



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Notifikasjonssenter

Notification center

Halvor Haaheim, Vegard Horgen og Stian Kvamme

Informasjonsteknologi / Dataingeniør

Fakultet for ingeniør- og naturvitenskap/Institutt for
datateknologi, elektroteknologi og
realfag/Informasjonsteknologi / dataingeniør

Veileder: Rogardt Heldal

Innleveringsdato: 04.06.2021

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Notifikasjonssenter / Notification center	<i>Dato:</i> 04.06.2021
<i>Forfatter(e):</i> Halvor Haaheim, Vegard Horgen og Stian Kvamme	<i>Antall sider u/vedlegg:</i> 24
	<i>Antall sider vedlegg:</i> 8
<i>Studieretning:</i> Informasjonsteknologi	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Rogardt Heldal	<i>Gradering:</i> Ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> WA Works	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Andreas Hammerbeck og Kristoffer Perminow	<i>Telefon:</i> 97661466 og 47396813

Sammendrag:

Hvordan kan et tilpasset notifikasjonssenter gjøre det enklere å formidle informasjon til kandidater og bedrifter? Målet med oppgaven vil være å løse dette på en enkel, intuitiv og funksjonell måte som bidrar til å gjøre nettstedet mer brukervennlig og effektivt.

How can a notification center adapted to a specific webpage make it easier to convey information from and to candidates and companies? The project goal is to solve this problem in an easy, intuitive and functional way, that makes the webpage more user friendly and efficient.

Stikkord:

Notifikasjoner	ReactTS, .NET	Rekrutteringsplattform
----------------	---------------	------------------------

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitenskap

Postadresse: Postboks 7030, 5020 BERGEN Besøksadresse: Inndalsveien 28, Bergen

Tlf. 55 58 75 00

Fax 55 58 77 90

E-post: post@hvl.no

Hjemmeside: <http://www.hvl.no>

FORORD

Rapporten dekker utviklingsprosessen av et notifikasjonssenter for WA.works, hvilket er en rekrutteringsplattform for bedrifter innenfor IT-bransjen. Oppgaven ble gitt av selskapet Wide Assessment AS og er gjennomført av Vegard Horgen, Halvor Haaheim og Stian Kvamme.

Vi ønsker å takke våre kontaktpersoner hos arbeidsgiveren, Kristoffer Perminow og Andreas Hammerbeck for deres grundige og hjelpsomme oppfølging gjennom prosjektet samt denne muligheten de har gitt oss. Vi vil i tillegg takke veilederen Rogardt Heldal fra HVL, for solid rådgivning og oppfølging.

INNHOLDSFORTEGNELSE

1. INNLEDNING	1
1.1 MOTIVASJON OG MÅL	1
1.2 KONTEKST	1
1.3 AVGRENSNINGER	2
1.4 RESSURSER	2
1.5 OPPBYGGING AV RAPPORTEN	2
2. PROSJEKTBEKRIVELSE	3
2.1 PRAKTISK BAKGRUNN	3
2.1.1 Prosjekteier	3
2.1.2 Tidligere arbeid	4
2.1.3 Initielle krav	4
2.1.4 Initiell løsnings-idé	4
3. DESIGN AV PROSJEKTET	5
3.1 FORSLAG TIL LØSNING	5
3.1.1 Alternativ løsning 1	5
3.1.2 Alternativ løsning 2	5
3.1.3 Diskusjon av alternativene	6
3.2 VALGT LØSNING	6
3.3 VALG AV VERKTØY	7
3.4 PROSJEKTMETODIKK	8
3.4.1 Utviklingsmetodikk	8
3.4.2 Prosjektplan	8
3.4.3 Risikovurdering	9
3.5 EVALUERINGSPLAN	9
4. DETALJERT DESIGN	10
4.1 DESIGN OG UTFORMING	10
4.2 BRUKSTILFELLEDIAGRAM	14
4.3 AKTIVITETSDIAGRAM	15
5. EVALUERING OG RESULTAT	17
5.1 EVALUERINGSMETODE	17
5.1.1 Manuell testing	17
5.1.2 Enhetstesting	17
5.2 EVALUERINGSRESULTAT	18
5.2.1 Resultat av manuelle tester	18
5.2.2 Resultat av enhetstester	18
6. DISKUSJON	19
6.1 GRUPPENS VALG	19

6.2 KONSEKVENNS AV VALG	19
6.3 FORBEDRINGSPOTENSIALE	20
7. KONKLUSJON OG VIDERE ARBEID	21
7.1 OPPSUMMERING	21
7.2 INITIELLE MÅL	21
7.3 RELEVANTE RESULTATER	21
7.4 VIDERE ARBEID	22
7.5 KONKLUSJON	23
8. REFERANSER	24
9. VEDLEGG	25
9.1 RISIKOLISTE	25
9.2 GANTT DIAGRAM	26
9.3 DETALJERT DESIGN	26

Begrepsliste

- Å sende en “poke” betyr at man sender et varsel til en annen part for å fange deres oppmerksomhet, for eksempel om en kandidat vil ha oppmerksomheten til en bedrift.
- Gantt-skjema er en form for søylediagram som illustrerer et prosjekts tidsplan. Det illustrerer datoer for start- og sluttidspunkt for oppgavene i prosjektet og viser et sammendrag av prosjektet.
- API - Et sett med funksjoner og prosedyrer som tillater opprettelse av applikasjoner som får tilgang til funksjonene eller dataene til et operativsystem, applikasjon eller annen tjeneste.
- REST API - Representational State Transfer - er en programvarearkitekturstil som bruker en delmengde av HTTP. Det brukes ofte til å lage interaktive applikasjoner som bruker webtjenester.
- React er et JavaScript bibliotek for å lage interaktive brukergrensesnitt.
- TypeScript er et programmeringsspråk. Det er et strengt syntaktisk superset av JavaScript og legger til valgfri statisk skriving til språket. Dette er en javascript variant med typer.
- .NET er en samling teknologier rundt programvareutvikling fra Microsoft.
- npm - Node Package Manager - er en pakkeleder for JavaScript-programmeringsspråket.
- Brukergrensesnitt/UI er det grensesnittet som gjør at en bruker kan kommunisere med maskiner. (User Interface).
- Material UI - Material Design er et designsystem som er utviklet av Google, og er utviklet for å kunne bruke dette designsystemet enkelt i React.
- Visual studio er et integrert utviklingsmiljø for Microsofts .NET plattform. Det hjelper utviklere å utvikle programmer for Windows, nettsider og mobil programvare.
- Visual studio code er en kildekoderedigerer laget av Microsoft.
- Github/Git er en vertstjeneste for programvareutviklingsprosessen med et versjonskontrollsystem ved bruk av Git.
- Discord er et kommunikasjonsverktøy brukt av gruppen.
- Slack er et kommunikasjonsverktøy brukt av bedrifter.
- Agile er en iterativ metode for å utvikle applikasjoner.
- SCRUM er et prinsipp som går ut på å dele opp større oppgaver i mindre deler som kan gjennomføres innen et satt tidsrom. De mindre delene kalles for sprinter, og varer en til fire uker.

- State - komponenter i React har innebygd state objekt, state objektet er der verdier som tilhører komponenten lagres.
- Docker image er en fil som det ikke går an å endre på og inneholder kildekode, biblioteker, avhengigheter, verktøy og andre filer som trengs for at applikasjonen skal kunne kjøres. Docker image er enkelt å dele, er alltid likt og settes opp på de aller fleste operativsystemer. På grunn av dette er det enkelt å dele images mellom ulike parter, dette fører til at alle har samme utgangspunkt.
- Container er en forekomst av Docker image, som bygger imaget til en kjørbare applikasjon. Hver container kjører separat og man kan endre på containeren mens den kjører. Containeren kan dele mapper med vert datamaskinen, så data kan lagres lokalt eller opprette et nytt image for å lagre dataene.
- HVL - Høgskulen på Vestlandet.

1. INNLEDNING

1.1 Motivasjon og mål

Flere store nettsteder som facebook, youtube og linkedin har til felles at alle har et notifikasjonssenter. Et notifikasjonssenter viser historikken av notifikasjoner en bruker har mottatt. Man kan se tilbake i tid og se hva man har gått glipp av, og eventuelt trykke på notifikasjonene for å bli sendt til de relevante sidene. Dette er veldig vanlig hos nettsider som har interaksjon mellom ulike brukere eller grupper. Hvorfor tilfører notifikasjoner økt brukervennlighet for disse nettsidene, slik at disse nettsidene implementerer et notifikasjonssystem?

Problemstillingen til oppgaven er hvordan et tilpasset notifikasjonssenter kan gjøre det enklere å formidle informasjon til kandidater og bedrifter. Målet med oppgaven vil være å løse dette på en enkel, intuitiv og funksjonell måte som bidrar til å gjøre nettstedet mer brukervennlig og effektivt.

Dette skal føre til at begge parter sparer tid og at jobbsøkende raskere kan nå ut til bedrifter med sine ressurser og kunnskaper. Samtidig blir prosessen for bedrifter raskere og mer effektiv slik at de kan nå frem til de kandidatene som passer dem best.

Wide Assessment har utviklet et rekrutteringsverktøy for bedrifter i IT-bransjen. De rangerer kandidater og gjør det enkelt for bedrifter å vite hvilke kandidater som passer dem best. Et notifikasjonssenter er et varslingsystem for å sende en melding fra en avsender til en eller flere mottakere, og vil gjøre rekrutteringsprosessen mer flytende. Dette vil gi Wide Assessment sitt rekrutteringsverktøy økt verdi.

1.2 Kontekst

Wide Assessment har allerede en nettside som lar kandidater innen IT lage profiler der man kan registrere ulike ferdigheter. Bedrifter som rekrutterer kan deretter se på profilene til de ulike kandidatene og ta kontakt med de som passer best ut i fra hvilke ferdigheter man har gitt på profilen.

Opgaven blir å videreutvikle dette ved å lage et notifikasjonssenter. Et notifikasjonssenter vil være svært nyttig fordi man da enkelt kan få beskjed om en bedrift viser interesse for kandidaten, eller motsatt. Wide Assessment vil også få mulighet til å

enkelt gi beskjed til alle brukere om nye funksjoner eller hendelser siden sist man har vært innlogget.

1.3 Avgrensninger

Prosjektet medfører få avgrensninger for gruppen, et medlem av gruppen har hatt praksis i Wide Assessment tidligere og derfor erfaring med deres system. Oppgaven er lagt opp til å være gjennomførbar innen gitt tidsplan og dermed vil hovedfunksjonene i oppgaven bli implementert innen da. Teknologien som brukes er implementert i WA nettsiden og dermed har gruppen mulighet for å spørre kontaktpersonene fra bedriften om deres løsninger.

Avgrensningene for prosjektet vil bli i form av testing og nøye evaluering av Wide Assessment, dette vil bli gjort muntlig, men mer nøyaktig gjennomgang av koden vil ikke bli gjennomført innen tidsfristen. Dette er fordi en slik gjennomgang er tidkrevende og mindre viktig for resultatet.

1.4 Ressurser

Ressursene bachelorgruppen har tilgjengelig er veiledning, kunnskap og rådgivning fra Wide Assessment, intern veiledning fra HVL, samt egen kunnskap.

Kontaktpersonene fra Wide Assessment brukes til tekniske spørsmål som har oppstått underveis og testing spesifikt mot deres systemer. Rådgiver fra høgskolen vil bidra mer innenfor prosjektrapporten.

1.5 Oppbygging av rapporten

Rapporten starter med en begrepsliste for begreper brukt i rapporten, der betydningen forklares.

Kapittel 1 inneholder målet med prosjektet, bakgrunnen for at Wide Assessment ønsker dette implementert, konteksten det skal implementeres i og ressurser som er brukt.

Kapittel 2 er en prosjektbeskrivelse som inneholder gruppens praktiske bakgrunn, hvem oppgaven er skrevet for, gruppens tidligere arbeid, prosjektets initielle krav og løsningen på problemet.

Kapittel 3 omhandler design av prosjektet. Det går over forslagene til løsning på prosjektet, samt alternative løsninger og en diskusjon av de ulike alternativene.

Kapittel 4 handler om detaljert design av prosjektet. Det går dypere inn i valg av design og utforming, og inneholder også brukstilfellediagram og aktivitetsdiagram.

I kapittel 5 diskuteres evalueringsmetodene for prosjektet. Her blir metodene for evaluering, som manuell testing og enhetstesting gjennomgått. Resultat av evalueringstestene blir også diskutert.

I kapittel 6 diskuteres valgene til gruppen og konsekvensen av valgene. Hvordan valgene har påvirket resultatet, samt potensielle forbedringsmuligheter.

Kapittel 7 inneholder konklusjon og oppsummering, samt om det kunne vært noe å arbeide videre med.

Kapittel 8 inneholder referanseliste.

Kapittel 9 Vedlegg, kapittelet inneholder risikoliste, GANTT diagram og bilder for design av nettstedet.

2. PROSJEKTBEKRIVELSE

2.1 Praktisk bakgrunn

2.1.1 Prosjekteier

Prosjekteier er Wide Assessment som har rekrutteringsverktøy for IT bransjen. Rekrutteringsverktøyet er innovativt ved at det rangerer kandidater etter ferdigheter som bedriften etterspør. Kandidatene kan liste opp sine egne ferdigheter og hvilket nivå de ligger på i de ulike ferdighetene, i tillegg til at kandidatene kan “poke” bedrifter de ønsker å legge inn sin søknad til. Dette betyr at man sender en varsel til en annen part for å

fange deres oppmerksomhet. Bedriftene kan også “poke” kandidater de er interessert i. Målet er å gjøre veien fra søker til ansatt effektiv for både for bedrifter og kandidater.

2.1.2 Tidligere arbeid

Wide Assessment er et rekrutteringsverktøy for bedrifter i IT bransjen som allerede er implementert og i bruk av bedrifter og kandidater.

2.1.3 Initielle krav

Arbeidsoppgavene i prosjektet er å implementere et notifikasjonssenter for Wide Assessment sitt system. Det vil være to typer notifikasjoner, en for kandidater og en for bedrifter. Når en kandidat blir kontaktet av en bedrift skal det komme opp en notifikasjon hos kandidaten. Det samme skal skje for en bedrift om en kandidat søker på en stilling eller “poker” bedriften. Når en notifikasjon er sett av brukeren skal den markeres som lest slik at uleste notifikasjoner kan fremheves. Det skal også være mulig for Wide Assessment å legge til notifikasjoner for alle kandidater, alle bedrifter eller alle brukere for å eksempelvis varsle om ny funksjonalitet på plattformen eller lignende.

2.1.4 Initiell løsnings-idé

Løsningen gruppen initielt ønsker å benytte er et notifikasjons system med funksjonalitetene som er beskrevet av bedriften. Kandidater og bedrifter skal motta notifikasjoner hvis de blir “poket” og bedriften skal kunne sende ut notifikasjoner til alle brukerne på nettstedet.

Løsningen som gruppen utvikler vil bli en egen komponent i React, slik at den er frittstående og kan enkelt implementeres og flyttes rundt på nettstedet om ønskelig. Gruppen følger her et konsept kalt “best practice” for å optimalisere komponenten. Dette innebærer elementer som navngivning, konsistent skriving og krav rundt rendering (Priya, 2020). For å oppnå dette planlegger gruppen å være strukturert i oppbyggingen av komponenten slik at hvert ledd er separert. Notifikasjons-ikonet vil være det ytterste leddet, og vise varsel om det er kommet notifikasjon, ved klikk på ikonet vil notifikasjonene vises. Dette blir nivået der det meste av logikken for react-komponenten vil foregå.

Backend ønsker gruppen å lage så lite komplisert som mulig for lettleselig kode, samtidig som all ønsket funksjonalitet oppnås. Databasen skal inkludere det som er nødvendig for at notifikasjonscenteret virker.

3. DESIGN AV PROSJEKTET

3.1 Forslag til løsning

3.1.1 Alternativ løsning 1

Et alternativ er å lage notifikasjon systemet uten bruk av ferdiglagde komponenter som tar seg av håndteringen av sending og mottagelse av notifikasjonene, dette er for at Wide Assessment har mest mulig frihet og mulighet til å bygge opp komponenten slik de vil og det gir gruppen mer frihet for å tilpasse komponenten.

I denne løsningen må både frontend, backend, og en databasetabell opprettes. I tillegg må REST API brukes for kommunikasjon mellom frontend og backend. En REST API er en programvarearkitekturstil som bruker en delmengde av HTTP. Det brukes ofte til å lage interaktive applikasjoner som bruker webtjenester. Dette gjør at verktøyene som Wide Assessment har brukt for å implementere deres allerede eksisterende ressurser blir brukt og at de har full kontroll på komponenten og ikke er avhengig av at tredjeparter fortsetter å støtte deler av løsningen.

3.1.2 Alternativ løsning 2

Et annet alternativ er å bruke en ferdiglaget React-komponent som lastes ned via en npm-pakke. Dette vil da være en npm-pakke som inkluderer et notifikasjonscenter for å generere notifikasjoner gjennom en REST API. Hele notifikasjonscenteret vil da være ferdig laget, men gruppen må lage de ulike typene notifikasjoner, som at man kan trykke på notifikasjoner, at de linker til andre steder på nettsiden, at man kan se at de er lest, o.l. Denne løsningen inkluderer også et dashboard hvor man kan se status og aktivitet på notifikasjoner, samt statistikk.

Dette vil dog inkludere en tredjepartsside. For å kunne bruke APIet til tredjepartssiden er man nødt til å ha en appId og en subscriberID, som vil si at man er nødt til å være registrert og logget inn hos denne tredjeparten.

3.1.3 Diskusjon av alternativene

Det er fordeler og ulemper ved begge alternativene. Alternativ to som bruker ferdiglagde React-komponenter kan det spares store mengder med tid, men samtidig vil dette begrense friheten til å utforme applikasjonen slik det er ønsket. Samtidig vil dette føre til at notifikasjonssystemet bli avhengig av tredjepartssider. Et annet potensielt problem med å bruke denne løsningen er å bruke noen andre sine tjenester til egne formål for økonomisk vinning. Dette må avklares med skaperne av det enkelte API og kan kreve kostbare kommersielle lisenser. Dette er noe gruppen ønsker å unngå, samtidig som det vil skape unødig bry for Wide Assessment.

Gruppen ønsker å lage et brukergrensesnitt, som er designet i samme stil som resten av nettsiden til Wide Assessment. Alternativ èn vil gi oss større frihet til å gjøre dette, men det vil være en større jobb og mer som skal utvikles. Det er også mer som kan gå gale under utviklingsprosessen, for eksempel at gruppen ikke har nok kunnskap om de utviklingsverktøyene som skal brukes i prosjektet.

Notifikasjonssenteret utvikles og designes i samme stil som resten av nettsiden. Dette oppnår gruppen ved å tilpasse komponenter fra Material UI, som er et populært React brukergrensesnittbibliotek. Material Design er et designsystem som er utviklet av Google, og Material UI er utviklet for å kunne bruke dette designsystemet enkelt i React. Material UI er allerede i bruk i enkelte deler av nettsiden. Dette hjelper å forsikre gruppen om at notifikasjonssenteret blir utviklet i samme standard som resten av nettsiden.

Alternativ èn vil også føre til en løsning for kommunikasjon mellom frontend og backend som er bygget opp på samme måte som resten av nettsiden. Dette vil gjøre koden og nettsiden enklere å vedlikeholde siden alt er samlet og laget på samme måte.

3.2 Valgt løsning

Gruppen har valgt å gå for alternativ èn, da dette vil gjøre utviklingen enklere og mer frittstående uten å være avhengig av tredjeparts sider for at notifikasjonssenteret skal oppnå ønsket funksjonalitet. I stedet vil det bli benyttet npm-pakker med React komponenter via for eksempel Material UI som gjør utviklingsprosessen enkel og fleksibel. Denne løsningen vil også gjøre det enklere å kombinere den nye koden med det som allerede eksisterer, uten å skape for mye ulikheter både i koden og brukergrensesnittet.

3.3 Valg av verktøy

Gruppen har valgt å bruke verktøyene Wide Assessment bruker i sin daglige drift. Gruppen har hatt frihet til å velge andre program hvis det oppfyller de samme funksjonaliteten, men har valgt å ikke gjøre det. Visual studio og Visual studio code har blitt brukt som kildekoderedigerer til backend og frontend.

For å ha kontroll på hva som skal gjøres i de forskjellige iterasjonene, samt hva som må fikses senere har Pivotal tracker blitt brukt. Dette er en sjekklister som gruppemedlemmene kan legge til punkter på underveis i iterasjonen.

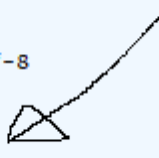
Cmder har blitt brukt istedenfor vanlig windows terminalvindu. Dette gjør behandlingen av kommandoer enklere, og Cmder har i tillegg flere funksjoner. Dette har blitt brukt til å laste ned npm-pakker som blant annet Material UI komponenter som har blitt brukt, samt til oppsett av GitHub repositories og til å kjøre både front-end og back-end. Material UI er brukt for å få enkelte ferdiglagde komponenter i React som deretter kan modifiseres og utformes slik at alt passer sammen.

For å gjøre det enklere å lage, distribuere og kjøre applikasjonen benytter Wide Assessment seg av et program som kalles Docker. Denne applikasjonen benytter seg av Docker image og container. Docker image er en fil som det ikke går an å endre på og inneholder kildekode, biblioteker, avhengigheter, verktøy og andre filer som trengs for at applikasjonen skal kunne kjøres. Docker image er enkelt å dele, er alltid likt og settes opp på de aller fleste operativsystem. På grunn av dette er det enkelt å dele images mellom ulike parter, dette fører til at alle har samme utgangspunkt. Container er en forekomst av Docker image, som bygger imaget til en kjørbare applikasjon. Hver container kjører separat og man kan endre på containeren mens den kjører. Containeren kan dele mapper med vert datamaskinen, så data kan lagres lokalt eller opprette et nytt image for å lagre dataene (*Sofija Simic 2019*).

Under prosjektet har gruppen benyttet seg av en REST API, hvilket gjør spørringer for webtjenester enklere. Det ble kommunikasjon mot både allerede eksisterende APIer, men og nyutviklede APIer som gruppen produserte. Back-end-metoder ble utviklet i samsvar med dette ved hjelp av HTTP. Hovedsakelig blir parametere gitt gjennom url mens protokoll for retur data er JSON format se figur 3.3.1 (flere bilder vedlagt i detaljert design 9.3.9-9.3.12).

```
HTTP/1.1 200 OK
X-Powered-By: Express
connection: close
date: Wed, 02 Jun 2021 13:41:21 GMT
content-type: application/json; charset=utf-8
server: Kestrel
transfer-encoding: chunked

32b
[{"id":26,"name":"WA","text":"222222222","read":0,"date":"21-05-26-10:22"},{"id":25,'
```



(Figur 3.3.1 respons data på JSON format)

PostgreSQL blir brukt som database for å lagre all dataen til notifikasjonssenteret. Github er brukt sammen med Git til versjonshåndtering av applikasjonsutviklingen.

Kommunikasjonsverktøyene som har blitt brukt er programmet discord for å kommunisere med hverandre når gruppen har jobbet hjemmefra. Kommunikasjon med Wide Assessment har blitt gjort gjennom Slack og Google meet.

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

Smidig utvikling (eng: "agile development") benyttes i kombinasjon av scrum-inspirert arbeidsmetode. Gruppen har fokus på å utvikle i mindre deler eller sprinter. Mot slutten av hver sprint testes det at delen oppnår ønsket funksjonalitet og funksjon (*Claire Drumond*). Dette vil også føre til at oppdragsgiver vil ha mulighet til å gi tilbakemeldinger underveis, og endringer kan gjøres lettere. Arbeidet for neste sprint ble valgt ut ifra status på utført arbeid, og deretter ble oppgaver valgt fra backlogen. En backlog er en liste over hva som kreves av produktet, eller i dette tilfellet arbeidsbeskrivelsen (*Scrum.org*). Det er på forhånd satt opp en plan for hele prosjektet. Hver sprint har en tidsplan basert på forventet arbeidsmengde, men er satt opp med forbehold om eventuelle problemer som kan oppstå underveis.

3.4.2 Prosjektplan

Gruppen har valgt å sette opp en tidsplan i form av et GANTT-diagram, som vil visualisere prosjektets tidsplan med aktiviteter og viktige milepæler. Dette gir en god oversikt og noe konkret å forholde seg til. Gruppen har møter med Wide Assessment etter behov, samt at de kan kontaktes via Slack hvis gruppen har spørsmål. Det er avtalt ukentlig møte med intern veileder. Gruppen har også valgt å bruke Pivotal Tracker, som er en scrum-tavle

designet for å hjelpe å visualisere arbeid, se hva som blir jobbet med, og maksimere effektivitet. Dette fungerer veldig bra sammen med smidig utvikling. Her skriver gruppen inn ulike deler som må utvikles, og velger så oppgaver de vil jobbe med. Resterende gruppemedlemmer kan her se hva resten jobber med, hva som er fullført, hva som må testes, o.l.

GANTT-diagrammet har blitt brukt som et overordnet mål, mens underveis i utviklingen av prosjektet har scrum-tavlen blitt fulgt, ut ifra hva som måtte jobbes med. Siden gruppen hovedsakelig har fulgt scrum-tavlen har det vært lettere å endre i prosjektplanen underveis, ut ifra hva som var nødvendig å jobbe mer med på det gitte tidspunkt. Tidsrammen på de ulike delene i GANTT-diagrammet stemte bra, og det har ikke vært noen store endringer underveis i prosjektet.

Hver mandag har gruppen et møte hvor det blir gjennomgått hva som ble gjort forrige uke, samt evaluering av arbeidet og arbeidsmengde. Gruppen legger så en plan for uken og arbeidet blir fordelt. Gruppen ser også over viktige frister og milepæler.

3.4.3 Risikovurdering

Etter samtale med veileder og arbeidsgiver har gruppen fastslått at det er relativt få risikoer rundt prosjektet. Oppgaven er konkret og rett frem, og gruppemedlemmene kjenner hverandre fra før og vet hva som kan forventes av hverandre. Alle tre medlemmene har også arbeidet med React i praksis tidligere. Til tross for dette kan det alltid være lurt å ta alle risikoer i betraktning når det kommer til prosjekter av en slik størrelse. Potensielle risikoer er analysert ved hjelp av risikomatrisen vedlagt i vedlegg. Se figur 9.1.1.

3.5 Evalueringsplan

For å evaluere applikasjonens funksjonalitet vil gruppen se på resultatene av testene som er blitt utført. Disse testene vil gi oss en god ide om hvor vidt applikasjonen fungerer slik det skal og vil gi et godt utgangspunkt for evalueringen. Det samme gjelder responsivitet og brukervennlighet. At en nettside er responsiv vil si at den ser bra ut på alle enheter. Nettsiden vil automatisk justeres ut ifra skjermstørrelse (W3 schools). Den beste måten å evaluere disse faktorene på er å fysisk teste applikasjonen med et bredt utvalg brukere

med ulik bakgrunn. Testing med et bredt utvalg brukere vil bli tidkrevende, noe det ikke er tid til, derfor vil denne delen av testingen eventuelt bli gjort av gruppen og bedriften.

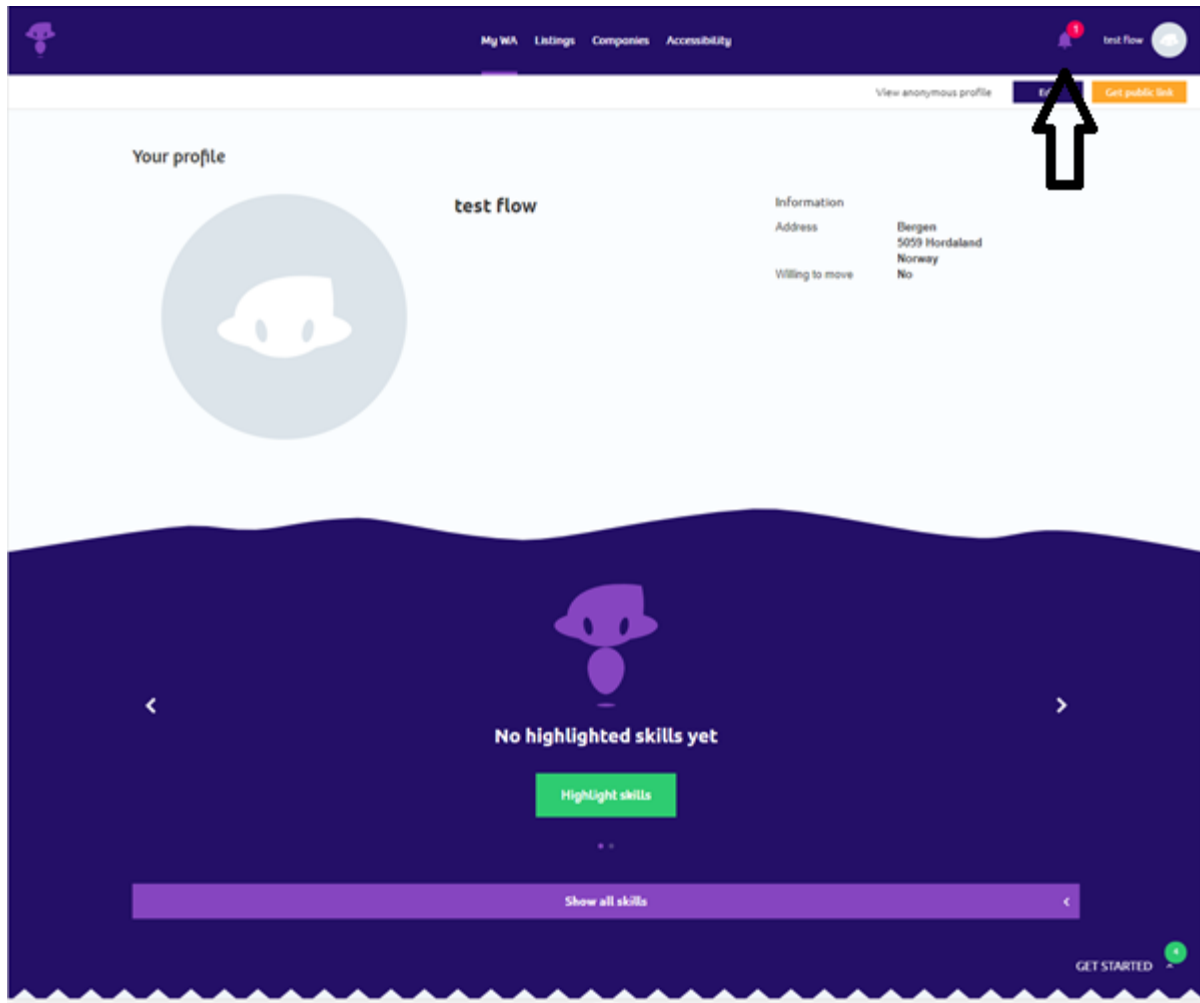
Ryddighet av kode kan evalueres med bistand fra utviklerne i WA. De vil ha en bestemt mening om hvordan de ønsker at koden skal se ut, og kan dermed gi oss konstruktiv evaluering både underveis og mot slutten av prosjektet.

4. DETALJERT DESIGN

4.1 Design og utforming

Ved utviklingen av designet og utformingen av notifikasjonssystemet var det i likhet med kodingen mye som var forhåndsbestemt av oppgavebeskrivelsen og den allerede etablerte applikasjonen sin utforming. Det ble derfor gjort sammenligninger med liknende menyer og fargevalg for å bestemme utseende på systemet. Fargevalgene hadde enkelte konflikter med HCI sine standarder, HCI er standarden for hvordan mennesker kan samhandle med datamaskiner på enklest mulig måte. Konflikten var med notifikasjonstilleren sin farge. Til tross for dette benyttet gruppen oss av fargestandarden til Wide Assessment for at applikasjonen ville samsvare godt med resten av systemet. Det fantes ikke en helt lik funksjon fra før, hvilket medførte at gruppen måtte utvikle nye komponenter slik som et notifikasjonssymbol og avgjøre hvor dette skulle plasseres.

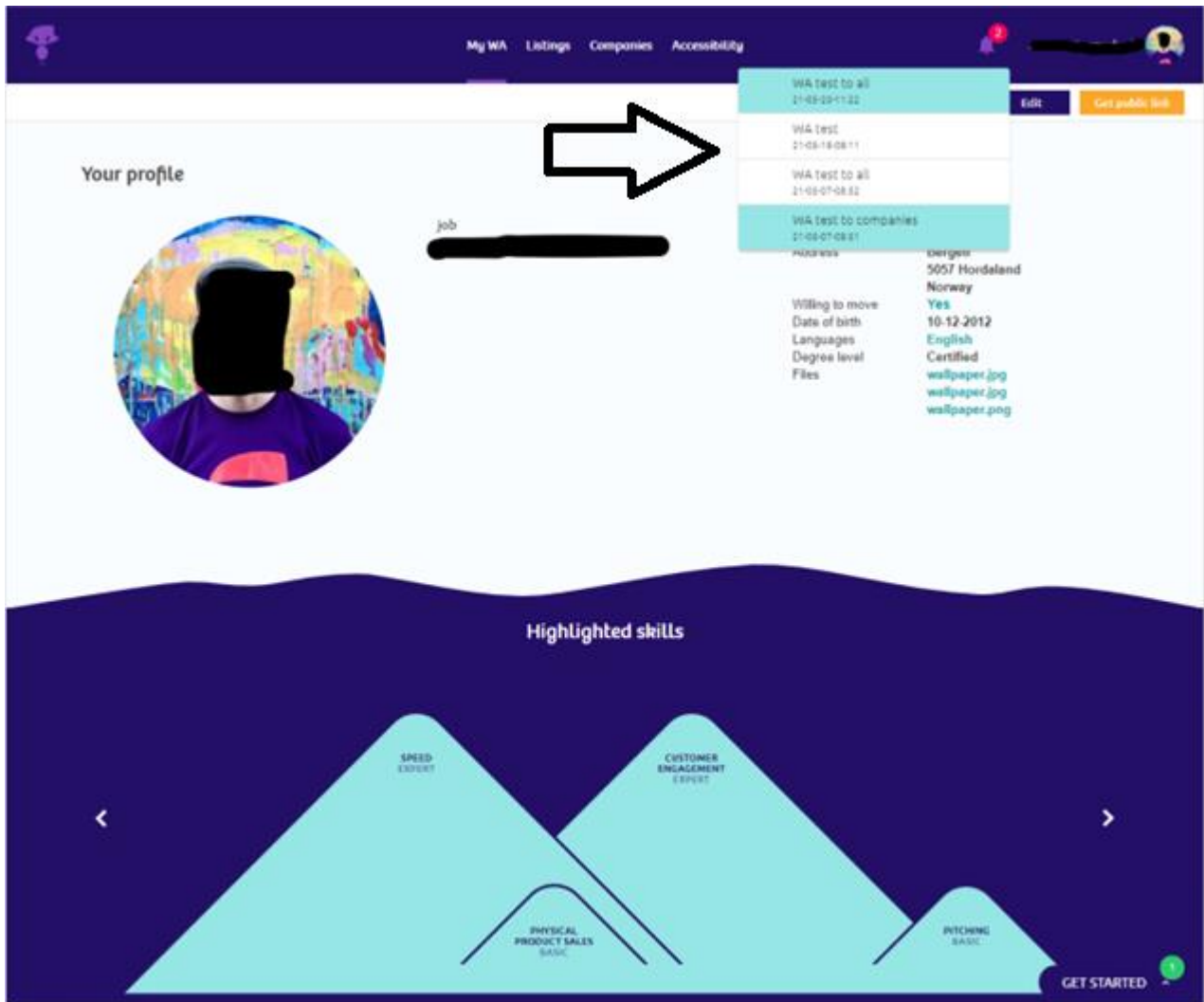
Det ble derfor undersøkt på anerkjente nettsteder slik som facebook, youtube og linkedin for å se hvordan deres notifikasjonssentre var utformet og designet. Etter denne undersøkelsen var det tydelig at det var påfallende likhetstrekk og utforming rundt dette, nemlig at ikonet symboliserte en bjelle og var plassert på menybarene oppe i høyre hjørne. Ved å følge gjeldende praksis, ble notifikasjonsikonet sin plassering på nettsiden forutsigbart og lettere for brukere å bli vant til (Babich N 2020, Page structure). Dette lot seg godt gjøre med tanke på WA sitt eksisterende design, gruppen konkluderte derfor med at notifikasjons-symbolet skulle plasseres ved profilmeny oppe til høyre. Se figur 4.1.1, flere bilder vedlagt i Detaljert design 9.3.1, 9.3.2.



(Figur 4.1.1 Notifikasjonssymbol og meny)

Som man ser i figur 4.1.1 har Wide Assessment allerede en menybar øverst på nettstedet. Det var derfor naturlig å legge inn notifikasjonssymbolet i denne menybaren, ved siden av profilmenyen til brukeren. Man får også opp antall notifikasjoner man har som er ulest.

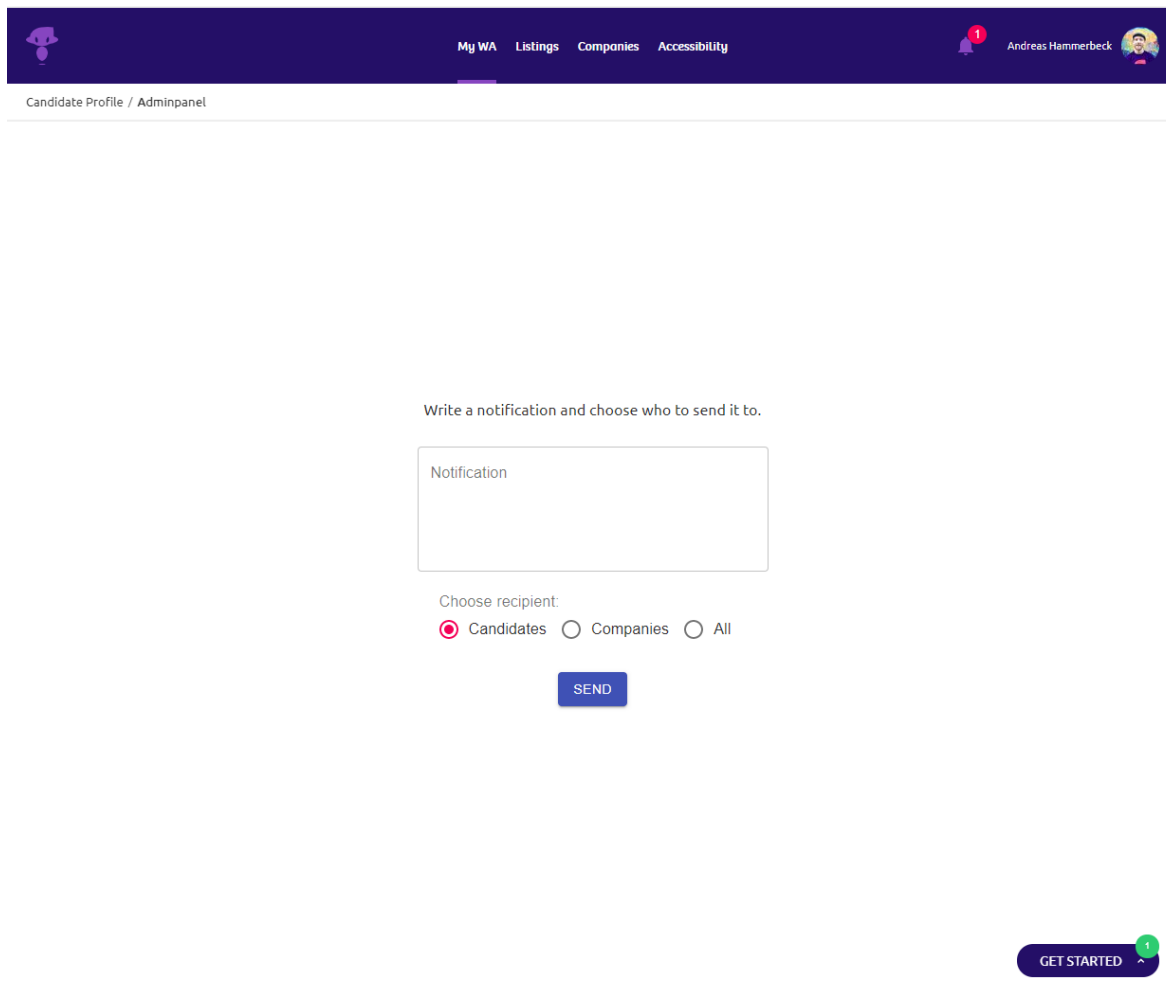
Det var også nødvendig å designe selve menyen, hvor det ble tatt inspirasjon fra Wide Assessment sin meny, med tilsvarende skriftstørrelse, skrifttype og fargevalg. Hver enkelt notifikasjon krevde en funksjon for å markere om meldingen var lest eller ulest, hvor det igjen ble gjort sammenligninger med anerkjente nettsteder og konkludert med at den vanligste fremgangsmåten er å markere med en annerledes bakgrunnsfarge. Følgende implementasjoner medførte et brukergrensesnitt som er intuitivt, funksjonelt og effektivt. Se figur 4.1.2, flere bilder vedlagt i detaljert design 9.3.2, 9.3.4.



(Figur 4.1.2 Notifikasjonslisten)

Som man ser i figur 4.1.2 vil man få opp en liste med notifikasjoner når man trykker på notifikasjonssymbolet. Denne listen inneholder notifikasjonene, hvem notifikasjonen er sendt fra og tidspunktet den ble sendt. Man kan også enkelt se om en notifikasjon er lest eller ulest ut fra bakgrunnsfargen på notifikasjonen.

Wide Assessment ønsket også implementasjon av et enkelt administratorpanel for utsending av notifikasjon til alle kunder, kandidater eller bedrifter. Utviklingen av panelet ble relativt simpelt, men samtidig intuitivt og med et større fokus på funksjonalitet enn design ettersom at funksjonen kun vil være tilgjengelig for bedriften og ikke brukerne. Det ble allikevel utformet i samsvar med det eksisterende designet og vil kun være synlig for administratorer i deres egen profilmeny. Administrator panelet var også et tillegg i forhold til oppgavebeskrivelsen, men Wide Assessment hadde ikke et fra før av, så gruppen valgte derfor å lage et enkelt administratorpanel for å få på plass funksjonaliteten. Figur 4.1.3 flere bilder vedlagt i detaljert design 9.3.5-9.3.8



The screenshot shows a web interface for an administrator. At the top, there is a dark blue navigation bar with a logo on the left and menu items: 'My WA', 'Listings', 'Companies', and 'Accessibility'. On the right side of the bar, there is a notification bell icon with a red '1' and a user profile for 'Andreas Hammerbeck'. Below the navigation bar, the page title is 'Candidate Profile / Adminpanel'. The main content area has a heading 'Write a notification and choose who to send it to.' followed by a text input field labeled 'Notification'. Below the input field, there is a section 'Choose recipient:' with three radio button options: 'Candidates' (which is selected), 'Companies', and 'All'. A blue 'SEND' button is positioned below the radio buttons. In the bottom right corner of the main content area, there is a dark blue button labeled 'GET STARTED' with a small green '1' icon next to it.

(Figur

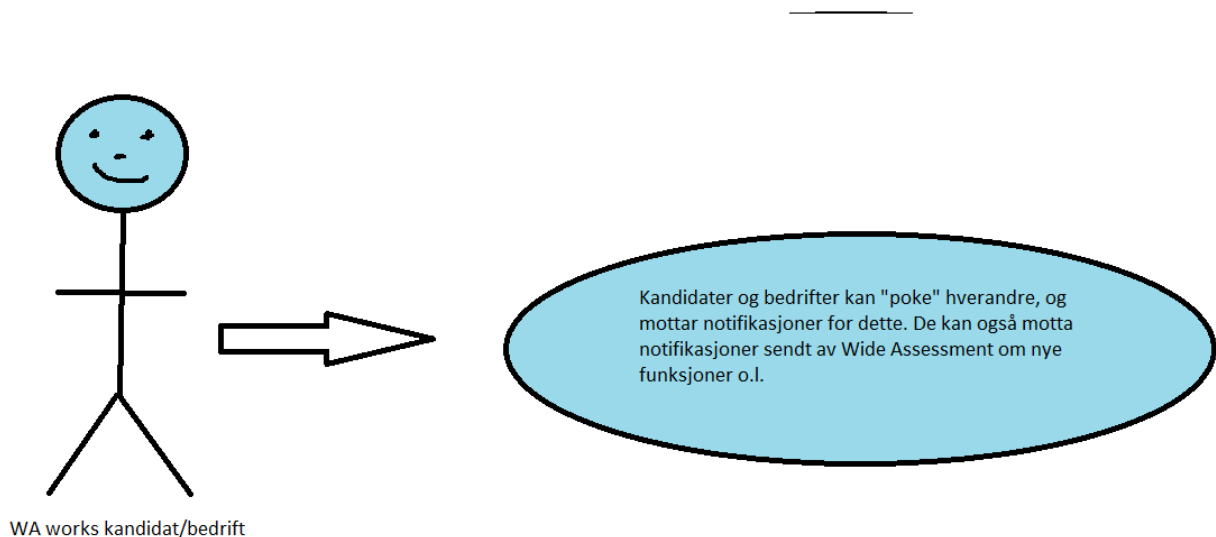
4.1.3 Administratorpanel)

Figur 4.1.3 viser administratorpanelet der Wide Assessment kan sende ut notifikasjoner til kandidater, bedrifter, eller alle.

4.2 Brukstilfellediagram

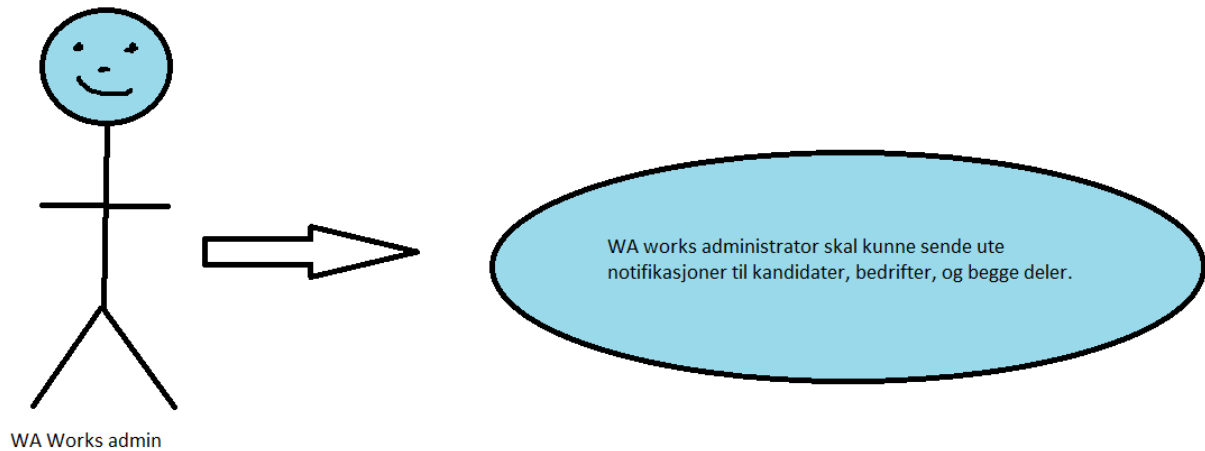
Brukstilfellediagrammet (eng: “use case diagram”) for tilleggsfunksjonen vår har ikke blitt endret på i løpet av utviklingsprosessen, ettersom oppgaven er veldig konkret og oppgavebeskrivelsen gir et helt klart bilde på hvilken funksjonalitet som skal oppnås.

Kandidater skal kunne “poke” bedrifter de er interessert i, og bedrifter skal kunne “poke” kandidater de er interessert i. Begge brukere skal få notifikasjon om at de har blitt “pocket”. De skal også motta notifikasjoner sendt fra Wide Assessment om for eksempel nye funksjoner og nyheter siden sist de var pålogget. Dette kan man se ut ifra figur 4.2.1.



(Figur 4.2.1, brukstilfellediagram for kandidat/bedrift)

Wide Assessment administrator skal også kunne sende ut notifikasjoner til enten kandidater eller bedrifter, eller begge deler. Dette kan man se i figur 4.2.2.



(Figur 4.2.2, brukstilfellediagram for WA administrator)

4.3 Aktivitetsdiagram

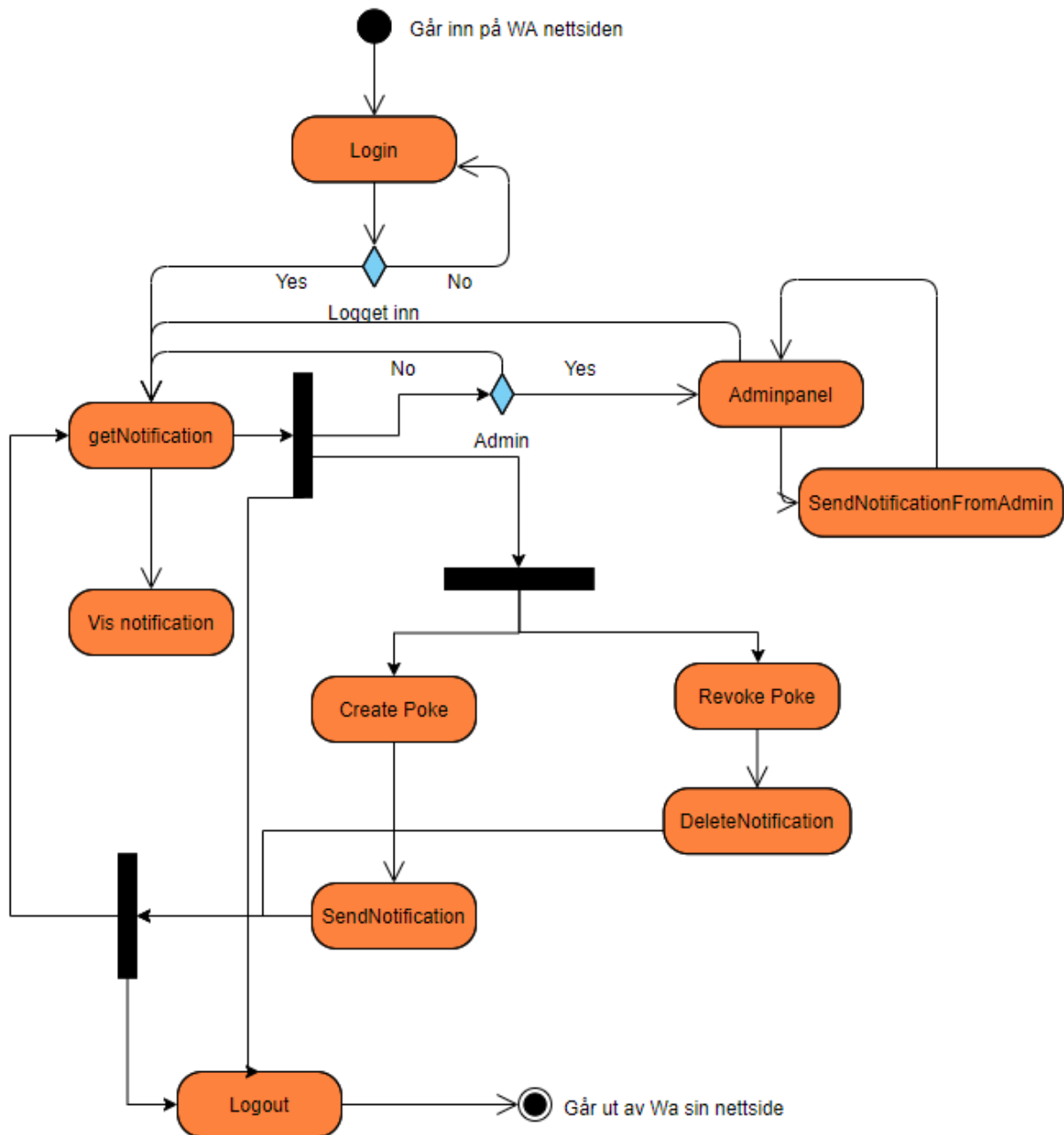
Gruppen har inkludert et aktivitetsdiagram for å illustrere flyten i funksjonen vår. Aktivitetsdiagrammet beskriver ikke andre funksjoner på nettsiden, kun sending av notifikasjoner.

Brukeren logger inn, og det blir gjort en sjekk på om man er logget inn. Er man ikke blitt logget inn blir man sendt tilbake til login siden. Når man er logget inn blir metoden "getNotification" kjørt, som henter alle notifikasjonene til en bruker. Brukeren kan da åpne notifikasjonene for å se dem.

Hvis man har administratorrettigheter kan man gå inn på administratorpanel, her blir det gjort en sjekk på om brukeren har administratorrettigheter. Har man ikke det, blir man sendt tilbake til hovedsiden. Fra administratorpanelet kan administrator sende notifikasjoner til kandidater, bedrifter, eller begge deler. Metoden "SendNotificationFromAdmin" blir da kjørt, og det lages en notifikasjon.

Hvis man er en kandidat, kan man søke på en stilling, og metoden "ApplyToListing" vil bli kjørt. Denne sender en søknad, og metoden "SendNotification" vil bli kjørt, som sender en notifikasjon til bedriften man søker hos. Man kan også "poke" en

bedrift eller kandidat, og dette vil også kjøre metoden "SendNotification". Det er også mulig å slette en "poke" da blir "DeleteNotification" metoden kjørt.



(Figur 4.3.1 Aktivitetsdiagram)

5. EVALUERING OG RESULTAT

5.1 Evalueringsmetode

5.1.1 Manuell testing

Gruppen har hovedsakelig brukt manuell testing til å evaluere om applikasjonen fungerer som den skal under utviklingsfasen. Spesielt i front-end har manuell testing vært mye brukt. Testing av applikasjonen ble gjort parallelt med utviklingen for å verifisere at koden fungerte som den skulle.

Manuell testing passet bra til å teste hver enkelt del av applikasjonen vår underveis, siden prosjektet er delt opp i sprinter. Da ble det laget en funksjon og testet denne, og eventuelt fikset på funksjonen før gruppen gikk videre, hvis den ikke funkete som planlagt. Dette sikret at løsningen oppfylte kravene fra oppdragsgiver, før neste iterasjon ble gjennomført.

5.1.2 Enhetstesting

Enhetstesting er en type programvaretesting hvor individuelle biter av applikasjoner, sammen med tilhørende kontrolldata, brukerprosedyrer og operasjonsprosedyrer, testes for å se om de er egnet for en gitt type bruk. Enhetstester er korte programfragmenter, laget av programmerer i løpet av utviklingsprosessen. Enhetstesting er grunnlaget i komponenttesting. Enhetstester blir typisk laget for å verifisere at kode tilfredsstillere krav til design og oppførsel (*Wikipedia, 2021, avsnitt 1*).

Enhetstestene for å teste notifikasjonsentralen er laget med en database i minnet (eng: "in-memory"), dette vil si at testene ikke gjøres direkte på databasen, men frittstående på en egen database etterligning. Fordelen med dette er at det ikke er unødvendig stress mot databasen når testingen utføres. Eksisterende data i databasen er urørt i enhetstestene og kluss med dataene i databasen unngås. Det som testes med enhetstestene på databasen i minnet er metodene som utfører spørringer mot databasen. Dette utføres ved at hver test har en oppgave om å teste en metode mot databasen, dette kan være innsetting, uthenting, sletting eller endring av notifikasjons objekter.

5.2 Evalueringsresultat

5.2.1 Resultat av manuelle tester

Som sagt tidligere ble det gjort manuelle tester av applikasjonen parallelt med utviklingen. For hvert punkt i spesifikasjonen måtte det bli evaluert om delene i applikasjonen fungerte som spesifisert.

Det skulle være to typer notifikasjoner, en for kandidater og en for bedrifter. Man skulle få opp notifikasjon når man blir kontaktet, enten hvis en bedrift kontakter en kandidat, eller motsatt. Når en notifikasjon er sett av brukeren skal den markeres som lest, slik at uleste notifikasjoner fremheves. Wide Assessment skal kunne sende ut globale notifikasjoner enten til alle kandidater eller bedrifter, eller til både kandidater og bedrifter. Alle spesifikasjonene ble manuelt testet underveis og fungerte som forventet. I tillegg til dette ble det laget et administratorpanel der Wide Assessment kunne sende ut notifikasjoner til kandidater/bedrifter, eller begge deler. Her ble det også testet at kun brukere med administratorrettigheter hadde tilgang til panelet.

5.2.2 Resultat av enhetstester

Resultatet fra testene var at alle metodene som utfører spørringer mot databasen gir ønsket resultat. Testene utførte spørringer som gikk på innsetting, uthenting og endring av data, deretter test på om ønsket resultat var oppnådd.

Gruppen valgte å lage tester som fokuserte på notifikasjonssentralen sine metoder mot databasen, og ikke generelt stress tester på databasen. Grunnen til dette er at notifikasjonstabellen i databasen er et tillegg på Wide Assessment sitt allerede eksisterende databasesystem.

6. DISKUSJON

6.1 Gruppens valg

Gruppen valgte å gå for en iterativ utviklingsmetode. Dette følte naturlig siden det ble testet og forandret ting kontinuerlig underveis. Oppgaven vår er veldig konkret på hva som skal utvikles, så det var ikke så mange valg å ta på akkurat det. Wide Assessment bruker til vanlig React, TypeScript og C# (.Net Core), samt Pivotal Tracker for scrum-tavle. Det måtte derfor programmeres i de samme programmeringsspråkene, og det ble valgt å sette opp gruppens egen sprinter i Pivotal Tracker.

En konsekvens av at det programmeres i samme språk som Wide Assessment er at gruppen bruker litt lenger tid, siden ingen på gruppen hadde programmert i TypeScript eller C# før. Gruppen måtte også sette seg inn i Wide Assessment sin kodebase, og gjøre ting likt som dem, som også førte til at det gikk med mer tid på enkelte ting. Til gjengjeld gjør dette at designet blir likt det Wide Assessment allerede bruker på nettsiden, samt at det programmeres i samme standard som allerede er gjort i kodebasen.

Det mest distinkte valget gruppen gjorde var å lage notifikasjonssentralen som funksjonell komponent. Wide Assessment har mest klasse komponenter i sin React-kode, men de hadde ingen innspill på valget gruppen gjorde. Grunnen til at gruppen valgte funksjonell komponent var ut av egen interesse og hva som er mest brukt i dag.

6.2 Konsekvens av valg

Valgene som er gjort har påvirket resultatet på flere måter, mange av valgene var gjort før gruppen begynte på oppgaven. Notifikasjonssentralen skulle legges til i en allerede eksisterende nettside ble en del av valgene selvsagte. Dette førte til at vår tilnærming til oppgaven ble til dels låst av eksisterende teknologier. Det gruppen utviklet i forbindelse med notifikasjonssentralen brukte eksisterende teknologier, noe som gjorde at gruppen måtte lære seg dette, i stedet for å bruke teknologier som de var kjent med fra før.

Metoden gruppen valgte for prosjektet var scrum inspirert metode. Der ukentlig arbeid ble delt opp til daglig arbeid som kunne gjennomføres i ukentlige sprinter. For å kunne

følge disse planene delte gruppen oppgavene opp i ulike iterasjoner, der hver iterasjon bygde på notifikasjonssentralen. Resultatet av ukjente teknologier og ukentlige sprinter førte til at når gruppen støtte på større problem kunne det føre til at tidsplanen ikke blir holdt.

Den iterative prosessen brukt under utvikling gjorde gruppen tidlig hadde resultater å vise til samt fungerende notifikasjonssentralen. Fordelene med dette var at gruppen underveis i prosessen hadde et produkt som virket. Gruppen kunne lære seg de nye teknologiene enklere på grunn av den iterative arbeidsmåten delt opp i daglig arbeid. Oppgavene ble konkrete, dette førte til at gruppemedlemmene kunne lære seg teknologiene litt etter litt.

Konsekvensen av valget om å bruke funksjonell komponent var ikke betydelig. Klasse komponenter og funksjonelle komponenter er laget slik at de virker begge veier, fra klasse til funksjonell komponent, og fra funksjonell til klasse komponent. Den største konsekvensen av valget var ved uthenting av data fra state. Dette måtte gjøres på en annen måte, men når gruppen fant ut av det var resten relativt rett frem. En ting å bemerke seg er at kall av post metoder fra notifikasjonssentralen sin "creator" ble gjort fra andre komponenter enn notifikasjons komponenten, ble det gjort på React klasse komponent måten. Dette resulterte i at gruppen måtte lære seg å gjøre det på begge måtene. Noe som er en fordel senere ved arbeid med klasse komponenter.

6.3 Forbedringspotensiale

Som tidligere nevnt er oppgaven relativt konkret og begrenset av det som allerede eksisterer i applikasjonen. Dette medfører at det ikke er mange alternativer til løsningen gruppen har gått for. Til tross for dette er det enkelte ting som kunne blitt gjort annerledes. Gruppen valgte å lage notifikasjonssenteret ved hjelp av material UI i stedet for å utforme det på samme måte som var gjort tidligere i liknende menyer i applikasjonen. Dette vil resultere i et svært likt resultat, men det visuelle vil være vanskeligere å gjengi gjennom material UI enn det hadde vært ved hjelp av den gamle koden.

Når det kommer til databaser og backend ville noen annen løsning enn å benytte seg av de allerede eksisterende metodene og databasene være lite gunstig, og er derfor en

selvforklarende løsning. Det vil si at resultatet med liten sannsynlighet ville blitt bedre om det ble gjort det på nytt med en annen fremgangsmåte.

7. KONKLUSJON OG VIDERE ARBEID

7.1 Oppsummering

Målet for prosjektet var å utvikle et notifikasjonssenter i systemet til Wide Assessment. Det skulle være to typer notifikasjoner, en for kandidater og en for bedrifter. Når en kandidat blir kontaktet av en bedrift skal det komme opp en notifikasjon hos kandidaten. Det samme skal skje for en bedrift dersom en kandidat søker på en stilling eller “poker” bedriften. Når en notifikasjon er sett av brukeren skal den markeres som lest, slik at uleste notifikasjoner kan fremheves. Det skal også være mulig for WA å sende ut notifikasjon til alle kandidater, alle bedrifter, eller alle brukere, for å eksempelvis varsle om ny funksjonalitet på plattformen eller lignende.

Nettstedet brukes i dag av flere bedrifter og kandidater, og Wide Assessment ønsker å gjøre prosessen fra å gå fra kandidat til ansatt så enkel som mulig. De ønsket derfor å implementere et notifikasjonssystem slik at en kandidat eller bedrift med en gang kan se hva som er nytt siden de sist var pålogget.

7.2 Initielle mål

De initielle målene ble nådd, alle funksjonene som ble beskrevet i oppgaveteksten ble implementert, i tillegg til at gruppen lagde admin side for utsending av driftsmeldinger til alle brukere eller kun kandidat/bedrifter.

7.3 Relevante resultater

Resultatene gruppen har oppnådd er svært relevante, ettersom et notifikasjonssystem er noe svært mange nettsteder har implementert. Teknologien som ligger bak er også svært relevant for nettside programmering, ved hjelp av TypeScript lages “creators”, “reducers” og “actions”. Dette gjør det mulig å bruke REST API i React, ved bruk av et HTTP-bibliotek kalt Axios. Axios spørringene blir utført i “creator”, ved assistanse fra “reducer”, “action” og en “store” som har

applikasjonen sin state (*Ibrahima Ndaw 2020*). Dette samler logikken for spørringer og state på ett sted, som fører til ryddig lettleselig og kode som er enklere å vedlikeholde.

7.4 Videre arbeid

Oppsummert ble målet for prosjektet nådd, gruppen implementerte de funksjonene som ble beskrevet i oppgaveteksten uten å måtte gjøre noen kompromisser. Med tanke på videre arbeid med notifikasjonssenteret hadde gruppen et møte med Wide Assessment. Her ble det gått gjennom arbeidet gruppen hadde utført, og det ble diskutert hva som var bra og hva som kunne forbedres.

Etter møte med Wide Assessment ga de gruppen tilbakemeldinger på det som var gjort. Tilbakemeldingen var at oppgaven var løst på en tilfredsstillende måte, med alle ønskede funksjoner implementert. Wide Assessment var også tilfreds med måten notifikasjonssentralen var integrert i nettsiden deres. Designet passet bra inn og tok seg godt ut i nettsiden deres, i henhold til farger og stilsettet som var brukt.

Under møtet ble det diskutert ulike forbedringspotensialer. Gruppen hadde blant annet en kolonne i databasen som ble kalt “createdby”, som settes til 0,1 eller 3. Dette er for å bestemme om en notifikasjon skulle sendes til kandidat (0), bedrift (1), eller alle (3). Dette var ikke lett å forstå ut ifra navnet på kolonnen, så her det ganske enkelt å endre navn på databasekolonnen til noe mer beskrivende.

Enkelte steder i databasen benyttet gruppen seg av tall verdier for å markere true, false og liknende i stedet for å bruke boolean. Det ble her forklart at det kan benyttes enum sammen med boolean verdier for lettleselig kode.

Det ble også oppdaget en feil som gjør at når en notifikasjon blir sendt vil ikke brukere få beskjed om dette før siden blir oppdatert. Dette gjør nettsiden mindre dynamisk, men kan fikses ved å legge til kode som poller med bestemt tidsintervall etter endring i notifikasjoner.

Til slutt ble det oppdaget en feil i administratorpanelet som gjør at når en bruker markerer en felles melding som lest, vil den bli markert som lest av alle i samme gruppering. Denne feilen kommer av at notifikasjonssystemet i administratorpanelet lagres i databasen på samme måte som de enkelte notifikasjonene. For å fikse et slikt problem vil en mulig løsning være å relatere lest/ulest med brukerens ID i en egen tabell, slik at det blir markert individuelt.

Grunnet tidsbegrensning ble det ikke være tilstrekkelig med tid til å fikse disse problemene før fristen, og feilene må fikses i etterkant av prosjektet av bedriften selv.

7.5 Konklusjon

Problemstillingen til oppgaven er hvordan et tilpasset notifikasjonssenter kan gjøre det enklere å formidle informasjon til kandidater og bedrifter. Målet med oppgaven var å løse dette på en enkel, intuitiv og funksjonell måte som bidrar til å gjøre nettstedet mer brukervennlig og effektivt.

Som nevnt tidligere i oppgaven har vi ikke fått andre enn gruppen selv til å teste notifikasjonssenteret. Gruppen kan derfor ikke med sikkerhet si at det har blitt enklere å formidle informasjon til kandidater og bedrifter, men ut ifra møtet med Wide Assessment så var de veldig positive til arbeidet som var gjort. Den ønskede funksjonaliteten var på plass, og designmessig passet notifikasjons komponenten inn i nettstedet. Det avsluttende møtet med Wide Assessment burde vært gjennomført tidligere, dette hadde gitt gruppen lengre tid til å fikse de fleste problemene nevnt i videre arbeid (kap 7.4). Grunnet korona ble møter med bedriften gjort digitalt, og dette fungerte greit. Hadde gruppen hatt mulighet til å arbeide noen dager i lokalene til Wide Assessment ville deler av utviklingen blitt gjort raskere. Samtidig ville oppdragsgiver hatt muligheten til å vurdere arbeidet fortløpende og eventuelle mangler ville blitt oppdaget tidligere.

8. REFERANSER

- *Harvard-eksempler*: <https://www.ntnu.no/viko/harvard-eksempler>
(hentet 01. Mars 2021)
- *.Net documentation*: <https://docs.microsoft.com/en-us/dotnet/fundamentals/> (Hentet: 15. Mars 2021).
- Ibrahim Ndaw (2020) *How to use redux in your react typescript app*: <https://www.freecodecamp.org/news/how-to-use-redux-in-your-react-typescript-app/> (Hentet 10. April 2021).
- Claire Drumond *What is scrum* <https://www.atlassian.com/agile/scrum>
(hentet 10. April 2021).
- *Unit testing 2021*: https://en.wikipedia.org/wiki/Unit_testing
(Hentet 22. April 2021).
- Babich N. (2020) *Web Page Design: A Comprehensive Guide*: <https://xd.adobe.com/ideas/principles/web-design/web-page-design/>
(Hentet: 29. April 2021).
- React State: https://www.w3schools.com/react/react_state.asp
(hentet 05. Mai 2021).
- Priya. (2020) *15 React Best Practices You need to follow in 2021*: <https://www.codeinwp.com/blog/react-best-practices/>
- Scrum.org *What is a product backlog* <https://www.scrum.org/resources/what-is-a-product-backlog>
(hentet 10. Mai 2021).
- W3 schools *Html responsive web design* https://www.w3schools.com/html/html_responsive.asp
(hentet 10. Mai 2021).
- Sofia simic (2019) *Docker Image VS Container: The Major Differences* <https://phoenixnap.com/kb/docker-image-vs-container>
(Hentet 10. Mai 2021)

9. Vedlegg

9.1 Risikoliste

Risiko	S	K	RF	Tiltak	Interessent	Fase
Sykdom/covid	3	3	9	Hjemmekontor, særlig ved symptomer	Gruppe	Kontinuerlig
Tekniske feil	1	3	3	Teknisk hjelp fra arbeidsgiver/ studiested	Gruppe	Kontinuerlig
Miskommunikasjon	1	3	3	Samarbeid og flittig bruk av kommunikasjonsverktøy	Gruppe/ arbeidsgiver	Kontinuerlig
Mangel på kunnskap	3	2	6	Benytte seg jevnlig av ressurser og veileder/ arbeidsgiver	Gruppe	Kontinuerlig
Endring på oppgavebeskrivelsen	1	4	4	God kommunikasjon med arbeidsgiver	Gruppe	Kontinuerlig

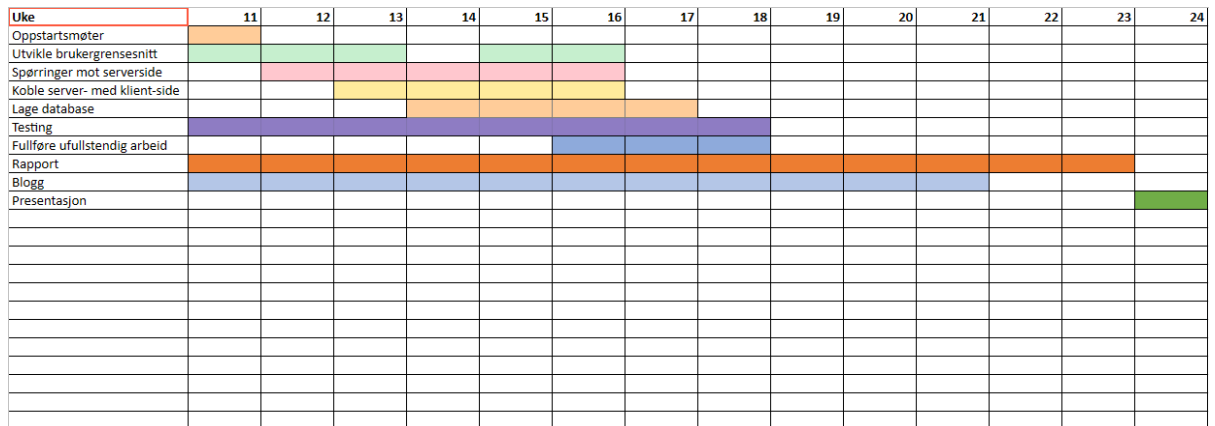
(Figur 10.1.1 Risikoliste)

S = Sannsynlighet at hendelsen antreffer (1-5)

K = Alvorlighetsgraden av konsekvensen (1-5)

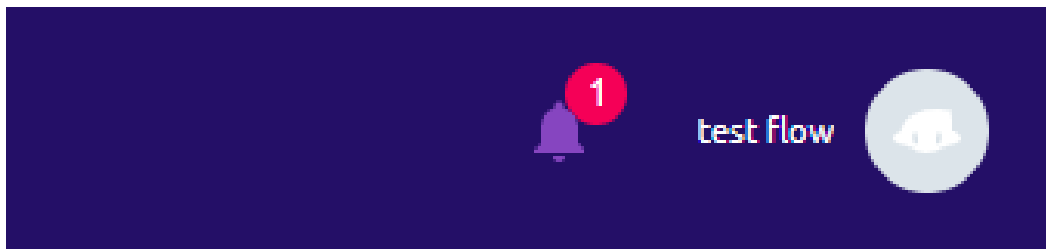
RF = Risk-faktoren (S ganget med K)

9.2 GANTT diagram

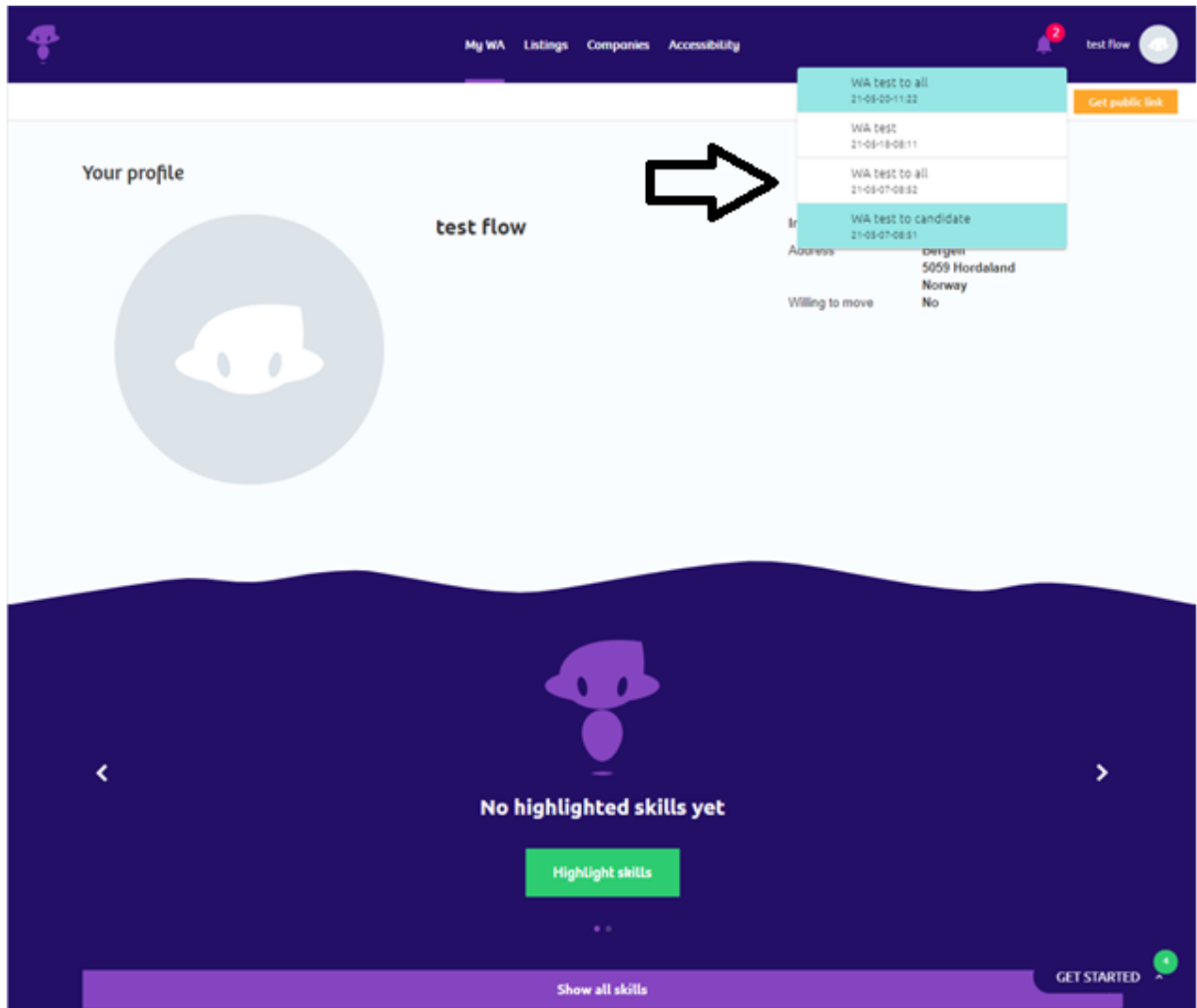


(Figur 9.2.1 Gantt-diagram)

9.3 Detaljert design

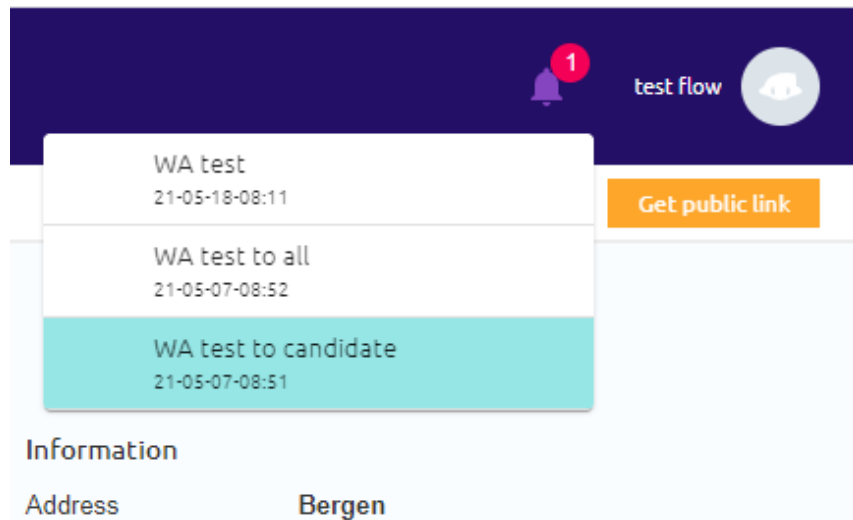


(Detaljert design 9.3.1 Notifikasjonssymbol)



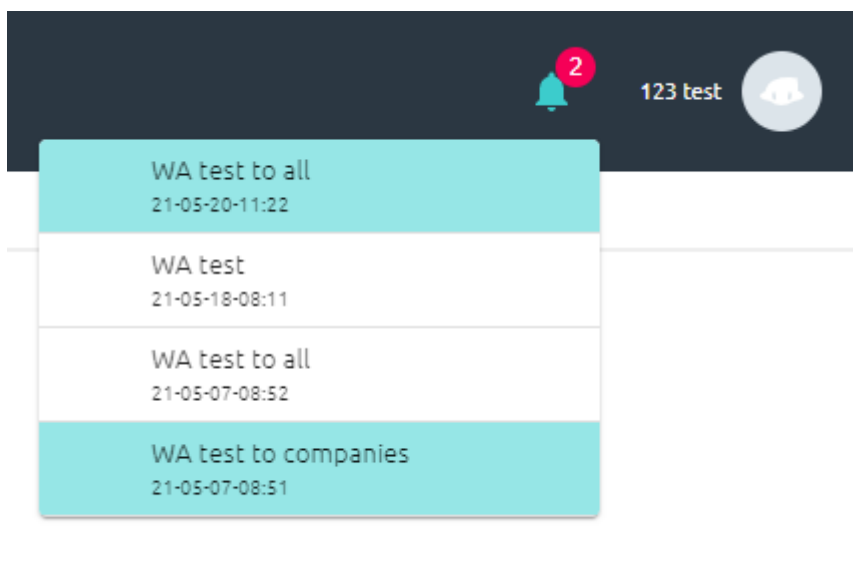
(Detaljert design 9.3.2 Notifikasjonslisten)

For kandidat:

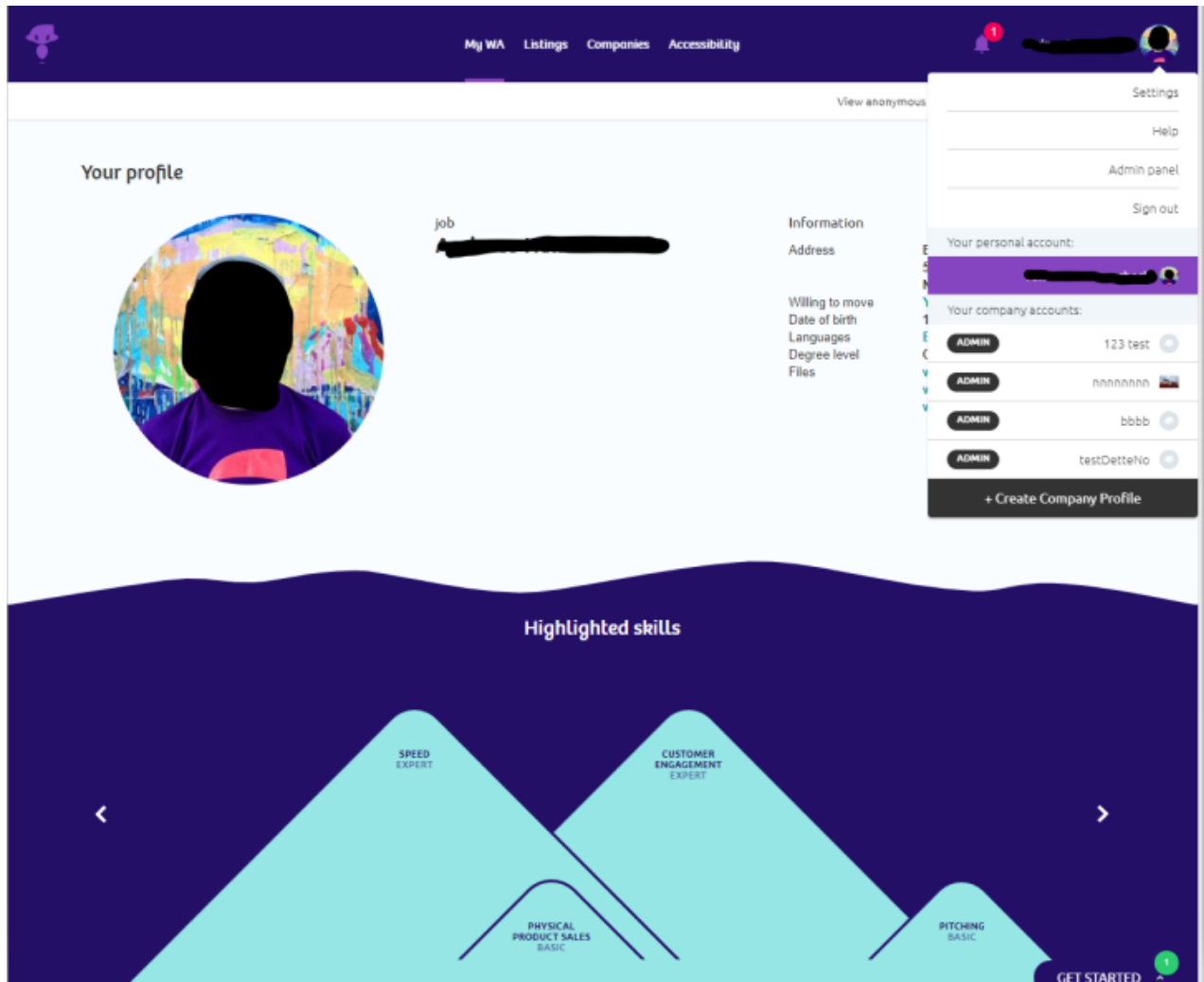


(Detaljert design 9.3.3 Notifikasjonslisten for kandidat)

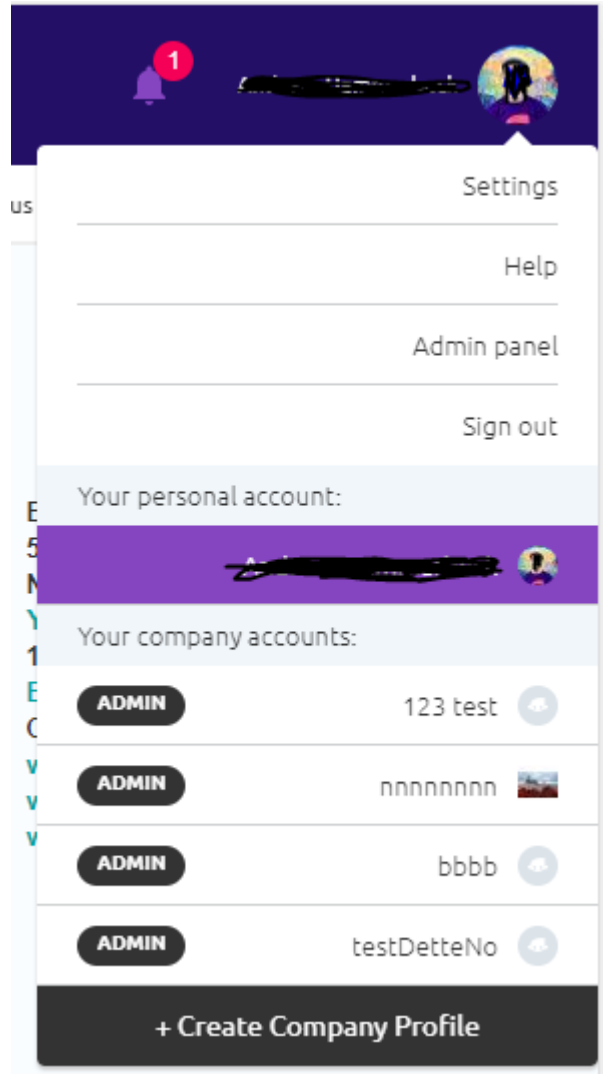
For bedrift:



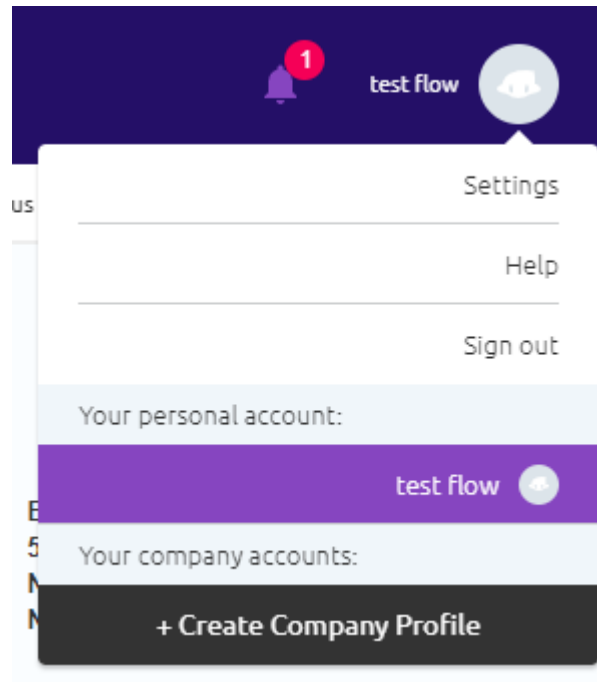
(Detaljert design 9.3.4 Notifikasjonslisten for bedrift)



(Detaljert design 9.3.5 Knapp for å gå til administratorpanel)



(Detaljert design 9.3.6 Med administrator rettigheter (admin panel knapp))



(Detaljert design 9.3.7 Uten administrator rettigheter)

Write a notification and choose who to send it to.

Notification

Choose recipient:

Candidates Companies All

SEND

(Detaljert design 9.3.8 Administratorpanel)

Request Headers

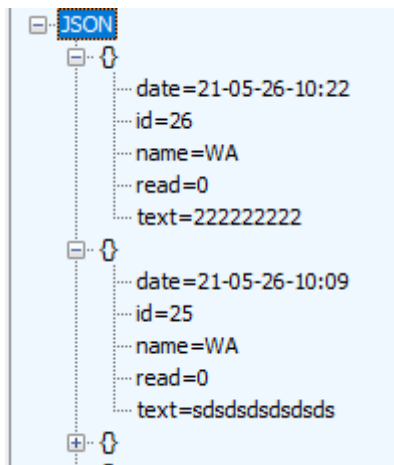
POST /api/notificationcenter/sendPokeFromUser/112 HTTP/1.1

(Detaljert design 9.3.9 post url for poking av bedrift)

```
HTTP/1.1 200 OK
X-Powered-By: Express
connection: close
date: Wed, 02 Jun 2021 13:41:21 GMT
content-type: application/json; charset=utf-8
server: Kestrel
transfer-encoding: chunked

32b
[{"id":26,"name":"WA","text":"222222222","read":0,"date":"21-05-26-10:22"}, {"id":25,
```

(Detaljert design 9.3.10 get response raw for henting av notifikasjoner)



(Detaljert design 9.3.11 get response JSON for henting av notifikasjoner)

Request Headers

POST /api/notificationcenter/sendNotificationToAll/send%20demo/0 HTTP/1.1

Client

(Detaljert design 9.3.12 post header for sending av notifikasjon med teksten: send demo)