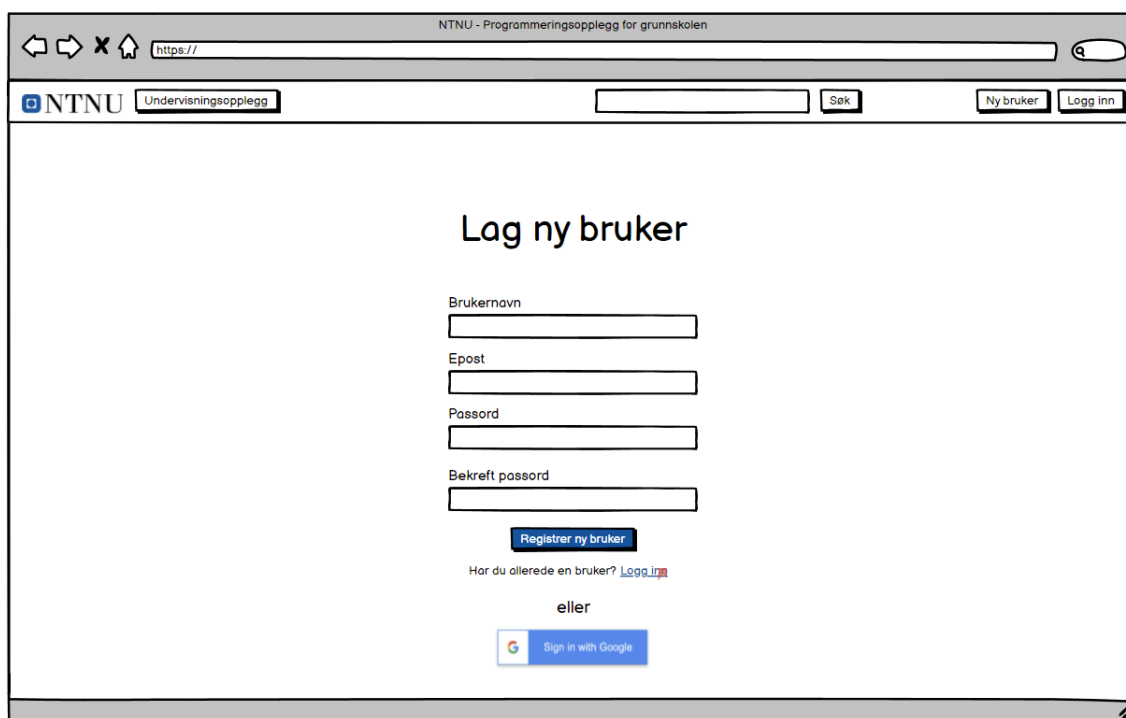
Siden for å lage ny bruker vil bli endret. Figur 4 viser den eksisterende applikasjonen mens figur 5



viser wireframe for ny side for å lage ny bruker. Den eksisterende applikasjonen har tekstfeltene for påloggingsinfo til venstre på siden, og et bilde til høyre. Dette vil bli justert slik at bildet fjernes og tekstfeltene sentreres i bildet. I tillegg blir det lagt til en knapp for pålogging via Google.

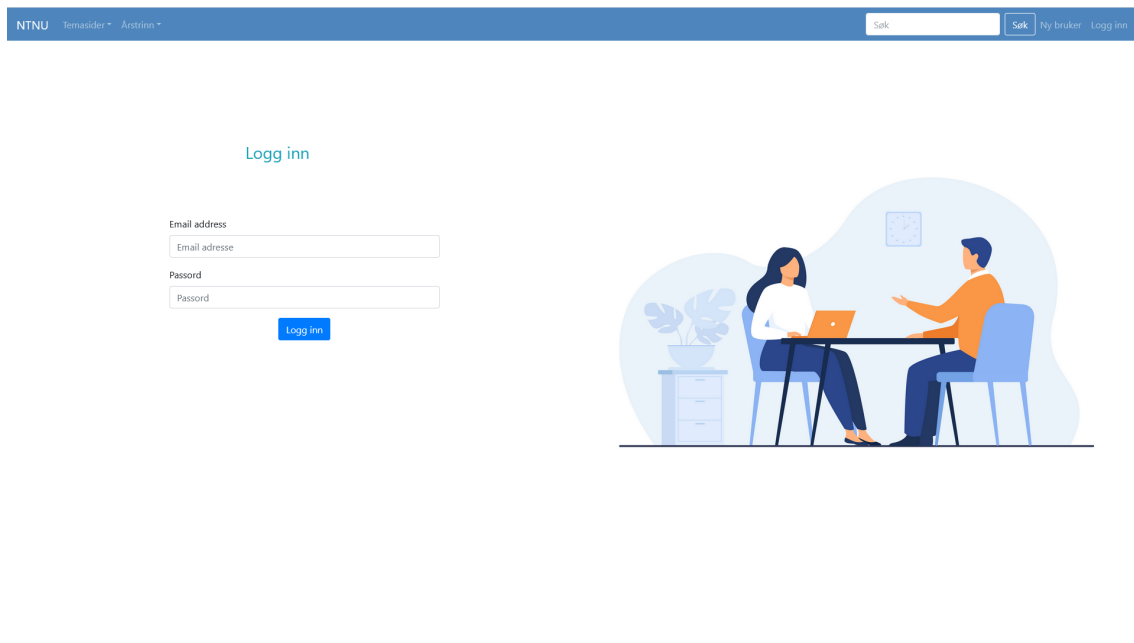


Figur 4: Eksisterende applikasjon - Lag ny bruker

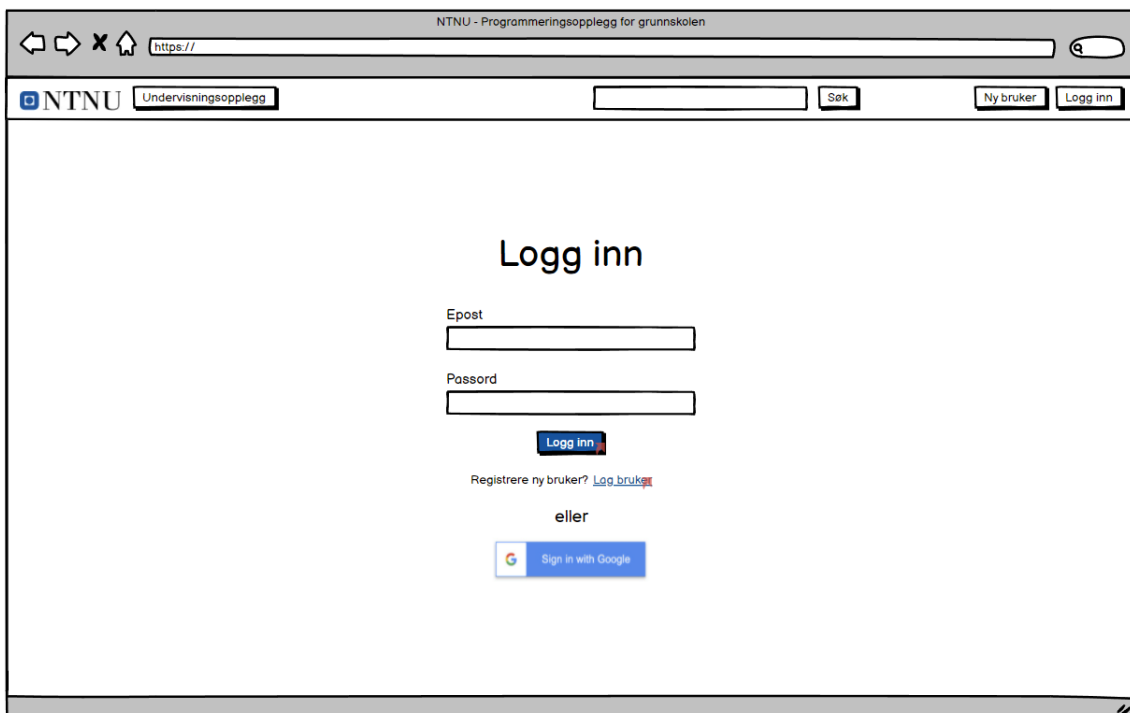


Figur 5: Wireframe - Ny side for å lage ny bruker

Siden for login vil endres på tilsvarende måte som siden for å lage ny bruker. Figur 6 viser eksisterende applikasjon mens figur 7 viser wireframe for ny side for å logge inn.



Figur 6: Eksisterende applikasjon - Logg inn



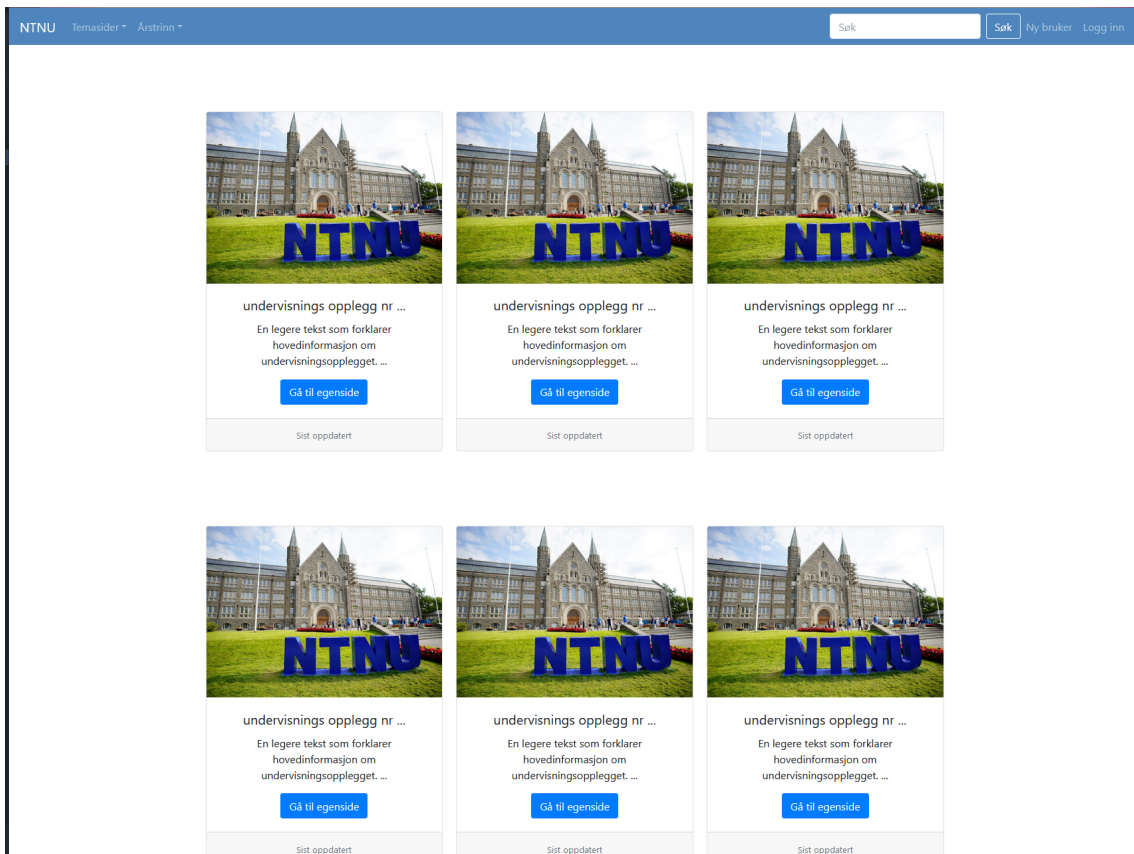
Figur 7: Wireframe - Ny side for å logge inn

Figur 8 viser forsiden slik den vil se ut for en administrator etter pålogging. Det blir lagt inn en dropdown-liste til høyre for søkefeltet, hvor brukernavnet står. I dropdown-listen vil man i tillegg kunne velge "Mine opplegg", "Legg til nytt opplegg" og "Rediger bruker". Forskjellen fra en vanlig bruker er at administrator har en ekstra knapp oppe til venstre i header, som heter "Undervisningsopplegg til godkjenning". Ved å trykke på denne kommer administrator til en side hvor de undervisningsopplegg som trenger godkjenning kommer opp.



Figur 8: Wireframe - Administrator pålogget

Figur 9 viser hvordan den eksisterende applikasjonens side for "alle undervisningsopplegg" ser ut. Her vil det bli gjort en forandring når det gjelder søkbarhet. Figur 10 viser wireframe for hvordan den nye siden for "alle undervisningsopplegg" vil se ut. Det blir lagt inn en kolonne til venstre på siden, hvor man kan filtrere søkeresultatene basert på årstrinn, fagområde, verktøy, filtype og estimert tidsbruk for opplegget. Det vil også bli lagt inn en dropdown-liste for sortering av resultatene basert på dato lagt til.



Figur 9: Eksisterende applikasjon - Alle undervisningsopplegg



Figur 10: Wireframe - ”ny side for alle undervisningsopplegg”

Figur 11 viser wireframe over søkeresultat. Hvis søketeksten matcher ord i undervisningsopplegg vil disse vises som resultater. Hvis det ikke blir funnet noen resultater, vil brukeren få beskjed om det.



Figur 11: Wireframe - Søkeresultat ikke funnet

Figur 12 viser den eksisterende siden for å lage et nytt undervisningsopplegg, og figur 13 viser wireframe for ny side for å lage et nytt opplegg. Her vil det bli lagt inn checkboxer i stedet for tekstfelt for årstrinn, fagområde og verktøy som brukes. I tillegg blir det lagt inn en checkbox for estimert tidsbruk for applikasjonen. Det som hukes av i checkboxene blir lagt inn som tagger ved opplasting av nytt opplegg, og taggene vil være tilgjengelige i form av filtreringsmulighet på siden for alle undervisningsopplegg.

The screenshot shows a web application interface for creating a new assignment. The header includes the NTNU logo, navigation links for 'Temasider' and 'Arstrinn', a search bar with the text 'Søk', and user options for 'Søk', 'HovedMeny', and 'Logg ut'. The main content area is divided into two columns. The left column contains three sections: 'Last opp bilde:' with a 'Browse...' button and the text 'No file selected.'; 'Last opp filer:' with a 'Browse...' button and the text 'No files selected.'; and 'Verktøy som brukes:' with a large empty text input field. The right column contains four sections: 'Oppgavetittel' with a text input field; 'Fagkode' with a text input field; 'Arstrinn' with a dropdown menu showing 'ingen årstrinn'; and 'Tema' with a text input field. Below the input fields are two blue buttons: 'Lagre' and 'Tilbake'.

Figur 12: Eksisterende applikasjon - Lag nytt undervisningsopplegg

The wireframe shows a web application interface for creating a new assignment. The browser address bar displays 'https://'. The page title is 'NTNU - Programmeringsopplegg for grunnskolen'. The header includes the NTNU logo, a search bar with the text 'Søk', and user options for 'Brukernavn' and 'Logg ut'. The main content area is titled 'Legg til nytt undervisningsopplegg'. The left column contains four sections: 'Oppleggstittel' with a text input field; 'Passer for årstrinn' with a list of 'Klasse 1' and 'Klasse 2' and checkboxes; 'Fagområde' with a list of 'Matematikk' and 'Fysikk' and checkboxes; and 'Verktøy som brukes' with a list of 'Python', 'JavaScript', and 'Annet hva' and checkboxes. The right column contains three sections: 'Legg til bilde:' with a 'Browse...' button and the text 'Ingen filer valgt.'; 'Legg til filer:' with a 'Browse...' button and the text 'Ingen filer valgt.'; and 'Estimert tidsbruk for opplegget' with a list of '1 time' and '2 timer' and checkboxes. Below the input fields are two blue buttons: 'Last opp' and 'Tilbake'.

Figur 13: Wireframe - Lag nytt undervisningsopplegg

## C Systemdokumentasjon



INSTITUTT FOR DATATEKNOLOGI OG INFORMATIKK

INFT2900 - BACHELOROPPGAVE I INFORMASJONSBEHANDLING

---

# Videreutvikling av webapplikasjon for undervisningsopplegg

---

SYSTEMDOKUMENTASJON

*Forfattere:*

Henna Fredrika Björkskog

Truls Henrik Jakobsen

Versjon 1.1

# Innhold

<b>1 Innledning</b>	<b>2</b>
<b>2 Arkitektur</b>	<b>2</b>
<b>3 Prosjektstruktur</b>	<b>2</b>
3.1 Frontend . . . . .	2
3.1.1 Rotmappe . . . . .	2
3.1.2 cypress . . . . .	3
3.1.3 public . . . . .	3
3.1.4 src . . . . .	4
3.1.5 Components . . . . .	4
3.2 Backend . . . . .	5
3.2.1 Rotmappe . . . . .	5
3.2.2 artifacts . . . . .	6
3.2.3 certificate . . . . .	6
3.2.4 src . . . . .	6
3.2.5 __test__ . . . . .	7
3.2.6 database . . . . .	7
3.2.7 routes . . . . .	8
3.2.8 controllers . . . . .	8
3.2.9 services . . . . .	9
3.2.10 types . . . . .	9
3.2.11 utils . . . . .	9
<b>4 Klassediagram</b>	<b>10</b>
<b>5 Databasemodell</b>	<b>10</b>
<b>6 Server-tjenester</b>	<b>11</b>
<b>7 Sikkerhet</b>	<b>11</b>
7.1 HTTPS . . . . .	11
7.2 Autorisasjon . . . . .	11
<b>8 Installasjon og kjøring</b>	<b>11</b>
<b>9 Dokumentasjon av kildekode</b>	<b>12</b>



<b>10 Testing</b>	<b>12</b>
10.1 Frontend . . . . .	12
10.2 Backend . . . . .	12

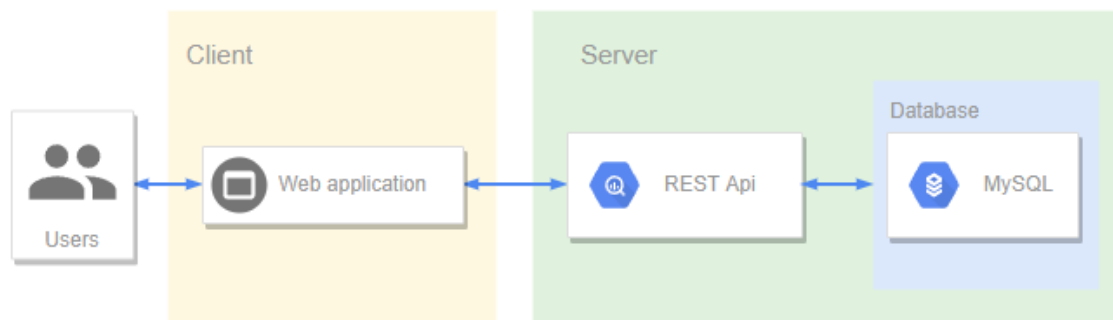
## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
7 april 2022	0.1	Første utkast av systemdokumentasjonen	Henna F. Bjørkskog og Truls H. Jakobsen
15. mai 2022	1.0	Utfyllende informasjon	Truls Henrik Jakobsen

# 1 Innledning

Denne systemdokumentasjonen er fremstilt i forbindelse med prosjektet ”Videreutvikling av webapplikasjon for undervisningsopplegg”, som utføres som bacheloroppgave ved NTNU våren 2022. Hensikten med dokumentet er å gi en teknisk presentasjon av systemet og gi leseren en fullstendig forståelse av hvordan systemet er satt sammen. Dokumentet tar for seg systemarkitektur og prosjektstruktur, databasemodell og klassediagram, dokumentasjon av kildekode samt en oversikt over server-tjenester og avhengigheter som kreves ved installasjon og kjøring av applikasjonen. I tillegg gis det også informasjon om kontinuerlig integrasjon og testing av applikasjonen.

# 2 Arkitektur



Figur 1: Systemarkitektur

Figur 1 viser hvordan arkitekturen til systemet er satt opp. Applikasjonen kjører som en webapplikasjon i en nettleser. En todelt arkitektur med klient og tjener sørger for kommunikasjonen mellom webapplikasjonen og databasen som lagrer informasjonen til applikasjonen. Kallene til databasen håndteres med Rest API.

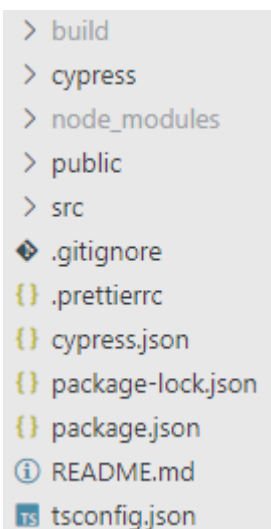
# 3 Prosjektstruktur

Her går vi gjennom hvordan prosjektet er strukturert og hva de ulike mappene inneholder. Vi nevner også de viktigste filene litt mer spesifikt.

## 3.1 Frontend

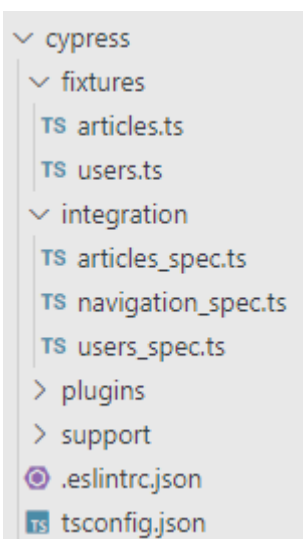
### 3.1.1 Rotmappe

Dette er rotmappen til frontend av prosjektet. Her ligger det litt forskjellige konfigurasjonsfiler og meta-data som f.eks **tsconfig.json**, **.gitignore**, **package.json**, og **cypress.json**.



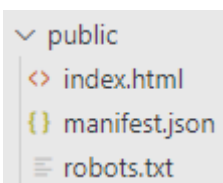
### 3.1.2 cypress

I denne mappen ligger alle filene som har med testing å gjøre. I **integration** ligger *specs* hvor vi definerer test-suiter som cypress går gjennom når den kjører alle testene. **fixtures** inneholder mock-data som kan brukes i testene, f.eks. falske brukere eller artikler, og hjelpefunksjoner.



### 3.1.3 public

**public** inneholder alt som skal eksponeres til offentligheten og være tilgjengelig via url. Vanligvis vil dette bilder osv, men bildene vi bruker lastes inn i JSX på forhånd så de ligger i **src** istedenfor.

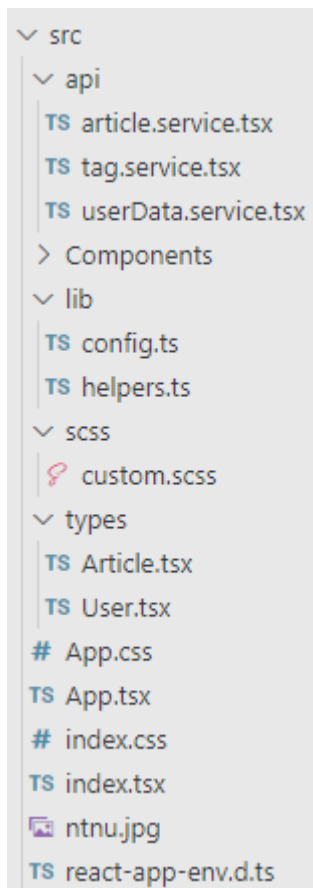


### 3.1.4 src

Tilnærmet all kildekoden ligger i **src**. Vi har delt filer som renderer ting til skjermen og filer som henter data fra API i mappene **api** og **components**.

I **api** har vi *.service.ts* filer som er koblingen mellom backend og komponentene som gjør ting på skjermen hos brukeren. Disse servicene eksporterer diverse asynkrone funksjoner som heter f.eks. `fetchArticles()` som henter data fra backend API'et.

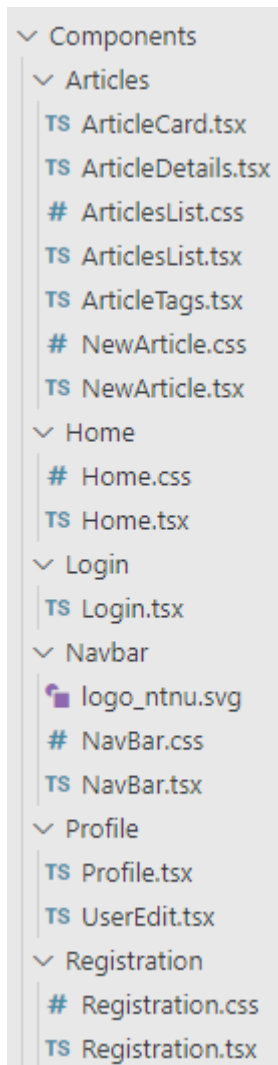
**lib** inneholder to filer. *config.ts* definerer noen globale konstanter som kan brukes i diverse andre steder i applikasjonen slik at man trenger bare endre dem på ett sted. *helpers.ts* eksporterer hjelpefunksjoner som har logikk som gjerne brukes i flere komponenter, eller store funksjoner som ville blitt rotete å ha i én komponent.



### 3.1.5 Components

**Components** er kanskje den viktigste mappen i **src**. Den inneholder alle komponenter som viser ting på skjermen til brukeren. Hver mappe her inne har relaterte komponenter inne i seg og deres tilhørende *.css* filer (hvis de finnes).

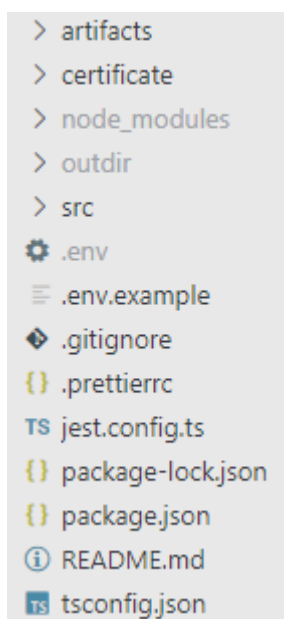
En forbedring vi gjerne kunne gjort her er å bryte ting mer opp og skille mellom hele *pages* og bare enkle *components*. Dette kunne blitt gjort ved å ha en egen mappe som het **pages** og hatt de større komponentene som bruker andre komponenter der. Et eksempel på noe som gjerne skulle vært en *page* og ikke en *komponent* er **ArticleList.tsx**.



## 3.2 Backend

### 3.2.1 Rotmappe

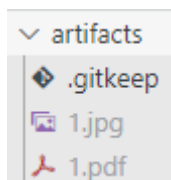
I rotmappen til backend ligger selvsagt alle de andre mappene innover, men også konfigurasjonsfiler som `tsconfig.json`, `.gitignore`, `package.json`, og `jest.config.ts`.



### 3.2.2 artifacts

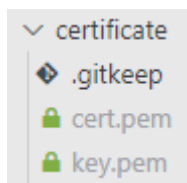
Denne mappen er hvor alle filer som blir lastet opp med artikler blir lagret. De får en hash som filnavn og blir lagret med riktig file-extension (jpg, pdf, osv...). Når noen ber om et spesifikt bilde lager vi pathen til bilde on the fly med å hente ut hash og filnavn fra databasen.

Mappen inneholder en **.gitkeep** fil fordi *git* ikke kan indeksere tomme mapper. Den filen har altså kun i funksjon at vi skal kunne la være å sjekke andre filer fra **artifacts** inn i git, men fremdeles beholder mappen.



### 3.2.3 certificate

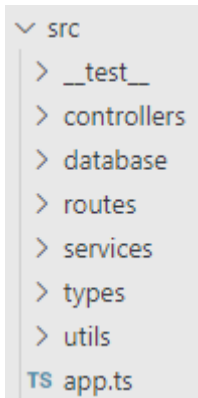
Her ligger sertifikat og nøkkel som skal til for at vi kan bruke HTTPS istedenfor HTTP når vi sender data. Disse filene blir ikke sjekket inn i *git* og vi har derfor en **.gitkeep** fil her også sann som i **artifacts**.



### 3.2.4 src

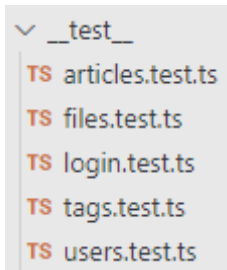
Ganske likt som i frontend så er det i **src** det skjer. Her er man ganske gode på å fordele ulike typer ansvar til mapper, disse dekkes i de neste punktene.

Kanskje den viktigste filen i hele applikasjonen ligger også her, **app.ts**. Her i denne filen blir HTTPS serveren initialisert, vi starter databasetilkoblingen, og vi konfigurerer en del ting her også. Bl.a. blir alle API-rutene hentet lastet inn her også.



### 3.2.5 \_\_test\_\_

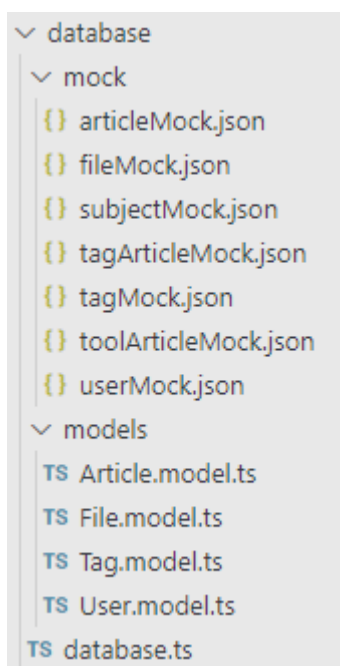
Her ligger alle test filene for *Jest*.



### 3.2.6 database

I **models** definerer vi alle objektene vi ønsker å ha i databasen ved hjelp av *Sequelize*. **mock** inneholder diverse *.json* filer med mock-data vi kan fylle databasen med slik at vi kan teste hvordan applikasjonen ser ut med ”ekte” data. **database.ts** inneholder en funksjon som gjør fullt oppsett av databasen og setter inn mock-data basert på environment variables satt i **.env**.



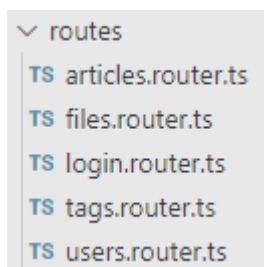


### 3.2.7 routes

I hver *.router.ts* fil definerer vi diverse API ruter som er relatert til en ressurs. **articles.router.ts** inneholder altså alle rutene som har med artikler og gjøre og sender forespørsler som kommer inn til de rutene videre til *controlleren* for den ressursen.

Hvis det kommer en GET forespørsel til `/articles` vil altså *articles.router.ts* sende forespørselen videre til *articles.controller.ts*.

Dette gjøres fordi vi ikke ønsker at controlleren skal trenge å vite hvilke ruter API'et bruker, det skal kun holde styr på *hva* vi kan gjøre med hver enkelt ressurs.



### 3.2.8 controllers

I *.controller.ts* filene ligger *controllere* for hver ressurs i backend. Disse controllerene holder styring på hva vi kan gjøre med disse ressursene, f.eks opprette, hente, slette, eller redigere en artikkel.

Den trenger ikke vite hvor forespørslene kommer fra eller hvordan disse operasjonene faktisk utføres i databasen, den trenger bare holde styr på at dette faktisk skjer når vi ber om det.

Selve implementasjonen av operasjonene er delegert til *services*.

```
▼ controllers
  TS article.controller.ts
  TS file.controller.ts
  TS login.controller.ts
  TS tag.controller.ts
  TS user.controller.ts
```

### 3.2.9 services

Her ligger alle implementasjonene av hvordan data blir hentet ut, redigert, eller slettet i databasen. Grunnen til at dette er ganske ålreit er at hvis vi skulle bytte fra sequelize til en annen ORM løsning, så trenger vi bare skrive om en *service* og ikke all den andre logikken rundt ressursene våre.

```
▼ services
  TS article.service.ts
  TS file.service.ts
  TS tag.service.ts
  TS user.service.ts
```

### 3.2.10 types

I denne mappen har vi alle typer som blir brukt rundt i applikasjonen. For de fleste ressursene har vi definert *DTO's* eller Data Transfer Objects. Disse er utrolig greie å bruke når modellen i databasen inneholder noen felter du ikke ønsker å sende til sluttbrukere. F.eks har vi ikke lyst å sende passordhashen til en bruker ut via API'et. Da kan vi hente ut et objekt fra databasen og så mappe det til en DTO slik at kun feltene vi ønsker kommer ut til verden.

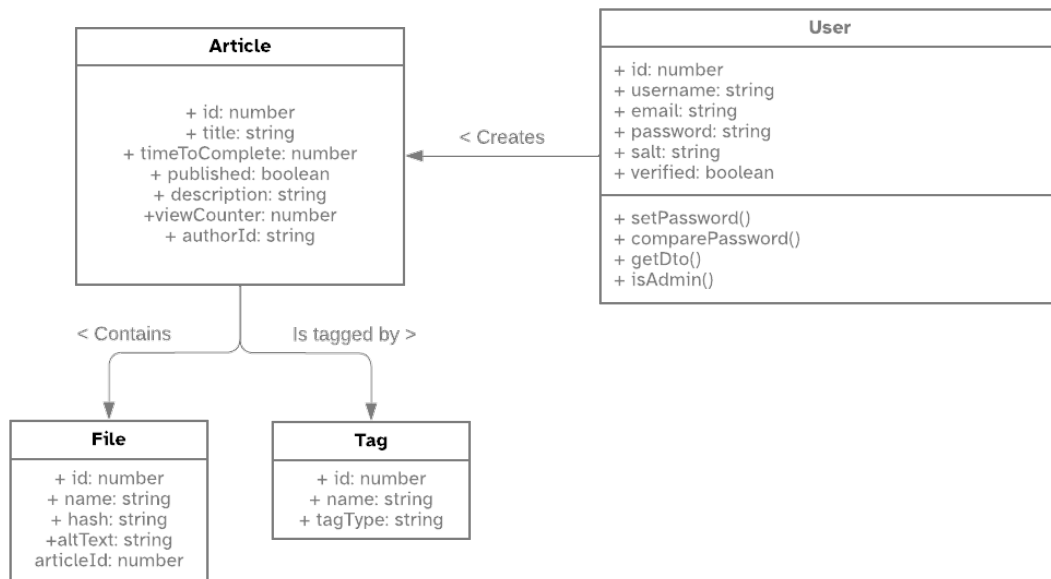
```
▼ types
  TS ArticleDto.ts
  TS FileDto.ts
  TS TagDto.ts
  TS UserDto.ts
```

### 3.2.11 utils

Her ligger kun **helper.ts** til dags dato. Den eksporterer kun én funksjon og det er en hjelpefunksjon som returnerer et databaseinstans slik at vi kan bruke det samme instanset over hele applikasjonen. Dersom vi noensinne trenger noen flere hjelpefunksjoner vil det være logisk å legge disse her.

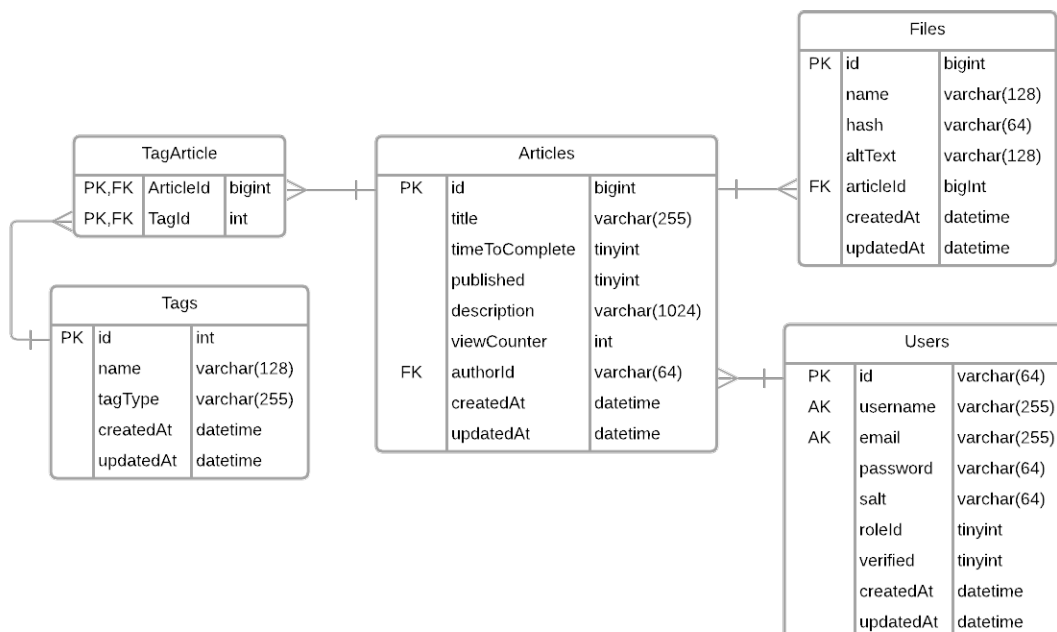
```
▼ utils
  TS helper.ts
```

## 4 Klassediagram



Figur 2: Klassediagram for applikasjonen

## 5 Databasemodell



Figur 3: Databasemodell for applikasjonen

Det er opprettet en relasjonsdatabase i MySQL for lagring av informasjon i applikasjonen. Figur 3 viser oversikten over denne. Her vises relasjonen mellom undervisningsoppleggene som lastes opp, brukerne som laster opp undervisningsopplegg og filene og bildene som lastes opp. I tillegg illustreres hvordan det i hvert enkelt undervisningsopplegg defineres fagområde, verktøy som brukes og årstrinn undervisningsopplegget passer for.

## 6 Server-tjenester

Serveren tilbyr en del ulike REST Paths, men det er egentlig så enkelt som CRUD-operasjoner for hver av de ulike objektene i klassediagrammet (Articles, Tags, Files, Users).

Mer utdypende dokumentasjon kan du finne ved å spinne opp backend og dra til `localhost:8080/api-docs`. Der skal det være API-dokumentasjon ved hjelp av Swagger.

## 7 Sikkerhet

### 7.1 HTTPS

Ettersom vi kommer til å sende passord frem og tilbake over nett er HTTPS nesten påbudt. Når man skal sette opp applikasjonen er man nødt til å generere sine egne SSL sertifikater. Dette kan greit gjøres manuelt med OpenSSL eller lignende, men det går også an å kjøre `npm run cert` som har en grei kommando for deg.

Sertifikatene ligger i `./certificate` og er to filer: `cert.pem` og `key.pem`.

### 7.2 Autorisasjon

Når brukere oppretter en konto i applikasjonen blir valgt brukernavn, epost, og passord sendt til serveren. Der blir det først sjekket om det finnes en bruker med samme brukernavn eller epost fra før. Dersom det ikke gjør det vil det genereres et salt som hashes sammen med passordet ved hjelp av SHA256. Saltet og hashen blir deretter lagret i databasen sammen med den nye brukerinformasjonen.

Når en bruker skal logge seg inn blir epost og passord sendt til serveren hvor vi henter ut saltet til eposten som ble skrevet inn og forsøker å hashe passordet sammen med det saltet. Deretter sammenligner vi den nye hashen med hashen som er lagret på brukeren og gir grønt lys hvis de matcher. Her er det forbedringspotensiale for å generere en session token og sende den til klienten, men til dags dato sender vi bare brukerens ID hvor den blir lagret i nettleserens localStorage.

Dersom en ondsinnet bruker skulle ønske kan de i teorien hente ut en artikkel og få tak i authorID som de kan sette i localStorage for å være "logget inn" som den brukeren.

Når man setter nytt passord så følger man samme prosess som ved registrering, nytt salt blir generert og passord blir hashet sammen med det og hash lagres.

## 8 Installasjon og kjøring

For å installere og kjøre applikasjonen er det best å kloner fra Github eller laste ned manuelt for så å følge instruksjoner i README videre.

## 9 Dokumentasjon av kildekode

Lenker til Github repositories hvor du kan finne all kildekoden og README:

Frontend: <https://github.com/trulshj/programming-syllabus-application-frontend>

Backend: <https://github.com/trulshj/programming-syllabus-application-backend>

Det finnes ingen utdypende dokumentasjon for frontend, men for backend henviser vi til swagger-dokumentasjonen som ble nevnt i 6. Server-tjenester.

## 10 Testing

### 10.1 Frontend

I frontend har vi integrasjonstester ved hjelp av Cypress. For å få testene til å kjøre må applikasjonen kjøre i bakgrunnen. Letteste måten å gjøre dette er `npm run start` i en egen terminal for så å deretter kjøre `npm run test` i et annet vindu. Testene her går gjennom de vanlige tingene som en bruker ville gjort som å lage bruker, logge inn, filtrere og sortere artikler, og lage nye artikler mm.

### 10.2 Backend

I backend kjører vi testing med Jest. Testingen gjøres på produksjons-builds av applikasjonen, derfor vil `npm run test` først bygge applikasjonen og deretter kjøre Jest kommandoen. Vi har en test-suite for hver service som kjører gjennom de essensielle funksjonene for servicen. Siden vi ikke har noen lokale database-instanser har vi ikke CI på backend ettersom NTNU sin database-server krever VPN for å være koblet til.

## D Testplan

Et av de viktigste suksesskriteriene ved applikasjonen er brukervennlighet. Brukertesting er derfor et viktig element i prosjektet. Hensikten med brukertestene er å få tilbakemelding på funksjonalitet og layout i applikasjonen, for å kunne forbedre produktet eller videreutviklinger av produktet på bakgrunn av tilbakemeldingene. Det var også et ønske å få generell tilbakemelding fra testpersonene om behovet for et slikt produkt, og hvilke aspekter og hvilken funksjonalitet som bør være med for at produktet ville bli brukt av personer innenfor målgruppen. Det ble vurdert som viktig at testpersonene for brukertesting av applikasjonen tilhører målgruppen for produktet.

### D.1 Testplan

1. **Formål med brukertesting:** Formålet med brukertesting er å finne ut hvor brukervennlig produktet oppleves og om den opplevde gevinsten er god nok til at de ville tatt produktet i bruk. I tillegg ønsker vi også å se hvordan brukere navigerer i applikasjonen, og på den måten kunne tilpasse produktet på best mulig måte.
2. **Funksjonalitet som skal testes:** Navigering i applikasjonen, søkefunksjonaliteten til applikasjonen og brukerhåndtering (opprettelse, på- og utlogging).
3. **Systemet som skal testes:** *Prosjekt Programmeringsreform* - En webbapplikasjon for undervisningsopplegg i programmering som er utviklet som bacheloroppgave i Informasjonsbehandling ved NTNU.
4. **Testbrukere:** Lærere i den norske skolen. Testteamet tok kontakt med Ryenberget Skole i Oslo, hvor tre lærere på ungdomstrinnene stilte opp som testpersoner.
5. **Testlokale:** Arbeidsplassen til testbrukerne.
6. **Testutstyr:** Testbrukerne vil ved testgjennomføringen bruke en laptop som testteamet har med seg. Denne laptopen har VPN for ekstern tilkobling til NTNU-nettet for tilgang til databasen.
7. **Testoppgaver til testbrukerne:** Oppgavene blir presentert i del 2 Oppgaver.
8. **Tidsplan for testgjennomføring:** Testgjennomføringen er ventet å ta ca 15 minutter per testbruker.
9. **Spørsmål til testbrukere før og etter testing:** Spørsmålene blir presentert i del 3 Spørsmål.
10. **Testteamet:** Testteamet består av gruppen som utvikler applikasjonen. Grunnet geografisk avstand vil det kun være én person fra testteamet tilstede ved enhver testsituasjon.
11. **Formidling av funnene i testen:** Funnene fra brukertestingen vil bli en del av grunnlaget for diskusjonen i hovedrapporten.
12. **Videre aksjonspunkter basert på resultater fra testen:** Videre aksjonspunkter som blir aktualisert som følge av brukertestene vil bli nedskrevet og presentert for oppdragsgiver. Disse vil kunne brukes ved en eventuell videreutvikling av applikasjonen.

### D.2 Oppgaver

1. Finn fram til oversikten over undervisningsopplegg.
2. Søk etter et undervisningsopplegg som passer for 7. klasse matematikk.
3. Last ned filene som hører til undervisningsopplegget.
4. Lag en bruker.
5. Logg inn i applikasjonen.

6. Legg til et nytt undervisningsopplegg:

- **Tittel:** Testopplegg brukertesting
- **Beskrivelse:** Dette er et opplegg som lastes opp i forbindelse med brukertesting
- **Klassetrinn:** 5. klasse
- **Fag:** Matematikk P
- **Verktøy:** Python
- **Thumbnail og filer:** Fra mappen i desktop som heter "Brukertestingfiler", sett inn "fox.jpeg" som thumbnail og "tekstfil.txt" som fil.

7. Logg ut av applikasjonen.

### D.3 Spørsmål

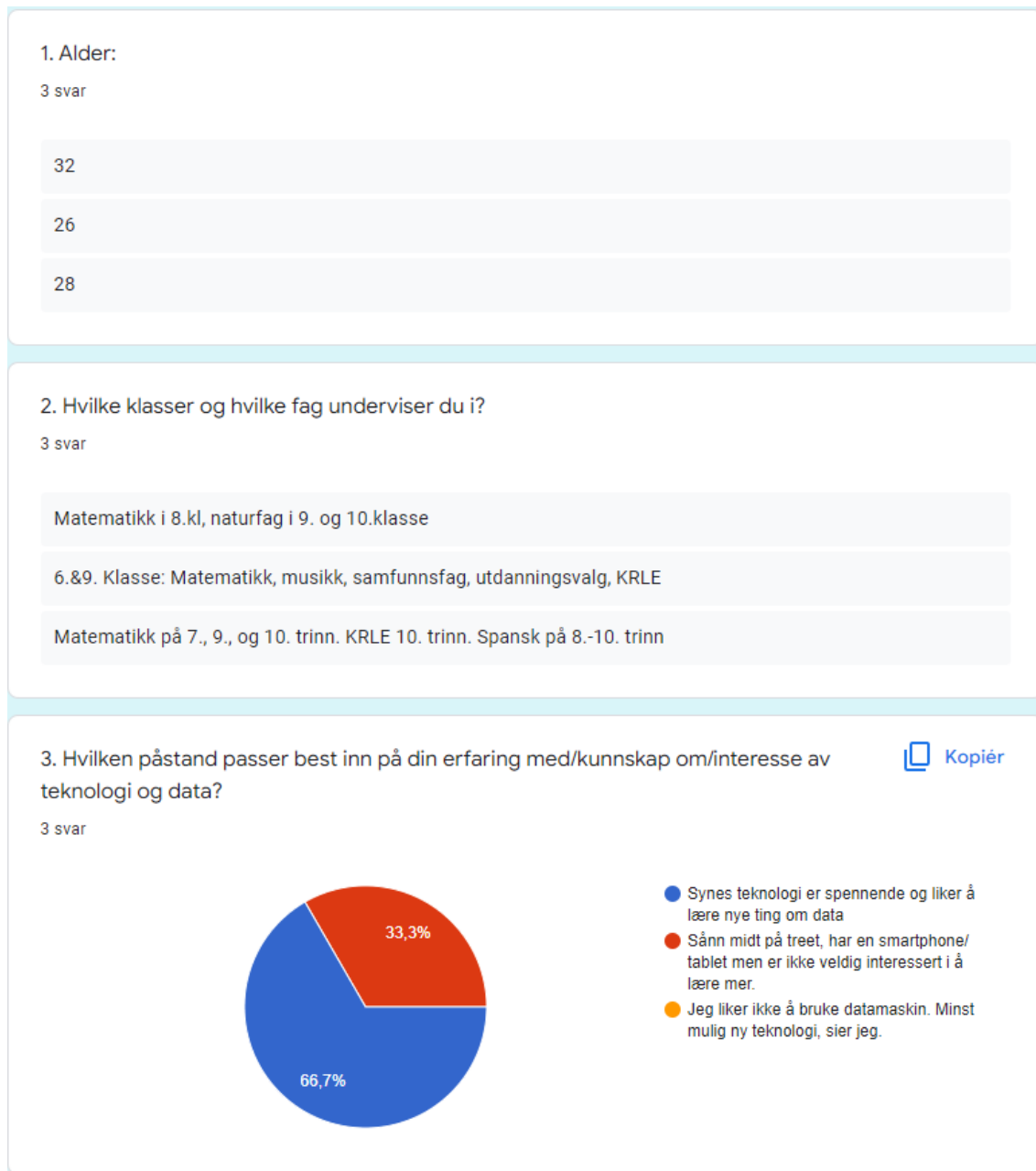
- Spørsmål som stilles før testen gjennomføres:
  - Alder?
  - Hvilke klasser og hvilke fag underviser du i?
  - Hvilken påstand passer best inn på din erfaring med/kunnskap om/interesse av teknologi og data?
    - \* Synes teknologi er spennende og liker å lære nye ting om data
    - \* Sånn midt på treet, har en smartphone/tablet men er ikke veldig interessert i å lære mer.
    - \* Jeg liker ikke å bruke datamaskin. Minst mulig ny teknologi, sier jeg.
  - Har du noen gang undervist i programmering?
    - \* Ja
    - \* Nei
    - \* Skal undervise i programmering, men har ikke gjort det ennå.
  - Dersom du hadde vært på leting etter et undervisningsopplegg som passer for 7. klasse matematikk, hvor ville du lett etter informasjon?
  - Hvor enkelt/vanskelig synes du det er å finne god informasjon på internett?
    - \* Veldig enkelt
    - \* Enkelt
    - \* Nøytral
    - \* Vanskelig
    - \* Veldig vanskelig
  - Har du vært borti et lignende produkt som dette tidligere?
  - Hva forventer du at et slikt produkt skal inneholde?
- Spørsmål som stilles etter at testen er gjennomført:
  - Hva er ditt helhetsinntrykk av produktet?
  - Dersom du hadde vært på leting etter et undervisningsopplegg for programmering, ville du brukt dette produktet?
    - \* Ja
    - \* Nei
  - Basert på ditt svar i forrige spørsmål, hvorfor/ hvorfor ikke?
  - Følte du at noe mangler i applikasjonen?
  - Navigering i applikasjonen: var ting der du forventet at de skulle være?
    - \* Ja
    - \* Nei

- Fungerte filterfunksjonene slik som du forventet?
  - \* Ja
  - \* Nei
- Hvordan opplevde du at påloggingen fungerte?
- Hvordan opplevde at det fungerte å legge til et nytt undervisningsopplegg?
- Har du andre innspill/forslag til forbedring?

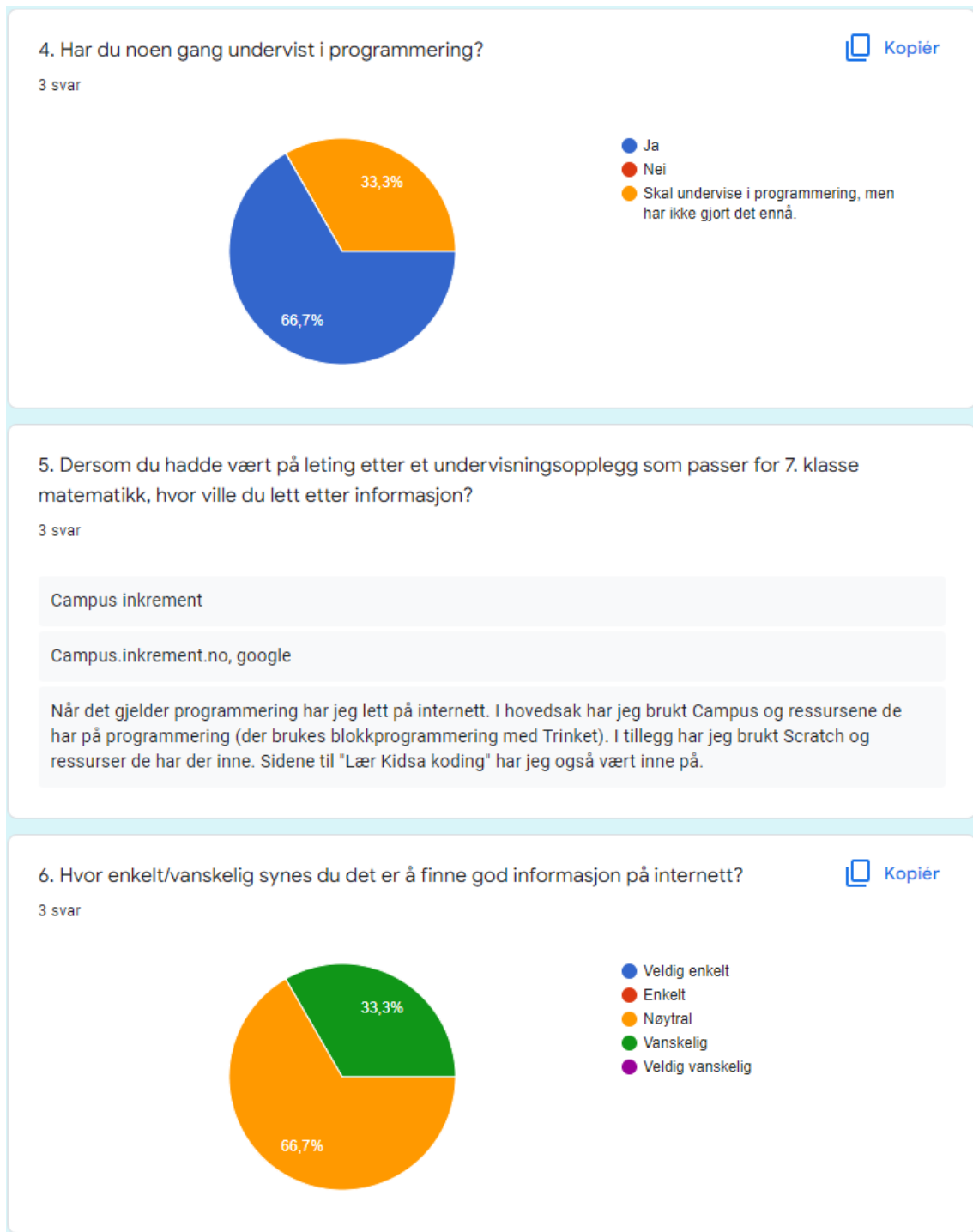


## E Respons på skjema fra brukertesting

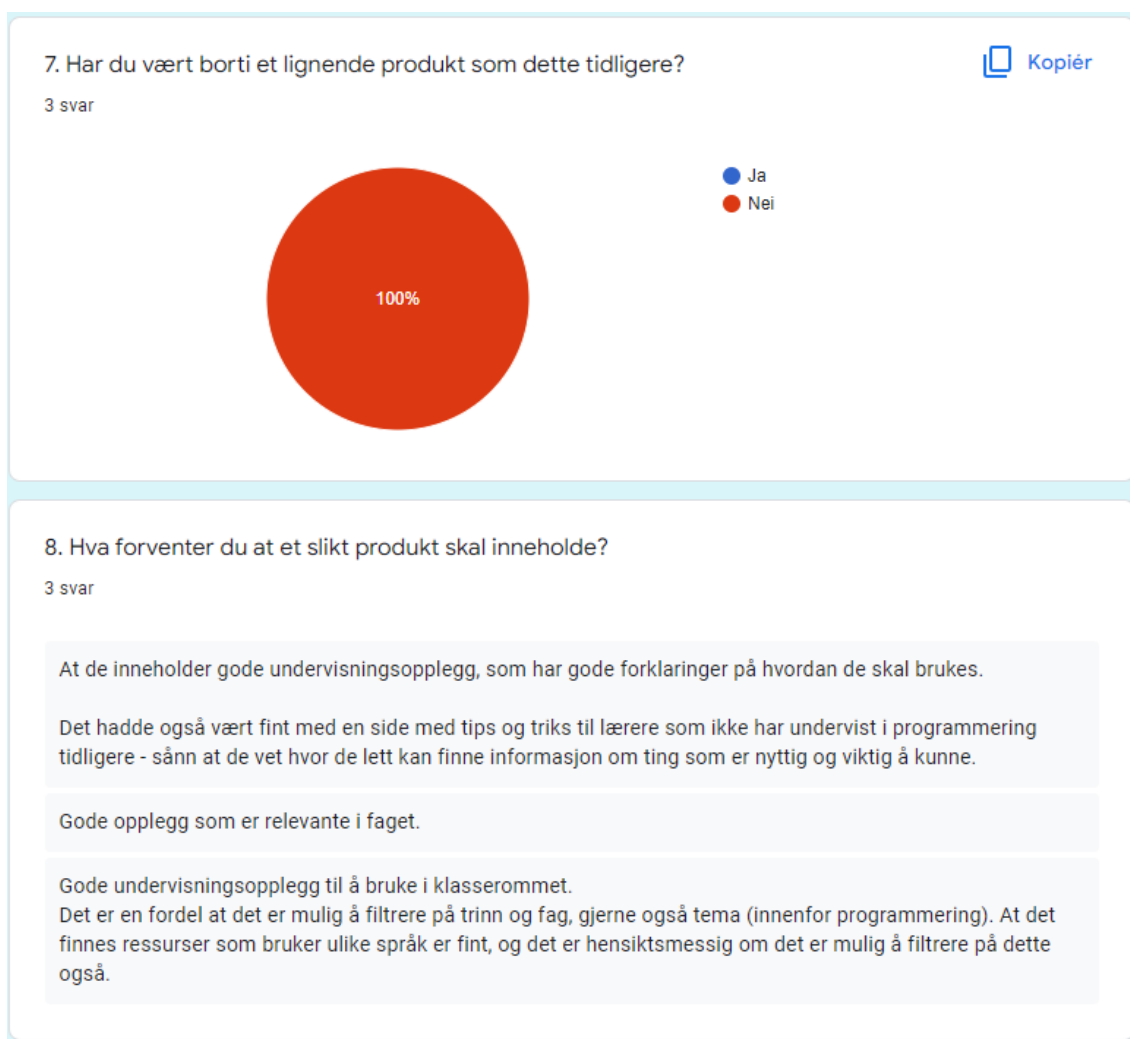
Testpersonene ble bedt om å fylle ut et Google Forms-skjema i etterkant av brukertesting. Et sammendrag av svarene på skjemaet fremkommer i figurene som følger.



Figur 29: Respons fra skjema brukertesting 1



Figur 30: Respons fra skjema brukertesting 2



Figur 31: Respons fra skjema brukertesting 3

### Etter testen

9. Hva er ditt helhetsinntrykk av produktet?

3 svar

Det virket som et godt produkt. Dette er et produkt jeg tenker vil være veldig nyttig for lærere som ikke kan så mye om programmering fra før.

Ser ryddig og pent ut.

Utgangspunktet er fint. Hvis det blir bygget ut med mange ressurser vil dette være noe som kan være nyttig.

10. Dersom du hadde vært på leting etter et undervisningsopplegg for programmering, ville du brukt dette produktet?

3 svar

[Kopier](#)

Svaralternativ	Prosent
Ja	100%
Nei	0%

11. Basert på ditt svar i forrige spørsmål, hvorfor/ hvorfor ikke?

3 svar

Fordi jeg vet at disse produktene vil være godt gjennomtenkt, og at de laget i det formål at de skal brukes som undervisningsopplegg for lærere i grunnskolen.

Virker oversiktlig, lett tilgjengelig og relevant

Det var enkelt å få oversikt. Forhåpentligvis vil det være gode opplegg som ligger der som kan brukes for å fremme elevenes læring i programmering.

Figur 32: Respons fra skjema brukertesting 4



Figur 33: Respons fra skjema brukertesting 5

Hvordan opplevde du at påloggingen fungerte?

3 svar

Den var veldig grei, som en hvilken som helst annen pålogging.

Kjapt og enkelt

Det fungerte fint

Hvordan opplevde at det fungerte å legge til et nytt undervisningsopplegg?

3 svar

Det synes jeg var intuitivt og virket som at det var enkelt å legge til nye opplegg.

Enkelt og greit

Det var oversiktlig og greit.

Har du andre innspill/forslag til forbedring?

3 svar

Som tidligere nevnt: Det hadde vært fint med en side inne på applikasjonen som kunne vært "programmering for dummies".

Ellers synes jeg det virka som en god side. Veldig fint å vite at undervisningen er lagd spesielt til barne- og ungdomsskolen, og at de er jobbet godt med av studenter som studerer programmering.

Håper på at plattformen blir tilgjengelig så fort som mulig!

Programmet blir ikke bedre enn oppleggene som ligger der. At dette vil brukes kommer an på hvordan ressursene oppleves, når brukervennligheten er på plass.

Figur 34: Respons fra skjema brukertesting 6

