



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Mobilapplikasjon for effektiv loggføring og kontroll
av arbeidstimer

Mobile application for effective logging and monitoring of
work hours

Mathias Nervik

Susanne Å. Løtvedt

Kristian A. Ølmheim

Fakultet for teknologi, miljø- og samfunnsvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Informasjonsteknologi

Veileder: Per Christian Engdal

Innleveringsdato: 13.05.2024

Vi bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Mobilapplikasjon for effektiv loggføring og kontroll av arbeidstimer	<i>Dato:</i> 13.05.2024
<i>Forfatter(e):</i> Mathias Nervik Susanne Å. Løvstedt Kristian A. Ølmheim	<i>Antall sider u/vedlegg:</i> 69
	<i>Antall sider vedlegg:</i> 69
<i>Studieretning:</i> Informasjonsteknologi	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Per Christian Engdal	<i>Gradering:</i> Ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> Retrams AS	<i>Oppdragsgivers referanse:</i> EF-1
<i>Oppdragsgivers kontaktperson:</i> Per Helge Litzheim Frøiland	<i>Telefon:</i> 48232041

<i>Sammendrag:</i>
<p>Målet med prosjektet er å lage en applikasjon for å registrere arbeidstimer. Den skal ha funksjonalitet som er enkel å bruke slik at brukerne kan registrere arbeidstimer og hvor mye de har jobbet på ulike prosjekter. Resultatet ble en ferdig Android applikasjon med de fleste funksjonalitetene gruppen hadde sett for seg.</p> <p>The goal is to create a mobile application for effective logging and monitoring of work hours. It contains functionalities that should make it easy for the users to register work hours and how much they have worked on each project. The result is an Android application with most of the functionalities the group expected.</p>

Stikkord:

Timeregistrering	Mobilapplikasjon	NativeScript-Vue
------------------	------------------	------------------

FORORD

Denne rapporten dokumenterer arbeidet som er gjort i bachelorprosjektet "Mobilapplikasjon for effektiv loggføring og kontroll av arbeidstimer" gitt av oppdragsgiver Retrams AS. Gruppen består av tre studenter, Mathias Nervik, Susanne Å. Løtvedt og Kristian A. Ølmheim som studerer informasjonsteknologi på HVL i Førde.

Vi ønsker å takke Retrams for å gi gruppen en spennende og lærerik oppgave, gode tilbakemeldinger og godt samarbeid under prosjektet.

Videre ønsker vi å takke vår veileder, Per Christian Engdal, for god oppfølging og tilbakemeldinger underveis i prosjektet.

INNHALDSFORTEGNELSE

ORDLISTE.....	1
FORKORTELSER.....	1
1. INNLEDNING.....	3
1.1 KONTEKST.....	3
1.2 MOTIVASJON.....	3
1.3 PROSJEKTEIER.....	3
1.4 PROBLEMBESKRIVELSE OG MÅL.....	4
1.5 OPPBYGGING AV RAPPORTEN.....	4
2. PROSJEKTBEKRIVELSE.....	5
2.1 PRAKTISK BAKGRUNN.....	5
2.1.1 Tidligere arbeid.....	5
2.1.2 Initielle krav.....	6
2.1.3 Initiell løsnings-idé.....	7
2.2 AVGRENSNINGER.....	7
2.3 RESSURSER.....	7
2.4 LITTERATUR OM PROBLEMSTILLINGEN.....	8
3. DESIGN AV PROSJEKTET.....	10
3.1 FORSLAG TIL LØSNING.....	10
3.1.1 Alternativ 1: Navigasjon ved bruk av bar-meny.....	10
3.1.2 Alternativ 2: Navigasjon ved bruk av skuff-meny.....	11
3.1.3 Alternativ 3: Navigasjon via flismeny på forside.....	12
3.1.4 Alternativ 4: Navigasjon gjennom å kombinere alternativene.....	13
3.1.5 Alternativ 5: Helt ny oppbygging.....	14
3.1.6 Diskusjon av alternativene.....	14
3.2 VALGT LØSNING.....	16
3.3 VALG AV VERKTØY.....	17
3.3.1 Figma.....	17
3.3.2 Collanote.....	17
3.3.3 Visual Studio Code.....	17
3.3.4 Typescript.....	17
3.3.5 NativeScript-Vue.....	17
3.3.6 Bitbucket.....	17
3.3.7 Google Drive.....	18
3.3.8 Trello.....	18
3.3.9 TailwindCSS.....	18
3.3.10 Xcode.....	18
3.3.11 ChatGPT.....	18
3.4 Prosjektmetodikk.....	18
3.4.1 Utviklingsmetodikk.....	18
3.4.2 Prosjektplan.....	20
3.4.3 Risikovurdering.....	20
3.5 EVALUERINGSPLAN.....	22
4. DETALJERT LØSNING.....	23
4.1 ARKITEKTURSKISSE.....	24

4.1.1	Frontend.....	25
4.1.2	Backend.....	26
4.2	WIREFRAME LAYOUT.....	27
4.3	WIREFRAME DESIGN.....	28
4.3.1	PinCodePage.....	28
4.3.2	StopwatchPage.....	29
4.3.3	NavBar.....	30
4.3.4	ProfileSettingsPage.....	31
4.3.5	AddHourPage.....	32
4.3.6	HoursPage.....	33
4.3.7	HourPage.....	34
4.3.8	ReportsPage.....	35
4.3.9	ProjectsPage.....	35
5.	RESULTATER.....	36
5.1	EVALUERINGSMETODE.....	36
5.1.1	Kontinuerlige tilbakemeldinger.....	36
5.1.2	Brukertesting.....	37
5.1.3	Integrasjonstesting.....	38
5.1.4	System Testing.....	38
5.2	EVALUERINGSRESULTAT.....	39
5.2.1	Kontinuerlige tilbakemeldinger.....	39
5.2.2	Brukertesting.....	40
5.2.3	Integrasjonstesting.....	42
5.2.4	System Testing.....	43
5.3	PROSJEKTRESULTAT.....	44
5.3.1	Funksjonelle krav.....	44
5.3.2	Logg Inn.....	45
5.3.3	Legg til time.....	46
5.3.4	Ikke-funksjonelle krav.....	46
5.4	PROSJEKTGJENNOMFØRING.....	47
6.	DISKUSJON.....	48
6.1	UTVIKLE FOR TO SYSTEMER.....	48
6.2	Nativescript Vue.....	48
6.3	Biblioteker.....	49
6.4	SIKKERHET.....	49
6.3.1	Pin-kode og biometri.....	49
6.3.2	Passende overskrift.....	50
6.5	Valg av design.....	50
6.5.1	VALGT NAVIGASJONSMETODE.....	50
6.5.2	Brukertesting.....	51
6.5.3	Farger / Tema.....	53
6.5.4	Rød-tråd.....	53
6.5	Utfordringer og muligheter for mobil.....	53
6.5.1	Nye funksjoner.....	53
6.5.2	Skalering.....	54

6.5.4 Tilgjengelighet.....	54
6.6 Prosjektgjennomføring.....	54
6.7 Oppsummering / Resultat.....	54
6.x.x Krav og mål/Resultat.....	54
6.x.x FORBEDRINGER OG VIDEREUTVIKLING.....	54
7. KONKLUSJON OG VIDERE ARBEID.....	56
7.1 KONKLUSJON.....	56
7.2 VIDERE ARBEID.....	57
7.2.2 <i>Dark mode</i>	57
7.2.3 <i>Leder og admin</i>	57
8. REFERANSER.....	58
9. VEDLEGG.....	59

ORDLISTE

Ord	Forklaring
JavaScript	Programmeringsspråk.
NodeJS	JavaScript runtime for å kjøre kode utenfor nettleseren.
Rammeverk	Samling av verktøy som forenkler utviklingen av et produkt
Nativescript	Et rammeverk for å utvikle native applikasjoner med JavaScript
Vue	Et JavaScript rammeverk for å bygge brukergrensesnitt med JavaScript
iOS og Android	2 av de mest brukte mobile operativsystemer
Service	En samling med gjenbrukbar kode til backend, ofte kode som sender forespørsler til en API.
Destinasjon	En side i en mobilapp som kan besøkes direkte av en bruker.

FORKORTELSER

Forkortelse	Forklaring
PC	“Personal Computer”, personlig datamaskin.
API	“Application Programming Interface”, en tjeneste som tilbyr kommunikasjon mellom programvaresystemer.
UX	“User Experience”, opplevelsen som en bruker har når de bruker et produkt, eller en tjeneste.
SUS	“System Usability Scale”, en standardmetode for å vurdere brukervennligheten til et system.
UCD	“User Centered Design”, en utviklingsmetode hvor brukeren er i fokus.
XSS	“Cross-Site Scripting”, en måte å kjøre JavaScript på en annen bruker sin datamaskin.

1. INNLEDNING

1.1 Kontekst

Retrams ønsker en mobilapplikasjon som bygger videre på deres web-løsning for timekontroll. Den opprinnelige løsningen ble utviklet som en løsning av en tidligere bachelor "Retrams Timeføring" (Ryland, Bibic, & Skårnes, 2023). Oppgaven omhandler en løsning de ansatte i Retrams kan bruke for å registrere og holde oversikt over arbeidstimer. Målet var å lage både en nettside og en mobil app, men sistnevnte ble det ikke tid til. Derfor ønsker Retrams å utvide webapplikasjonen med en mobilapp.

1.2 Motivasjon

Retrams er drevet av et klart formål når de starter dette prosjektet: å forbedre og effektivisere tidsrapportering for sine ansatte. Selv om den eksisterende løsningen er tilgjengelig på laptop / PC, er den ikke optimalisert for mobilbruk. Dette begrenser tilgjengeligheten for ansatte som ønsker å føre timer direkte og umiddelbart, spesielt i situasjoner de ikke har tilgang til laptop eller PC, som på reise eller utenfor kontoret. Derfor kreves en løsning som er skreddersydd for effektivitet og brukervennlighet på mindre skjermer.

Ved å utvikle en mobilapp ønsker Retrams å gjøre det mulig for ansatte å enkelt og effektivt føre timer, uavhengig av hvor de befinner seg når de ønsker å gjøre det.

Til slutt har Retrams også som motivasjon å styrke verdien av produktet de tilbyr ved å utvikle en egen løsning som Retrams ikke bare har mulighet til å forbedre med interne prosesser, men også til å kommersialisere løsningen.

1.3 Prosjekteier

Retrams er en IT-bedrift som ble etablert i 2018 og består av 7 ansatte (Retrams, 2024). Hovedkontoret deres ligger i Florø hvor de jobber med å utvikle tjenester til ulike bedrifter. Retrams er en prosjektdrevet organisasjon, dette gjør at en mobilapplikasjon som gir oversikt over alle arbeidstimer på de ulike prosjektene er ekstra viktig.

Retrams spesialiserer seg på å utvikle digitale løsninger rettet mot helsesektoren og har blant annet utviklet en scanner som gjør det mulig å identifisere kirurgisk utstyr uten å påføre fysiske merkelapper. Skanneren gjør at sykehus får raskt tilgang til riktig utstyr, reduserer smitterisiko, unngår over-innkjøp av utstyr og gir en full oversikt over varelageret. De har inngått en langsiktig avtale med Helse Bergen HF om å bruke denne løsningen på Haukeland universitetssykehus.

1.4 Problembeskrivelse og mål

Hovedutfordringen med den eksisterende web-applikasjonen for timekontroll er at den ikke er tilpasset visning på mindre skjermer, som betyr at den ikke er tilgjengelig på en mobiltelefon. Nettsiden kan fungere på et nettbrett hvis denne ligger sidelengs, men dette er ikke optimalt ettersom brukerdialogen ikke er beregnet for effektiv bruk og navigasjon på touchskjermer. Ansatte må derfor ha tilgang til PC for å kunne bruke nettsiden og registrere sine timer.

En sentral funksjon i dagens webløsning er en stoppeklokke som en ansatt kan anvende for å automatisk måle eget tidsbruk i en arbeidsøkt. Med dagens webløsning oppleves det tungvint å måtte benytte nettleseren på PC, for å starte og stoppe tidtaket. Ved å gi enkel og rask tilgang til denne fra mobiltelefon, vil ansatte sannsynligvis kunne forbedre nøyaktigheten rundt registrering av eget tidsbruk.

Det er dermed et behov for å kunne ha umiddelbar tilgang til de funksjonene dagens webløsning tilbyr, slik at ansatte effektivt kan registrere timer og ha kontroll over egen arbeidstid innenfor et prosjekt.

Målet er å utvikle en mobilapplikasjon, basert på eksisterende funksjonalitet fra dagens webløsning, som gjør det effektivt for ansatte å registrere sine timer.

1.5 Oppbygging av rapporten

Denne rapporten oppsummerer oppgaven bachelorgruppen arbeidet med våren 2024. Hvert kapittel har ulike fokusområder om hvordan gruppen løste oppgaven.

Kapittel 1. Innledning: Innledningen består av hvilken kontekst oppgaven skal bli løst i og litt om oppdragsgiveren.

Kapittel 2. Prosjektbeskrivelse: Dette kapitlet handler om den tidligere løsningen, hvilke omstendigheter gruppen skal løse prosjektet under og litteratur om problemstillingen.

Kapittel 3. Design av prosjektet: Inneholder forslag til de visuelle løsningene gruppen kan implementere og hvordan prosjektet er gjennomført.

Kapittel 4. Detaljert løsning: En detaljert forklaring om hvordan løsningen er designet ved hjelp av arkitekturskisser og wireframes.

Kapittel 5. Resultater: Hvordan prosjektet er gjennomført, resultater fra brukertester og forklaring av løsningen.

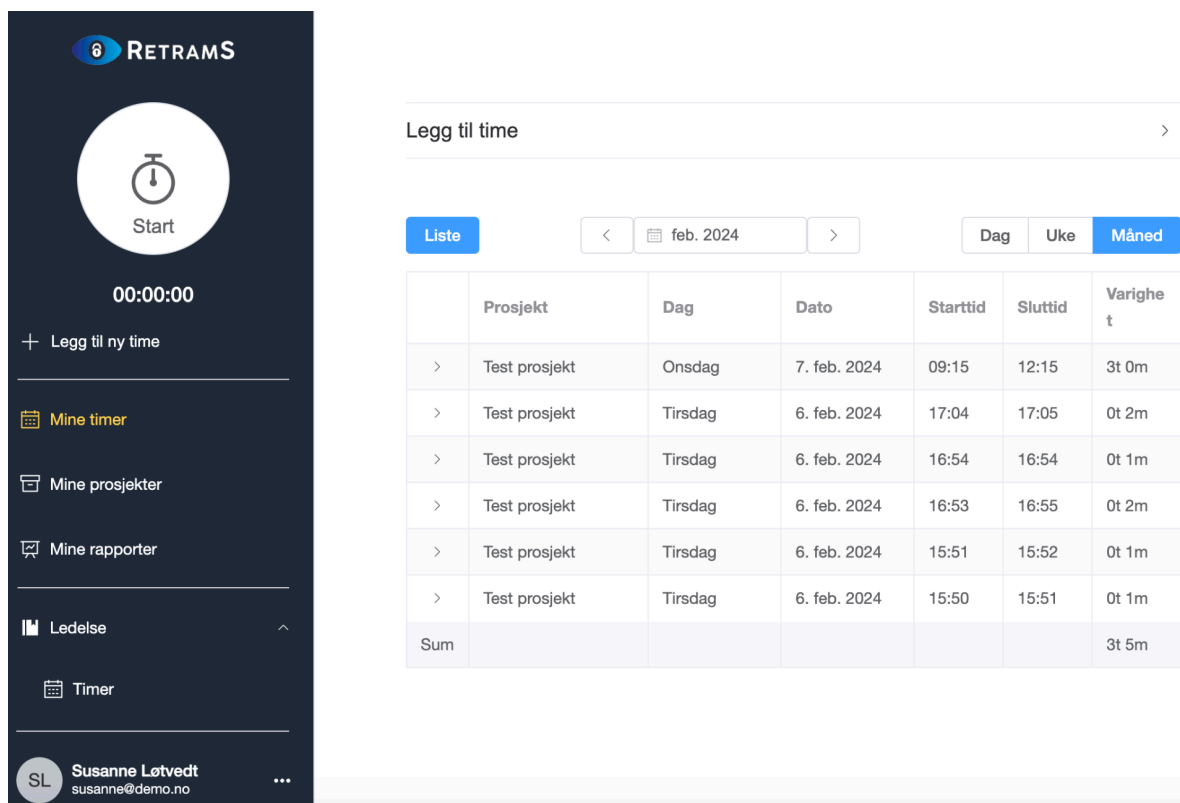
Kapittel 6. Diskusjon: Drøfting av løsning og resultater.

Kapittel 7. Konklusjon og videre arbeid: Dette kapittelet handler om hvordan gruppen syntes prosjektet har blitt gjennomført og hvordan applikasjonen kan jobbes videre på.

2. PROSJEKTBEKRIVELSE

2.1 Praktisk bakgrunn

2.1.1 Tidligere arbeid



The screenshot displays the RETRAMS mobile application interface. On the left is a dark sidebar with navigation options: 'Start' (with a stopwatch icon and '00:00:00'), 'Legg til ny time', 'Mine timer', 'Mine prosjekter', 'Mine rapporter', 'Ledelse', and 'Timer'. The user profile 'Susanne Løtvedt' is visible at the bottom. The main content area shows a 'Legg til time' button, a 'Liste' view selector, a date filter for 'feb. 2024', and view toggles for 'Dag', 'Uke', and 'Måned'. Below is a table of time entries.

	Prosjekt	Dag	Dato	Starttid	Sluttid	Varighet
>	Test prosjekt	Onsdag	7. feb. 2024	09:15	12:15	3t 0m
>	Test prosjekt	Tirsdag	6. feb. 2024	17:04	17:05	0t 2m
>	Test prosjekt	Tirsdag	6. feb. 2024	16:54	16:54	0t 1m
>	Test prosjekt	Tirsdag	6. feb. 2024	16:53	16:55	0t 2m
>	Test prosjekt	Tirsdag	6. feb. 2024	15:51	15:52	0t 1m
>	Test prosjekt	Tirsdag	6. feb. 2024	15:50	15:51	0t 1m
	Sum					3t 5m

Figur 2.1 Eksisterende løsning

Prosjektet er en fortsettelse av en tidligere bacheloroppgave, hvor det ble utviklet en web-løsning slik at ansatte lettere kunne registrere sine timer. Før denne løsningen ble laget brukte bedriften Excel til å føre og lagre arbeidstimer. Dette fungerte dårlig og de ønsket dermed en bedre løsning. Denne oppgaven handlet derfor om å utvikle en egen

applikasjon for timekontroll, som skulle gjøre det lettere for de ansatte å føre timer. Løsningen skulle også inneholde funksjoner for ledelse og administrasjon, slik at de kan opprette prosjekter og kontrollere registrerte timer.

I utgangspunktet var målet å utvikle en løsning for både PC og mobil, men det var ikke nok tid til å lage begge deler. Resultatet ble derfor en prototype av en web-applikasjon som bare fungerte på PC.

I ettertid har oppdragsgiver jobbet med å videreutvikle nettsiden og legge til funksjoner og sider som ikke ble ferdig.

Vår løsning skal bygge videre på de eksisterende funksjonene for de ansatte og gjøre disse tilgjengelig på mobil.

2.1.2 Initielle krav

Det primære fokuset ligger i å automatisk og enklest mulig registrere timer i appen, på ulike prosjekter. Dette er en løsning som finnes i den eksisterende løsningen, men som skal forenkles og tilpasses i den nye løsningen for å tillate mer effektiv registrering av timer på mobil-enheter. Appen skal også ha flere innloggingsmetoder for ulike sikkerhetsbehov. Dette inkluderer innlogging med pin-kode samt biometrisk autentisering som fingeravtrykk eller ansiktsgjenkjenning.

Oppdragsgiver ønsker at det benyttes Nativescript-Vue for mobilløsningen da de benytter Vue som rammeverk for sine webapplikasjoner. Dette sikrer at Retrams kan enklest mulig videreutvikle løsningen.

Løsningen skal

- fungere på iOS og Android
- en bruker skal kunne bytte mellom lys og mørk modus
- en bruker skal kunne logge inn med brukernavn og passord
- en bruker skal kunne logge inn med pin-kode
- en bruker skal kunne logge inn med biometri
- en bruker skal kunne registrere en time i et prosjekt
- en bruker skal kunne registrere ny time i et prosjekt
- en bruker skal kunne endre eksisterende time i et prosjekt
- en bruker skal kunne slette eksisterende time i et prosjekt
- en bruker skal kunne vise en oversikt over alle sine registrerte timer
- en bruker skal kunne vise en oversikt over sine registrerte timer per prosjekt
- en bruker skal kunne starte en stoppeklokke for beregning av tidsperiode

2.1.3 Initiell løsnings-idé

Den initielle løsningsideen er å implementere den gjeldende funksjonaliteten fra dagens webløsning slik at applikasjonen kan benyttes på en mobiltelefon. Dette blir gjort for å gjøre applikasjonen mer effektiv og tilgjengelig for brukerne. Siden gruppen skal lage en mobilapplikasjon vil dette kreve tilpasninger som sikrer en effektiv og optimal løsning. Dette inkluderer å implementere et godt brukergrensesnitt som gjør det enkelt for brukerne å navigere rundt i appen.

Det vil også utvikles mer funksjonalitet enn det som finnes i webløsningen som for eksempel innlogging med biometri. Dette bidrar med å forbedre sikkerheten og brukervennligheten til applikasjonen, hvor brukerne kan logge inn raskt.

2.2 Avgrensninger

En sentral avgrensning i prosjektet er at mobilapplikasjonen kun skal inneholde de eksisterende funksjonalitetene fra webløsningen og løsningen skal begrenses til å inkludere funksjoner som er relevant for de ansatte i bedriften. Gruppen skal derfor ikke jobbe med utvikling av webapplikasjonen, men bare fokusere på mobilapplikasjonen.

Mobilapplikasjonen skal kunne benyttes på både iOS og Android telefoner, og Nativescript-Vue skal brukes for begge plattformene. Det grafiske grensesnittet skal være mest mulig lik webløsningen og benytte det samme APIet. Gruppen har fått beskjed av oppdragsgiver til å bruke bitbucket for utveksling av kode.

2.3 Ressurser

For å sikre en god gjennomføring av prosjektet, vil det benyttes ulike ressurser. Den tidligere løsningen blir brukt som kunnskapsgrunnlag og inspirasjon, og det eksisterende APIet blir brukt for å implementere funksjonalitetene fra webløsningen. Den blir også brukt til å kjøre demoer for en bedre forståelse av hvordan applikasjonen skal fungere.

Gruppen har tilgang til brukerrepresentanter hos oppdragsgiver og har mulighet til å få teknisk veiledning ved behov.

Direkte kontakt med oppdragsgiver, veileder og brukerrepresentanter gjennom Teams og Discord, gir gruppen god veiledning for rapportskrivning og utvikling. Møterom og kontorplass hos Retrams og HVL, gir gruppen mulighet til å holde møter og presentasjoner.

Gruppemedlemmene har tilgang til iPhone, MacBook, Windows og Android. Dette gjør at gruppen kan teste mobilapplikasjonen på begge plattformer og tilpasse applikasjonen basert på kravene de har.

For å hente inspirasjon om hvordan gruppen skal skrive og løse oppgaver blir det brukt tidligere bacheloroppgaver for å sammenligne struktur, metoder og resultater. Dette gir innsikt i hvordan en bacheloroppgave kan se ut og gruppen får mulighet for eventuelle forbedringer.

2.4 Litteratur om problemstillingen

I 2007 revolusjonerte Apple smarttelefonmarkedet ved å introdusere berøringsskjerm med fingerbasert grensesnitt, som tidligere hadde vært pennbasert (s. 118)). I dag er berøringsgrensesnittet dominerende, og smarttelefoner har blitt så kraftige at de nå fungerer som små personlige datamaskiner. Dette har videre introdusert økt mobilitet og tilgjengelighet innenfor den digitale verden. Statistikk fra 2021 viser at 96 prosent av befolkningen i Norge eier en smarttelefon (SSB, 2022).

Hverdagen blir stadig mer digitalisert på mindre og mindre skjermer, der smarttelefonen har blitt et nødvendig verktøy for å kunne navigere seg gjennom dagliglivet. Mobilapplikasjoner basert på fysiske eller utdaterte løsninger gjør det mulig å spare tid på ellers krevende oppgaver. Mindre skjermer har derimot begrenset plass, som gjør det utfordrende å designe et passende brukergrensesnitt.

Mobildesign blir også påvirket av at brukere ønsker å utføre raske, fokuserte handlinger, med minimal interaksjon, ofte mens de er på farten (Taivalsaari, u.å.). Når man planlegger designet er det viktig å ta hensyn til at en bruker interagerer med brukergrensesnittet gjennom berøring. Det skal være lett for brukeren å forstå innholdet og navigere til ulike funksjoner, der det viktigste innholdet burde vises først (Grolid, 2021). Klikkbarhet skal være tydelig og brukeren skal alltid være orientert om hvor de befinner seg i hierarkiet. Innholdet bør også være universelt utformet, "slik at de kan brukes av så mange mennesker som mulig på en likeverdig måte" (Lid, 2024). Dette betyr at tekst skal være lesbart, og at brukergrensesnittet ikke krever avanserte bevegelser.

Det er også forskjeller mellom å designe en mobilapplikasjon og en mobilvennlig løsning for nettleser. En applikasjon vil utvikles ulikt basert på hvilke operativsystem den skal kjøres på, som primært i Norge er iOS eller Android (Statcounter, u.å.). Det er mulig å utvikle en applikasjon som fungerer på begge systemene, men dette kan være komplisert på grunn av unike forskjeller i utviklings- og kompileringsteknikker, og generell applikasjonsprogrammering (Hansen, Grønli, Ghinea, 2018). Designet på brukergrensesnittet kan derfor bli påvirket av operativsystemenes ulike standarder for design.

En web-løsning vil bli påvirket av hvilke nettleser den vises på, men ikke i like stor grad som en applikasjon blir påvirket av operativsystemet. Utfordringen med en web-løsning er at den burde fungere på flere skjermstørrelser. Det vil si, hvis man utvikler en mobilvennlig web-løsning, burde denne også være brukbar på større skjermer som på nettbrett og PC.

3. DESIGN AV PROSJEKTET

3.1 Forslag til løsning

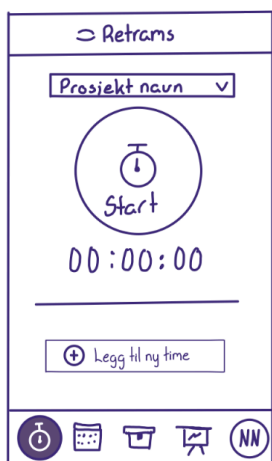
Mobilapplikasjonen skal inneholde de samme funksjonene som ansatte har tilgang til i web-løsningen. Den skal også kunne utvides med nye funksjoner i fremtiden, som for eksempel tilgang til Ledelse-funksjoner, eller å kunne få mer informasjon ved å trykke på et prosjekt osv.

Å lage en mobilapp for en nettside kan by på utfordringer. Dette er fordi mobiltelefoner har mindre skjerm som begrenser hvor mye plass det er til ulike funksjoner.

Alternativer til løsning vil derfor innebære hvordan ansatte skal kunne navigere og finne fram til de ulike funksjonene. Dette vil si å utforske ulike former for interaksjonsdesign, for å finne ut hvordan appen skal struktureres.

Det ble funnet fem ulike alternativer på hvordan navigasjonen kan fungere. Disse blir videre presentert gjennom enkle skisser.

3.1.1 Alternativ 1: Navigasjon ved bruk av bar-meny



Figur 3.1 bar-meny

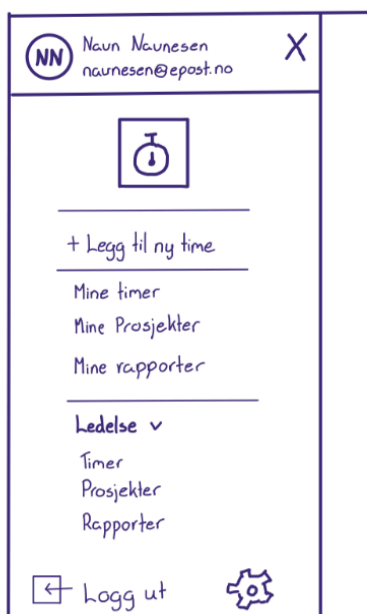


Figur 3.2 bar-meny, mine timer

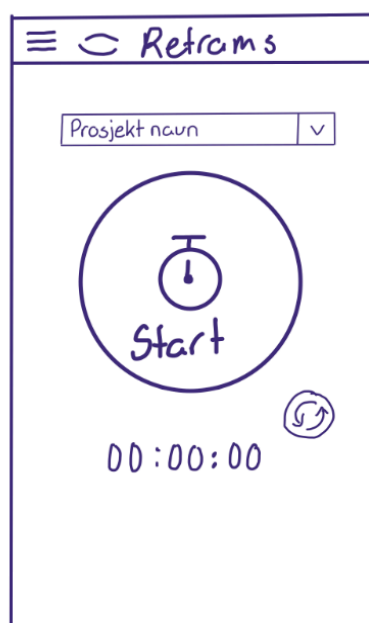
Det første alternativet er å kunne navigere gjennom en navigasjonsbar plassert på bunnen av skjermen som vist i figur 3.1 og figur 3.2. Denne typen navigasjon fungerer ved at brukere kan trykke på de ulike ikonene i menyen for å navigere til tilhørende sider.

Navigasjonsbaren vist i eksempelet inneholder Forside for stoppeklokke, Mine timer, Mine prosjekter, Mine rapporter og en profilside. Forsiden er den som vises først etter innlogging der ansatte umiddelbart kan starte stoppeklokken eller legge til en ny time.

3.1.2 Alternativ 2: Navigasjon ved bruk av skuff-meny



Figur 3.3 skuffmeny



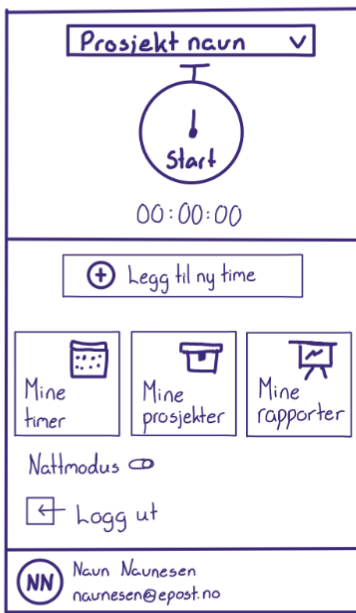
Figur 3.4 skuffmeny, gjemt



Figur 3.5 skuffmeny, mine timer

Dette alternativet ligner mest på dagens web-løsning som vist i kapittel 2.1.1. Denne menyen, også kjent som hamburgermeny, er tilgjengelig ved å trykke på de tre strekene plassert øverst til venstre på skjermen, som vist i figur 3.4 og figur 3.5. Dette gjør dermed at den tar lite plass og samtidig kan inneholde mange destinasjoner. Den vil kunne inneholde de samme destinasjonene som navigasjonsbaren i alternativet over, i tillegg til å ha plass til lederfunksjoner. "Legg til ny time" vil også plasseres her, som dermed gjør at stoppeklokkefunksjonen får sin egen dedikerte side.

3.1.3 Alternativ 3: Navigasjon via flismeny på forside



Figur 3.6 flismeny, nattmodus



Figur 3.7 flismeny, ledelse



Figur 3.8 flismeny, mine timer

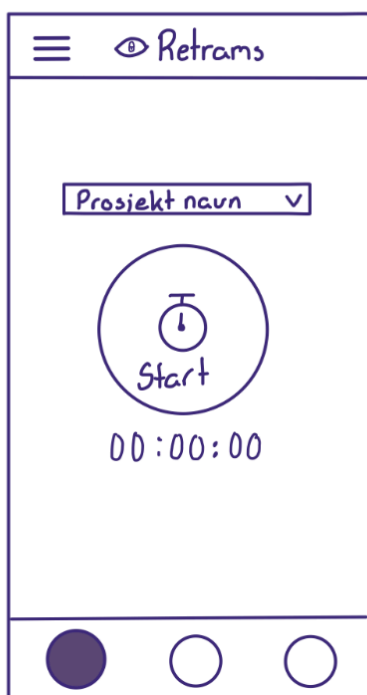


Figur 3.9 mine timer (alt.)

Det tredje alternativet er en meny som bare er tilgjengelig fra forsiden. Dette er en hierarkisk strukturert meny der bruker alltid må tilbake til forsiden for å kunne navigere til andre sider.

Denne menyen gir mulighet for å kunne velge om ansatte alltid vil kunne ha tilgang til stoppeklokken, eller at den bare skal vises på forsiden, som vist figurene (over). Alternativt kan det også implementeres en funksjon for å minimere den.

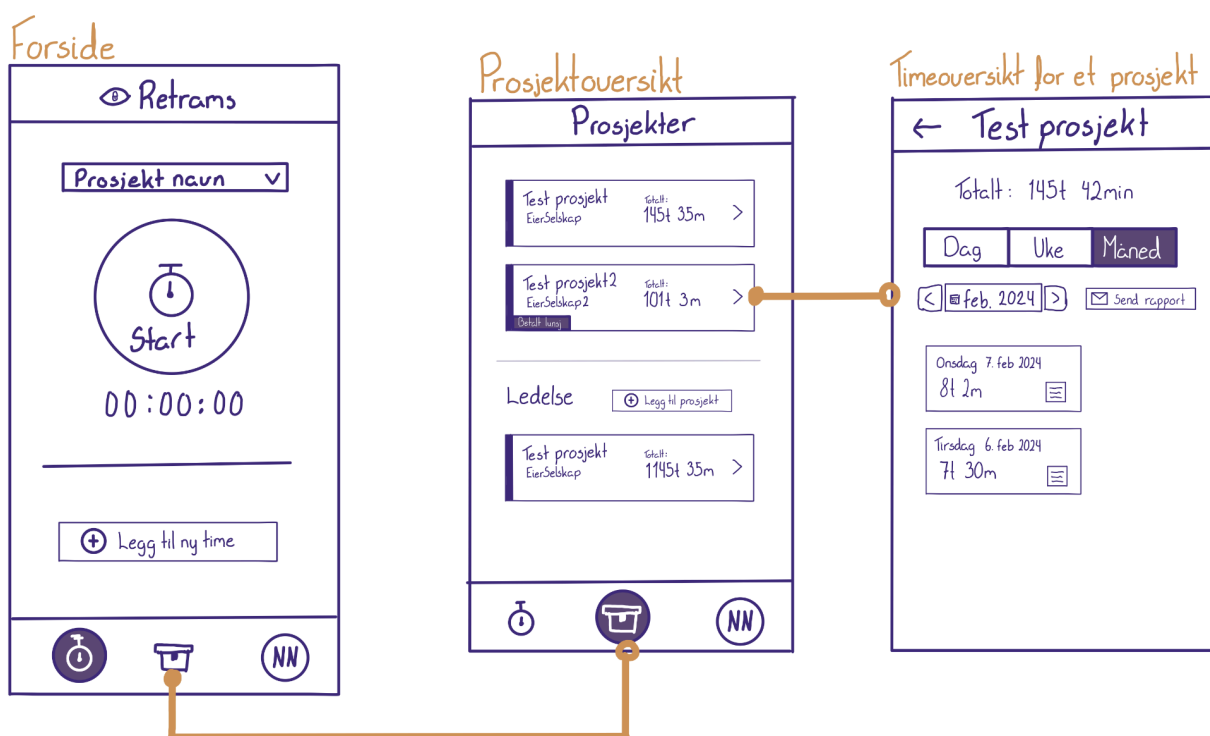
3.1.4 Alternativ 4: Navigasjon gjennom å kombinere alternativene



Figur 3.10 kombinasjon av navigasjonsbar og skuffmeny

Det fjerde alternativet er å kombinere navigasjonsbaren med skuffmenyen. Her vil de funksjonene som anses som viktigst alltid være tilgjengelig i navigasjonsbaren, og resten plasseres i skuffen. Eksempelvis, fra "Mine timer" vil en bruker kunne navigere direkte til stoppeklokken gjennom navigasjonsbaren, men må åpne skuff-menyen for å navigere til "Mine rapporter".

3.1.5 Alternativ 5: Helt ny oppbygging



Figur 3.11 ny løsning

Det siste alternativet utfordrer den originale web-løsningen med en ny måte å visualisere innholdet på. Her brukes en navigasjonsbar med bare tre alternativer; Forside, Prosjekter og en profilside. Forsiden og profilsiden inneholder det samme som i det første alternativet. Forskjellen er at under "Prosjekter" vil en ansatt kunne se en oversikt over alle sine prosjekter, og videre ved å trykke på dem, vil de kunne se de registrerte timene innenfor det valgte prosjektet. Dette betyr dermed at det ikke er nødvendig å gjøre timeoversikt og rapportvisning tilgjengelig gjennom navigasjonsbaren.

3.1.6 Diskusjon av alternativene

	Bar	Skuff	Flis	Kombinert	Ny oppbygging
Destinasjoner	3 - 5	5+	2+	8+	3 - 5
Utvidbar	✗	✓	✓	✓	◇
Rød-tråd	✗	✓	✗	◇	✗
Tommel- rekkevidde	✓	✗	✓	◇	✓
Kompleksitet	Lav(1)	Lav(2)	Middels(3)	Høy(4)	Høy(4)

✓ = Ja

✗ = Nei

◇ = Kanskje

Tabell 3.1 Sammenligner de ulike alternativene

Navigasjonsmetodene vurderes basert på om de oppfyller ulike kriterier, som vist i høyre kolonne av tabellen over. Disse er sortert basert på prioritet, der det viktigste kriteriet er at løsningen er utvidbar og dermed tilrettelegger for videreutvikling.

Rød-tråd handler her om at mobil-løsningen skal ligne på dagens web-løsning, det skal se ut som om det er en forbindelse mellom dem og at de hører sammen.

Tommel-rekkevidde vil si at om en bruker holder mobilen med en hånd, skal de kunne navigere ved å bare bruke tommelfingeren.

Kompleksitet handler om vanskelighetsgraden for utviklingen av navigasjonsmetoden, og dermed hvorvidt denne kan implementeres innenfor prosjektets tidsrammer.

Alternativ 1: Navigasjon ved bruk av bar-meny

Dette er en løsning som gjør det lett for å navigere mellom sidene, ettersom alle destinasjonene er innenfor tommel-rekkevidde. Det finnes også mye dokumentasjon som dermed gjør at den har lav kompleksitet. Derimot sier Googles Material Design spesifikasjoner at en navigasjonsbar ikke burde inneholde mer enn 5 eller mindre enn 3 destinasjoner (Material Design 3, u.å.). Som beskrevet tidligere, vil navigasjonsbaren for mobilløsningen inneholde akkurat 5 elementer. Dette betyr at den ikke er utvidbar, og at det derfor kan bli vanskelig å legge til lederfunksjoner når applikasjonen videreutvikles.

Alternativ 2: Navigasjon ved bruk av skuff-meny

Denne ligner mest på dagens web løsning og vil derfor ha en sterk rød-tråd. Den kan inneholde minst 5 destinasjoner(Material Design 3, u.å.), og er derfor utvidbar. Derimot er den plassert på toppen av mobilskjermen som gjør det vanskelig å bruke løsningen med en hånd da den er utenfor tommel-rekkevidde.

Likheten med dagens løsning og tilgjengelig dokumentasjon betyr at den vil kunne være lett å utvikle, og har derfor lav kompleksitet. Likheten betyr at gruppen ikke trenger å tenke over hva menyen skal inneholde og hvordan denne kan designes. Den vil derimot være litt vanskeligere å implementere enn det første alternativet, men denne forskjellen er minimal.

Alternativ 3: Navigasjon via flismeny på forside

Det er estimert av gruppen at denne menyen vil kunne inneholde mer enn 2 destinasjoner, og er derfor utvidbar. Derimot hvis den inneholder mer enn 6 vil ikke brukeren kunne se alle valgalternativene på skjermen, som vil gjøre den vrien å navigere. Her må brukere også alltid gå frem og tilbake for å kunne navigere til andre funksjoner. Den gir derimot umiddelbar tilgang til stoppeklokken uavhengig av hvor man befinner seg i appen. Å alltid ha tilgang til denne funksjonen er noe dette alternativet har til felles med web-løsningen, men ellers er de to veldig ulike og har derfor ikke noen rød-tråd.

Flis menyen har middels kompleksitet fordi den mangler tilgjengelig dokumentasjon, som også kompliserer videreutviklingen.

Alternativ 4: Navigasjon gjennom å kombinere alternativene

En kombinert meny vil kunne løse problemene til navigasjonsbaren og skuff-menyen. Denne kombinasjonen kan inneholde minst 8 elementer, som er summen av de to alternativene. Dette betyr at den derfor er utvidbar, som da løser problemet med navigasjonsbaren. Videre, ved å legge de viktigste funksjonene i navigasjonsbaren, vil disse være innenfor tommel-rekkevidde. Dermed løser dette også problemet med skuff-menyen.

Derimot oppstår det et problem når det skal bestemmes hvilken meny de forskjellige destinasjonene skal plasseres, og hva som er de viktigste funksjonene. Stoppeklokken og Legg til ny time kan ligge i navigasjonsbaren, men utfordringen er å velge det siste elementet. Løsningen vil også bare inneholde 6 destinasjoner, som betyr at en av menyene ikke vil møte spesifikasjonene. Dermed er konsekvensen at dette alternativet har høy kompleksitet, fordi det vil ta lang tid å bestemme hvordan den skal organiseres og implementeres. Det kan også føre til at gruppen tar dårlige valg for å oppnå 8 destinasjoner.

Alternativ 5: Helt ny oppbygging

Kan være forvirrende for brukere som er vandt med web-løsningen. Den løser problemet med alternativ 1, og er derfor utvidbar. Derimot vil det fortsatt være problemer for videreutviklingen, ettersom den er bygget opp på en helt annen måte enn webløsningen. Dette gjør det også vanskeligere å videreutvikle begge løsningene synkront. Det er derfor ikke en rød tråd mellom dette alternativet og web-løsningen. Store forandringer kan også være ubehagelig for brukere som er vandt med nåværende løsning.

3.2 Valgt løsning: skuff-meny

Det valgte alternativet er alternativ 2 fordi denne ligner mest på dagens løsning, og dermed en av de letteste menyene å designe i forhold til webløsningen. Videre vil det også derfor være mer forståelig å utvide den med nye utvalg i fremtiden.

3.3 Valg av verktøy

3.3.1 Figma

Figma er et designverktøy som brukes for å skape detaljerte wireframes og bestemme design. Figma blir brukt i en tidlig fase i prosjektet for å sikre at alle designelementer er godt gjennomtenkt og med positive tilbakemeldinger fra oppdragsgiver før det blir brukt i utviklingen.

3.3.2 Collanote

Collanote er en notatapp tilgjengelig på iOS, og blir brukt for å rask skissere og dele idéer tidlig i designfasen. Den blir også brukt under brukertesting for å lage oppgavene/spørsmålene og notere resultatene.

3.3.3 Visual Studio Code

Visual Studio Code er en kodeeditor valgt for utviklingen av appen. Den støtter et bredt spekter av programmeringsspråk og rammer, inkludert Typescript som er valgt for dette prosjektet.

3.3.4 Typescript

Typescript er valgt som programmeringsspråket for apputviklingen. Det tilbyr strengere syntaks for JavaScript som resulterer i bedre kodekvalitet og færre kjørefeil. Siden appen sannsynligvis skal videreutvikles etter endt bachelor, er det også en viktig fordel at koden blir mer vedlikeholdbar.

3.3.5 NativeScript-Vue

Nativescript er et rammeverk som blir brukt til å utvikle mobilapper på tvers av operativsystem ved hjelp av JavaScript eller Typescript. Dette verktøyet har blitt brukt til å bygge og kjøre appen, og sikrer at appen kan fungere sømløst både på Android og iOS.

3.3.6 Bitbucket

For lagring og synkronisering av kildekode benyttes Bitbucket, etter forslag fra oppdragsgiveren. Bitbucket tilbyr et gitt-basert versjonskontroll som er viktig for effektiv samhandling i gruppen, samt for at oppdragsgiver etter endt produkt kan videreutvikle vår løsning.

3.3.7 Google Drive

Google Drive blir brukt for å dele filer og for å skrive og lagre dokumentasjon relatert til prosjektet. Dette verktøyet muliggjør enkel tilgang til alle dokumentene samt at alle gruppe medlemmene kan skrive i samme dokument samtidig.

3.3.8 Trello

For oppsett av et kanban-board og for å holde oversikt over oppgaver, fremgang, og tidsfrister, benyttes Trello. Dette verktøyet støtter agile arbeidsmetoder og hjelper gruppen med å organisere arbeidsflyten og prioritere arbeidsoppgaver effektivt.

3.3.9 TailwindCSS

For implementeringen av designet på en effektiv måte, blir TailwindCSS brukt. Dette er et css-verktøy som lar gruppen raskt og effektivt lage gode brukergrensesnitt samt at relatert kode ikke er splittet opp i flere filer.

3.3.10 Xcode

For å kjøre og teste appen på iOS, er Xcode et krav. Xcode er Apples offisielle integrerte utviklingsmiljø for macOS, som inkluderer verktøyene som trengs for å kjøre en app på iOS.

3.3.11 ChatGPT

ChatGPT blir brukt for å tilby forslag til formuleringer, korrigerer av grammatikk, og forslag til både kode og innhold.

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

I valg av metode for prosjektet, er det flere særegenheter som kan tas i betraktning. Gruppen har tilgang til brukerrepresentanter via Retrams, og kan samarbeide tett med brukerne under utviklingen av prosjektet for at resultatet skal bli likt deres krav og behov. Gruppen følger prinsipper som design thinking og "User Centered Design" (Interaction Design Foundation, u.å.). Disse prinsippene blir brukt for å forstå brukernes behov og skape optimale og kreative løsninger for applikasjonen. Dette peker mot en iterativ prosess hvor gruppen tar til seg tilbakemeldinger fra brukeren og fortsetter å utvikle applikasjonen med hensyn til disse. Gruppen bruker wireframes og system testing underveis for å utvikle applikasjonen i henhold til tilbakemeldinger fra oppdragsgiver.

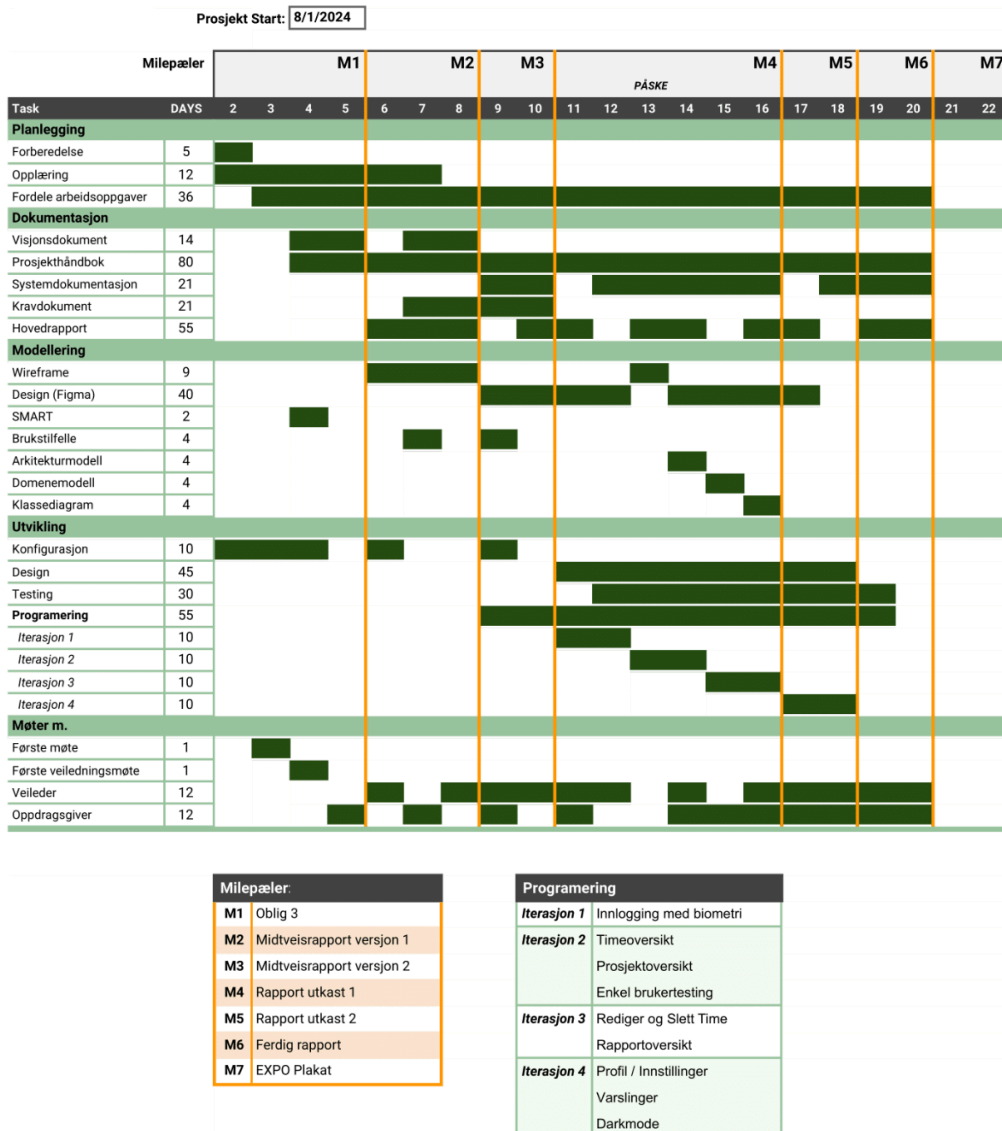
Gruppen bruker en hybrid modell av UCD hvor prosjektet har faste definerte milepæler for gjennomføringen: prosjektdesign og evaluering. Prosjektdesign handler om å sette seg inn i brukerens behov, forstå brukermiljøet i applikasjonen og hvordan brukeren skal

benytte seg av appen på best mulig måte. Denne iterasjonen handler om å planlegge applikasjonen slik at den er klar til å utvikles. Under evaluering bruker gruppen brukerundersøkelser og møter for å se hvordan appen har utviklet seg og hvordan brukerne reagerer på applikasjonen så langt. Gruppen kan se hva som er gjort bra og hva som burde endres på basert på disse testene og dermed gjenta prosessen med prosjektdesign og evaluering til brukerne er fornøyd med løsningen.

Gruppen bruker Kanban for å dele opp arbeidsoppgavene og få en oversikt over hvor langt vi har kommet i prosjektet. Oppgavene kan ha status som "To Do", "Doing" og "Done", og ut ifra dette kan vi se hvem som jobber på hvilke oppgaver og hvilken status de har i prosjektet.

3.4.2 Prosjektplan

Timekontroll



Figur 3.12 Hentet fra VEDLEGG 9.1.1 Prosjekthåndbok

Prosjektets fremdriftsplan er representert i Figur 3.12. Figuren viser et GANTT-diagram med syv forskjellige milepæler fordelt utover hele prosjektet. Hver milepæl har tilknyttede ukenumre som angir planlagt tidspunkt for utførelsen av de ulike oppgavene og aktivitetene. Utviklingen er delt opp i iterasjoner for hva som er planlagt å gjennomføre.

3.4.3 Risikovurdering

		Årsak	Sannsynlighet	Konsekvens	Risiko- produkt	Tiltak
1	Misforstått hva oppdragsgiver ønsker	Dårlig kommunikasjon	Lav (2)	Svært Høy (5)	10	Gjennomføre flere møter. Planlegge agenda og skrive ned spørsmål gruppen kommer på under utvikling. Brukerundersøkelse.
2	Oppdrag blir vanskeligere	Oppdragsgiver kommer med nye krav underveis	Svært Lav (1)	Svært Høy (5)	5	Avklare sannsynligheten for nye krav på forhånd.
3	Ikke får utført brukertesting	Usikkerhet rundt kontakt, ikke får laget enkel prototype,	Lav (2)	Middels (3)	6	Kommunisere med oppdragsgiver og få hjelp av veileder
4	Løsning blir ikke ferdig i tide	Avhengig av web-løsning som ikke er ferdig	Lav (2)	Middels (3)	6	Oppdragsgiver forventer ikke at løsning blir helt ferdig
5	Ikke får fullført alle målene	Dårlige prioriteringer	Middels (3)	Høy(4)	12	Identifisere prioriteringer og vanskelighetsgrad av funksjoner. Lage gode mål
6	Frafall av gruppemedlem	Sykdom, ulykke, eksterne livshendelser, uenigheter	Svært Lav (1)	Høy (4)	4	Åpne for digital kommunikasjon Lage SMART-modell for forventninger i gruppen Møte og kommunisere flere ganger i uken
7	Problemer med utstyr/programvare	Skade, dårlig/utdatert programvare, bugs	Høy (4)	Middels (3)	12	Undersøke programvare Gruppen hjelper hverandre med å finne feil. Kjøre nytt
8	Ikke får til fysisk møte i Florø	Dårlig vær, stengt vei, passer ikke for alle, utsettes for lenge	Høy (4)	Middels (3)	12	Spørre ofte. Planlegge tidlig. Følge med på værmelding.
9	Kompatibilitetsproblemer	Løsning fungerer ikke på alle enheter	Høy(4)	Høy (4)	16	Teste på ulike enheter. Planlegge og undersøke tidlig.
11	Problemer med datasikkerhet	Svake sikkerhetsprotokoller eller svakhet i koden.	Lav (2)	Høy (4)	8	Lage robuste sikkerhetsprotokoller og sjekke sikkerheten jevnlig.
12	Problemer underveis med utviklingsmiljøet	Utviklingsmiljøet er i beta	Lav (2)	Høy (4)	8	Jevnlig testing av funksjonalitet, og følge nøye med på nye oppdateringer til utviklingsmiljøet.

Tabell 3.1.3 Hentet fra VEDLEGG 9.1.1 Prosjekthåndbok

S a n n s y n l i g h e t	Svært Høy (5)	5	10	15	20	25
	Høy (4)	4	8	12	16	20
	Middels (3)	3	6	9	12	15
	Lav (2)	2	4	6	8	10
	Svært Lav (1)	1	2	3	4	5
		Svært Lav (1)	Lav (2)	Middels (3)	Høy (4)	Svært Høy (5)
Konsekvens						

Tabell 3.1.4 Risikomatrise

Fra tabellen kommer det fram at det å ikke nå alle satte mål representerer en betydelig risiko. For å adressere dette er det viktig med klare prioriteringer og mål for å sikre god fremdrift og resultat.

En annen høy risiko er relatert til uorden i prosjektets løsninger og arbeidsmetoder, som kan ha en høy kostnad både i tid og ressurser. Det er foreslått å dra nytte fra tidligere prosjekterfaringer og å legge til rette for grundig planlegging og estimering fra prosjektets begynnelse.

Kompatibilitetsproblemer fremstår også som en betydelig risiko. Tidlig testing på varierte plattformer og enheter er nødvendig for å sikre at produktet fungerer sømløst for alle brukere uavhengig av type mobil.

Fordi stabilitet er nødvendig for å opprettholde prosjektets fremdrift, utgjør utviklingsmiljøet en betydelig risiko. Effektive tiltak for å redusere en slik risiko er å utføre jevnlig testing og oppdatering av miljøet som gruppen bruker til å utvikle appen.

3.5 Evalueringsplan

Evaluering er en viktig del av utviklingsprosessen. Evalueringsmetoder fungerer som hjelpemiddel til å måle sluttresultatet opp mot krav og mål som ble satt i starten av prosjektet. Evalueringsplanen består av metoder innenfor brukertesting og programvare testing, som brukes iterativt gjennom hele prosjektet for å kunne evaluere løsningen.

Brukergrensesnittet vil kontinuerlig evalueres gjennom tilbakemeldinger gitt i faste møter med oppdragsgiver.

Midtveis i prosjektet vil detaljerte wireframes bli vist for å få tilbakemeldinger fra andre ansatte. Dette kan gjennomføres gjennom fysisk møte på bedriften, eller ved å sende inn en presentasjon med valgmuligheter.

Ved slutten av prosjektet er det planlagt å utføre en grundig evaluering av løsningen gjennom brukertesting og skjema. Denne testen handler om brukeropplevelsen og

applikasjonens funksjonalitet.

Spontan brukertesting vil også utføres dersom muligheten oppstår.

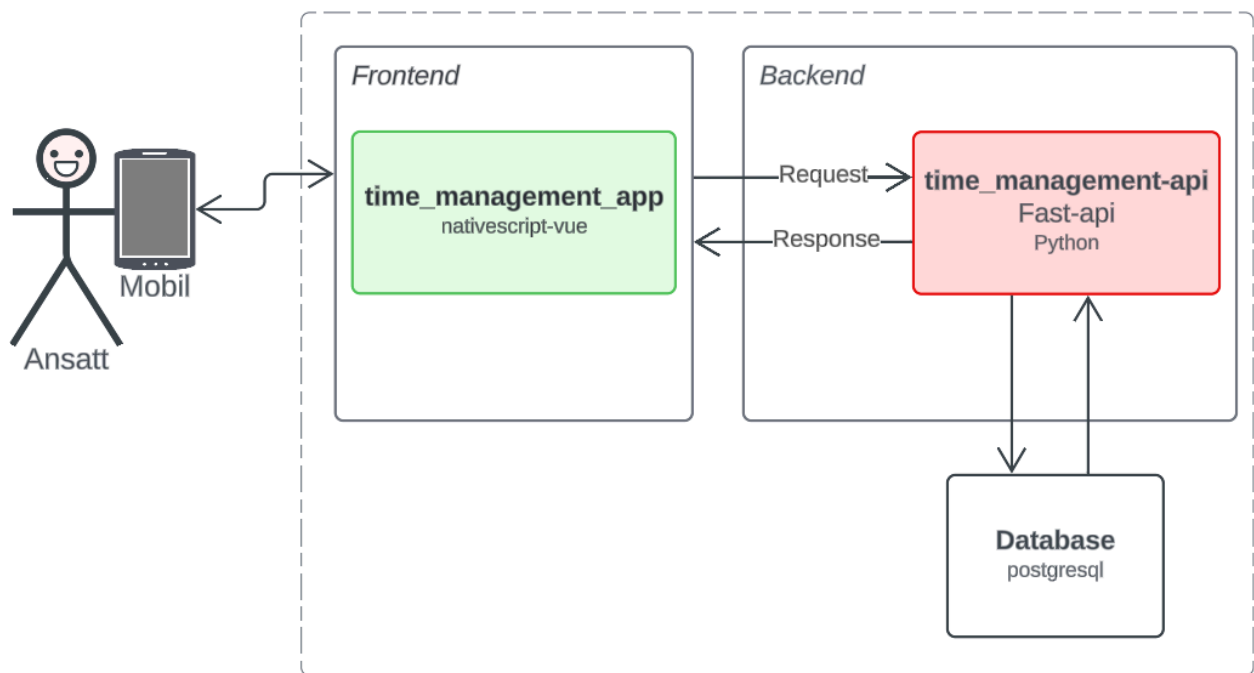
Under utvikling av applikasjonen er det planlagt å bruke to former for programvare testing. Den første er integrasjonstesting, som vil bli mest brukt i starten av utviklingen for å evaluere individuelle funksjoner. System testing vil videre bli brukt midtveis under utviklingen og videre mot slutten av utviklingen for å teste hvordan funksjoner samhandler.

4. DETALJERT LØSNING

Følgende funksjoner inngår som en del av løsningen:

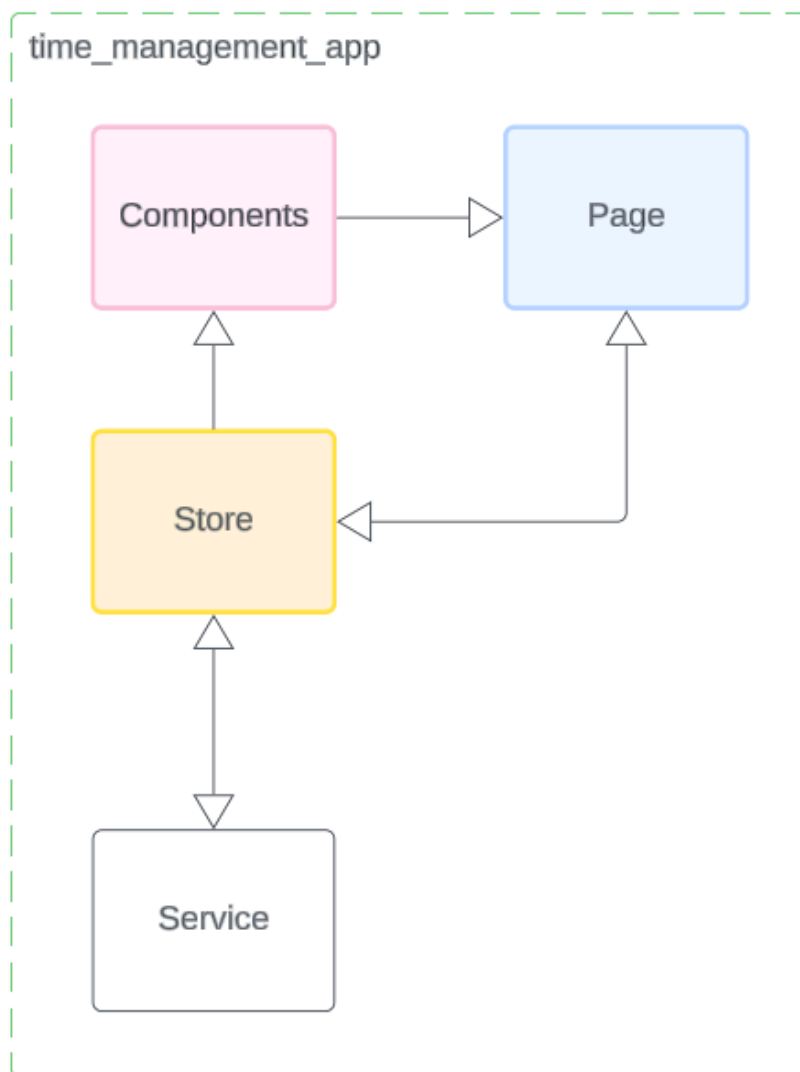
- Logge inn med biometri eller pin-kode
- Legge til en time manuelt
- Legge til en time med stoppeklokken
- Nullstille stoppeklokken
- Slette eller endre en time innen 10 minutter
- Vise en liste av timer basert på prosjekt
- Vise en liste av totalt antall timer per prosjekt
- Vise en liste over alle prosjekter, og om de har betalt lunsj

4.1 Arkitekturskisse



Figur 4.1 Hentet fra VEDLEGG 9.1.4 Systemdokumentasjon

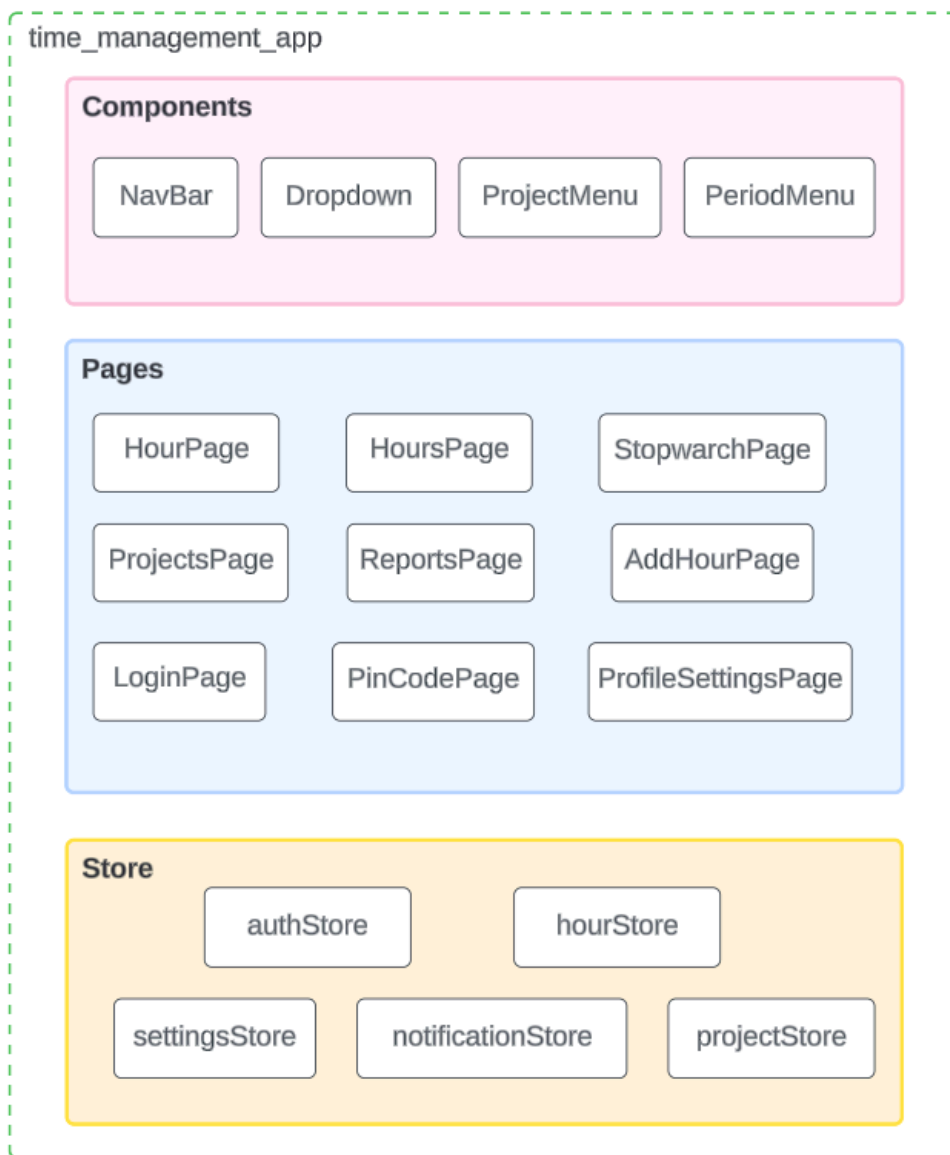
Figuren illustrerer de viktigste komponentene i løsningen, og hvordan de samhandler. Applikasjonen, her kalt "time_management_app", vises i figuren med grønn farge og er løsningen som er utviklet i dette prosjektet. Applikasjonen installeres på brukernes mobil, hvor den kommuniserer med backend-delen. Backend kommuniserer videre med databasen for data. API-et, her kalt "time_management-api" og representert med rød farge i figuren, gjenbrukes i dette prosjektet fra den forrige løsningen.



Figur 4.2 Applikasjonsarkitektur

4.1.1 Frontend

Appen struktureres i henhold til NativeScript Vue sin arkitektur med pages, komponenter, og stores. Hver side består av flere komponenter. Disse komponentene er byggesteinene som samler funksjonalitet og grensesnittelementer. For å håndtere og dele tilstand og logikk på tvers av forskjellige komponenter bruker en "store". En "store" fungerer som et sentralt lagringssted for tilstand i appen, hvor komponenten kan hente og oppdatere data dynamisk.



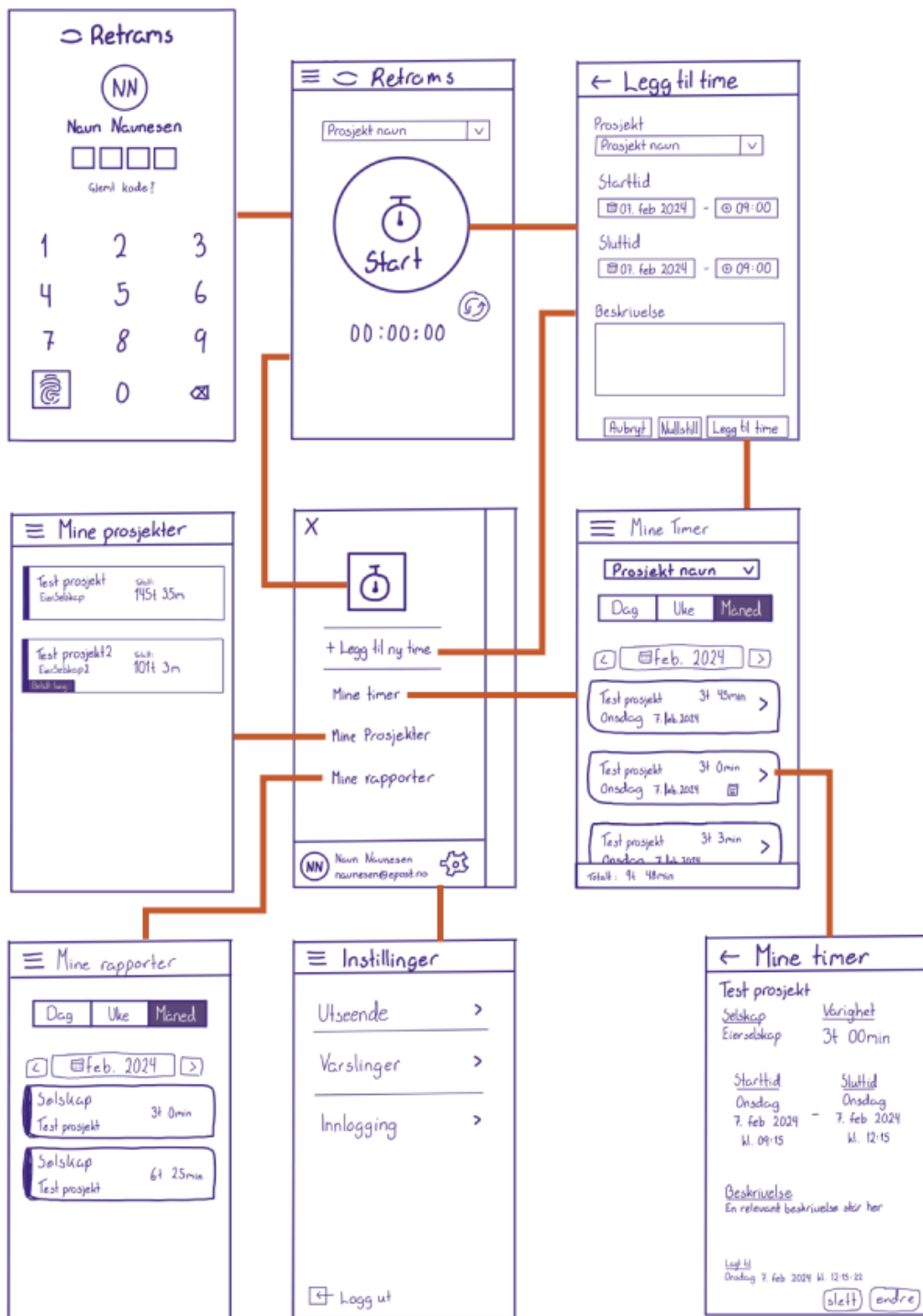
Figur 4.3 Detaljert applikasjonsarkitektur

Figuren viser det viktigste komponentene, pages og stores. Verdt å trekke fram er “NavBar” som definerer den overordnede strukturen på de fleste sidene, og Dropdown som definerer en selvlaget dropdown-meny. Fra store er det `authStore` som er verdt å merke seg, ettersom den brukes på alle sidene, og i flere av komponentene. Den definerer logikk for behandling av brukerdata og autentisering av innloggingsdata.

4.1.2 Backend

I løsningen er noen endepunkt brukt. `auth/login` blir brukt for å logge inn i systemet, `auth/me` blir brukt for å hente informasjon om en innlogget bruker, `hours` blir brukt for å opprette en tom time. `hours/employee` blir brukt for å hente en liste av timer for en bruker, `projects/employee` blir brukt for å hente prosjekter, og `hours/ID` blir brukt for å hente informasjon om en time.

4.2 Wireframe layout



Figur 4.4 flytdiagram for wireframes

Skissen som vist i figuren ovenfor representerer hovedsidene og flyten i applikasjonen. Her ble det bestemt hvordan brukergrensesnittet skulle organiseres, slik at gruppen videre kunne lage detaljerte design for applikasjonen.

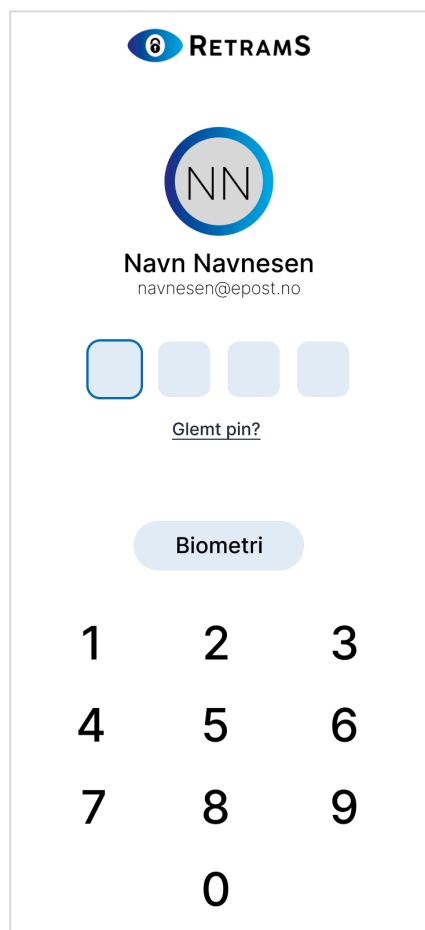
4.3 Wireframe design

De detaljerte designene er laget ved hjelp av Figma, som kan direkte kopieres under utviklingen av applikasjonen. Dette skaper dermed et større rom for fokus på funksjonalitet under utviklingen av applikasjonen.

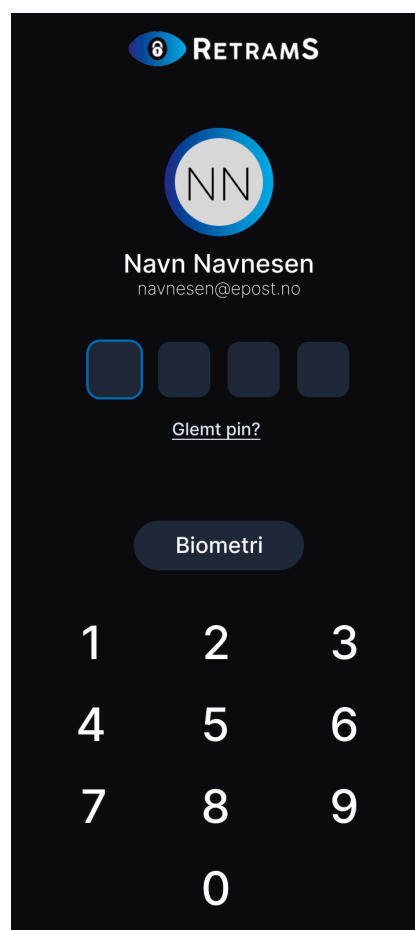
Designet for brukergrensesnittet er laget for å kunne fungere med både lyst og mørkt tema. Videre presenteres de fleste av design-prototypene, basert på skissene som vist i figur 4.4.

Merk! Fargene vil vises ulikt fra skjerm til skjerm og reflekterer derfor nødvendigvis ikke hvordan det faktisk ser ut på en mobilskjerm. Se vedlegg 9.2.5 for fargekoder

4.3.1 PinCodePage



Figur 4.4 Innlogging (lyst tema)

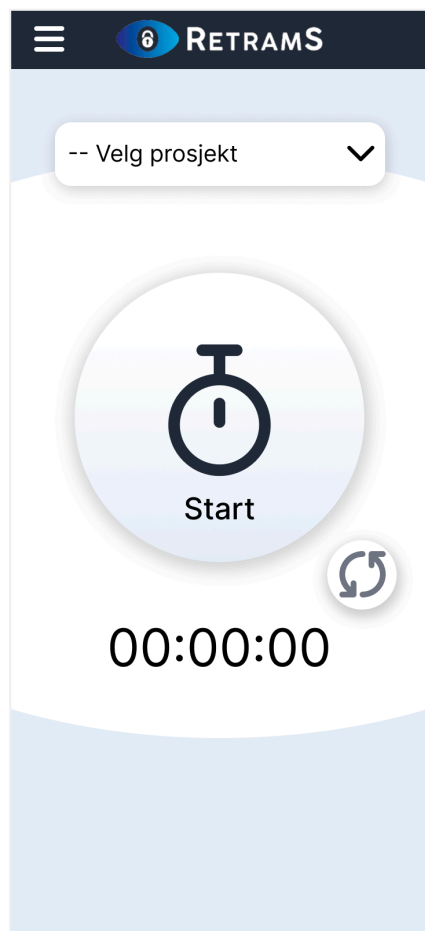


Figur 4.5 Innlogging (mørkt tema)

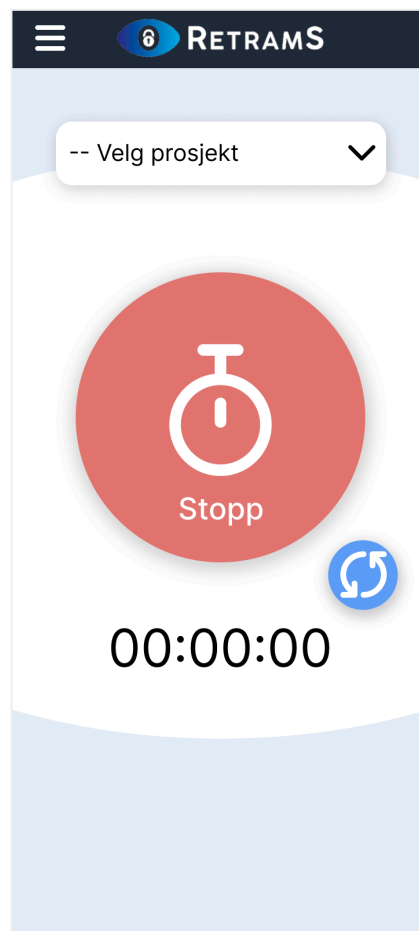
Dersom en bruker allerede er logget inn vil innloggingssiden vise dette med brukerens navn og epostadresse. Initialene i en sirkel er den samme som på nettsiden, der rammen rundt bruker de samme fargene som logoen.

Ved å trykke på "Glemt pin?" vil en bruker bli logget helt ut, og må logge inn med brukernavn og passord for å deretter kunne opprette en ny pinkode.

4.3.2 StopwatchPage



Figur 4.6 Stoppeklokke (lyst tema)



Figur 4.7 Startet stoppeklokke

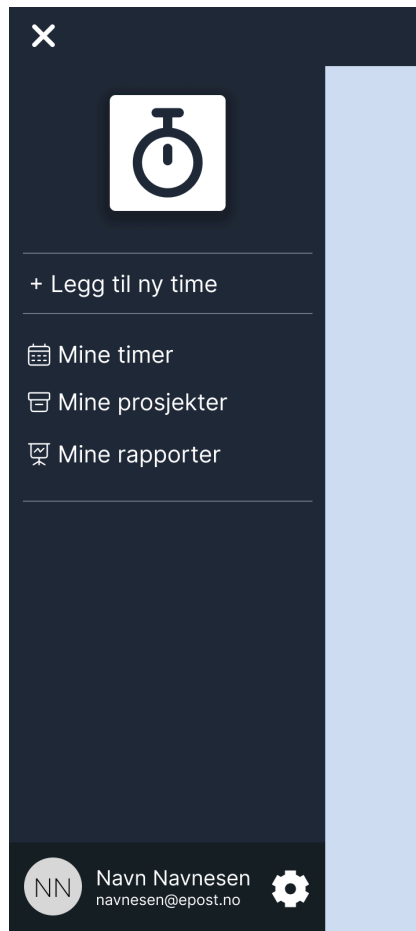
I motsetning til nettsiden har stoppeklokkefunksjonen en egen side i mobilappen. Den har her fått et nytt design inspirert av logoen til bedriften, som minner om et øye. Logoen vises også på denne siden ettersom at dette er forsiden.

Her ble det også valgt å implementere en ny funksjon der en bruker vil kunne oppgi prosjektet på forhånd. De vil dermed ikke trenge å velge dette når de stopper stoppeklokken for å legge til arbeidstiden.

Det skal også være mulig for brukere å kunne resette stoppeklokken. Grunnlaget for denne funksjonen er at det er lettere for brukere å starte stoppeklokken ved en feiltakelse på mobile touch-skjermer. Uten denne vil en bruker først måtte legge til timen for så å slette den. Dette kan oppleves som irriterende for brukerne, og mer krevende å utføre på mobil løsningen.

Det er viktig at stoppeklokkefunksjonen er lett tilgjengelig, ettersom denne er en av de mest brukte funksjonene i løsningen. Det er også et ønske fra oppdragsgiver at denne er tilgjengelig direkte etter innlogging.

4.3.3 NavBar

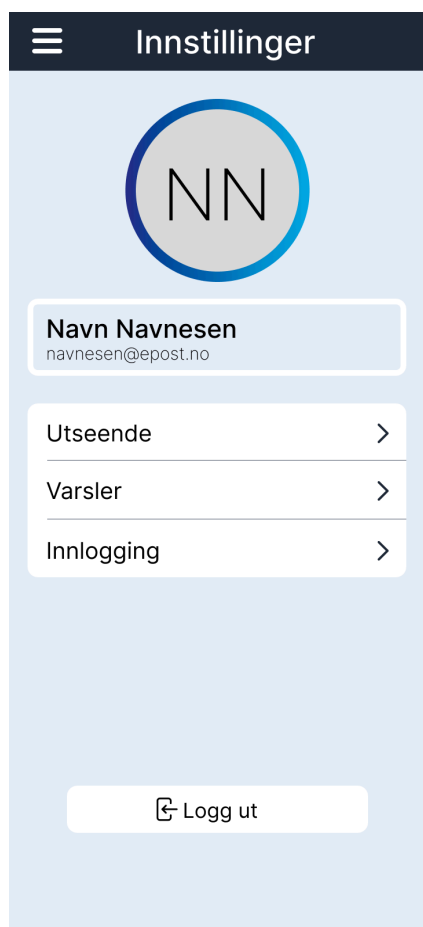


Figur 4.8 meny

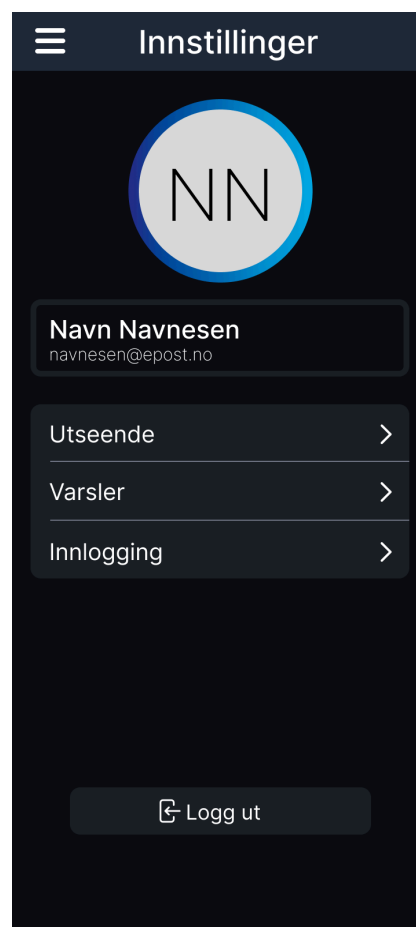
Menyen har for det meste det samme utseende som nettsiden, men med noen utvalgte endringer. Ettersom stoppeklokken har sin egen side, er denne representert som en firkantet knapp i menyen. Grunnen til at denne ikke er rund er for at brukere ikke skal forveksle denne med den faktiske funksjonen.

Et problem ved mobiløsningen er at den ikke har tilgang til de samme symbolene som web løsningen, dette er fordi web løsningen bruker element-plus. Element-plus er et bibliotek med ferdiglagde komponenter laget med html og css (Element Plus, u.å.). Siden en native-mobilapp ikke støtter html, men i stedet krever utvikling med egne native-komponenter, er det ikke mulig å bruke den samme løsningen som i web løsningen.

4.3.4 ProfileSettingsPage



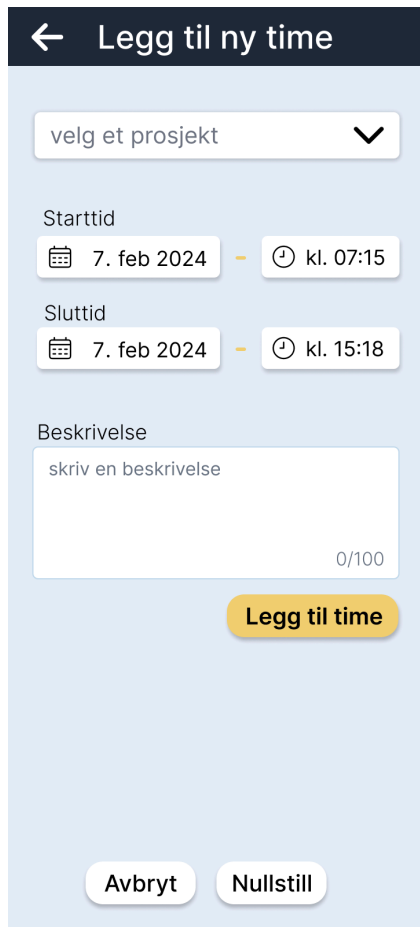
Figur 4.9 profilside (lyst tema)



Figur 4.10 profilside (mørkt tema)

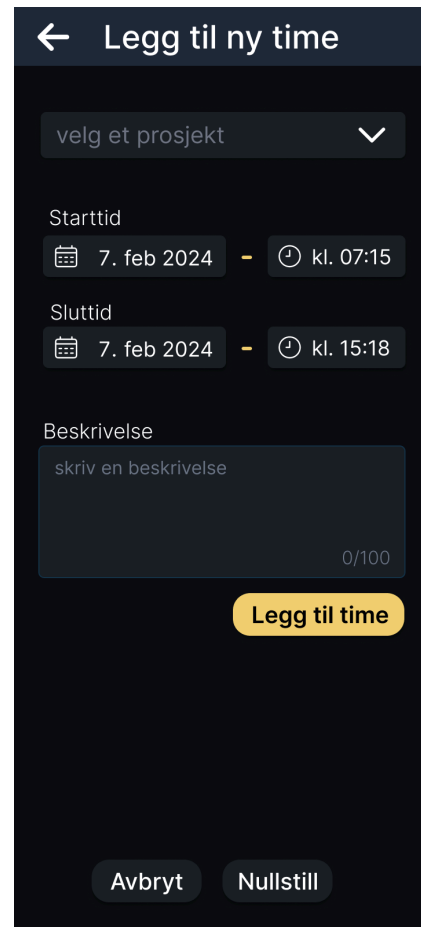
Innstillinger og profilinformasjon ble kombinert til en side. Den ansatte vil kunne se initialene sine omringet av den samme rammen som er brukt på innloggingssiden, Siden en bruker automatisk blir "logget ut", ble det valgt å plassere en logg ut knapp her istedenfor i menyen. Ved å trykke på denne knappen vil en bruker bli fullstendig logget ut, og må da logge inn med brukernavn og passord neste gang de ønsker å bruke appen. Gjennom innstillingene vil en ansatt kunne bestemme tema, varsler fra stoppeklokken, aktivere biometrisk innlogging og endre pinkode. Disse funksjonene har lavest prioritet innenfor prosjektet(SE VEDLEGG 9.2.21, 9.2.22, 9.2.23).

4.3.5 AddHourPage



The screenshot shows a mobile application interface for adding a new hour. The title bar is dark blue with a white back arrow and the text "Legg til ny time". Below the title bar is a white dropdown menu labeled "velg et prosjekt". The "Starttid" section includes a date picker set to "7. feb 2024" and a time picker set to "kl. 07:15". The "Sluttid" section includes a date picker set to "7. feb 2024" and a time picker set to "kl. 15:18". Below these is a text input field for "Beskrivelse" with the placeholder "skriv en beskrivelse" and a character count "0/100". A yellow button labeled "Legg til time" is positioned below the text field. At the bottom, there are two white buttons: "Avbryt" and "Nullstill".

Figur 4.11 Legg til time (lyst tema)



The screenshot shows the same mobile application interface as Figure 4.11, but in a dark theme. The title bar is dark blue with a white back arrow and the text "Legg til ny time". The dropdown menu, date and time pickers, and text input field are all dark grey. The "Legg til time" button is yellow, and the "Avbryt" and "Nullstill" buttons are white.

Figur 4.12 Legg til time (mørkt tema)

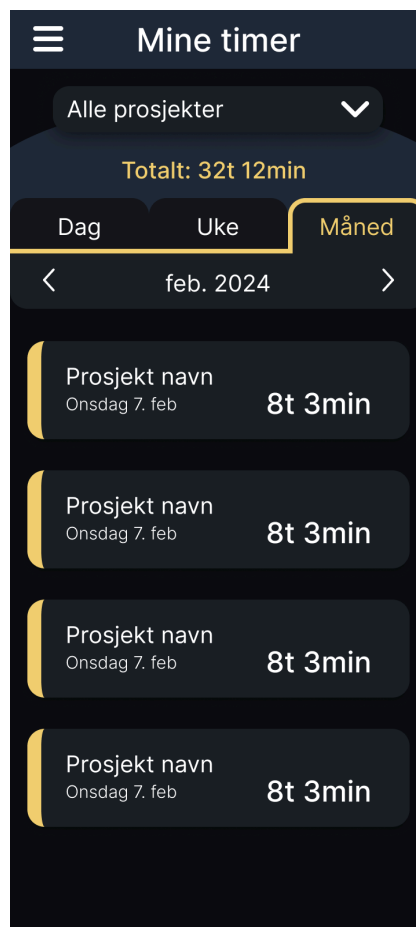
På denne siden vil en bruker kunne registrere arbeidstiden for en spesifikk dag og prosjekt. Hvis en bruker har valgt et prosjekt på forhånd for stoppeklokken, vil dette være valgt som standard. De vil ha mulighet til å bekrefte innholdet og legge til beskrivelse for å legge til ny time.

Dersom bruker velger å legge til en ny time fra navigasjonsmenyen, vil denne siden ikke ha et valgt prosjekt, og start- og sluttiden vil ta utgangspunkt i brukers nåværende dato og klokkeslett.

4.3.6 HoursPage



Figur 4.13 Oversikt over timer (lyst tema)



Figur 4.14 Oversikt over timer (mørkt tema)

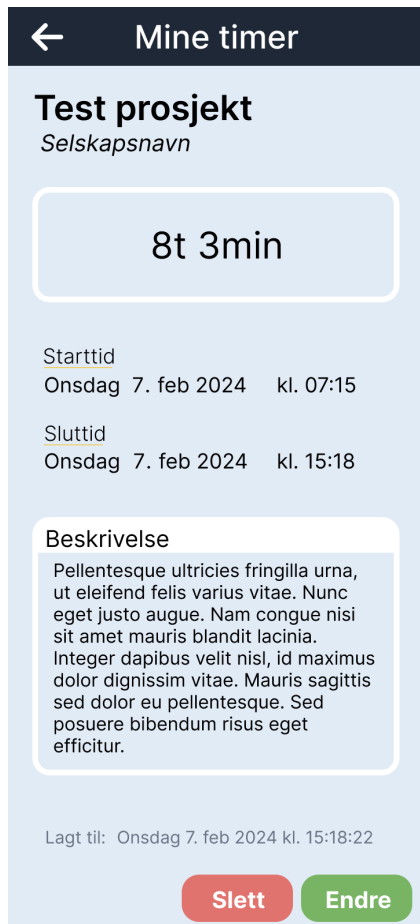
Gjennom denne siden vil ansatte få en oversikt over registrerte arbeidstimer.

Tabellen som brukes i dagens web-løsning for å vise registrerte timer, har her fått et mobilvennlig design i form av en liste. Hver økt vil vises som et eget listeelement, med informasjon om hvilket prosjekt denne tilhører, antall timer og minutter og hvilken dag timen er registrert for. En ansatt vil kunne filtrere oversikten basert på dag, uke eller måned, og der totalt antall arbeidstimer for den valgte perioden vises ovenfor.

Gjennom en nedtrekksmeny vil det også være mulig å sortere listen etter et spesifikt prosjekt, der total tiden vil vises for det valgte prosjektet.

Videre skal en ansatt kunne trykke på en registrert arbeidstid for å kunne se mer informasjon om denne.

4.3.7 HourPage



Figur 4.15 mine timer (lyst tema)



Figur 4.16, mine timer (mørkt tema)

På denne siden vil en ansatt kunne få detaljert informasjon om en enkelt arbeidstime. De vil blant annet kunne se start- og sluttid, hvilket selskap prosjektet tilhører, beskrivelsen de la til og når timen ble registrert. Her vil det også være mulig å endre og slette timen, så lenge det er innenfor 10 minutter etter at den er lagt til. Tekst har fått en ramme for å vise at de ikke skal kunne trykkes på.

4.3.8 ReportsPage

Rapport siden har fått samme design som HoursPage, bare her uten totaltid (se VEDLEGG 9.2.19). Det er heller ikke nødvendig for en bruker å kunne sortere etter prosjekt ettersom disse blir summert her.

I fremtiden skal det også være mulig å trykke på en rapport for å sende denne videre, men denne funksjonaliteten er ikke et krav i dette prosjektet.

4.3.9 ProjectsPage



Figur 4.17 mine prosjekter (lyst tema)



Figur 4.18 mine prosjekter (mørkt tema)

Prosjektsiden har samme design som de andre listene, men med omvendt farger. Her er designet også lagt opp for at en bruker skal kunne trykke på et prosjekt for å få mer informasjon. Denne funksjonen er heller ikke et krav og blir ikke implementert i dette prosjektet.

5. RESULTATER

I dette kapitlet handler om hvilke metoder som ble brukt under evalueringen av prosjektet og resultatene av disse. Evalueringsmetodene vil først bli forklart gjennom hva de handler om og hvordan de ble brukt. Deretter vil resultatene bli presentert.

5.1 Evalueringsmetode

5.1.1 Kontinuerlige tilbakemeldinger

Etter hver iterasjon under utviklingen presenterte gruppen sitt arbeid til oppdragsgiver via faste møter. Det ble diskutert farger, funksjonalitet og eventuelle utfordringer som ble møtt på i sprinten. Gruppen viste wireframe-skisser av brukergrensesnittet for godkjenning, og sendte inn ulike valgalternativer av detaljerte designforslag.

I møtene med veileder ble det også diskutert hva som har blitt gjort, og de ble gitt ulike forslag til hvordan prosjektet kan gjennomføres.

5.1.2 Brukertestning

1. Legg til ny time med stoppeklokken for Test prosjekt med en liten beskrivelse
2. Legg til en ny time for prosjektet den 2 . mars fra kl. 7:01 til 14:02
3. Finn hvor mange timer som ble registrert for Test prosjekt i uke 17. Skriv svaret her: _____
4. Finn timen registrert 2. februar for prosjektet Hihahaha Skriv svaret her: _____
5. Har prosjektet Hihahaha betalt lunsj? Ja <input type="checkbox"/> Nei <input type="checkbox"/>

Figur 5.1 brukertestning

Disse testene gjennomføres på to ulike måter; moderert brukertestning og umoderert brukertestning.

Moderert brukertestning "innebærer at en moderator guider brukerne gjennom oppgaver mens de observerer og tar notater" (Pixel Studio, 2023). Her vil deltakerne bli gitt et sett med 5 oppgaver som de skal utføre eller finne svaret på gjennom å bruke applikasjonen (se figur 5.1). Disse oppgavene er laget basert på ulike brukstilfeller og tilgjengelig funksjonalitet i appen, for å teste hvordan brukere navigerer og forstår applikasjonen. Deltakerne vil bli gitt en mobiltelefon der de får tilgang til applikasjonen, sammen med oppgavesettet på en iPad. Målet med denne testen er å finne mulige svakheter i løsningen, og notere brukeropplevelsen rundt tilgjengelige funksjoner og designvalg.

Umoderert brukertesting handler om å la deltaker utføre testen alene (NTNU, 2023). Denne utføres digitalt uten guide. I dette prosjektet blir denne metoden kombinert med evaluering gjennom SUS-skjema. "The System Usability Scale (SUS)" er en enkel skala med ti elementer som gir et globalt syn på subjektive vurderinger av brukervennlighet (Brooke, 1996). Dette betyr at skjemaet vil kunne gi et generelt innblikk i brukernes meninger om løsningen. Målet med denne testen er å se om applikasjonen møter forventningene til oppdragsgiveren, og om den dermed tilfredsstiller behovene og kravene fra problemstillingen.

Det ble også utført en spontan brukertest for skuff-menyen, der en person med erfaring innenfor apputvikling ble spurt hva en bruker må trykke på for å navigere til stoppeklokke-funksjonen.

5.1.3 Integrasjonstesting

Denne type testing handler om hvordan to ulike moduler interagerer med hverandre og tester om de fungerer riktig. Gir riktig resultat. Identifisere om det oppstår problemer når moduler kombineres. (GeeksforGeeks, 2024a)

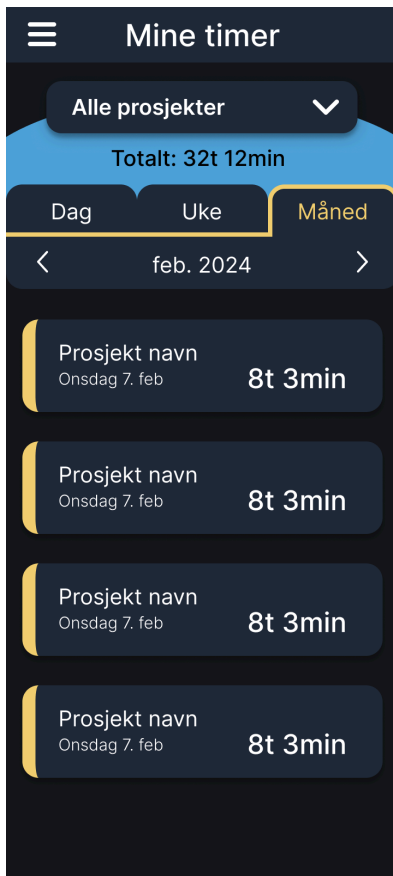
Skrive litt om hvordan vi utførte dette. Mest sannsynlig hybrid testing. Varierer mellom å teste "top-level" moduler og "bottom-level" moduler (samme kilde som over).

5.1.4 System Testing

Denne handler om å teste og evaluere den generelle funksjonaliteten til en fullt integrert programvareløsning, for å se om denne møter forventningene til systemet. (GeeksforGeeks, 2024b). Mest brukt er "Performance Testing" som tester hastighet, skalerbarhet, stabilitet og pålitelighet for programvaren. Testene er utført på Android og iOS, og er basert på de funksjonelle kravene for prosjektet.

5.2 Evalueringsresultat

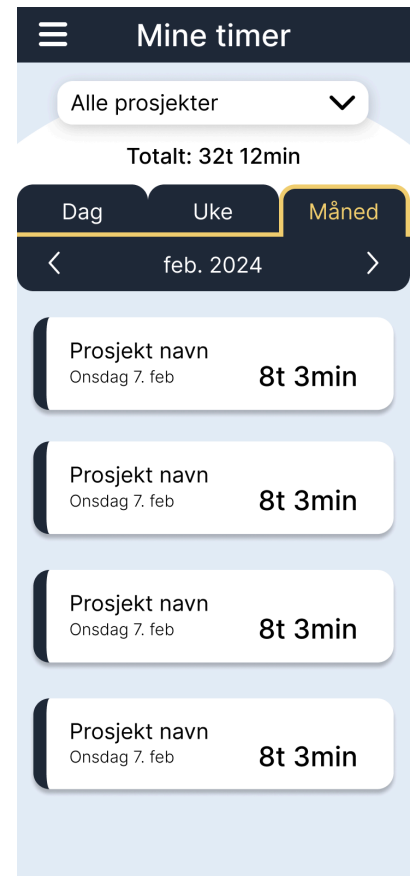
5.2.1 Kontinuerlige tilbakemeldinger



Figur 5.2 valgt løsning (mørkt tema)



Figur 5.3 alt. løsning med gul tekst

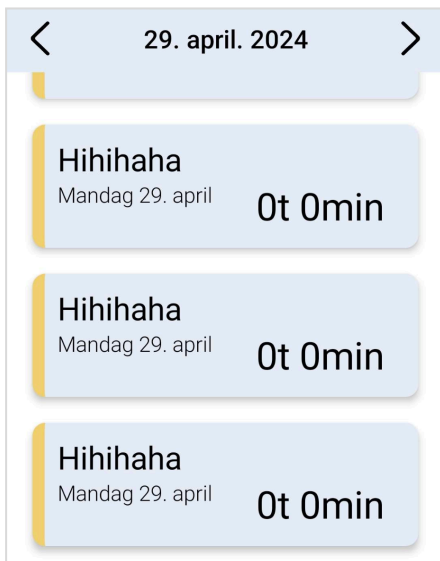


Figur 5.4 alt. løsning med kantlinje

Oppdragsgiveren var fornøyd med skissene som ble vist av brukergrensesnittet, og hadde ingen ønsker om endringer.

Gruppen fikk positive tilbakemeldinger om de detaljerte designene, der de tre figurene vist over ble utpekt som favoritter. Spesielt likte de den gule fargen, og ønsket å ha den gule kantlinjen som markerer valgt tidsperiode på begge tema-alternativene (figur 5.2 og 5.4).

5.2.2 Brukertesting



Figur 5.5 brukertest, oppgave 1



Figur 5.6 brukertest, oppgave 2

Moderert brukertest

Fire medstudenter ble valgt ut til å delta i den modererte brukertesten. Se vedlegg 9.2.6, 9.2.7, & 9.7.8 for utfylte skjema.

Generelt oppdaget gruppen gjennom brukertestene at applikasjonen var treg, og at det dermed tok lang tid før den navigerte videre.

I oppgave 1 var det en deltaker som klarte å legge til 3 nye timer med samme innhold (se figur 5.5). De andre deltakerne hadde ingen problemer med denne oppgaven.

Under oppgave 2 brukte de fleste deltakerne litt tid på å finne ut hvor de kunne legge til en blank time, men de virket ikke frustrerte over å finne denne i menyen. Alle klarte å legge til riktig tid, utenom en deltaker som glemte å endre dato for sluttid (figur 5.6).

Det riktige svaret for oppgave 3 var 0 timer og 22 minutter, der halvparten av deltakere svarte riktig. En deltaker sorterte ikke etter prosjektet, og beregnet tiden manuelt (vedlegg 9.2.6). Den andre deltakeren som svarte feil trykket ikke på "velg" knappen i dropdown-menyen for å sortere prosjektene, og skrev dermed ned total tiden for alle prosjektene den dagen (vedlegg 9.2.8). Ingen av deltakerne gikk til "ReportsPage" for å finne svaret.

Alle deltakerne svarte riktig på oppgave 4, men det var vanskelig for den første deltakeren å finne fram til riktig dato. Dette var fordi det da ikke var mulig å velge en spesifikk dato gjennom en kalender-meny. Denne var tilgjengelig for de andre deltakerne.

I den siste oppgaven ble det observert at alle deltakerne prøvde først å trykke på individuelle registrerte timer for å se om denne hadde betalt lunsj. Når de da ikke fant svaret på denne siden, utforsket de applikasjonen videre og fant svaret ganske raskt. Alle svarte derfor riktig. Flere prøvde også å trykke på individuelle rapporter og prosjekter, og lurte på hvorfor disse så trykkbare ut.

Umoderert brukertest

	Veldig Uenig	Noe Uenig	Verken enig eller uenig	Noe enig	Veldig enig
1. Jeg tror jeg vil bruke dette verktøyet ofte.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2. Jeg syntes appen var unødvendig komplisert.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Jeg syntes appen var enkel å bruke.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4. Jeg tror jeg trenger støtte fra en utvikler for å bruke denne appen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Jeg syntes de ulike funksjonene i appen fungerte bra sammen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6. Jeg syntes det var for mye ustabilitet i appen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Jeg tror de fleste vil lære seg å bruke appen raskt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8. Jeg syntes appen var veldig vanskelig å bruke.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Jeg følte meg veldig trygg på å bruke appen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. Jeg trengte å lære meg mye før jeg kunne bruke appen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

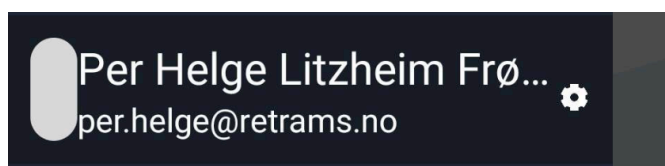
Hvor sannsynlig er det at du anbefaler denne appen til andre (Vennligst sett ring rundt svaret ditt)

Ikke sannsynlig 0 1 2 3 4 5 6 7 8 9 **10** Veldig sannsynlig

Figur 5.7 SUS-skjema oversatt fra engelsk, med resultater fra oppdragsgiver

Det ble bare mottatt en tilbakemelding gjennom SUS-skjemaet, der deltakeren var oppdragsgiveren. Resultatene var veldig positive (se figur 5.7).

Oppdragsgiveren sendte også skjermbilder fra appen som viste noen problemer med skalering (se figur 5.8).



Figur 5.8 Beskåret bilde som viser at menyen ikke blir skalert riktig. Se VEDLEGG 9.2.24 for fullt skjermbilde

Spontan brukertest

Deltakeren klarte ikke å finne ut hvordan man navigerer til stoppeklokkefunksjonen (fra figur 4.8). Videre kunne hen også fortelle at regler for denne meny-typen sier at alle knapper i denne navigasjonsmenyen må ha et beskrivende navn(må finne kilde).

5.2.3 Integrasjonstesting

Modul	Test	Resultat
LoginPage.vue	Logge inn med epost og passord	✓
PinCodePage.vue	Opprette pin-kode	✓
	Logge inn med pin-kode	✓
	Glemme bruker fra systemet	✓
StopwatchPage.vue	Starte stoppeklokke uten et prosjekt	✓
	Starte stoppeklokke med et prosjekt	✓
	Nullstille stoppeklokken	✓
AddHourPage.vue	Bruke stoppeklokken som start- og slutt-tidspunkt	✓
	Sette start- og slutt-tidspunkt	✓
	Velge prosjekt fra menyen	✓
	Nullstille skjema	✓
HoursPage.vue	Vis alle timer for et prosjekt	✓
	Beregn antall timer i en periode	✓
	Endre periode	✓
	Knapper for å redigere/slette time før 10min	✓
ReportsPage.vue	Vise betalt-lunch status på et prosjekt	✓
	Endre periode	✓
ProfileSettingsPage.vue	Glemme bruker fra systemet	✓

Tabell 5.1 integrasjonstest

5.2.4 System Testing

Modul	Test	Android	iOS	Avvik
LoginPage.vue, PinCodePage.vue	Logge inn	✓	✓	
StopwatchPage.vue	Legge til en time uten slutt-tidspunkt	✓	✓	
	Nullstille stoppeklokken	✓	✓	
AddHourPage.vue	Legge til en time med slutt-tidspunkt	✓	✗	På iOS er det ikke mulig å lukke varsel-bokser som oppstår ved ugyldige valg.
	Rediger en eksisterende time	✓	✗	
HoursPage.vue	Vise alle timer i en periode	✓	✓	
HourPage.vue	Vise detaljer om en time	✓	✓	
	Redigere en time innen 10min	✗	✗	Knappen for å redigere timen blir utilgjengelig dersom skjermen er for liten, eller om beskrivelsen er for lang.
	Slette en time innen 10min	✗	✗	På iOS klarer ikke brukeren å lukke bekreftelsen om at timen er slettet og må dermed avslutte appen for å gå videre. Slett time knappen har også samme avvik som rediger time.
ProjectsPage.vue	Vise alle mine prosjekter	✓	✓	
ReportsPage.vue	Vise alle timer på prosjekter i en valgt periode	✓	✓	

Tabell 5.2 system test

5.3 Prosjektresultat

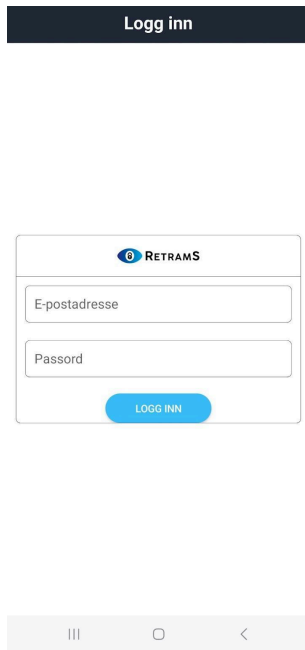
5.3.1 Funksjonelle krav

Nr	Krav	Delkrav			
		Nr.	Beskrivelse	Planlagt	Fullført
1	Appen skal fungere på iOS og Android	1.1	Appen skal fungere på iOS	✓	✗
		1.2	Appen skal fungere på Android	✓	✓
2	En bruker skal kunne bytte mellom lys og mørk modus	2.1	Appen skal ha et design for mørk-modus	✓	✓
		2.2	Appen skal kunne bytte mellom lys og mørk modus	✗	✗
3	En bruker skal kunne logge inn	3.1	En bruker skal kunne logge inn med brukernavn og passord	✓	✓
		3.2	En bruker skal kunne logge inn med firesifret pin-kode	✓	✓
		3.3	En bruker skal kunne logge inn med biometri (fingeravtrykk, ansikt, etc.)	✓	✓
4	En bruker skal kunne registrere ny time i et prosjekt			✓	✓
5	En bruker skal kunne endre eksisterende time i et prosjekt			✓	✓
6	En bruker skal kunne sette eksisterende time i et prosjekt			✓	✓
7	En bruker skal kunne vise en oversikt over alle sine registrerte timer			✓	✓
8	En bruker skal kunne vise en oversikt over registrerte timer per prosjekt			✓	✓
9	En bruker skal kunne starte en stoppeklokke for beregning av tidsperiode			✓	✓

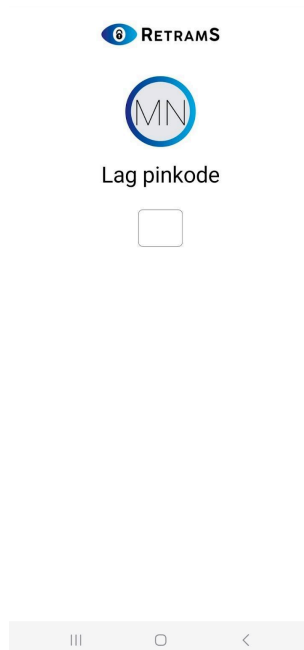
Tabell 5.3 Hentet fra VEDLEGG 9.1.2 Visjonsdokument

Tabellen ovenfor viser at iOS-versjonen ikke ble fullført. Ulike utfordringer med biometri gjorde at gruppen ikke kom i mål med dette. Utvikling og test på Android plattformen ble derfor prioritert. Tabellen viser også at alle de resterende planlagte funksjonelle kravene ble fullført.

5.3.2 Logg Inn



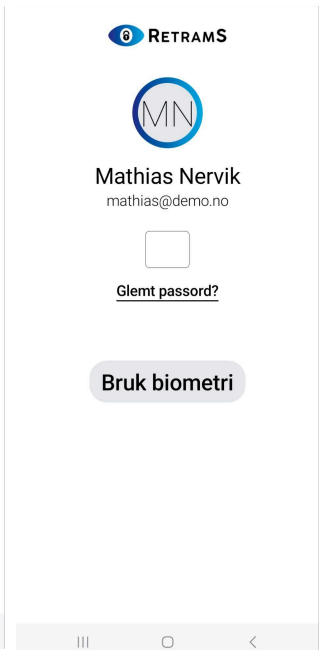
Figur 5.7 Logg inn



Figur 5.8 lag pinkode



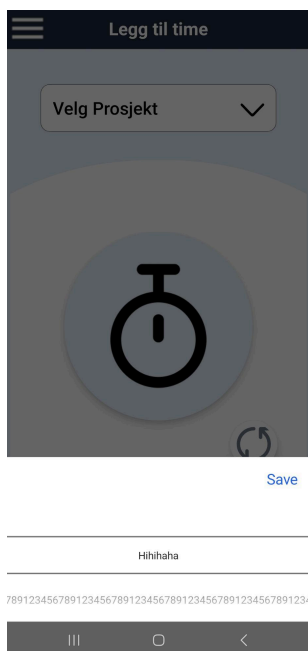
Figur 5.9 gjenta pinkode



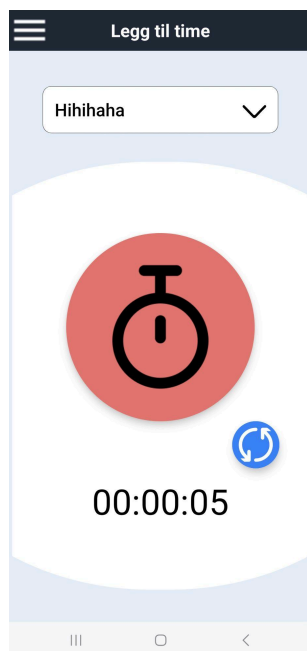
Figur 5.10 biometri

Figurene ovenfor viser “Logg inn” usecase. (9.1.3 Kravdokument). Når en bruker åpner appen for første gang blir en innloggingsside med e-post og passord vist, som vist i figur 5.7. Deretter blir brukeren sendt til “Lag Pinkode” siden, hvor brukeren må opprette en ny pin-kode. Denne siden blir alltid hoppet over, hvis brukeren allerede har opprettet en pin-kode. Deretter må brukeren repetere den samme pin-koden på nytt, dersom feil pin-kode blir tastet inn, blir brukeren sendt tilbake til figur 5.8. Etter at pin koden har blitt repetert vises den egentlige innloggingsskjermen, her kan brukeren velge å logge inn med pin-koden eller med biometri hvis dette er tilgjengelig.

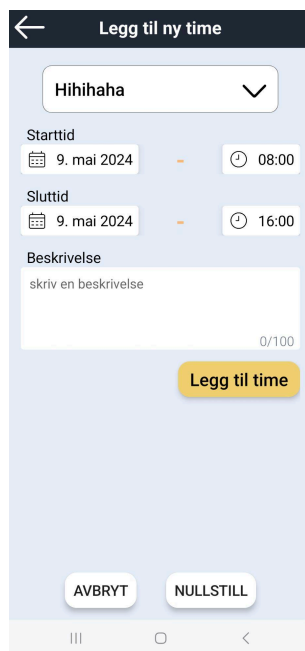
5.3.3 Legg til time



Figur 5.11 velg prosjekt



Figur 5.12 startet stoppeklokke



Figur 5.13 lagre sluttidspunkt



Figur 5.14 timer som er registrert

Figurene ovenfor viser “Legg til time” usecase (9.1.3 Kravdokument). Når en bruker har logget inn vises skjermen, som vist i figur 5.11. Her kan en ansatt velge et prosjekt før stoppeklokken blir startet, en ansatt kan deretter starte stoppeklokken ved starten av arbeidsdagen. Stoppeklokken endrer farge ved start for å illustrere at stoppeklokken nå har blitt startet. Når arbeidsdagen er ferdig kan en ansatt klikke på stoppeklokken igjen, for å lagre sluttidspunkt for timen. Etter at sluttidspunktet har blitt registrert, som vist i figur 5.13, vises en liste av alle timene registrert denne måneden, som vist i figur 5.14.

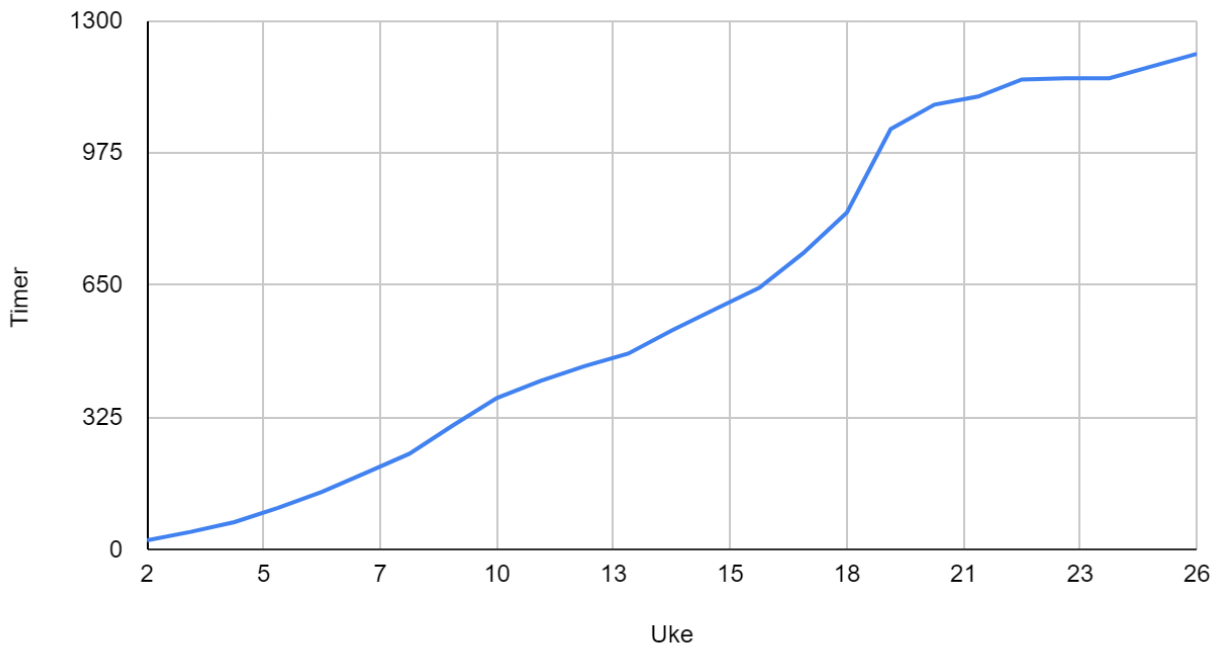
5.3.4 Ikke-funksjonelle krav

Nr	Beskrivelse	Planlagt	Fullført
1	Programmet skal skrives i NativeScript-Vue med NodeJS	✓	✓
2	Programmet skal koble til et eksisterende API	✓	✓
3	Programmet skal være brukervennlig å bruke	✓	✓
4	Bruker skal kunne logge inn med biometri eller pin-kode	✓	✓
5	Bruker skal kunne registrere timer uten å føle å møte hinder	✓	✓

Tabell 5.4 Hentet fra VEDLEGG 9.1.2 Visjonsdokument

5.4 Prosjektgjennomføring

Akkumulert timeforbruk



Figur 5.15 akkumulert timeforbruk for prosjektet

Grafen viser akkumulert timeforbruk for gruppen gjennom hele prosjektet. (figurtekst)

Generelt tidsforbruk i prosjektet ligger under forventet tidsforbruk som er på 1620 timer. Underveis i prosjektet ble dette identifisert og iverksatt tiltak, men gruppen endte likevel på litt under forventet tidsforbruk.

Gruppen gjennomførte en risikoanalyse i starten av prosjektet for å identifisere ulike risikoer gruppen kunne møte på. Basert på denne analysen implementerte gruppen flere tiltak for å forhindre at disse skulle påvirke prosjektet negativt.

Til tross for forebyggende tiltak, oppsto det likevel noen av de identifiserte risikoene underveis, noe som krevde ytterlige inngrep for å holde prosjektet på rett kurs.

6. DISKUSJON

I dette kapitlet blir det diskutert hvordan gruppen opplevde gjennomføringen av prosjektet, hvilke resultater som ble oppnådd, valg av design og sikkerhet. Gruppen går over hva som gikk bra og hva som kunne blitt gjort annerledes.

6.1 Utvikle for to systemer

Å utvikle for to systemer viste seg å være krevende. Gruppen måtte først skrive koden og deretter teste på to ulike enheter, iOS og Android. Gruppen prøvde å kjøre den samme koden på begge enheter, med det fungerte sjeldent. Det var enten problemer på iOS eller Android og gruppen måtte finne ut hva som forårsaket feilen og tilpasse koden til den spesielle enheten. Dette gjorde at utviklingen tok lengre tid og gruppen måtte tilslutt bestemme seg for å bare fokusere på Android utvikling da den hadde minst feil og krevde mindre tid. Det ble derfor bare en ferdig løsning på android.

Noen av problemene gruppen møtte på under utviklingen av iOS var knyttet til godkjenning av appen for utvikling. For å kunne kjøre appen på iPhone etter endringer i koden, måtte gruppen godkjenne Apple-development i innstillinger på iPhone. Dette var et problem som oppsto ofte og var tidkrevende, noe som gjorde utviklingen mindre fleksibel.

6.2 Nativescript Vue

Å utvikle en app i nativescript-vue viste seg å være litt krevende. Medlemmene i gruppen hadde ingen tidligere erfaringer, hverken med vue eller med nativescript, men noen av gruppemedlemmene hadde erfaring med react-native, som er ganske likt nativescript-vue, men det krevde fortsatt en del ekstra arbeid.

Tidlig i prosjektet ble det prioritert å sette opp et utviklingsmiljø og tilrettelegge for videre utvikling. Dette viste seg å være en god prioritering ettersom gruppen på flere tidspunkt møtte på problemer knyttet til utviklingsmiljøet, som derfor tidlig kunne bli håndtert.

Den eksisterende løsningen bruker en innebygd routing-funksjon for å navigere mellom de forskjellige sidene. Dette er en kjernefunksjon i vue, men en funksjon som ikke ennå er kompatibel med Nativescript-vue. Dette krevde en total ombygging av den eksisterende kodebasen for å lage en tilsvarende funksjonalitet. I tillegg brukte den eksisterende løsningen bibliotekene *pinia* og *axios*, dette er to biblioteker som

6.2.1 Biblioteker

I den eksisterende løsningen ble det brukt to biblioteker *pinia* og *axios*, som henholdsvis er kode for å definere gjenbrukbar logikk og hjelpefunksjoner for å koble seg til et API. Disse to bibliotekene er ikke kompatible med nativescript og krevde derfor en total ombygging, derfor ble en del ekstra tidsbruk for å få dette til å fungere optimalt i nativescript.

6.3 Sikkerhet

6.4.1 Pin-kode og biometri

Under utviklingen av mobilapplikasjonen ble det tatt en avgjørelse om at all biometri bruker den innebygde funksjonaliteten fra den mobile enheten. Det ble derfor mindre komplekst under utviklingen, ettersom gruppen trengte kun å bruke eksisterende kode for å verifisere en ansatt med biometri.

På grunn av vanskeligheter med kryptering av en kort tallkode, ble det derfor valgt å lagre pin-koden direkte på mobilenheten, som betyr at når en ansatt taster inn en pin-kode, blir denne sammenlignet med den verdien som er lagret i minnet på telefonen. Dette utgjør en sikkerhetsrisiko fordi pin-koden blir ikke kryptert når den lagres. Applikasjonen er derfor sårbar mot angrep, der det kan bli mulig å stjele data. Sikkerhetsbruddet kan oppstå dersom en bruker laster ned en app som inneholder skadelig programvare, som kan brukes til å stjele brukerinformasjon (Ahvanooy, Rabbani, Rajput, 2017).

Sikkerhetsrisikoen er imidlertid lav fordi databasen ikke inneholder veldig sensitiv informasjon, som ikke allerede er tilgjengelig på bedriftens nettside. Derimot kan dette være et problem dersom bedriften ønsker å holde de ansatte hemmelig, for eksempel for å beskytte dem mot personer med kontaktforbud. Det kan også være en risiko hvis sikkerhetsbruddet skjer for ansatte som jobber med hemmelige prosjekter, da det blir mulig å hente navnet og beskrivelser for registrerte timer under dette prosjektet.

På en annen side, er det ikke mulig å utgjøre stor skade på systemet eller bedriften gjennom tilgang til applikasjonen. Videre, siden pin-koden kun lagres i minnet og ikke i en database, vil den bare være gyldig for mobilenheten den ble opprettet på. Hvis en bruker ønsker å logge inn på en annen mobiltelefon, må de derfor opprette en ny pin-kode for denne enheten. Fordelen er at pin-koden derfor ikke har noen verdi utenfor den tilhørende mobilen. Tilgang til applikasjonen beskyttes også gjennom at mobiler ofte har en egen låseskjerm. Spørsmålet er dermed om det er nødvendig å kreve brukerautentisering for å få tilgang til løsningen.

Det er flere applikasjoner der det bare er nødvendig å logge inn en gang. Dette er gjeldende for de fleste sosiale medier, som for eksempel Facebook og TikTok. De er derfor utsatt for angrep dersom telefonen blir kompromittert av skadelig programvare.

Planday er en applikasjon som er identifisert som en lignende løsning på prosjektets problemstilling. Gjennom denne appen kan ansatte stemple inn når de ankommer

arbeidsplassen for å registrere arbeidstid. Den krever heller ikke at bruker må autentisere seg hver gang de skal bruke mobilappen.

Et problem som kan oppstå er hvis en ansatt låner bort mobilen, spesielt til barn som ikke alltid forstår konsekvensene av sine handlinger.

Gjennom Planday er det bare mulig å stemple inn dersom man er satt opp på en vakt, der den også bruker GPS for å sjekke om den ansatte er på arbeidsplassen (Planday, u.å.). Dersom et barn åpner Planday, vil de derfor ikke ha mulighet til å opprette nye arbeidstimer eller stemple inn.

Timekontroll derimot, gjør det mulig å starte stoppeklokken når som helst på døgnet uavhengig av lokasjon, der det også er mulig å legge til flere timer på en dag. Stoppeklokken er veldig stor og brukervennlig, og kan derfor være veldig lett og underholdende for små barn å legge til masse timer. Selv om dette ikke er farlig, er det fortsatt irriterende å måtte fikse.

Det er derfor nødvendig å kreve pin-kode for å få tilgang til løsningen. For videreutvikling bør denne krypteres. Videre burde det også bli gjort en vurdering om denne skal lagres i minnet på mobilenheten eller i en database.

6.4.2 Forebygging av vanlige sårbarheter

SQL-injection og cross-site scripting (XSS) er to sikkerhetstrusler som må håndteres.

XSS er primært en risiko for webapplikasjoner, hvor skadelig kode kan sendes inn i nettsider og vises for andre brukere. Siden vår løsning er en mobilapp, er dette en svært lav risiko (Gupta, Gupta, 2017).

SQL-injection er en teknikk hvor uautoriserte brukere kan skrive database-kode i et tekstfelt for å få direkte tilgang til databasen. Dette er motvirket i vår løsning på tilsvarende måte som i den eksisterende web-løsningen (Patel, et al., 2018).

6.4 Valg av design

6.4.1 Valgt navigasjonsmetode

Valg av navigasjonsmetode ble diskutert mye mellom gruppen. Spesielt var det skuff-meny og kombinert meny som ble vurdert. Da gruppen prøvde å tegne et brukergrensesnitt for kombinert meny, ble det oppdaget at det var vanskelig å bestemme innholdet i de to menyene. Det ble også nevnt under diskusjon av alternativene, at det ikke var nok funksjoner til å fylle begge menyene. Gruppen ville derfor brukt mye tid på å løse dette problemet, som kunne hatt store konsekvenser for prosjektgjennomføringen.

Det oppstod noe problemer når menyen skulle utvikles. Ved utvikling av en tilsvarende løsning i Android og iOS uten å bruke et rammeverk, er det mulig å koble seg direkte til historikken til enheten. Dette ble en utfordring når vi skulle navigere fra en side til en annen, ettersom menyen alltid ble holdt åpen dersom en bruker klikket på

tilbakeknappen. Det oppstod også en utfordring når menyen skulle inkluderes på flere sider, det viste seg at menyen krevde en spesifikk implementasjon for å sikre at elementene ble vist som planlagt. Til slutt oppstod det også en utfordring hvor tittel-elementet på mobilappen alltid ble vist over skuff-menyen. Denne utfordringen ble løst ved å konstruere et helt nytt tittel-element som alltid ble tegnet bak skuff-menyen.

Under brukertestene ble det observert at deltakerne holdt mobilen med to hender når de utførte oppgavene. Dette betyr at tommelfinger-rekkevidden ikke er et stort behov for å bruke applikasjonen. En navigasjonsbar vil også på bunnen av skjermen, og derfor gi mindre plass for de andre funksjoner.

Resultatene fra systemtestene viser også applikasjonen har problemer med skalering der elementer kan forsvinne. En navigasjonsbar ville videre forverret disse problemene fordi den krever en plass kilde her finn. Fordelen med å velge en skuff-meny er derfor at gruppen har hatt større område å jobbe med.

Som vist under detaljert design, var det planlagt å ha en knapp uten tekst for å navigere til stoppeklokkefunksjonen. Designet ble laget for å ligne på web-løsningen, der stoppeklokkefunksjon er tilgjengelig i menyen, og hadde derfor bare et estetisk grunnlag. Deltakeren i den spontane brukertesten trodde at denne knappen var en logo, og klarte derfor ikke å finne riktig svar. Siden designet også ikke følger retningslinjene for navigasjonsmetoden, ble det besluttet å endre designet. Dette var en enkel og rask endring, og hadde derfor ingen konsekvens for prosjektgjennomføringen. Endringen har også ført til at menyen nå har mer plass til nye funksjoner.

6.4.2 Brukertesting

Testresultatene i de modererte brukertestene ble påvirket av at applikasjonen var treg, men dette viste seg å være positivt. Deltakerne ble også ikke fortalt dersom de svarte feil, slik at testen fikk realistiske svar.

Som nevnt klarte en deltaker å legge til 3 timer fra samme stoppeklokke-økt. Siden applikasjonen ikke responderte med en gang, ble det mulig for deltakeren å trykke på "legg til"-knappen flere ganger. Gruppen løste dette ved å deaktivere knappen etter at den er trykket på, og dersom alle detaljene i skjemaet er gyldige.

Testene ble også påvirket av rekkefølgen av oppgavene. Etter å ha lagt til en ny time vil en bruker bli ført direkte til timeoversikten. Siden testen startet med å legge til en time via stoppeklokkefunksjonen, ble det ikke nødvendig å forlate denne siden deltakerne før på siste oppgave. I retrospekt, burde brukertesten startet med å la deltakerne utforske løsningen før de begynte med oppgavene.

Opgavesettet ble redigert etter den første testen slik at deltakerne fikk bedre plass å skrive på. Utførelsen av brukertesten ble også gjort annerledes for hver deltaker. De to første deltakerne leste ikke spørsmålene nøye og svarte derfor feil. Moderatoren måtte derfor lese spørsmålene høyt.

I oppgave 3 la den første deltakeren ikke merke til at totaltid sto opplyst øverst på "HoursPage". Derimot etter å ha fullført testen fikk de lov til å undersøke løsningen, og la da merke til at de kunne bruke dette for å svare på oppgaven. De andre deltakerne la merke til denne ganske fort, og det ble derfor vurdert som et vellykket designvalg. Det ble også oppdaget at deltakerne ønsket å kunne velge et prosjekt fra dropdown menyen ved å bare trykke på valget. De syntes det var tungvint å måtte trykke på et prosjekt for så å trykke på lagre knappen. Det ble ikke tid til å fikse dette problemet, og må dermed løses under videre utvikling av applikasjonen.

Etter å ha lagt til en kalender-meny, ble oppgave 4 lettere å løse for de neste deltakerne. De løste derfor denne oppgaven ganske fort som viser at løsningen er effektiv.

I den siste oppgaven ble det observert at alle deltakerne prøvde først å trykke på individuelle registrerte timer for å se om denne hadde betalt lunsj. Derfor er det mulig det er nødvendig å legge til denne informasjonen her også.

Et mål med designet var at rapporter og prosjekter skulle se trykkbare ut. Siden flere av deltakerne prøvde å trykke på disse, har dette designet derfor vært vellykket.

De modererte brukertestene var vellykket, på tross av at det oppsto problemer.

Gruppen hadde også planer om å besøke arbeidsplassen til Retrams, men dette ble ikke mulig på grunn av dårlig vær og kollisjoner i kalenderen. Gruppen fikk til et fysisk møte på skolen, men dette var bare med Per Helge. Videre var det også meningen å få flere svar på SUS-skjemaet, slik at gruppen kunne bedre vurdere om løsningen var vellykket.

6.4.3 Farger / Tema

Gruppen valgte å beholde og bruke den originale mørke blåfargen fra web-løsningen på hovedmenyen.

Etter tilbakemeldinger om detaljerte design, valgte gruppen å bruke den gule fargen mer i designet siden de likte denne så godt.

Gruppen fikk derimot problemer med å bruke den sterke blåfargen vist i figur 5.2. I utgangspunktet hadde gruppen planlagt å bruke den samme blåfargen som ble brukt i menyen for store deler av mørk-temaet. Dette skapte problemer når resten av designet skulle lages. Det ble derfor bestemt en mørkere bakgrunnsfarge, og elementene har fått en gråere blåfarge. Sirkelen på HourPage har samme farge som menyen for begge temaene.

Til sammen ble disse valgene og vurderingene gjort for at applikasjonen skulle ligne på web-løsningen.

6.5 Utfordringer og muligheter for mobil

En av utfordringene ved dette prosjektet var å lage et brukergrensesnitt tilpasset små skjermer. Mobiltelefoner har ulike begrensninger på grunn av størrelse og hvordan den brukes. Men den tilbyr også unike muligheter. Innlogging med pinkode og biometri er eksempler på funksjoner som er mulig å implementere på grunn av mobiltelefon.

Mindre skjerm betydde også at det var vanskelig å gjøre noen av funksjonene fra web løsningen tilgjengelig i applikasjonen. Det var derfor nødvendig å implementere noen nye funksjoner.

6.5.1 Nye funksjoner

Forhåndsvelge prosjekt

Denne funksjonen ble funnet på tidlig i prosjektet når gruppen utforsket web løsningen. Denne funksjonen gjør det raskere å legge til en ny time, og som dermed gjør det effektivt å legge til ny time.

Nullstille stoppeklokken

I web løsningen er det ikke mulig å nullstille stoppeklokken. Dette betyr at dersom man starter tiden er man nødt til å legge til en ny time der man kan slette den etterpå. På web løsningen har ikke dette store konsekvenser fordi man som regel ikke kommer til å trykke på stoppeklokken med mindre man ønsker å bruke den. Mobiltelefoner derimot har berøringsskjerm, der stoppeklokkefunksjonen tar opp store deler av forsiden. Det er derfor større mulighet for brukere å aktivere denne med en feiltakelse, der de ville blitt nødt til å legge til og deretter slette økten hver gang. Det ble derfor besluttet å legge til en knapp slik at stoppeklokken kan nullstilles. Selv om denne knappen er mindre, har den også samme problem der brukeren kan berøre denne og dermed slette den tiden som er tatt. Derfor vil en bruker måtte bekrefte handlingen gjennom en dialogboks for å nullstille stoppeklokken.

Sortere etter prosjekt

I web løsningen blir de registrerte arbeidstimene kategorisert under hvilke prosjekt de tilhører. Som et forsøk på å gjenskape dette i det nye designet, ble sorteringsfunksjonen lagt til.

Notifikasjoner

Denne ideen ble funnet på sent i prosjektet, og det ble derfor ikke tid til å gjøre denne funksjonell for den endelige løsningen. Etter tilbakemelding fra oppdragsgiver ble det likevel bestemt å implementere deler, og gjøre denne klar for videreutvikling. Det er lagt

til et bibliotek i applikasjonen som brukes til å legge til notifikasjoner. Den funksjonaliteten som er at en bruker gi appen tillatelse for å kunne gi varsler. En ansatt kan velge mellom å mota varsel etter 8, 4 eller 2 timer. Valgene blir også lagret i løsningen. I den endelige løsningen vil ikke dette ha noen effekt. For å videreutvikle denne funksjonen må logikken for å sende ut varslene implementeres på stoppeklokke siden.

6.5.2 Skalering

Testene viste at noen ting i designet skalerer dårlig og må fikses i videre utvikling.

6.6 Prosjektgjennomføring

Gruppen har holdt seg til gantt skjemaet og jobbet med det som var planlagt i hver iterasjon. Alle løsningene ble ikke ferdig innenfor den forventede tidsrammen, noe som gjorde at gruppen måtte prioritere de viktigste funksjonene i prosjektet for å få tid til å ferdigstille applikasjonen. Et eksempel på dette er utviklingen på iOS som ikke ble ferdigstilt på grunn av ulike utfordringer. For å unngå dette kunne gruppen fokusert på å utvikle for kun en enhet og dermed fått bedre tid til å utvikle en mer fullstendig app med flere funksjoner enn det som var krevd. Gruppen kunne også disponert tiden bedre da den totale timeforbruken for gruppen er mindre enn kravet til bacheloren. Utviklingen kunne også startet tidligere slik at gruppen fikk bedre tid.

7. KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Målet for prosjektet var å lage en mobilapp for registrering av timer, nærmere forklart i kapittel 1.4. Tidlig i prosjektprosessen ble et utviklingsmiljø satt opp og fungerende mobilapp produsert. I samarbeid med oppdragsgiver ble prosjektet avgrenset til å kun inkludere funksjoner for en ansatt.

I løpet av prosjektperioden ble det designet flere varianter av sidene i mobilappen. Resultatet av prosjektet har vært stort sett vellykket. Gjennom en uassistert brukertest, ble mobilappen testet av oppdragsgiver. Tilbakemeldingen fra oppdragsgiver har vært svært positiv.

Ved ferdigstilling av prosjektet ble en kjørbare mobilapplikasjon for Android overlevert, og et utvidet design for dark mode. Mobilapplikasjonen ble levert med funksjoner fra web-applikasjonen, men med en optimalisert bruker-dialog for mindre skjermer.

7.2 Videre arbeid

7.2.1 iOS

Videre arbeid for iOS er å utvide biometri-innloggingen, og la brukeren komme seg videre etter bekreftelsesmeldinger.

7.2.2 Dark mode

Gruppen leverte en mobilapp designet både for- og uten dark-mode, men denne funksjonen ble ikke ferdig implementert ettersom annen funksjonalitet ble prioritert. Alt av design produsert for appen har blitt overlevert, og en implementasjon av dette vil derfor være et betydelig forbedringspotensial.

7.2.3 Leder og admin

Gruppen har diskutert med oppdragsgiver om en leder og admin funksjon som kan utvikles for å gi appen mer funksjonalitet og brukervennlighet for ledere og administratorer. Dette vil gi ledere og administratorer i bedriften økt kontroll og mer effektiv håndtering av prosjekter.

7.2.4 Pin-kode

En sikkerhetsrisiko i appen er lagringen av pin koden. Denne kan videreutvikles til en mer optimalisert lagring av pin-kode hvor koden ikke blir lagret på enheten.

7.2.5 Logo

I begynnelsen av prosjektet lagde gruppen wireframes med logoen til Retrams på forsiden med stoppeklokken. Dette ble nedprioritert og kan bli tatt med som videre arbeid.

8. REFERANSER

- Ahvanooey, M., Li, Q, Rabbani, M., & Rajput, A. (2017) *A Survey on Smartphone Security: Software Vulnerabilities, Malware, and Attacks*. Tilgjengelig fra: <https://arxiv.org/pdf/2001.09406> (Hentet: 12. mai 2024)
- Brooke, J. (1996) *SUS - a quick and dirty usability scale*. Tilgjengelig fra: https://www.researchgate.net/publication/319394819_SUS_-_a_quick_and_dirty_usability_scale (Hentet: 12. mai 2024)
- GeeksforGeeks (2024a) *Integration Testing - Software Engineering*. Tilgjengelig fra: <https://www.geeksforgeeks.org/software-engineering-integration-testing> (Hentet: 12. mai 2024)
- GeeksforGeeks (2024b) *System Testing - Software Engineering*. Tilgjengelig fra: <https://www.geeksforgeeks.org/system-testing> (Hentet: 12. mai 2024)
- Grolid, S., (2021) *Planlegg innhold for mobil*. Tilgjengelig fra: <https://ndla.no/subject:a6b56b7e-2149-4216-92b6-3095feb870f3/topic:010fd9d7-aa5a-49b6-b3ee-5887c34c97a5/resource:5955f2bf-5a3b-47f2-a9fe-026e02409f7a> (Hentet: 12. mai 2024)
- Gupta, S., Gupta B. (2017) *Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art*. <https://link.springer.com/article/10.1007/s13198-015-0376-0> (Hentet: 12. mai 2024)
- Hansen, B., Grønli, A., & Ghinea, G. (2018) *A survey and Taxonomy of Core Concepts and Research Challenges in Cross-Platform Mobile Development*. Tilgjengelig fra: 10.1145/3241739
- Interaction Design Foundation (u.å.) *what is user centered design (UDC)?* Tilgjengelig fra: <https://www.interaction-design.org/literature/topics/user-centered-design> (Hentet: 4. mai 2024)
- Lid, I. (2024) *universell utforming*. Tilgjengelig fra: https://snl.no/universell_utforming (Hentet: 12. mai 2024)
- Material Design 3 (u.å.) *Navigation bar*. Tilgjengelig fra: <https://m3.material.io/components/navigation-bar/guidelines> (Hentet: 12. mai 2024)
- NTNU (2023) *Metoder for brukerinnsikt - kvalitative*. Tilgjengelig fra: <https://i.ntnu.no/wiki/-/wiki/Norsk/Metoder+for+brukerinnstkt++kvalitative> (Hentet: 12. mai 2024)
- Patel, D., Dhamdhare, N., Choudhary, P, & Pawar, M. (2018) *A System for Prevention of SQLi Attacks*. <https://ieeexplore.ieee.org/abstract/document/9215361> (Hentet: 12. mai 2024)
- Pixel Studio (2023) *Brukertesting av system: Identifisering av brukerbehov og forbedringsmuligheter*. Tilgjengelig fra: <https://pxstudio.no/brukertesting-av-system> (Hentet: 12. mai 2024)

Planday (u.å.) *Et stemplingsur som gjør timeregistrering og lønnskjøring til en lek*. Tilgjengelig fra: <https://www.planday.com/no/stemplingsur/> (Hentet: 12. mai 2024)

Retrams (2024) *Om Retrams*. Tilgjengelig fra: <https://retrams.no/om-retrams> (Hentet: 25 januar 2024)

Ryland, S., Bibic, A. & Skårnes, Ø. (2023). *Retrams Timeføring*. HVL. Upublisert.

SSB, (2022), *Norsk Mediebarometer 2021*. Tilgjengelig fra: <https://www.ssb.no/kultur-og-fritid/tids-og-mediebruk/artikler/norsk-mediebarometer-2021> (Hentet: 3. mai 2024)

Statcounter (u.å.) *Mobile Operating System Market Share Norway*. Tilgjengelig fra: <https://gs.statcounter.com/os-market-share/mobile/norway> (Hentet: 12. mai 2024)

Taivalaari, A., (u.å.) *Apps vs. Open Web: The battle of the Decade*. Tilgjengelig fra: <https://webpages.tuni.fi/livelykernel/publications/BattleOfTheDecade-Mikkonen-Taivalaari.pdf> (Hentet: 12. mai 2024)

Element Plus, (u.å.) *Icon*. Tilgjengelig fra: <https://element-plus.org/en-US/component/icon.html> (Hentet: 12. mai 2024)

Hannemyr, G., Liestøl, G., Luders, M., Rasmussen, T. *Digitale Medier*. Universitetsforlaget 2015.

9. VEDLEGG

9.1 Støttedokumenter

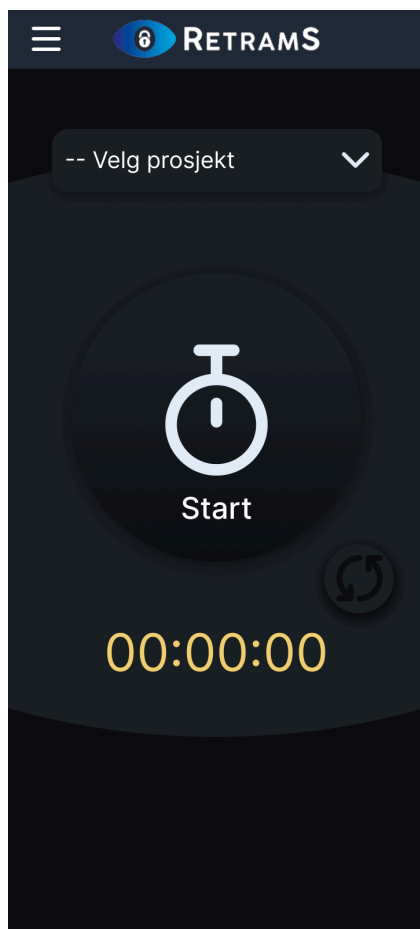
9.1.1 Prosjekthåndbok

9.1.2 Visjonsdokument

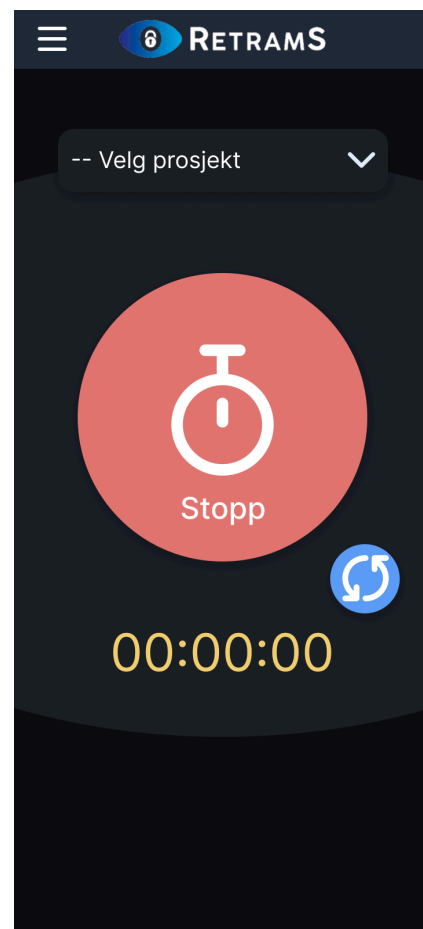
9.1.3 Kravdokument

9.1.4 Systemdokumentasjon

9.2 Figurer og Skjema



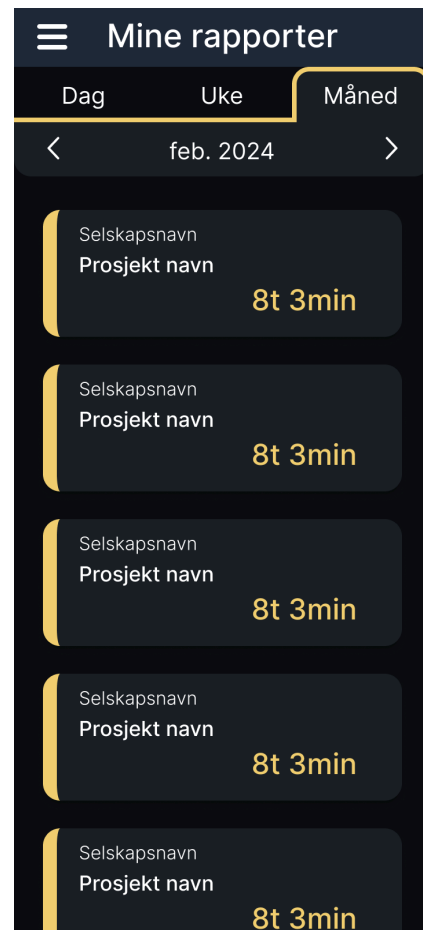
9.2.1 design, Stoppeklokke (mørkt tema)










9.2.2 design, Startet



9.2.3 mine rapporter design, lyst tema



9.2.4 mine rapporter design, mørkt tema

 Hoved blå #212937	 Gul #F1D070
 Lys blå #E5EDF9	 Rød #E37570
Hvit #FFFFFF	 Grønn #7CB867
 Mørkest blå #0C0E12	
 Mørk blå #1B2027	

9.2.5 farger som er brukt i designet

1

1. Legg til ny time med stoppeklokken for Test prosjekt med en liten beskrivelse
2. Legg til en ny time for prosjektet Test prosjekt den 23. mars fra kl. 7:01 til 14:02

total antall

3. Finn hvor mange timer som ble registrert for Test prosjekt i uke 17. Skriv svaret her: 87 33 min

4. Finn timen registrert 2. februar for prosjektet Hihahaha. Skriv svaret her: 87

5. Har prosjektet Hihahaha betalt lunsj?

Ja

Nei

3

1. Legg til ny time med stoppeklokken for Test prosjekt med en liten beskrivelse

2. Legg til en ny time for prosjektet Test prosjekt2 den 25. mars fra kl. 7:01 til 14:02

3. Finn total antall timer som ble registrert for Test prosjekt i uke 17.

Skriv svaret her: 22 min

4. Finn timen registrert 2. februar for prosjektet Hihahaha

Skriv registrert tid her: 8t

5. Har prosjektet Hihahaha betalt lunsj?

Ja

Nei

4

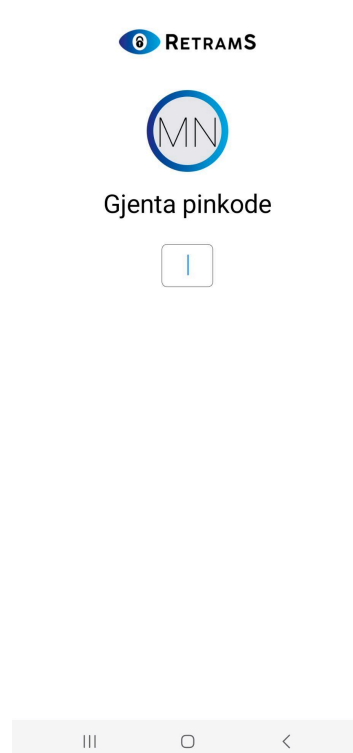
1. Legg til ny time med stoppeklokken for Test prosjekt med en liten beskrivelse
2. Legg til en ny time for prosjektet Test prosjekt den 26. mars fra kl. 7:01 til 14:02
3. Finn total antall timer som ble registrert for Test prosjekt i uke 17.
Skriv svaret her: 39 timer
4. Finn timen registrert 2. februar for prosjektet Hihahaha
Skriv registrert tid her: 05 timer
5. Har prosjektet Hihahaha betalt lunsj?
Ja
Nei



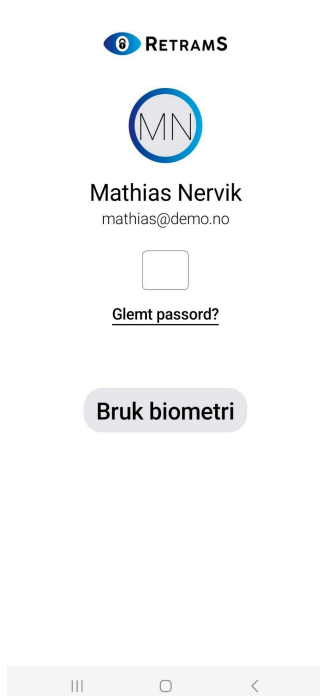
9.2.9 ferdig løsning for logg inn siden



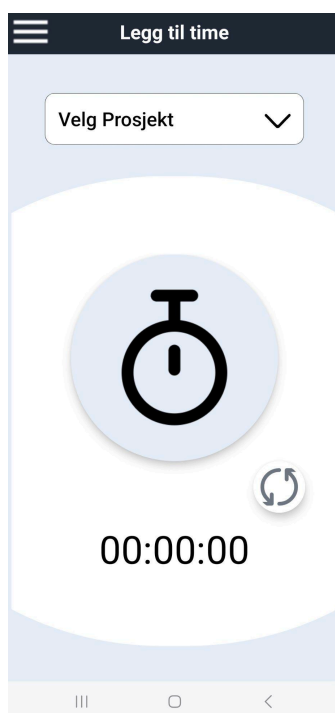
9.2.10 ferdig løsning for lag pin kode



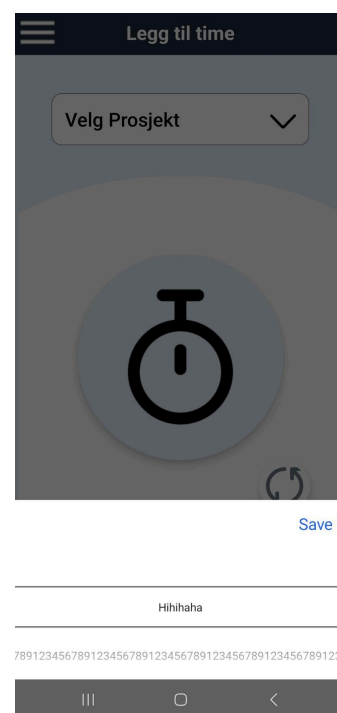
9.2.11 ferdig løsning for gjenta pin kode



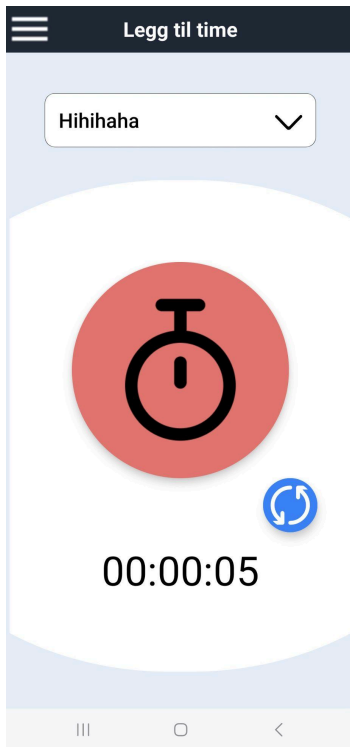
9.2.12 ferdig løsning for biometri innlogging



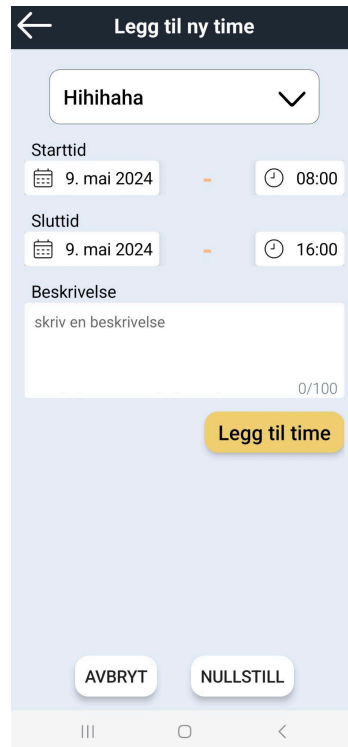
9.2.13 ferdig løsning for stoppeklokke siden



9.2.14 ferdig løsning for å velge prosjekt



9.2.15 ferdig løsning for å starte stoppeklokke



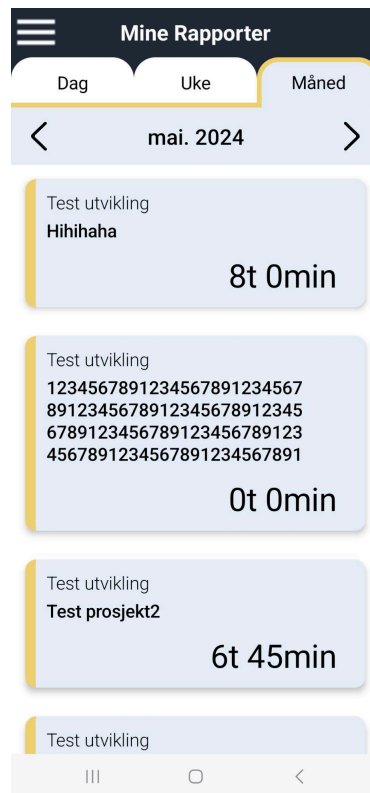
9.2.16 ferdig løsning for å legge til ny time



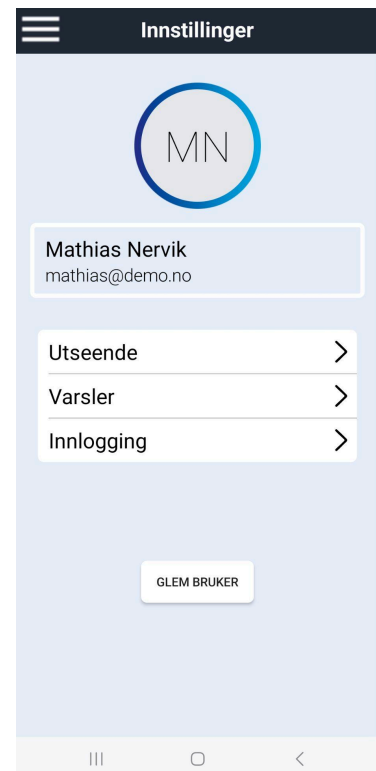
9.2.17 ferdig løsning for mine timer siden



9.2.18 ferdig løsning for mine prosjekter siden



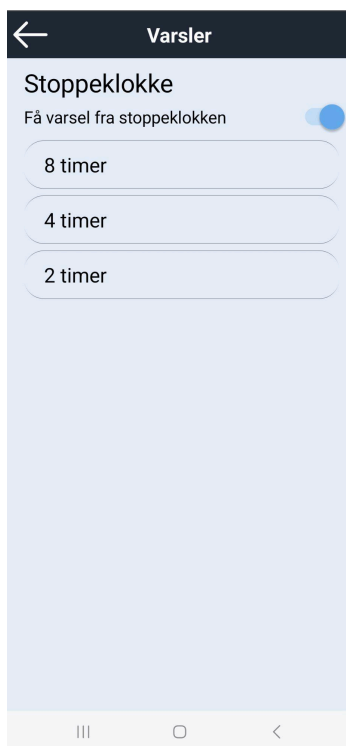
9.2.19 ferdig løsning for mine rapporter siden



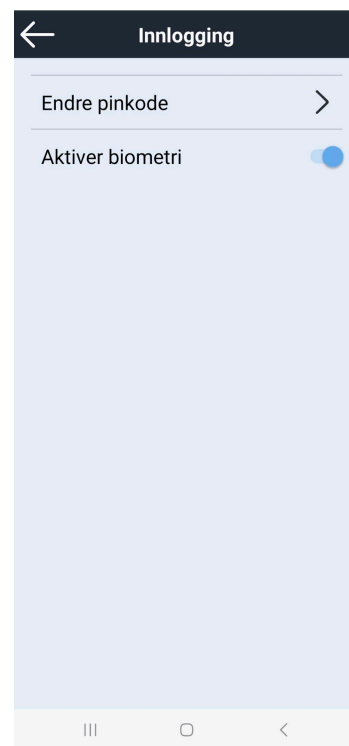
9.2.20 ferdig løsning for innstillinger siden



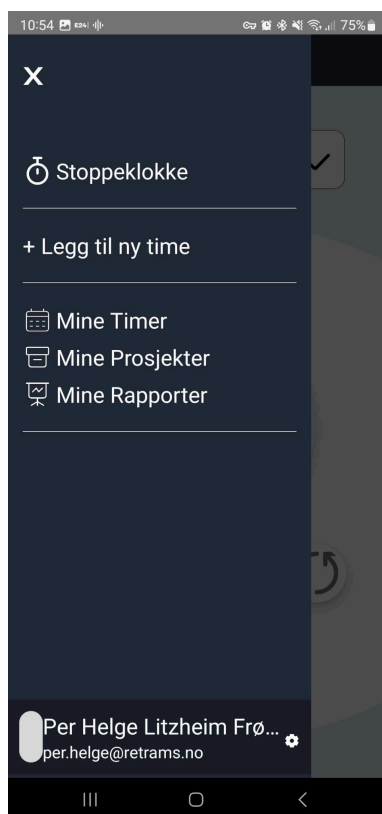
9.2.21 ferdig løsning for utseende siden



9.2.22 ferdig løsning for varsler siden



9.2.23 ferdig løsning for instillinger for innlogging siden



9.2.24 tilbakemelding fra oppdragsgiver om feil i menyen