



Høgskulen
på Vestlandet

BACHELOROPPGAVE

LinkUp - plattformen som fremmer
relasjoner

LinkUp - the platform promoting relations

**Adrian Berget, Amund Fremming & Dennis
Rizah Nordvi Osmani**

DAT191 Bacheloroppgave

Fakultet for teknologi, miljø- og samfunnsvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Veileder: Volker Stolz

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, § 10.

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> LinkUp - plattformen som fremmer relasjoner	<i>Dato:</i> 10.05.2024
<i>Forfatter(e):</i> Adrian Berget, Amund Fremming og Dennis Rizah Nordvi Osmani	<i>Antall sider u/vedlegg:</i> 51
	<i>Antall sider m/vedlegg:</i> 83
<i>Studieretning:</i> Dataingeniør	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Volker Stolz	<i>Gradering:</i> Ingen
<i>Merknader:</i> Ingen	

<i>Oppdragsgiver:</i> Ingen	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Ingen	<i>Telefon:</i> Ingen

<i>Sammendrag:</i> Bacheloroppgaven gikk ut på å implementere og dokumentere utviklingen av en fullstack Native Applikasjon som tok sikte på å skape en dedikert plattform for sosialisering gjennom eventer. Applikasjonen var en MVP av et større system, men skulle likevel omfatte et stort antall funksjonaliteter, slik at den var mulig å ta i bruk uten umiddelbart behov for videre utvikling.
--

Stikkord:

Native Applikasjon	Fullstack	Eventer	Sosial plattform	React Native	C# - ASP.NET Core
--------------------	-----------	---------	------------------	--------------	-------------------

Høgskulen på Vestlandet, Fakultet for teknologi, miljø- og samfunnsvitenskap

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

Denne rapporten dokumenterer arbeidet som har blitt gjort ifm. bachelorprosjektet *LinkUp - Plattformen som fremmer relasjoner*. Valg- og utforming av oppgave ble gjort av studentene selv, etter godkjenning av emneansvarlig ved HVL. Prosjektet er en avsluttende del av studieløpet for bachelor i dataingeniør ved Høgskulen på Vestlandet.

Vi ønsker å takke vår veileder Volker Stolz for god oppfølging og tilbakemeldinger gjennom hele perioden, i form av møter og annen dialog. I tillegg til dette ønsker vi å takke hverandre for samarbeidet gjennom prosjektperioden. Nye teknologier og omfattende prosjektvalg førte til en tidkrevende prosess hele semesteret, og ville ikke vært mulig å gjennomføre uten en stor dedikasjon fra samtlige gruppemedlemmer.

Ordliste

Ord/Forkortelse	Beskrivelse
Feed	En liste med innlegg på sosiale medier, oftest lagt ut av applikasjonens brukere.
Skår	Fra engelsk "Score", brukt til å referere til poengsystemet.
Native applikasjon	Applikasjon utviklet spesielt for en bestemt plattform som iOS eller Android.
HTTP	Hypertext Transfer Protocol - en protokoll for å sende data over internett.
Token (JWT)	En standardisert metode for sikker overføring av informasjon mellom to parter som JSON-objekter, vanligvis brukt til autentisering og autorisasjon.
Bearer Authentication Token	En type autentiseringstoken som brukes til å gi tilgang til ressurser på nettet, vanligvis brukt sammen med HTTP-forespørsler.
UI-komponent	En gjenbrukbar byggeblokk i brukergrensesnittet (UI) som utfører spesifikke funksjoner eller viser innhold på skjermen.
Navigasjonsbar	Meny som gir brukeren mulighet for å flytte seg mellom ulike sider i en applikasjon.
Liveserver	En server på en pc som kontinuerlig leverer og oppdaterer en applikasjon eller nettside i sanntid, slik at utviklere kan se endringer umiddelbart mens de arbeider på koden.
Xcode	En IDE (integreert utviklingsmiljø) som kan simulere en Iphone med full funksjonalitet.
Ingress plattform	En programvareløsning som regulerer ekstern tilgang til en applikasjon/tjeneste ved å tilby autentiserings- og autorisasjonsmekanismer
Interfaces	En definisjon som beskriver de metoder og egenskaper som et objekt kan støtte
Modal	Et vindu som dukker opp over elementet som er på skjermen i utgangspunktet.

Azure App Service	En plattform i Microsoft Azure som lar utviklere bygge, distribuere og skalere web- og mobilapper.
App Store Connect	En plattform utviklet av Apple som tillater utviklere å legge ut, oppdatere og overvåke applikasjoner.
Testflight	Plattform utviklet av Apple som muliggjør kontrollert testing av beta-versjoner før offisiell lansering av en applikasjon.
DTO	Et designmønster i softwareutvikling der et objekt (DTO) brukes til å bære spesifikke data mellom ulike deler av et program.
PWA	En type webapplikasjon som gir brukere en opplevelse som likner på en native applikasjon, samtidig som den kan nås via en nettleser på ulike enheter.
SOLID	Et akronym som står for fem designprinsipper innen objektorientert programmering, laget for å standardisere god praksis på oppsett av kode.
Task-based asynchronous Pattern (TAP)	Et designmønster i .NET som gjør det mulig for utviklere å skrive asynkrone metoder ved å returnere objekter av typen Task eller Task<T>.

INNHALDSFORTEGNELSE

1	INNLEDNING	1
1.1	Kontekst	1
1.2	Motivasjon.....	1
1.3	Prosjekteier.....	2
1.4	Problembeskrivelse og mål	2
1.5	Oppbygging av rapporten.....	3
2	PROSJEKTBEKRIVELSE	4
2.1	Praktisk bakgrunn.....	4
2.1.1	Initielle krav	4
2.1.2	Initiell løsnings-idé	4
2.2	Avgrensninger	5
2.3	Ressurser	7
2.3.1	Ressurser for utvikling.....	7
2.3.2	Litterære ressurser.....	7
2.4	Empirisk utforskning av problemstillingen	7
3	DESIGN AV PROSJEKTET	9
3.1	Forslag til løsning.....	9
3.1.1	Alternativ løsning 1 - Webapplikasjon/PWA.....	9
3.1.2	Alternativ løsning 2 - Native applikasjon.....	10
3.1.3	Diskusjon av alternativene.....	10
3.2	Valgt løsning	11
3.3	Valg av verktøy	11
3.4	Prosjektmetodikk.....	13
3.4.1	Utviklingsmetodikk	13
3.4.2	Valg av utviklingsmetodikk.....	15
3.4.3	Prosjektplan	16
3.4.4	Risikovurdering	16
3.5	Evalueringsplan.....	17
4	DETALJERT LØSNING	19
4.1	Overordnet arkitektur.....	19
4.1.1	Systemarkitektur	19

4.1.2	Applikasjonsflyt.....	20
4.2	Backend-API og database	22
4.2.1	Rammeverk og biblioteker	22
4.2.2	Arkitektur.....	23
4.2.3	API og swagger.....	24
4.2.4	Sikkerhet og tokens.....	26
4.2.5	Database.....	27
4.3	Frontend og brukergrensesnitt	29
4.3.1	Rammeverk og biblioteker	29
4.3.2	Oppbygning og struktur.....	31
4.3.3	Brukergrensesnitt.....	32
5	RESULTATER	38
5.1	Evalueringsmetode.....	38
5.2	Evalueringsresultat.....	39
5.3	Prosjektresultat.....	41
5.4	Prosjektgjennomføring.....	42
6	DISKUSJON.....	43
6.1	Fremgangsmåte og fremdrift.....	43
6.2	Negative konsekvenser	43
6.3	Mulige forbedringer	44
7	KONKLUSJON OG VIDERE ARBEID.....	46
7.1	Konklusjon	46
7.2	Videre arbeid.....	47
8	REFERANSER	49
	Figurer.....	51
9	VEDLEGG.....	52

1 INNLEDNING

I det innledende kapittelet forklares bakgrunnen til prosjektet. Dette omfatter motivasjonen for prosjektet, sammenhengen oppgaven er skrevet i, samt en beskrivelse av problemet som ligger til grunn for gjennomføringen av prosjektet. Det blir også beskrevet hvordan rapporten er strukturert.

1.1 Kontekst

Bacheloroppgaven ved Høgskolen på Vestlandet gir studenter mulighet til å arbeide med et egendefinert prosjekt dersom bakgrunnen og ideen er solid nok. Denne rapporten er et eksempel på en slik oppgave, og er et resultat av en studentgruppe som over en lengre periode har drøftet ulike teknologiske mangler i samfunnet. Ideen ser på muligheten og potensielle løsninger for en bedre plattform for planlegging av aktiviteter i grupper. Årsaken til at valget falt på nettopp dette prosjektet er en kombinasjon av behovet og nytteverdien gruppen ser av et slikt produkt i samfunnet, samt relevans til studiet.

1.2 Motivasjon

Basert på egne erfaringer og diskusjoner har prosjektgruppen observert at dagens løsninger for å lage og delta i sosiale arrangementer ikke er gode nok. Dette påvirker både personer i nye omgivelser som ønsker å møte og bli kjent med nye mennesker, og allerede etablerte grupper og gjenger. Gruppen mener at nåværende løsninger er forbundet med en høy terskel for å ta i bruk verktøyene til det de er ment for.

For gruppen mennesker som søker nye relasjoner, er dagens tilbud hovedsakelig bestående av arrangementer som arrangeres i regi av bedrifter, og naturligvis store offentlige arrangementer som f.eks. konserter og festivaler. Når det gjelder allerede etablerte kretser, blir det gjerne benyttet andre uegnede sosiale medier til planlegging, med lite oversikt og overflødig kommunikasjon. For gruppen har denne tydelige mangelen på en applikasjon som svarer på disse behovene, dannet motivasjonen for utviklingen av det aktuelle prosjektet.

Plattformen kan både hjelpe til med å effektivisere prosesser, samtidig som den kan bidra til å forbedre psykisk helse gjennom sosialisering. Å skape en fullstendig applikasjon fra bunnen av

som er av nytte for samfunnet, er også en viktig del av motivasjonen som ligger til grunn for prosjektet.

1.3 Prosjekteier

Prosjektgruppen, bestående av Adrian Berget, Amund Fremming og Dennis Rizah Nordvi Osmani er prosjekteierne i dette prosjektet. Ettersom at det ble tatt et valg om en egendefinert bacheloroppgave, som nevnt i delkapittel [1.1](#), er ingen eksterne selskaper involvert i utviklingen. Etter prosjektperioden, vil gruppen selv ha eierskap til produktet, og dermed mulighet og frihet til å videreutvikle produktet dersom dette er ønskelig. Høgskolen på Vestlandet hadde en veiledende rolle gjennom utviklingen av prosjektet, representert av Volker Stolz.

1.4 Problembeskrivelse og mål

Det er godt dokumentert innenfor moderne antropologi at sosialisering bidrar til å forhindre mental nedbrytning, og dermed er en sentral del av psykisk velvære og livskvalitet (2022, Bisbal). Nåværende løsninger for å oppsøke sosialisering består av store sosiale medier, hvor noen av disse har innebygd funksjonalitet for å opprette og delta i arrangementer, og andre ikke. Disse, sammen med et antall dedikerte plattformer for planlegging av arrangementer, gir et tilbud som hovedsakelig er representert av bedrifter og fagrelaterte hendelser med økonomiske og markedsføringsrelaterte insentiver. Det er gruppens oppfatning at terskelen forbundet med løsningene også fører til at allerede etablerte sosiale grupper tyr til ikke-dedikerte plattformer for planlegging og kommunikasjon.

Problemstillingen for prosjektet blir dermed: Hvordan kan det utvikles en plattform dedikert til arrangementer, som gir en lav terskel både for mennesker som oppsøker nye sosiale kretser, og allerede etablerte grupper som ønsker mer effektiv planlegging?

Målet for oppgaven er å svare på problemstillingen gjennom en fungerende applikasjon som tilfredsstillende teknologiske og ikke-teknologiske krav som er satt av prosjektgruppen. De mest essensielle funksjonalitetene som behøves for å kunne svare på problemet vil være en lokasjonsbasert event-strøm som gir forslag basert på hvor brukeren befinner seg, invitasjonslenker for enkel deling gjennom andre kanaler, og ulik synlighet for ulike typer arrangementer.

1.5 Oppbygging av rapporten

Rapporten er delt inn i ni kapitler, inkludert dette kapitlet som er en innledningsdel. Kapittel 2 inneholder en beskrivelse av valgene som ble tatt i forbindelse med prosjektet, herunder krav, avgrensninger og ressurser. Kapittel 3 begrunner den valgte løsningen for prosjektet, samt verktøy og prosjektmetodikker som har blitt tatt i bruk. I kapittel 4 beskrives den valgte løsningen av problemstillingen i detalj, og tekniske valg og spesifikasjoner vil bli gjort rede for. Kapittel 5 viser evalueringen og resultatet av prosjektet sett i sammenheng med problemstillingen. I Kapittel 6 og 7 vil prosjektet bli drøftet gjennom diskusjon, og deretter følger en konklusjon og plan for videre arbeid. Kapittel 8 og 9 viser referanser og vedlegg som har blitt benyttet i rapporten.

2 PROSJEKTBEKRIVELSE

Dette kapitlet fokuserer på fundamentet og den praktiske bakgrunnen som prosjektet er bygget på. Prosjektets opprinnelige krav og ideer vil bli beskrevet, samt avgrensningene som har blitt gjort for å skape en realistisk og gjennomførbar prosjektplan.

2.1 Praktisk bakgrunn

2.1.1 Initielle krav

De initielle kravene baserer seg på hvordan en bruker skal kunne navigere i- og benytte seg av applikasjonen. En bruker skal kunne registrere seg selv, enten manuelt eller ved tredjeparts innlogging med Google, Apple eller Vipps. Dette tillater en rask registreringsprosess, og deretter kan brukeren benytte seg av applikasjonens funksjonaliteter. Brukeren skal kunne opprette eventer for små og store sammenkomster, og begrense synligheten til privat, venner eller offentlig. Brukere skal kunne opprette relasjoner seg imellom, og invitere andre ved å sende invitasjonlenker for enkel spredning av arrangementer. Eventer i nærområdet skal dukke opp i en event-feed. Videre skal brukere kunne delta på eventer, enten de er offentlige eller om de har mottatt en invitasjon. Hver bruker har hver sin bail-skår ifm. oppmøte og svarpresisjon på eventer. Dette registrerer tilfeller hvor en bruker melder seg på uten å stille opp, slik at brukerne stilles ansvarlig for å svare presist inne i applikasjonen. Etter eventer skal deltakere i eventet vurdere arrangøren og eventet basert på egen opplevelse.

Det er også et krav om at brukere skal kunne interagere med eventer de skal delta i. Dette innebærer at det skal være mulig å kommunisere med andre deltakere vha. en meldingstjeneste/chat, samt svare på meningsmålinger. Applikasjonen skal også potensielt kunne samarbeide med andre bedrifter eller organisasjoner, slik at disse skal kunne promotere, arrangere, og drive markedsføring via applikasjonens plattform.

2.1.2 Initiell løsnings-ideé

Den initielle løsnings-ideén er forankret i de initielle kravene, og tar i tillegg til å løse problemstillingen, sikte på å være en høykvalitets-applikasjon som inneholder alt en bruker kan dra nytte av i forbindelse med eventer. Kjernen i applikasjonen var opprettelse og deltakelse i

eventer med tilstrekkelig informasjon, og mulighet for brukere til å velge om de skal delta eller ikke på disse. I tillegg til dette var også relasjoner mellom brukere i form av vennskap en sentral del. Applikasjonen skulle inneholde ulike event-strømmer: offentlige åpne eventer for alle, eventer kun for venner og eventer kun for inviterte. Den offentlige strømmen skulle ha ulike sorteringsparametere for å la brukere finne eventer basert på egne preferanser.

Det var også planlagt å implementere kommunikasjon mellom brukere i form av chat, i tillegg til statusoppdateringer og meningsmålinger i eventer. Rangeringssystemet som lar brukere gi skår til både event- deltakere og arrangører, var tenkt som en innovativ løsning for å fremme ærlighet og troverdighet. Den initielle løsningen involverte også tredjepartssystemer i forbindelse med autorisering og kartintegrasjon. Applikasjonen skulle være tilgjengelig på iOS og Android, via henholdsvis App Store og Google Play Store, og dermed gi brukere mulighet til å få varslinger fra appen.

2.2 Avgrensninger

Den initielle løsningsidéen som er beskrevet ovenfor i delkapittel [2.1.2](#) utgjør en stor applikasjon med mye funksjonalitet og mange små detaljer. Grunnet det store omfanget av den fullstendige løsningen, og den begrensede varigheten av prosjektperioden, har det blitt gjort en rekke avgrensninger for å definere et gjennomførbart prosjekt. For å kunne avgrense applikasjonen ble den initielle funksjonaliteten til løsningen sortert ved bruk av et MOSCOW diagram, se figur 2.1. Dette deler funksjonalitet inn i kategoriene “must have”, “should have”, “could have” og “won’t have”.



Figur 2.1 - Moscow Diagram utviklet for å spesifisere avgrensningen

Bruken av et slikt diagram har gjort prosessen med å avgrense prosjektet oversiktlig. I tillegg gir det mulighet til å kunne legge til funksjonalitet fra “should have” og “could have”, dersom produktet skal videreutvikles på et senere tidspunkt. Punktene som er oppgitt under “must have” representerer det som skal implementeres gjennom prosjektperioden. Denne funksjonaliteten er også definert som krav, og beskrives mer omfattende i vedlegg 2 - Kravdokument. De definerte kravene utgjør en MVP som tilbyr de mest essensielle funksjonalitetene for systemet, samtidig som at løsningen er omfattende nok til å svare på problemet som beskrives i problemstillingen. Funksjonaliteten som blir utelatt fra prosjektet er blant annet meldingstjenester, meningsmålinger, samtaletråder i eventer, tredjepartsintegrasjoner og en rekke systemer for å rangere brukere på troverdighet og ærlighet.

2.3 Ressurser

2.3.1 Ressurser for utvikling

Prosjektet ble utformet, planlagt og implementert ved hjelp av en kombinasjon av Windows- og iOS-maskiner. For tilgang til database og skylagring ble HVL sin PostgreSQL database tatt i bruk, som blir beskrevet nærmere i delkapittel [3.3](#). For hosting av applikasjonen valgte gruppen å benytte seg av Azure App Service.

2.3.2 Litterære ressurser

I løpet av prosjektet ble det utforsket en rekke nye teknologier, som gruppen hadde ingen eller lite erfaring med fra tidligere. I den forbindelse ble det investert tid i å lese dokumentasjon fra både Microsofts sider om ASP.NET Core, i tillegg til React Native sin dokumentasjon. Før utviklingsarbeidet startet, ble det også gjennomført en fase med praktisk testing av teknologiene. For skriving av rapport, dokumentasjon, og planlegging av prosjektet, benyttet gruppen teori fra tidligere studerte emner, informasjon fra forelesninger i DAT191, samt en grundig gjennomgang av tidligere oppgaver.

2.4 Empirisk utforskning av problemstillingen

Problemstillingen for prosjektet var i utgangspunktet bygget på antakelser og påstander. For å bekrefte og underbygge at problemet var reelt, samt justere problemstillingen, ble det tidlig gjennomført en spørreundersøkelse som tok sikte på å gi en dypere innsikt i brukeres behov og vaner i de aktuelle bruksområdene til produktet. Resultatene fra undersøkelsen ble også en viktig ressurs i forbindelse med valg av løsning og krav. Undersøkelsen skilte mellom de to ulike bruksområdene; etablerte- og ukjente sosiale kretser. Det var totalt 90 respondenter i undersøkelsen, og majoriteten oppga en alder på mellom 18 og 60 år. Hele undersøkelsen med svar er vedlagt i vedlegg A.

Det ble avdekket et tydelig behov og potensial i delen som omhandlet løsninger for å utforske nye sosiale kretser. Halvparten av respondentene var nøytrale, lite-, eller veldig lite fornøyde med dagens løsninger, og den samme mengden svarte at de synes det er vanskelig å finne arrangementer som passer egne interesser. Disse tilbakemeldingene viser at det overordnede problemet er reelt og at det er behov for produktet. I tillegg til dette viste det seg at halvparten

deltar i slike arrangementer “noen ganger”, og over 80% opplever en høy eller svært høy terskel knyttet til eksisterende løsninger. Av dette ble oppfattelsen om at det eksisterer en tydelig terskel knyttet til bruk av dagens tilbud bekreftet.

Delen av undersøkelsen som tok for seg eventer i etablerte sosiale kretser bekreftet også prosjektgruppens oppfatning av dagens situasjon. 50% av deltakerne i undersøkelsen planlegger slike arrangementer en gang i uken eller oftere, og ytterligere 29% gjør det minst annenhver uke. I tillegg svarte over 80% at de benytter seg av Messenger og Snapchat for slik planlegging, som ikke har innebygget funksjonalitet for planlegging av eventer. Disse tilbakemeldingene bekrefter at det er et tydelig behov for å planlegge hendelser og eventer, men at det ikke benyttes en effektiv kommunikasjonskanal for dette. Om lag 40% svarte at de ofte eller hver gang hadde problemer med oversikt og informasjonsflyt, samt ytterligere 17% var usikre på spørsmålet. Dette bekrefter behovet og forbedringspotensialet ytterligere.

Avslutningsvis svarte over 60% av respondentene at det var svært sannsynlig eller sannsynlig at de ville tatt i bruk en ny applikasjon for å opprette eller delta i små og store eventer. Til tross for at dette er en positiv tilbakemelding som bekrefter behovet i markedet, erkjenner prosjektgruppen at det faktiske prosenttallet vil være betraktelig lavere. Dette fordi det er lettere å svare ja i en slik setting, enn å faktisk laste ned en ny ukjent applikasjon.

3 DESIGN AV PROSJEKTET

I dette kapitlet utforskes kjernevalgene som styrer prosjektets utførelse, inkludert teknologi, metodikk, og planlegging. Gruppen vil skissere hvordan disse valgene, sammen med risikoanalysen og evalueringsplanen, danner en tilnærming for å nå og bekrefte målene for prosjektet.

3.1 Forslag til løsning

Løsningsalternativene som blir diskutert nedenfor har til felles at de er avhengig av et backend API. Dette implementeres i C#-rammeverket ASP.NET Core, for kommunikasjon mellom database, server og klient. Utformingen av dette API-et vil være tilnærmet likt uavhengig av hvilken løsning som velges for klientsiden av applikasjonen, og blir dermed ikke nevnt ifm. løsningsalternativene.

3.1.1 Alternativ løsning 1 - Webapplikasjon/PWA

Gjennom å utnytte teknologi som gruppen allerede har kjennskap til, var det en mulighet å utvikle en webapplikasjon, i form av en PWA (Progressive Web Application). Dette utvikles ved hjelp av React, et JavaScript-rammeverk spesialisert for utvikling av frontend. Webapplikasjoner tilbyr en enkel og rask løsning for de som ikke ønsker å stille spesifikke krav til operativsystemer eller enheter for bruk av applikasjonen. En stor fordel er at applikasjonen ikke trenger å lastes ned for at brukerne skal få tilgang. Den kan manuelt legges til på hjemskjermen og fremstå som en vanlig applikasjon.

Webapplikasjoner kan ofte utvikles raskt i forhold til andre løsninger, men dette er ikke alltid tilfellet. Fordelen med alternativet ligger i enklere testing, oppdateringer, og publisering. Under utvikling krever en webapplikasjon lite bruk av verktøy, da det ikke er nødvendig med simulatorer eller andre eksterne verktøy, kun en nettleser. Det er også fordelaktig ved at det muliggjør en desktop-versjon av applikasjonen uten behov for store endringer eller tillegg i koden.

3.1.2 Alternativ løsning 2 - Native applikasjon

Tradisjonelle apper som er installert direkte på enhetens operativsystem kalles native applikasjoner. Disse applikasjonene vises ofte som ikoner på enhetens hjem-skjerm, og lastes ned via operativsystemets app-butikk, som for eksempel App Store (iOS) eller Google Play Store (Android). Ettersom at applikasjonene kjører direkte på operativsystemet, gir det løsningen mulighet til å ha en mer tilpasset brukeropplevelse med blant annet varslinger, widgets, lokasjon og enhetstilpasset skalering.

Til tross for at utvikling av native apper ofte krever to separate kodebaser, kan denne utfordringen løses ved bruk av rammeverket React Native. Dette rammeverket gjør det mulig for gruppen å bygge applikasjonen for både iOS og Android fra samme kodebase, noe som gjør utviklingen overkommelig til tross for mangel på forkunnskaper. React Native og tilhørende verktøy forklares mer utdypende i delkapittel [3.3](#).

3.1.3 Diskusjon av alternativene

For å ta en beslutning la gruppen vekt på en rekke ulike faktorer: tilgjengelig tid, nødvendig kunnskap, ønsket utfall, samt brukerens terskel for å benytte seg av applikasjonen.

Løsning 1 kan potensielt redusere tidsbruken i forbindelse med opplæring i ny teknologi, rammeverk, testing, og lansering. Til tross for dette, kan en slik løsning imidlertid gå på bekostning av produktets verdi. Gruppens ambisjon er å etablere en robust plattform for håndtering av arrangementer, som kan videreutvikles over tid. Det er også vesentlig å fremheve pressemeldingen fra Apple i februar 2024, som meddelte at selskapet vil opphøre støtten for PWA-er på Apple-produkter i Europa (Apple, 2024). Dette tiltaket kan påvirke anvendeligheten av løsning 1 negativt ved å redusere produktets tilgjengelighet, øke brukerterskelen, og komplisere integrasjonen av varslinger i applikasjonen. En PWA er også nødvendig for å tillate brukere å legge til webapplikasjonen på egen hjem-skjerm, slik at det tilsynelatende er en tradisjonell app.

Løsning 2 krever mer bruk av ukjent teknologi for utviklingsgruppen. Likevel er det relevant å legge til at gruppen allerede behersker React-teknologien, noe som innebærer at overgangen til React Native forventes å være overkommelig Dette fordi disse teknologiene er relativt like og bygger på mange av de samme prinsippene. Native applikasjoner krever ofte godkjenning fra en "reviewer" og må oppfylle diverse krav både ved lansering og oppdatering av applikasjonen. For å

realisere målet om å utvikle en sentralisert eventplattform der brukeropplevelsen er i fokus, er det en nøkkelfunksjonalitet at applikasjonen integrerer enhetens varslinger og widgets, både på låst og aktiv skjerm. Dette er kun mulig ved bruk av løsning 2.

Det er også viktig å vurdere brukernes generelle tidsbruk på native apper sammenlignet med webapplikasjoner. Statistikk viser at native applikasjoner står for 80-90% av skjermtiden til en gjennomsnittlig telefonbruker (Buck, 2024; Wurmser, 2020).

3.2 Valgt løsning

Etter evalueringen av løsningsalternativer for prosjektet, valgte gruppen å utvikle en native applikasjon. Dette valget, selv om det potensielt forlenger utviklingstiden, ble foretrukket for å balansere mellom prosjektets mål, løsningens funksjonalitet og tilgjengelig data om gjennomsnittlig tidsbruk på native applikasjoner kontra webapplikasjoner. Strategien for løsningen går ut på å lage en Minimum Viable Product (MVP) basert på "must-have" funksjonene fra MOSCOW-diagrammet, som blir vist og forklart i delkapittel [2.2](#).

Valget ble også i stor grad tatt i forbindelse med en vurdering av de siste reguleringene fra Apple som påvirker PWA-er i Europa. Dette førte til beslutningen om å velge en native applikasjon for å sikre applikasjonens potensial i fremtiden. Gjennom å utvikle en MVP, legger gruppen et solid fundament for fremtidig utvikling, testing og iterasjoner basert på tilbakemeldinger fra brukere. Det er derfor særdeles viktig at applikasjonen bygges på et fundament som har en sikker fremtid, og ikke står i fare for å bli avvirket.

3.3 Valg av verktøy

Visual Studio Code

Visual Studio Code er en teksteditor utviklet av Microsoft, som er svært tilpasningsdyktig ved å støtte en rekke programmeringsspråk og rammeverk. Den tilbyr et enkelt og intuitivt oppsett, integrerer med terminalen, og er et verktøy gruppen er godt kjent med.

PostgreSQL

PostgreSQL er en "open-source" relasjonsdatabase som støtter avanserte datatyper og transaksjoner, og er kjent for sin pålitelighet, fleksibilitet og støtte for SQL-standarder. Valget av

en relasjonsdatabase skyldes prosjektets behov for å etablere og administrere avhengigheter mellom objekter, noe som ville vært utfordrende med graf- eller dokumentdatabaser.

PgAdmin

PgAdmin er et verktøy som brukes for opprettelse, redigering og administrasjon av PostgreSQL-databasen, samt enkel manipulering av data. Bruken av pgAdmin grunnes av tidligere erfaring med verktøyet.

ASP.NET Core

ASP.NET Core er et rammeverk for å bygge moderne webapplikasjoner. Det har et aktivt utviklermiljø og et stort samfunn, som gjør det mulig å bygge og skalere applikasjoner på en effektiv måte. Det tilbyr enkel oppkobling med databasen, og integrering av sikkerhet. ASP.NET Core tilbyr integrasjon med Swagger.

Swagger

Swagger tilbyr testing av API-endepunktene til applikasjonen, som forenkler testing og feilsøking under utviklingen. Dette gjør det ved å tilby et intuitivt brukergrensesnitt som representerer API-endepunkter, samt hvilke data som er forventet i forbindelse med disse.

React Native

React Native er et rammeverk for å bygge mobile applikasjoner ved hjelp av JavaScript og React. Ved å bruke React Native kan samme kodebase brukes på både iOS- og Android, som gir betydelige tidsbesparelser. React Native sine tredjepartsbiblioteker utgjør en omfattende samling av verktøy, komponenter og moduler som gir en god mulighet til å skreddersy applikasjonen.

Expo

Expo er et åpent rammeverk og en plattform for universell React-applikasjonsutvikling, som forenkler prosessen med å bygge, teste og utgi applikasjoner for iOS og Android. Gruppen valgte Expo fordi det muliggjør utvikling på tvers av ulike operativsystemer, noe som er ideelt for gruppens varierte utviklingsmiljøer.

Git

Git er et versjonskontrollverktøy, som gir mulighet til å lagre, håndtere og samarbeide om koden på en effektiv måte. Det blir opprettet en "branch" for hver oppgave, noe som lar flere utviklere arbeide parallelt uten å oppnå konflikter med hverandre. Git gir også muligheten til å dele kode med veileder underveis i prosjektutviklingen.

Azure

Azure er en skyplattform og tjeneste som tilbyr en rekke verktøy og tjenester for å utvikle,

distribuere og administrere applikasjoner og tjenester i skyen. Med tanke på hosting, ble Azure-portalen valgt fordi den tilbyr en intuitiv plattform for å implementere CI/CD ved hjelp av Github Actions til Azure App Service.

Figma

Figma er en designplattform som blir brukt for å samarbeide i sanntid på modeller og diagrammer, samt opprette prototyper og annet design for applikasjonen.

Discord

Discord er en gratis kommunikasjonsplattform, som tilbyr kommunikasjon via videosamtaler og meldinger. Det er opprettet en egen server med ulike kanaler, for å enkelt samarbeide og formidle viktig informasjon under utviklingen av prosjektet.

Trello

Trello er en nettbasert prosjektstyringsplattform, hvor oppgaver organiseres i visuelle tavler for å effektivisere samarbeidet og arbeidsflyten. Dette forenkler arbeidsfordelingen, og gir et helhetlig overblikk over både planlagte og utførte arbeidsoppgaver.

Testflight

TestFlight er en plattform utviklet av Apple som tillater utviklere å distribuere beta-versjoner av iOS-apper til et utvalgt testpublikum før de lanseres offisielt på App Store. Dette gjør det mulig for gruppen å gjennomføre brukertester og samle tilbakemeldinger fra testere uten en offisiell lansering av applikasjonen.

3.4 Prosjektmetodikk

I denne seksjonen presenteres den valgte prosjektmetodikken, i tillegg til prosjektplanlegging og risikovurdering. Det blir videre lagt frem hvordan disse elementene samhandler for å forme en strategi og tilnærming til prosjektgjennomføringen.

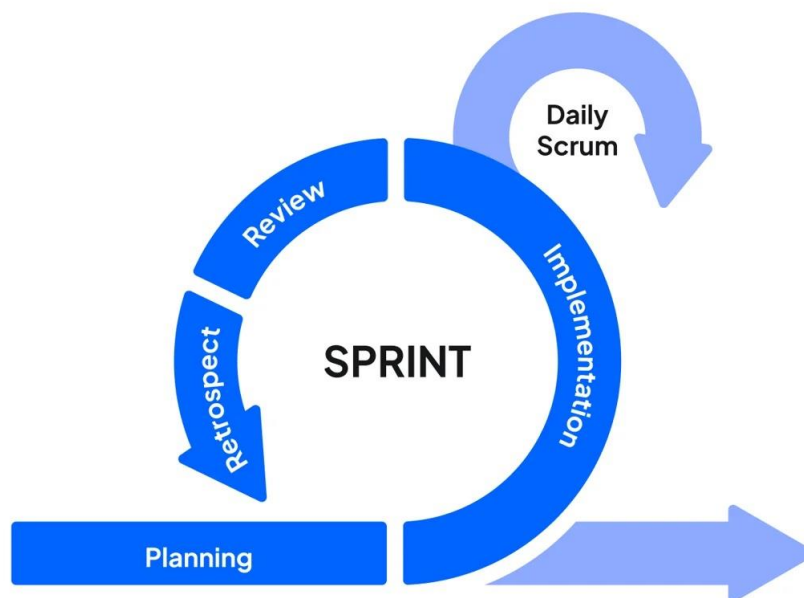
3.4.1 Utviklingsmetodikk

I startfasen av prosjektperioden ble det bestemt at det skulle jobbes etter de smidige (agile) utviklingsmetodene Scrum og Kanban. En sentral egenskap ved smidige utviklingsmetoder er erkjennelsen av at programvareutvikling er en iterativ og kontinuerlig læringsprosess, der løsningen på problemer oppdages og forbedres gjennom hele utviklingsforløpet. Smidig utvikling er i dag en bransjestandard for utvikling av alle typer digitale produkter. (Dingsøy, 2024). I en

undersøkelse utført av Digital.ai, 16. Januar 2024, kommer det fram at rundt 71% bruker smidige utviklingsmetoder i deres SDLC (Software Development Lifecycle) (Martin, 2024).

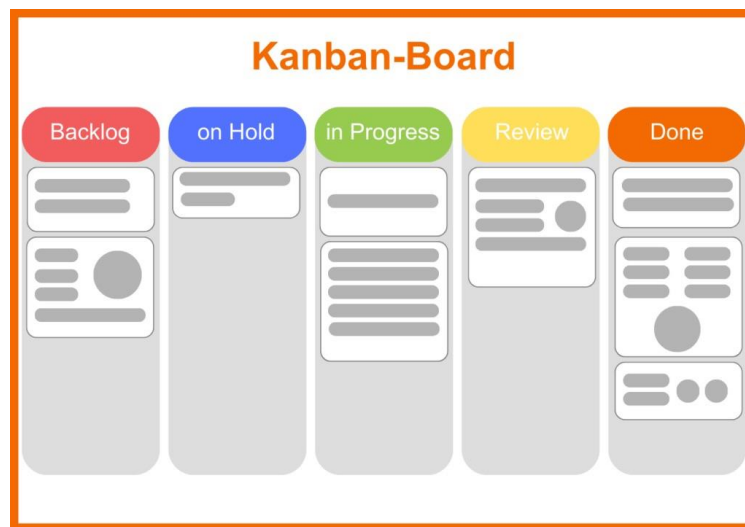
Scrum er den mest brukte smidige utviklingsmetoden (Dingsøy, 2024). Det er et rammeverk som brukes til å organisere, administrere og levere komplekse produkter og prosjekter. Det gjør det mulig for team å tilpasse seg endringer, levere verdifull funksjonalitet raskt og effektivt, og tilpasse kundens behov og tilbakemeldinger underveis. Scrum bruker klare roller og ansvarsområder for å sikre effektiviteten og ansvarligheten. Produktutviklingen organiseres etter korte, tidsbestemte iterasjoner, definert som sprints.

Hendelsesløpet til en sprint i Scrum, se figur 3.1, har vanligvis en varighet på 1 til 4 uker. Det begynner med sprintplanlegging, der oppgaver og mål settes for perioden. Deretter arbeides det daglig med oppgavene, holdes korte stand-up møter, og leverer inkrementer av produktet. Ved slutten av sprinten gjennomføres en gjennomgang for å vise frem resultater, etterfulgt av en retrospektiv for å evaluere sprinten og identifisere forbedringsmuligheter.



Figur 3.1 - Illustrasjon av hendelsesløpet i en Sprint i Scrum (Ones, 2023)

Kanban er et velkjent rammeverk innenfor smidig utvikling, som ble utviklet av Toyota på slutten av 1940-tallet for å optimalisere sine ingeniørprosesser. Det er et visuelt rammeverk for styring av arbeid som brukes til å organisere arbeidsflyten, begrense arbeidsmengden og kontinuerlig forbedre prosessene. I programvareutvikling fremmer Kanban-flyten dynamisk oppgavestyring, fremskynder leveringssykluser og forbedrer kundetilfredshet gjennom fokusert, uavbrutt arbeid. For å visualisere arbeidet blir det brukt Kanban-tavle, se figur 3.2, hvor arbeidsoppgavene organiseres i kolonner som representerer ulike stadier i arbeidsflyten. De ulike stadiene er “To do”, “On hold”, “In Progress”, “Review” og “Done” (Radigan, 2024).



Figur 3.2 - Eksempel på Kanban-tavle (121watt, 2023)

3.4.2 Valg av utviklingsmetodikk

I oppstartsfasen ble valget om utviklingsmetode tatt, hovedsakelig basert på tidligere erfaringer med smidige utviklingsmetoder fra andre prosjekter. Det var viktig å planlegge utviklingsarbeidet tidlig og grundig, for å oppnå en visjon og et ønsket resultat med prosjektutviklingen. Smidig utvikling forenkler samarbeid og tillater endringer basert på ny kunnskap som opparbeides underveis.

Med Scrum er det vanligvis klart definerte roller, men grunnet en prosjektgruppe på kun tre personer hadde alle like mye ansvar for gjennomføringen av prosessene i prosjektet. Det ble avtalt å arbeide i ukentlige sprints, med daglige stand-up møter for å sørge for at alle til enhver tid hadde oversikt over arbeidet. Gruppen utførte retrospektiv på slutten av hver sprint for å evaluere ukesarbeidet, og legge en plan for neste sprint.

I tillegg til dette ble også prinsipper fra Kanban-rammeverket implementert for at gruppen enkelt skulle kunne visualisere arbeidsoppgavene. Ved bruk av Kanban-tavle i Trello, kunne gruppen organisere og arbeide med de ulike arbeidsoppgavene som tilhørte hver sprint. En slik tavle gjorde det også enkelt å dele store arbeidsoppgaver opp i mindre, konkrete oppgaver som skulle utføres innenfor sprintintervallet. Dette skapte en effektiv arbeidsflyt, enkel arbeidsfordeling, gode resultater, og kontinuerlig oversikt i utviklingsprosessene.

Basert på avsnittene ovenfor er det rimelig å påstå at utviklingsmetodikken var sterkt preget av prinsipper fra Scrum og Kanban uten at noen av rammeverkene ble direkte implementert som arbeidsmetodikk. Kombinasjonen av disse er vanlig, og har fått navnet ScrumBan (Atlassian, u.å).

3.4.3 Prosjektplan

For å utforme en fremdriftsplan for prosjektutviklingen har gruppen utformet et Gantt-diagram, se vedlegg B. Dette er et horisontalt stolpediagram som brukes for prosjektplanlegging. Diagrammet gir en visuell tidslinje med start- og sluttdatoer for ulike oppgaver innenfor prosjektet. Som det kommer frem i diagrammet vil systemets backend bli implementert først, etterfulgt av design og frontend. En iterativ fremgangsmåte ved å utvikle komponenter både i front- og backend samtidig kunne bidratt til enklere og bedre testing. Gruppen valgte likevel å utvikle hele front-, og backend separat, da dette antas å gå raskere enn den iterative fremgangsmåten.

I det anvendte Gantt-diagrammet til gruppen, representerer de grå stolpene overordnede hovedkategorier, som definerer arbeidsområdene for hver periode. Innenfor disse hovedkategoriene symboliserer hver uke en sprint, med en kort beskrivelse av oppgavene som skal gjennomføres. Lengden på hver stolpe indikerer varigheten av oppgavene. Det er viktig å merke seg at hovedkategorien "Bachelor Writing" er planlagt for å foregå parallelt med den øvrige utviklingen.

3.4.4 Risikovurdering

“En risikovurdering er et verktøy for å identifisere uønskede hendelser og risikoen for at disse skal inntreffe. Risiko betegner forholdet mellom sannsynlighet for at en uønsket hendelse vil inntreffe og konsekvenser av en slik hendelse.” (Datatilsynet, 2019). I oppstartsfasen ble det utviklet en risikoanalyse for å få en oversikt av hendelser som ble antatt relevante i prosjektsammenhengen.

Dette innebar årsaken til at de kunne forekomme, hva konsekvensene ville være og tiltakene for å håndtere dem. For å indikere alvorlighetsgraden av en risiko benyttes tall og fargekoder, se figur 3.3. Sannsynligheten og konsekvensen av en risiko rangerer mellom 1 og 5, hvor risikofaktoren defineres av å multiplisere sannsynligheten og konsekvensen. Grønn viser til lav risiko, gul indikerer middels risiko og rød viser til høy/ekstrem risiko.

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						

Figur 3.3 - Risikoanalyse med tall og fargekoder for å kategorisere ulike risikoer.

Det har blitt identifisert ulike risikoer i form av utfordringer og trusler som kan påvirke prosjektets suksess eller sluttresultat, disse er å finne i vedlegg C.

3.5 Evalueringsplan

I evalueringsfasen av prosjekter som omhandler systemutvikling, er det viktig å både verifisere og validere løsningen. Verifikasjon bekrefter at løsningen har en korrekt oppbygging, mens validering handler om oppnåelse av målsettingen for prosjektet (Helgesen, 2024).

For å verifisere løsningen vil det bli gjennomført integrasjonstester og systemtester under og etter utvikling av de ulike delene av systemet. Integrasjonstestene sørger for at komponentene fungerer i sammenheng med hverandre, og gjennomføres ved å teste API-endepunktene som håndterer dataene. Testene må gjennomføres kontinuerlig ved implementering av ny kode. Ettersom at disse endepunktene har lik funksjonalitet under utvikling og i et produksjonsmiljø, vil dette også sørge for at funksjonaliteten er intakt i den fullstendige løsningen. Systemtester vil bli gjennomført manuelt av prosjektgruppen, samt gjennom brukertester som også er en del av valideringsfasen.

Systemet skal valideres gjennom observasjon og intervju med brukere. Under utviklingen vil det bli gjennomført en breddeundersøkelse for å kartlegge meninger, vaner og behov hos potensielle brukere. Tilbakemeldingen blir brukt til å justere og videreutvikle ideen og problemet. På et senere tidspunkt valideres systemet gjennom brukertesting i slutfasen av utviklingen, som sørger for at systemet er tilnærmet komplett i testsammenhengen. Her blir systemet validert opp mot de konkrete målene for funksjonaliteten fra kapittel 1, samtidig som at brukerens respons på et utvalg av spørsmål gir en god indikasjon på hvorvidt løsningen er et godt svar på problemstillingen.

4 DETALJERT LØSNING

Dette kapittelet tar for seg løsningen som har blitt implementert, og gir en teknisk forklaring på rollen til de ulike elementene i systemet. Det blir gjort rede for hvordan disse fungerer på et individuelt nivå, samt hvordan de samarbeider og påvirker hverandre. Gjennom kapittelet blir det også begrunnet hvorfor ulike valg ble gjort underveis, for å oppnå et best mulig resultat.

4.1 Overordnet arkitektur

4.1.1 Systemarkitektur

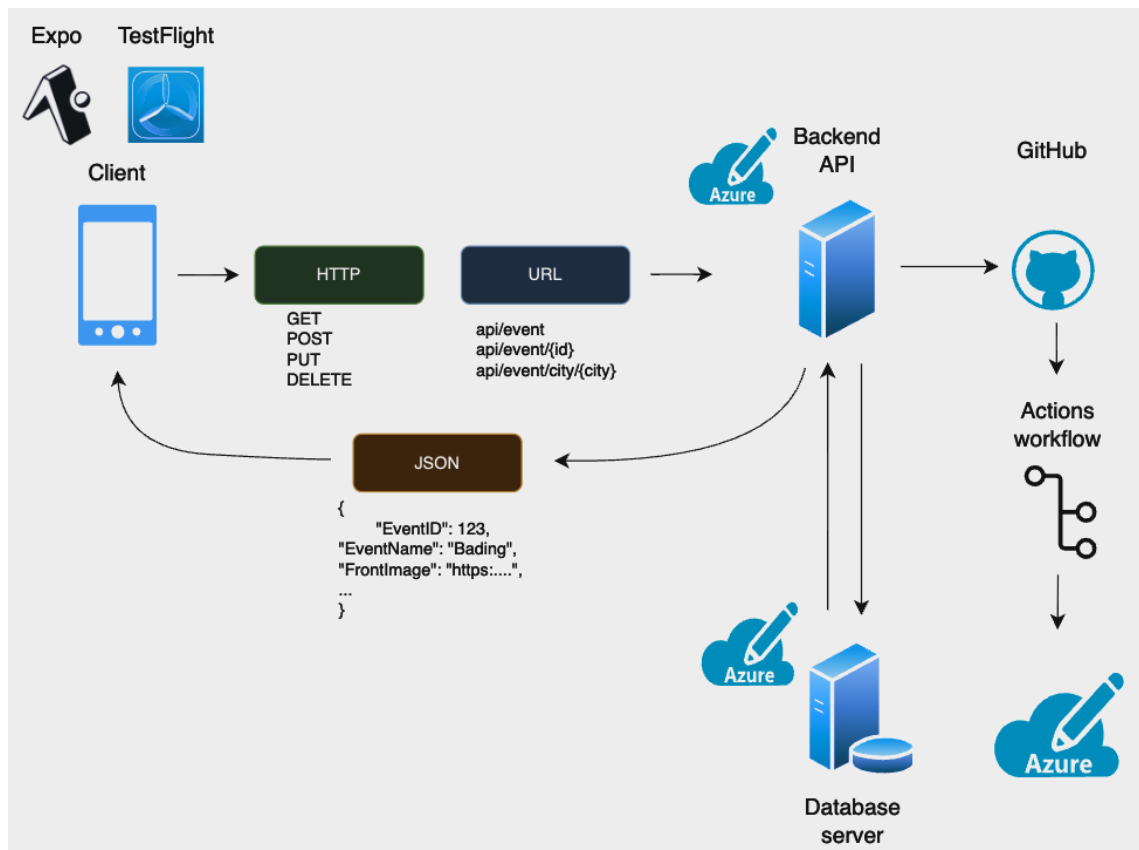
Applikasjonens systemarkitektur, skissert i figur 4.1, har to primære komponenter; en serverdel og et klientgrensesnitt. Disse blir videre referert til som henholdsvis backend og frontend. I utviklingsfasen blir disse behandlet som én kodebase, til forskjell fra produksjonsmiljøet hvor det hostes og lagres som separate kodebaser. Backend-koden som bygger på Microsofts rammeverk ASP.NET Core, definerer i stor grad utformingen på systemet, og er ansvarlig for å gjøre systemet sikkert og konsistent. I tillegg kommuniserer denne delen toveis med databasen, slik at data kan lagres og hentes trygt. Frontend definerer applikasjonsdelen som kjører på brukernes enheter, også kjent som klienten. Denne delen av koden er ansvarlig for å tolke og håndtere handlingene brukeren utfører på brukergrensesnittet, samt sende og hente data til og fra backend. Denne kommunikasjonen blir nærmere forklart i delkapittel [4.2.3](#).

Produksjonsmiljø for testing

Ved en testlansering av applikasjonen vil systemets omgivelser endre seg, da koden ikke lenger kjøres lokalt på en PC. Backend hostes med Azure App Service og benytter en automatisert utrullingsprosess konfigurert gjennom GitHub Actions. Denne prosessen aktiveres ved hver oppdatering som pushes til hovedgrenen, og fører til at endringer automatisk blir bygget og lansert til Azure. Denne dynamikken er med på å muliggjøre raske og direkte oppdateringer.

Videre kommuniserer backend direkte med HVL sin PostgreSQL-database, som gjøres tilgjengelig via en separat server på Azure. Databasen ivaretar systemets funksjonalitet og integritet ved å lagre og håndtere alle nødvendige data på en sikker og effektiv måte.

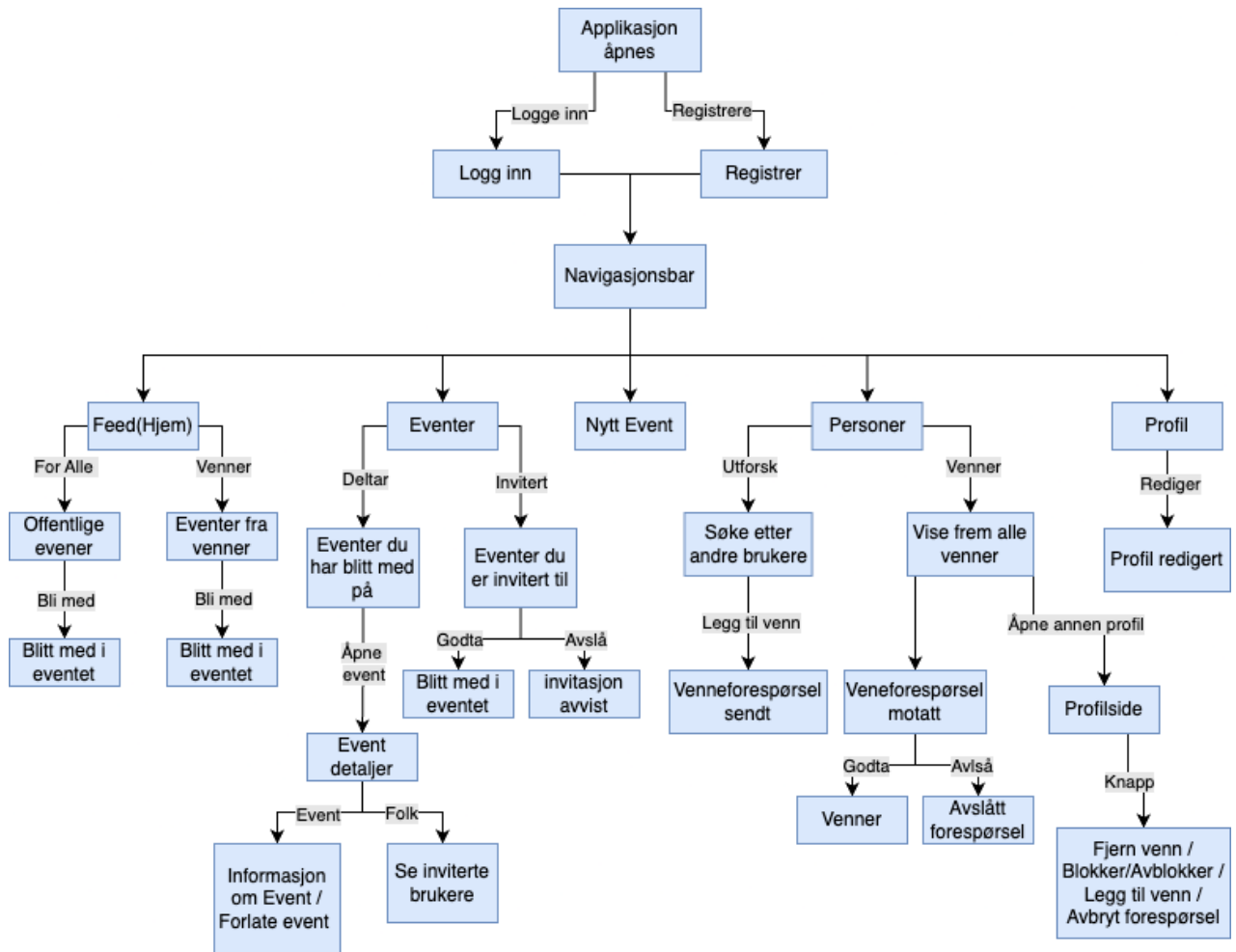
Ved fullføring vil klientgrensesnittet undergå en byggeprosess før det lastes opp til App Store Connect og deretter distribueres via TestFlight. Denne tilnærmingen muliggjør omfattende testing av applikasjonen for prosjektgruppen, og bidrar til å sikre best mulig funksjonalitet og brukervennlighet før eventuell lansering.



Figur 4.1 - Applikasjonens systemarkitektur

4.1.2 Applikasjonsflyt

I figur 4.2 vises de ulike applikasjonsflytene en bruker kan følge ved bruk av applikasjonen. Det er naturligvis mulig å gjennomføre mange slike "flyter" etter hverandre, men modellen gir et bilde på dybden i de ulike delene av systemet. Teksten under figuren beskriver overordnet hvilke funksjonaliteter en bruker har tilgang til inne i applikasjonen.



Figur 4.2 - Oversikt over applikasjonens flyt

Etter vellykket innlogging eller registrering vil bruker få tilgang til selve applikasjonen. Nederst på skjermen ligger en navigasjonsbar som holder lenker til de viktigste sidene i applikasjonen: “Feed”, “Eventer”, “Nytt event”, “Personer” og “Profil”. Den første skjermen som møter brukeren er “Feed”, hvor det er mulig å utforske og melde seg på både offentlige og venners arrangementer. Ved påmelding blir det valgte eventet borte fra denne feeden, og vil dukke opp under “Eventer”.

"Eventer" viser arrangementer brukeren er påmeldt i, og brukes også til å holde oversikt over og administrere event-invitasjoner. På denne skjermen kan brukere også åpne utvidet informasjon om eventer de deltar i. "Nytt event"-siden tilbyr mulighet for å opprette nye arrangementer og invitere venner til disse om ønskelig. "Personer"-seksjonen er utformet for å søke etter profiler, legge til venner, se venners forespørsler, og navigere gjennom egen venneliste. Brukere kan også navigere seg til "Profil"-siden som en personlig visning der brukere kan redigere egen informasjon.

4.2 Backend-API og database

Backend-API-et og den tilknyttede databasen er en av to hovedkomponenter i fullstack-applikasjonen. I dette kapittelet forklares teknologiene som brukes for å utvikle denne delen av prosjektet, og hvordan disse samhandler med hverandre. Det gir en oversikt over backend-strukturen, oppbyggingen av API-er og sikkerhetsimplementering.

4.2.1 Rammeverk og biblioteker

Prosjektet sin backend-løsning er utviklet i ASP.NET Core rammeverket, som beskrevet i delkapittel [3.3](#). Bruken av ASP.NET Core gir tilgang til .NET-økosystemet, som tilbyr nyttige biblioteker for å gi forbedret funksjonalitet og effektivisere utviklingen av prosjektet.

Prosjektet tar i bruk OOP (Object Oriented Programming) prinsipper, som går ut på at modeller blir definert i samsvar med databasens tabeller. For kommunikasjon mellom backend og database benyttes ORM (Object-Relational Mapping) for å utføre CRUD-operasjonene på dataene i relasjonsdatabasen. ORM er et verktøy som forenkler samspillet mellom relasjonsdatabaser og OOP-språk (Abba, 2022). EF Core (Entity Framework Core) er et moderne, kryss-plattform ORM-bibliotek utviklet av Microsoft. Det gir gruppen mulighet til å jobbe med databaser ved hjelp av .NET-objekter, som fjerner behovet for å manuelt skrive SQL-spørringer. EF Core er fleksibelt, og lar utviklere velge den mest passende databaseløsningen og -leverandøren for prosjektet, for eksempel MySQL, PostgreSQL osv. (Ajevnikers, et al., 2021).

EF Core var et sentralt hjelpemiddel under utviklingen av prosjektet, da det bidro til å effektivisere koblingen mellom database og backend. For å opprette databasen installeres EF Core-pakken, og det implementeres modellklasser som representerer databasetabellene. Det opprettes også en DbContext klasse som representerer databasen sin kontekst. Klassen definerer hvilke modeller som skal oversettes til de ulike databasetabellene, samt hvilke relasjoner og nøkler som skal opprettes. Basert på dette, genererer EF-core automatisk tabellene i databasen. Ved endringer i koden som påvirker oppsettet av databasen, er det også mulig å oppdatere databasen. EF-core tar vare på informasjonen om de ulike versjonene vha. såkalte migrasjoner som lagres i prosjektet. Dette eliminerer behovet for manuelt å håndtere database-skript og forenkler vedlikeholdet av databasen i løpet av applikasjonens levetid.

EF Core støtter LINQ (Language Integrated Query), som gjør det mulig å skrive spørringer direkte i C# for å utføre databaseoperasjoner. Det gjorde koden mer lesbar, pålitelig, og enklere å vedlikeholde. I tillegg kunne gruppen benytte seg av asynkrone spørringer som gir forbedret ytelse og responsivitet i applikasjonen.

4.2.2 Arkitektur

Arkitekturen på backend-delen av applikasjonen bærer preg av Model-View-Controller design mønsteret. View blir ikke generert i form av HTML-sider på serveren, men i stedet brukes React Native til å bygge grensesnittet på klientsiden. Med MVC-arkitekturen blir koden organisert i separate lag for å oppnå god lesbarhet, vedlikeholdbarhet samt overholdelse av SOLID-prinsippet om enkelt ansvar. Prinsippet om enkelt ansvar innebærer at en klasse skal ha ansvar for kun én del av systemet (Wikipedia, 2023). De ulike lagene i backend forklares nedenfor:

Models representerer de definerte datatypene som brukes i applikasjonen. Disse klassene reflekterer innholdet i databasetabellene, hvor de definerer strukturen og egenskapene til dataene som lagres i databasen.

Repositories er ansvarlige for å håndtere dataaksesslogikken i applikasjonen. De gir et abstraksjonslag over databasetilkoblingen og tilbyr metoder for å utføre vanlige databaseoperasjoner og CRUD-operasjoner. Dette innebærer henting, oppretting, oppdatering og sletting. Repositories bruker modellklassene for å kommunisere med databasen gjennom EF Core, noe som isolerer resten av applikasjonen fra detaljene i databasen.

Services er et mellomledd mellom Repositories og Controllers. De inneholder forretningslogikk og funksjonalitet som ikke direkte er relatert til dataaksess. De er ansvarlige for å håndtere komplekse oppgaver og operasjoner som skal brukes til å svare på klientforespørsler. Servicene koordinerer flere Repositories for å utføre oppgaver på tvers av de ulike datamodellene.

Controllers er ansvarlige for å motta forespørsler fra klienten, tolke dem, validere input og deretter samhandle med de riktige servicene for å behandle forespørslene. Til slutt må de også gi en respons til klienten, som indikerer om operasjonene var vellykket eller ikke. De fungerer som et ledd mellom klienten og resten av applikasjonen. Controllerne følger tradisjonelle REST API-prinsipper, som beskrives nærmere i delkapittel [4.2.3](#).

Felles for alle metodene som utvikles i de ulike lagene i backend, er bruken av Async/await og ExceptionHandling. TAP (Task-based Asynchronous Pattern) brukes for å utføre asynkrone operasjoner i applikasjonen. Det tillater at metoder kjører uten å blokkere hovedtråden (BillWagner, pkulikov, 2023). Dette er nyttig for gruppens applikasjon ettersom det er mange I/O-operasjoner, som for eksempel å hente data fra databasen og utføre nettverksskall. ExceptionHandlers benyttes for å håndtere forskjellige typer feil som kan oppstå under utførelse av metoder. Ved at feil oppfattes og håndteres på en riktig måte, kan brukere få informative feilmeldinger, som bidrar til en god brukeropplevelse (BillWagner, et al., 2023).

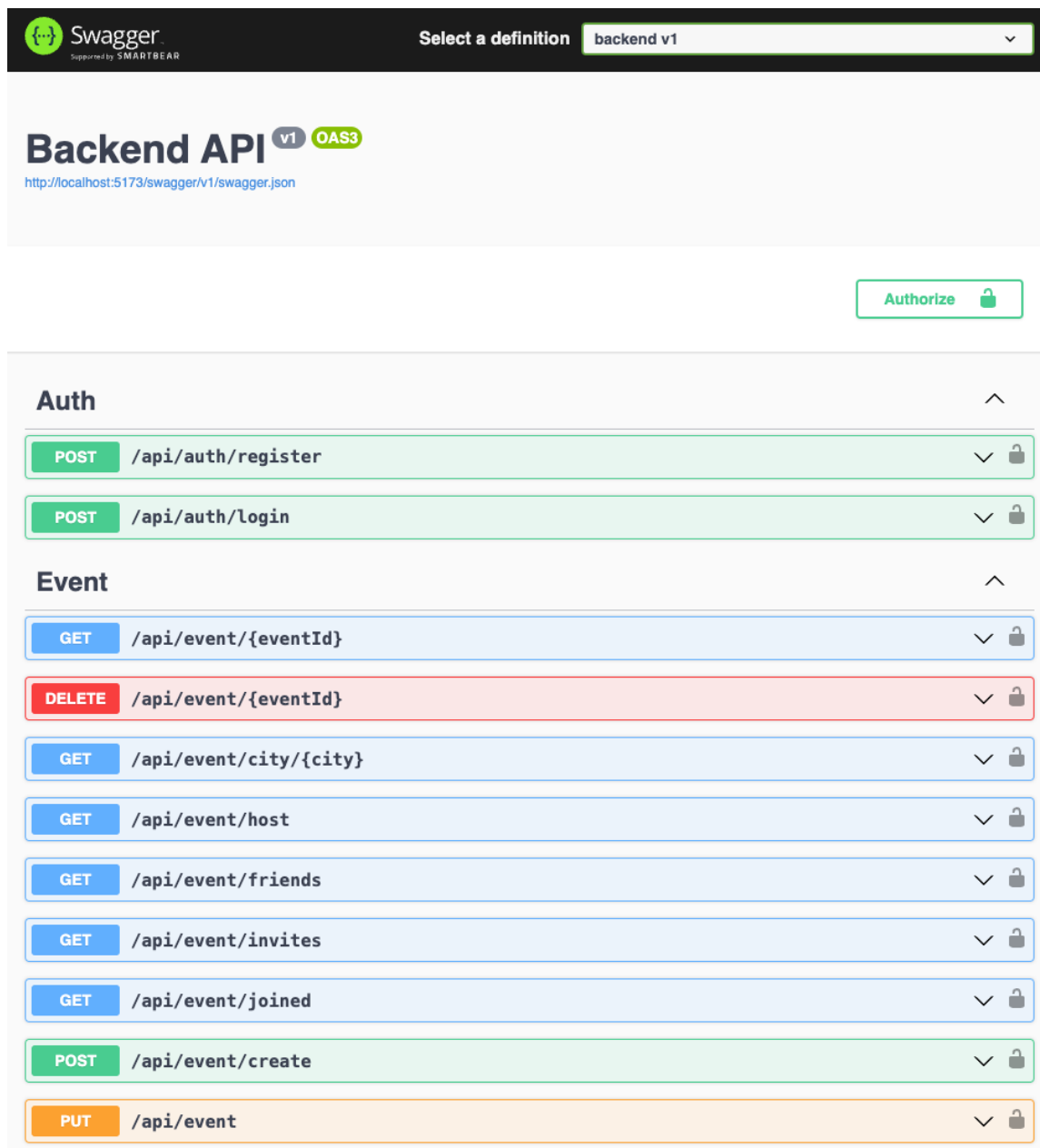
Prosjektet består også av Enums, DTOs (Data Transfer Objects) og Interfaces. Dette bidrar til en mer strukturert, modulær og vedlikeholdbar kodebase. Enums inneholder de ulike konstante verdiene brukt til utviklingen av prosjektet. Et eksempel er de ulike fastbestemte rollene “Creator”, “Host” eller “Participant” en bruker kan ha i et event. DTO-ene representerer en forenklet versjon av dataene som sendes over nettverket gjennom API-et. Interfaces brukes til å definere en kontrakt eller et sett med metoder som en klasse må implementere (GitHubPang, et al., 2023). Alle Service-klassene har hvert sitt Interface, hvor hver Interface-klasse inneholder alle metodene med dokumentasjon om hva de gjør. Dette gir en god forståelse og oversikt over hva som er hensikten og funksjonaliteten til de ulike metodene i applikasjonen.

4.2.3 API og swagger

Et API fungerer som en bro, og muliggjør interaksjon mellom forskjellige programvarekomponenter gjennom forhåndsdefinerte metoder og protokoller. Begrepet API står for Application Programming Interface og innebærer et sett med regler og spesifikasjoner som programvarekomponentene følger for å kommunisere med hverandre (AWS, u.å.).

I konteksten til prosjektet er det brukt et REST (Representational State Transfer) API. Dette benytter seg av HTTP/HTTPS-protokoller for dataoverføring, med metodene GET, POST, PUT og DELETE for å tilrettelegge for manipulasjon av data. Klienten kommuniserer med backend API-et gjennom URL-adresser definert i backend-koden. Dette muliggjør datautveksling mellom klient- og server-siden.

Swagger er et omfattende verktøy for API-utvikling og spiller en nøkkelrolle i utvikling, testing, og dokumentasjon av REST API-endepunkter i prosjektet. Ved å automatisere dokumentasjons-genereringen fra prosjektets kildekode, tilbyr Swagger en interaktiv dokumentasjon som er tilgjengelig for både backend- og frontend-utviklere. Dette muliggjør enklere utvikling for begge parter. (Swagger, u.å.) Verktøyet letter testingen av API-endepunkter og bidrar til en effektiv integrasjon mellom ulike systemkomponenter uten å avhenge av et separat miljø som f.eks. Postman.



Figur 4.3 - Swagger dashboard som viser noen av applikasjonens endepunkter

4.2.4 Sikkerhet og tokens

Passordlagring er en risiko som gruppen har vært nødt til å adressere og ta hensyn til. For å oppnå en sikker lagring av passord, har det blitt benyttet passord hashing. Dette er en prosess som anvender en kryptografisk hashfunksjon, noe som transformerer det opprinnelige passordet til en enveiskryptert hashverdi. Gruppen har også benyttet seg av salting. Dette innebærer å legge til en tilfeldig generert tekststreng til passordet før hash-prosessen utføres (Arias, 2019).

Prosjektet er et delt system som opererer over internett gjennom APIer, noe som introduserer diverse sikkerhetsmessige utfordringer som må håndteres for å beskytte data i applikasjonen. For å sikre sikker dataoverføring, kommuniserer systemet over HTTPS (Hypertext Transfer Protocol Secure), som er en sikker forlengelse av HTTP. Her blir data kryptert i overføringen. Dette sikrer at all data som overføres mellom klienten og serveren er kryptert og dermed beskyttet mot mulig avlytting eller manipulasjon (AWS, u.å.).

For å styre tilgangen til applikasjonens ressurser benyttes JWT (JSON web token). Et slikt token fungerer som en nøkkel som inkluderes i HTTPS-forespørselen, og lar backend-API-et validere om brukeren har tilgangsrettigheter til de forespurte ressursene. JWT inneholder såkalte claims, som ofte inkluderer brukerens ID. For å forhindre uautorisert manipulasjon, signeres tokenet med en digital signatur. Dette lar mottakeren bekrefte at tokenet ikke er endret etter utstedelsen. Tokenet sin signatur beskytter mot modifikasjoner, og HTTPS sørger for at dataoverføringen er sikker (JWT, u.å.).

Prosessen som gir bruker et gyldig token blir initiert i backend etter en vellykket forespørsel for innlogging eller registrering. Etter at tokenet har blitt generert, sendes det tilbake til klienten i responsen til den initielle forespørselen slik at klienten kan bruke denne “nøkkelen” for å få tilgang til ressurser

Implementeringen av JWT krevde en konfigurasjonsprosess av selve teknologien. Dette innebar definering av metode for tokendekoding, hemmelige verdier for sender og mottaker, samt definisjon av selve nøkkelen. For generering av token ble det implementert en tokenservice. Dekoding og validering av token håndteres gjennom en etablert konfigurasjon i Program.cs-filen,

som en fil for grunnleggende oppsett av prosjektet. Dette inkluderer verifisering av tokens autentisitet og integritet i henhold til de definerte sikkerhetsparameterne som vises i figur 4.4.

```
builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>
    {
        options.TokenValidationParameters = new TokenValidationParameters
        {
            ValidateIssuer = true,
            ValidateAudience = true,
            ValidateLifetime = true,
            ValidateIssuerSigningKey = true,
            ValidIssuer = builder.Configuration["Jwt:Issuer"],
            ValidAudience = builder.Configuration["Jwt:Audience"],
            IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(builder.Configuration["Jwt:Key"])),
            ClockSkew = TimeSpan.Zero
        };
    });
```

Figur 4.4 - konfigurasjon av JWT i program.cs

Fra backends perspektiv vil ikke brukere ha tilgang til data fra API-ene uten et tilhørende token, som også må være gyldig. Tokenet utstedes til brukeren ved innlogging eller registrering og inneholder påstander (claims) som speiler brukerens ID. Denne ID-en i tokenet brukes til å begrense tilgang til data, basert på brukerens tilgangsrettigheter.

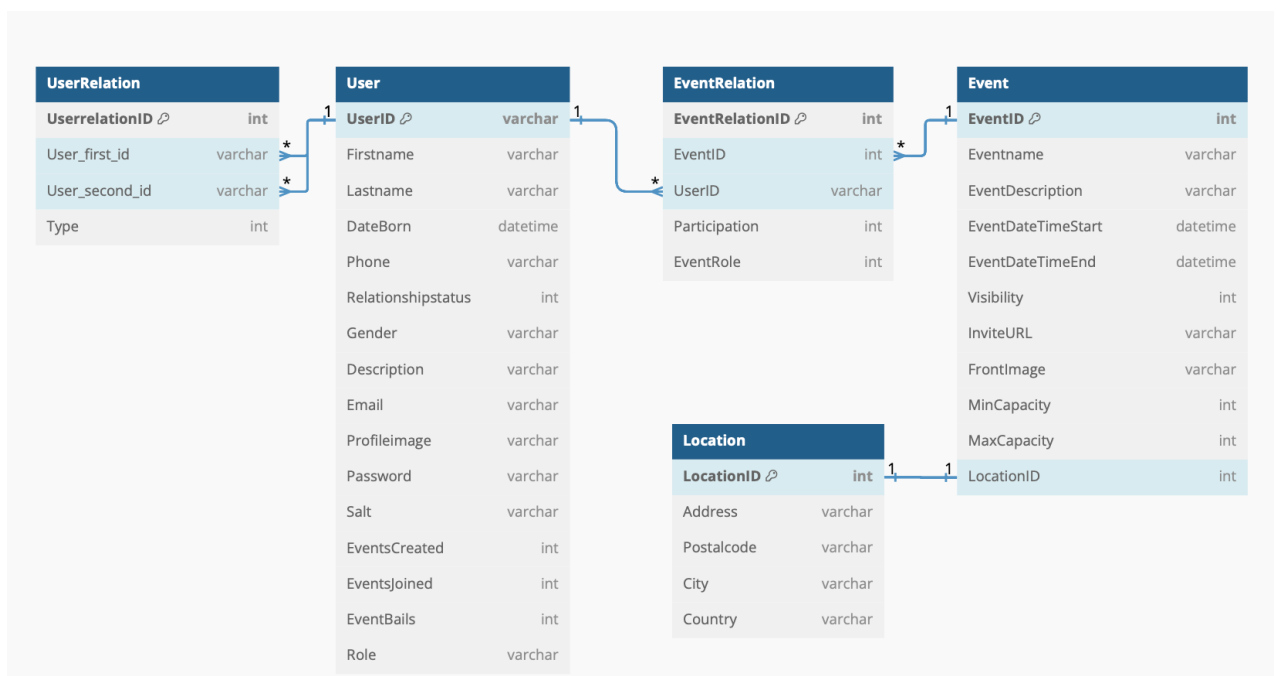
4.2.5 Database

Databasen er lagringsplassen hvor all data som blir brukt i applikasjonen hentes fra, oppdateres og lagres. Gruppen brukte PostgreSQL som relasjonsdatabase, med pgAdmin som verktøy for å administrere databasen. Verktøyene er beskrevet nærmere i delkapittel [3.3](#). Relasjonsdatabasen organiserer dataene i tabeller, der hver rad er et objekt, og dataene lagres som attributter (Bratbergsengen, 2017). Dette tillater å strukturere dataene på hierarkisk måte og opprette relasjoner mellom ulike tabeller, som gjør det mulig å hente data på en effektiv måte gjennom SQL-spøringer.

Det er utviklet et ER-diagram (Entity-Relationship diagram), se figur 4.5, for å skape en visuell representasjon av tabellene i databasen, og deres relasjoner med hverandre. Hvert attributt har en spesifisert datatype, for å forhindre feil og sikre dataintegritet. De markerte cellene i tabellene definerer primær- og fremmednøkler. Primærnøkkel er en unik identifikator for hver rad i en

tabell, og sikrer at hver datainnføring identifiseres entydig. Fremmednøkkel er en kolonne som refererer til primærnøkkelen i en annen tabell (Bratbergsengen, Tøssebro, 2023).

Et eksempel på dette er blant annet i Event-tabellen hvor «EventID» er primærnøkkelen, og «LocationID» er fremmednøkkelen. Denne refererer til primærnøkkelen i «Location»-tabellen, og sørger for at den relevante lokasjonen kan relateres til eventet. Linjene mellom tabellene spesifiserer forholdene. Ved å ta i bruk samme eksempel, er det tydelig at det er en linje og derav en relasjon mellom Location-tabellen og Event-tabellen. Dette er et en-til-en forhold, som indikerer at et event har kun én lokasjon og en lokasjon tilhører kun ett event. Mellom «Event» og «EventRelation» er det et en-til-mange forhold, da et event kan ha mange eventrelasjoner.



Figur 4.5 - ER-diagram med oversikt over databasetabellene til prosjektet

Denormalisering av databaser handler i korte trekk om å legge til overflødig data i en eller flere tabeller i en database for å oppnå mer effektive spørringer (Visual Paradigm, 2023). Dette ble gjort enkelte steder for å muliggjøre raskere datahenting ved å unngå behovet for komplekse spørringer som går på tvers av flere tabeller i databasen. For eksempel i «User»-tabellen, har det blitt lagt til kolonner for opprettede-, deltatte-, og forlatte eventer. Denne denormaliseringen tillater direkte tilgang til denne informasjonen fra en brukeroppføring, i stedet for å måtte søke gjennom flere relasjoner mellom brukere og eventer.

4.3 Frontend og brukergrensesnitt

Frontend og brukergrensesnitt regnes som den andre hovedkomponenten i fullstack-applikasjonen. Som forklart tidligere i kapittelet, er det nødvendig å implementere funksjonalitet på klientsiden, i tillegg til logikken som ligger i backend-delen av applikasjonen. Denne logikken er viktig for å hente, sende, og redigere data før og etter kommunikasjon med backend API-et. Utforming og design på hvordan data skal vises frem til bruker er også et nøkkelaspekt ved frontend-delen til applikasjonen.

4.3.1 Rammeverk og biblioteker

I forbindelse med utviklingen av frontend ble det benyttet en rekke rammeverk og biblioteker. React Native og Expo er de mest fundamentale rammeverkene i applikasjonens frontend. Årsaken til at disse ble benyttet, og grunnleggende informasjon om hvordan de påvirker utviklingsprosessen, er også nevnt tidligere i delkapittel [3.3](#).

React Native er det grunnleggende rammeverket i applikasjonen, og gjør det mulig å enkelt skape en applikasjon for både Android og iOS fra samme kodebase. Komponentene og bibliotekene som medfølger ved bruk av React Native er viktige for å skreddersy utviklingsgruppens muligheter under utviklingen. Applikasjonen tar i stor grad i bruk innebygde UI-komponenter, som gir en generalisert struktur både på koden generelt, men også på de individuelle komponentene i koden. Dette er komponenter som er optimalisert for å gi applikasjonen en rask ytelse, samt sørger for at applikasjonen følger god praksis og kodestandarder innen apputvikling for både Android og iOS. Dette involverer blant annet komponenter for å hente inndata fra bruker, knapper, tekst og liknende. I tillegg til de innebygde bibliotekene som medfølger av å bruke React Native, benytter gruppen et bibliotek fra det tradisjonelle React rammeverket. Dette biblioteket heter React Navigation, og brukes for å kunne skape en sømløs navigering mellom de ulike sidene som ligger i navigasjonsbaren nederst på skjermen.

Expo fungerer som en augmentasjon av React Native rammeverket, og gir gruppen en rekke verktøy som effektiviserer utviklingsprosessen. Expo Go er et av verktøyene, og benyttes for å teste applikasjonen i sanntid under utviklingen. Ved hjelp av dette kan gruppen se endringer som gjøres på en mobilskjerm, da verktøyet kommer i form av en applikasjon som kan lastes ned på

mobiltelefoner. Denne kobles enkelt opp mot liveserveren som kjører på PC-en, og tilbyr sanntidsoppdateringer og andre tradisjonelle utviklerverktøy. For gruppemedlemmene som utviklet på Mac var det mulig å simulere en mobiltelefon vha. Xcode, og ha Expo Go på denne istedenfor en ekte telefon. Da applikasjonen skulle testes med API-kall til backend, var dette enkelt på Mac, da backend kjører lokalt på samme maskin som Xcode simulatoren. For gruppemedlemmene med Windows ble det benyttet Ngrok. Dette er en ingressplattform som gjør det mulig å synliggjøre den lokale backend-serveren på PC-en til internett, slik at Expo Go applikasjonen på mobiltelefonen kan bruke denne adressen til å gjøre API-kall.

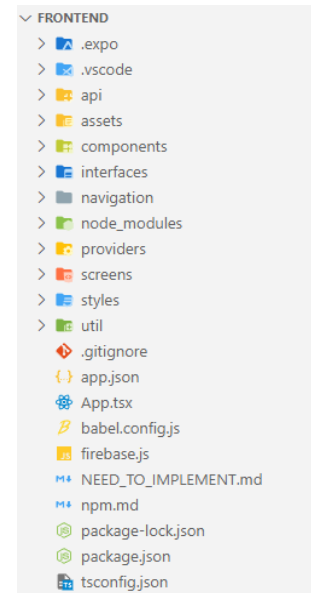
I tillegg til dette har også Expo en rekke innebygde biblioteker som kan benyttes for å forbedre og skreddersy applikasjonen. Expo Vector Icons og Expo Location er to av bibliotekene som blir benyttet. Expo Vector Icons tilbyr enkel integrasjon av simple ikoner til ulike visuelle detaljer i applikasjonen. Expo Location gjør det mulig å be om brukerens lokasjon på en standardisert måte, noe som ellers kan være en utfordring ettersom applikasjonen skal fungere på tvers av operativsystemer. Denne lokasjonen blir brukt til å gi brukere hendelser basert på hvor de befinner seg. I tillegg til disse, blir også biblioteket Expo Date-Time-Picker brukt. Dette gjør det enkelt å motta input fra bruker i forbindelse med dato og tidspunkt for hendelser som skal opprettes.

Expo tilbyr også verktøy og tjenester for å håndtere, bygge og utgi applikasjonen. EAS (Expo Application Services) er eksempel på dette, og gjør prosessen med å publisere applikasjonen til App Store enklere. I tillegg gir det også gruppen mulighet til å gjennomføre oppdateringer på applikasjonen “over the air”. Dette betyr at innehavere av applikasjonen ikke behøver å laste ned en ny oppdatering, men at det kommer direkte inn på den installerte applikasjonen på deres enhet.

4.3.2 Oppbygning og struktur

Ved bruk av rammeverkene fra forrige delkapittel, og generelt ifm. applikasjonsutvikling, er det viktig å ha en fast struktur og et ryddig oppsett på kodebasen. I moderne rammeverk for web-utvikling som f.eks. React, blir det opprettet klasser vha. JavaScript eller TypeScript som representerer en side på nettstedet. I native-utvikling er dette veldig likt, da React Native er laget for å være så fleksibelt som mulig.

Prosjektet har en mappe med “screens”, videre omtalt som “skjerm” som representerer alle sider i applikasjonen som dekker hele mobilskjermen. Alle andre synlige elementer på mobilskjermen som ikke er assosiert med en slik skjerm, er definert som en gjenbrukbar komponent. Slike gjenbrukbare komponenter er en sentral del av moderne mobilutvikling. Gruppen har valgt å ha generiske komponenter i en mappe på øverste nivå i frontend-koden, og komponenter som er spesifikke for en skjerm i skjermens egne mappe. Dette danner den grunnleggende strukturen for logikken og utseendet på de visuelle elementene i applikasjonen. Ellers følger prosjektet en standard struktur for React prosjekter, med en App-fil som inngangspunkt til applikasjonen.



Figur 4.6 - mappestruktur frontend

Det er også en rekke andre elementer i applikasjonen som sørger for at den oppnår systemkravene, samtidig som den er robust og følger gode praksiser. Det har blitt laget en mappe med Interfaces på toppnivå. Dette er gjort for å generalisere dataobjektene som skal interagere med backend API-et, slik at disse samsvarer med entitetene i databasen. Videre brukes disse i en annen mappe på høyeste nivå, som heter API. Dette er en av de viktigste delene av applikasjonen, og definerer metoder for å hente og sende data til og fra API-endepunktene i backend. Dette gjøres ved å benytte “Fetch”, som er en innebygget funksjon i JavaScript. Fetch API-et fungerer som et bindeledd til backend, og lar gruppen sende egendefinerte HTTP forespørsler direkte til backend API-et ved å bruke de ulike API-endepunktens URL (React, 2023).

En annen sentral del av kodebasen for frontend er håndtering av tokens, som forklares fra backend-perspektivet i delkapittel [4.2.4](#). Som det blir forklart der, er API-endepunktene beskyttet

mot uautorisert tilgang ved hjelp av en Bearer Authentication Token. Etter innlogging eller registrering i backend blir det sendt et token tilbake til klientsiden, som blir verifisert hver gang en bruker åpner applikasjonen. Etersom mange ulike klasser avhenger av dette tokenet for å kalle på backend API-et, er det viktig å håndtere tokens på et globalt nivå i frontend-delen av applikasjonen. Dette oppnås ved å lage en provider, som benytter seg av context API-et til React. Ved bruk av context-API i en komponent, vil alle barnekomponentene som omringes av dette komponenten ha tilgang til denne konteksten (React Native, 2023). Prosjektet har dermed en Token Provider, som benytter seg av en slik kontekst. Denne ligger på øverste nivå i komponent-treet i App-filen, slik at tokenet er tilgjengelig globalt i frontend-prosjektet, altså for alle klasser. Det samme er gjort for lokasjonen til brukeren.

4.3.3 Brukergrensesnitt

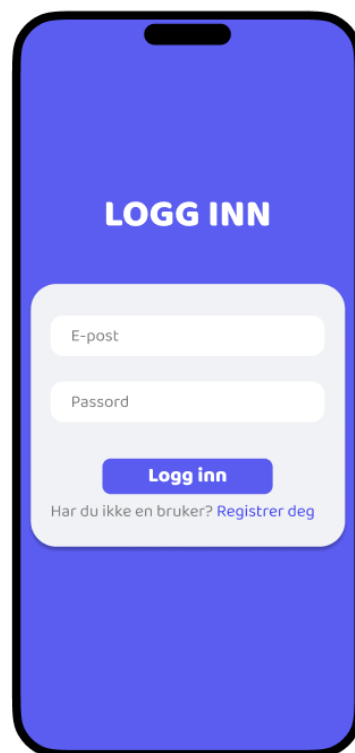
Applikasjonens brukergrensesnitt ble designet før utviklingen av frontend ved bruk av det digitale verktøyet figma. Alle prototyper fra tidlig fase i prosjektperioden er å finne under kapittel 4 i vedlegg 2 - Kravdokument. Som det også nevnes her, er det mange fordeler ved å ha designskisser som utvikler seg i takt med resten av prosjektet. Brukergrensesnittet var gjennom to iterasjoner før det resulterte i den versjonen som ble tatt i bruk i applikasjonen. Den tidligste av disse ble hovedsakelig benyttet for å skape en felles overordnet visjon i gruppen om utseendet, samtidig som den var viktig for å utforme backend med utgangspunkt i ønsket funksjonalitet. Andre iterasjon ble utviklet på et senere stadium i utviklingsfasen, og tok derfor hensyn til ulike endringer av krav og funksjonalitet i applikasjonen som ble gjort underveis. I tillegg ble det brukt betydelig mer tid på denne versjonen, da denne skulle representere det faktiske utseendet på applikasjonen, i motsetning til første iterasjon som kun ble brukt internt.

For å oppnå en responsiv utforming på brukergrensesnittet ble det implementert gjenbrukbare funksjoner som regner ut størrelsen på skjermen til brukerne. Funksjonene henter ut skjermstørrelsen, og beregner deretter relative størrelser i horisontal, vertikal og diakonal retning. All styling ble skrevet ved å plassere størrelser på innsiden av en slik beregning. På denne måten vil for eksempel en padding på 10 piksler ha et likt utseende, relativt, på to mobiltelefoner med ulik størrelse. Skaleringen er ikke tilpasset til å fungere på nettbrett.

Applikasjonen består av en rekke brukstilfeller, og hver av disse krever et eget design med funksjonalitet på frontend-siden. Nedenfor blir det redegjort for designet til mange av de viktigste sidene i applikasjonen, samt hvilke brukstilfeller disse svarer til. Alle brukstilfeller med beskrivelser er tilgjengelig i vedlegg 2 - Kravdokument, og design for hele applikasjonen vises i vedlegg E. En sentral del av utseendet på applikasjonen er at mange av sidene inneholder faner øverst, som gir enkel navigasjon for brukeren mellom relatert eller liknende innhold.

Innlogging

Figur 4.7 viser siden som møter brukeren første gang applikasjonen blir åpnet på en enhet. Her er det også mulig å registrere nye brukere ved å følge lenken for dette. Bruker må også logge inn på nytt hvis autentiserings-tokenet har utløpt. Dette svarer naturligvis på brukstilfellene for å logge inn og registrere.

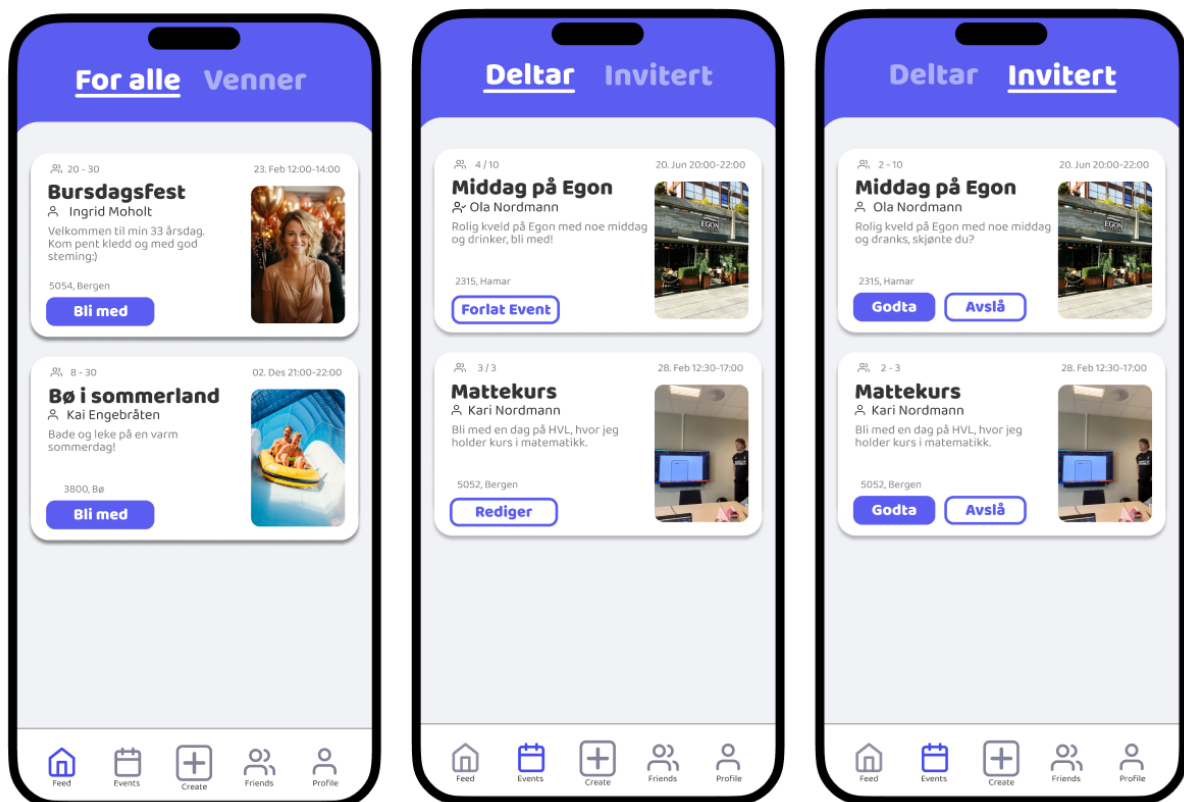


Figur 4.7 - Innlogging

Feed og events

Etter vellykket innlogging eller registrering vil en bruker møtes av en event-strøm, videre omtalt som feed, med eventer basert på lokasjonen til brukeren. Denne feeden ligger til venstre i figur 4.8. Det er også en fane for venner, som er identiske i utseende, men inneholder venners arrangementer som er synlige kun for venner.

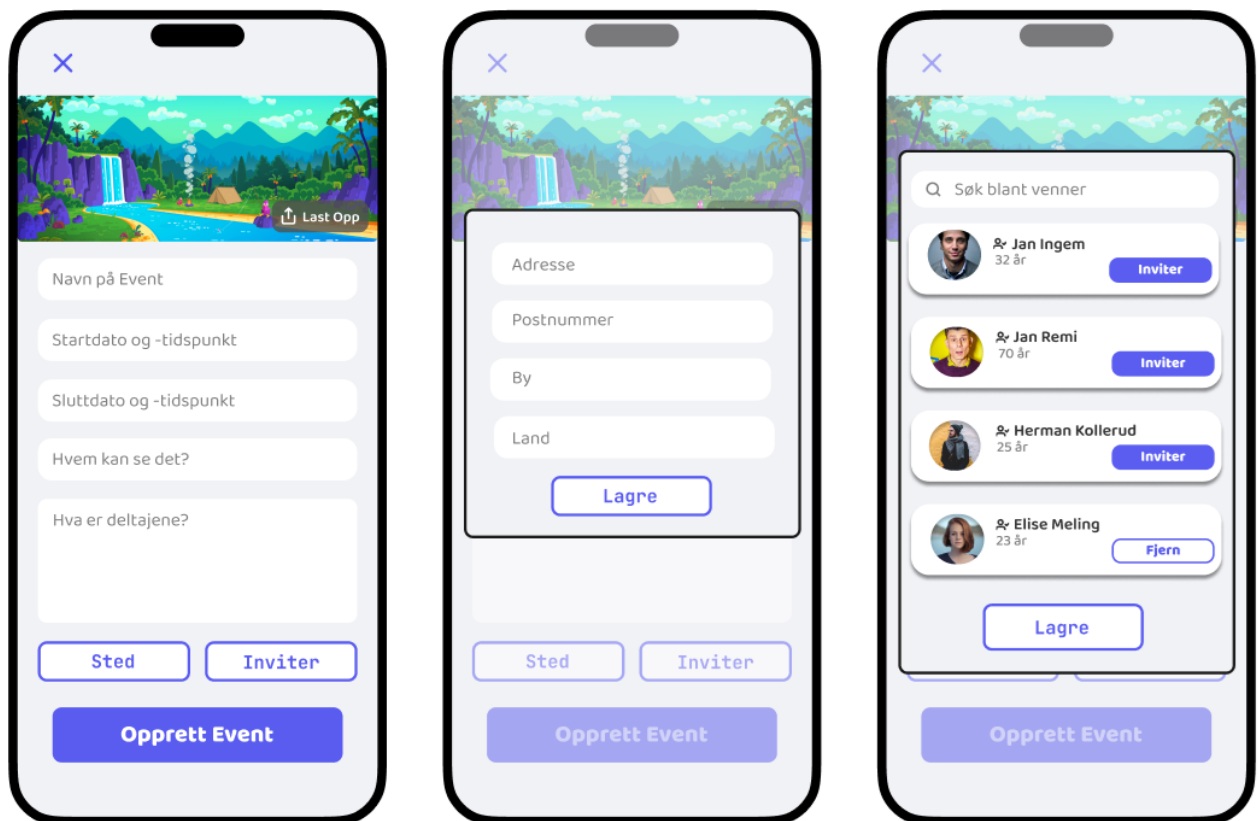
De to bildene til høyre i figuren viser innholdet som ligger under “Eventer” i navigasjonsbaren, og inneholder en liknende oversikt. Også denne siden inneholder faner for ulikt innhold. Her får brukeren oversikt over eventinvitasjoner, samt mulighet til å se eventer brukeren allerede har godtatt. For eventene i “joined”-fanen er det mulig å trykke seg inn på en egen side for eventet for mer informasjon. Funksjonaliteten i denne delen av applikasjonen svarer til brukstilfeller for å utforske-, se-, og svare på invitasjon til eventer. Hvis brukeren som trykker seg inn på et event er verten, er det også mulig å slette og oppdatere informasjon, noe som også er sentrale brukstilfeller.



Figur 4.8 - Ulike Feeds i applikasjonen

Opprette event

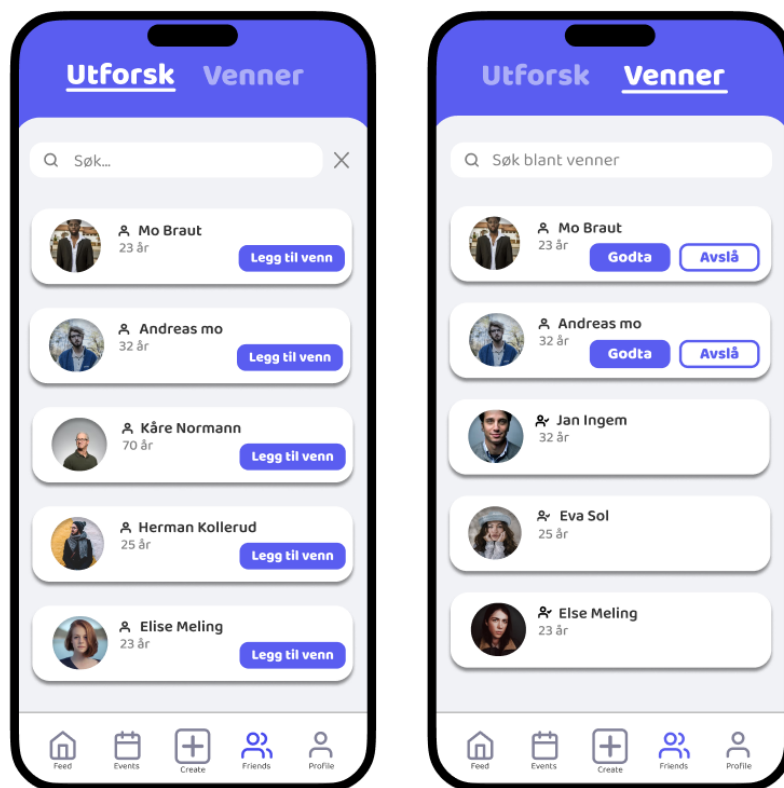
I midten av navigasjonsbaren ligger en knapp som fører brukeren til oppretting av nye eventer. Denne siden dekker hele skjermen uten de tradisjonelle fanene og navigasjonsbaren. Her vil bruker legge inn nødvendig informasjon for å opprette et event. For sted og personer som skal legges til, er det opprettet modaler for å kunne gi rikelig med plass til dette, uten å skape et rotete design for siden ellers. Oppsettet med modaler vises frem i figur 4.9. Siden svarer til brukstilfellet for å opprette eventer og invitere brukere til eventer. I tillegg lar den også brukere oppdatere eventer, da siden vil brukes på samme måte til dette. Ved oppdatering vil feltene ha innhold når man kommer inn i siden, som samsvarer med eksisterende data for eventet som skal oppdateres.



Figur 4.9 - Sider for å opprette event

Personer

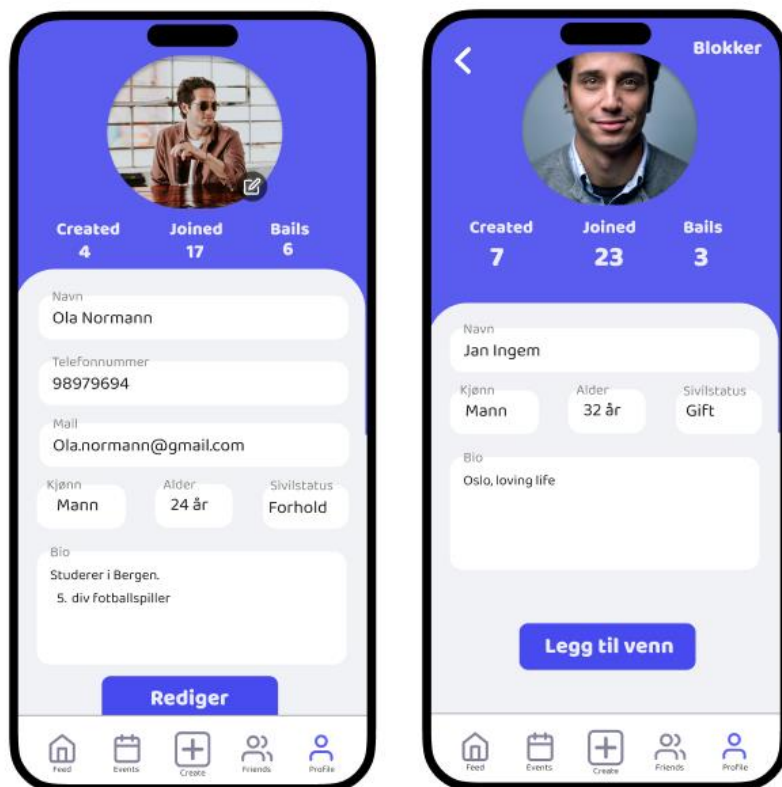
I figur 4.10 vises det hvordan gruppen har løst utforskning av andre brukere og venner. Her er det delt inn i “søk” og “venner” i fanene, hvor søk er laget for å utforske nye personer, mens venner kun viser en brukers venner. Sistnevnte fane inneholder også ubesvarte venneforespørsler fra andre brukere, som kan godtas eller avslås. Søk-fanen tillater brukere å søke etter og legge til nye venner gjennom søkefeltet øverst. Det er også mulig å søke i vennefanen på tilsvarende måte. Siden for personer sørger for at brukstilfeller for å utforske brukere, utforske venner og legge til venner er tilfredsstillt.



Figur 4.10 - Side for å utforske andre brukere og venner

Profil

En annen side som er sentral for å tilfredsstille brukstilfellene er profilsiden, se figur 4.11. Denne sørger for at brukere har mulighet til å vise og oppdatere brukerinformasjon, se egen og annen bruker, oppdatere egen bruker og fjerne venner. Ved visning av egen profil vil bruker ha muligheten til å redigere informasjonen som ligger på siden. Hvis brukeren er inne på profilsiden til en annen profil, vil det være mulig å legge til/fjerne venn, basert på hva forholdet mellom brukerne er ved tidspunktet for profilbesøket.



Figur 4.11 - Profilsider

5 RESULTATER

5.1 Evalueringsmetode

Evalueringen av prosjektet ble i stor grad basert på evalueringsmetodene som var planlagt og dokumentert i delkapittel [3.5](#). Som det nevnes her, er det ønskelig med både verifisering og validering av systemet. Evalueringen ble gjennomført både kontinuerlig underveis i prosjektperioden, samt mot slutten av prosjektperioden med en tilnærmet fullstendig versjon av applikasjonen.

Verifiseringen bestod primært av manuelle integrasjons- og systemtester. Ettersom backend-delen av applikasjonen følger MVC-liknende struktur ble det naturlig å implementere ny funksjonalitet i alle “lagene” av denne strukturen før testing. Deretter ble denne koden testet vha. API-endepunktene i Swagger. Denne tilnærmingen tillot gruppen å bekrefte at ny funksjonalitet fungerte som forventet på hvert nivå, samtidig som det også ble testet at kommunikasjonen mellom lagene var korrekt. Utdypende feilmeldinger førte til enklere feilhåndtering når dette oppstod. Ettersom at mange av API-endepunktene også benytter seg av funksjonalitet som ikke er direkte laget for det aktuelle endepunktet, førte API-testingen til at gruppen fikk verifisert både enkelte enheter og moduler i systemet, samt deres integrasjon med hverandre.

Under utviklingen av frontend ble applikasjonen, som nevnt tidligere, kjørt gjennom en iPhone simulator. Denne kommuniserte med backend API-et, noe som sørget for at integrasjonen mellom serveren og klienten ble regelmessig testet. Det førte også til en kontinuerlig manuell verifisering av inkrementelle versjoner av systemet som helhet, da all implementert funksjonalitet var tilgjengelig ved kjøring av applikasjonen i simulatoren. Mot slutten av utviklingsperioden ble det gjennomført omfattende systemtester både av prosjektgruppen selv, og gjennom brukertester.

Brukerintervjuer ble gjennomført for validering av systemet, tre uker før prosjektets sluttdato. Det valgte tidspunktet førte til testing av et tilnærmet fullstendig produkt, samtidig som at avdekkede feil kunne rettes opp i. Testene ble gjort vha. en kombinasjon av observasjon og dybdeintervju. Intervjuet var viktig for å få svar på i hvilken grad systemet tilfredstilte målene og kravene til prosjektet. Intervjuobjektene fikk først en veldig kort introduksjon til hva systemet gikk ut på.

Etter dette observerte prosjektgruppen, mens objektene testet all funksjonalitet i applikasjonen. Selve testingen av applikasjonen ble muliggjort gjennom Testflight, slik at testobjektene kunne teste applikasjonen på egen mobiltelefon. Prosjektgruppen noterte observasjoner, samt stilte nøytrale spørsmål da det var tydelig at noe var uklart. Avslutningsvis ble det gjennomført et dybdeintervju med ulike spørsmål som var relevante for å tolke prosjektets grad av måloppnåelse iht. initielle krav og mål. Gruppen hadde som mål å unngå ledende spørsmål, for å unngå forutinntatte svar.

5.2 Evalueringresultat

Den kontinuerlige valideringen underveis i utviklingsfasen førte til en fungerende og omfattende applikasjon som tilfredstilte samtlige funksjonelle krav utenom ett. Alle operasjoner på bruker, eventer, og deres relasjoner fungerer på tiltenkt måte. I tillegg viste evalueringen også at intern funksjonalitet i systemet som scoring-mekanisme, responsivt design og Admin-funksjonalitet fungerer i henhold til kravene.

Brukertesten som ble gjennomført ga en god indikasjon på hvordan det endelige produktet svarte på problemstillingen og målene som ble satt for prosjektet innledningsvis. Det er viktig å påpeke at antall respondenter på brukertesten var seks, og at alle respondentene var i aldersgruppen 23-25 år. Et større og bredere utvalg ville gitt mulighet for en mer presis konklusjon, noe som blir diskutert i kapittel [6](#). Hele undersøkelsen med spørsmål og svar er å finne i vedlegg D.

Problemstillingen gikk ut på hvordan det kunne skapes en plattform som gir lav terskel for å delta i eventer både for nye og eksisterende sosiale relasjoner. Løsningen svarer på problemstillingen ved at det har blitt utviklet en fullt funksjonell plattform dedikert for eventer, som er tilpasset for begge disse gruppene. Hvorvidt plattformen gir brukere en lav terskel for å delta i eventer, innhentet gruppen svar på gjennom spørreundersøkelsen. Noen av resultatene er presentert nedenfor i tabell 5.1 og viser at et flertall av respondentene mente at plattformen gir lav terskel for åpne eventer, samtidig som at den er godt tilrettelagt for vennegrupper. Merk at noen av svarene er omformulert fra vedlegget grunnet muntlig språk ved direkte sitering av respondenter.

Spørsmål:	Svar:
Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?	Veldig lav terskel, hadde gjerne gjort dette for å knytte nye kontakter, og håper at dette produktet blir tilgjengelig så fort som mulig.
– –	Relativt lav, så lenge jeg finner noe jeg synes er interessant.
– –	Ganske lav, så lenge det er typiske åpne aktiviteter. Hadde vært veldig lav terskel for å sende eventer til venner.
– –	På Facebook er det mye større, mens her er det mer å gjøre i form av mindre eventer og aktiviteter, som kanskje vil bidra til lavere terskel for å bli med på eventer med bare nye og ukjente personer. Jeg tror dette ville bidratt til å skape en tryggere plattform for å bli kjent med nye personer i form av å delta på spennende og åpne eventer i eget nærområde.
Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?	Ville helst benytte en dedikert plattform, ettersom at dette gir arrangementet en ny vri, og det blir mer spennende samt ryddig å følge med i en slik plattform.
– –	Ville brukt appen, enkel og har alt samlet på en plass.
– –	Hvis appen ble brukt i vennegjengen så hadde jeg brukt denne, det er passe antall for å bruke denne. Snapchat hadde blitt slitsomt, men hadde det vært mange flere så er jeg usikker.
– –	Jeg ville brukt denne, slippe det å sende melding til ulike personer, eller meddele eventet som man har opprettet.

Tabell 5.1 - spørsmål og svar fra brukerintervju

Målene for prosjektet var hovedsakelig å fullføre systemet ved å tilfredsstille alle funksjonelle og ikke-funksjonelle krav, med ekstra fokus på en lokasjonsbasert event-feed, invitasjonslenker og ulik synlighet for ulike arrangementer. Gjennom evalueringen av prosjektet har det blitt tydelig at all funksjonalitet som er implementert fungerer, og at alle de funksjonelle kravene utenom ett er implementert i systemet. Målet om invitasjonslenker for lavere terskel ifm. deling og deltakelse i eventer ble ikke tilfredsstilt. Årsaken til dette var en kombinasjon av løsningsvalget og begrenset tid, og blir ytterligere diskutert i kapittel [6](#).

5.3 Prosjektresultat

Resultatet av prosjektet er en fungerende applikasjon i form av en MVP av et større system, som også var planen for prosjektet. En bruker kan enkelt opprette en bruker, navigere seg rundt på applikasjonen, opprette eventer og brukerrelasjoner, samt utforske og melde seg på eventer i nærområdet. Den er også brukervennlig og sikker. Brukergrensesnittet skalerer likt til alle skjermstørrelser for mobiltelefoner, men ikke til nettbrett. Applikasjonen er koblet opp med Azure, hvor backend blir hostet og gjør det mulig at godkjente brukere kan nedlaste applikasjonen på egen telefon gjennom Testflight. Denne konfigurasjonen gjør det også enkelt å senere lansere applikasjonen på App-Store om dette er ønskelig. Som det kommer frem fra evalueringresultatene i punkt [5.2](#), kan applikasjonen bidra til lavere terskel på å delta i ulike typer eventer. Dette kan føre til knytting av nye bekjenskaper og sosiale kretser, samtidig som det fremmer sosiale aktiviteter i allerede etablerte vennegrupper.

Likevel har applikasjonen enkelte mangler som må utbedres før den kan kalles et fullstendig produkt som er klart for en offentlig lansering. Det funksjonelle kravet om at brukere kan invitere andre ikke-brukere ved hjelp av invitasjonslenker uteble, samtidig som at brukerintervjuet avdekket tilleggsfunksjonalitet som ville gjort applikasjonen mer brukervennlig og gjennomført. I tillegg er det behov for finpussing og forbedring av applikasjonens komponenter, samt refaktorering av enkelte elementer i kodebasen.

5.4 Prosjektgjennomføring

Prosjektet ble gjennomført ved å følge prosjektplanen gruppen utarbeidet i form av et Gantt-diagram, som forklart i del [3.4.3](#). Planen har blitt fulgt i stor grad, men etter hvert som gruppen kom lenger ut i prosjektutviklingen, ble det vanskeligere å overholde den initielle planen. I starten ble store oppgaver inndelt i mindre og enklere oppgaver, som igjen gjorde det lettere å gi hverandre gode code-reviews og jevnlig refaktorere prosjektet. Nærmere slutten ble oppgavene større, og hvert gruppe-medlem fikk hvert sitt ansvarsområde for å ferdigutvikle applikasjonen. Dette gjorde arbeidet mer effektivt, men førte også til at code-reviews og refaktoreringen ble mindre vektlagt, noe som gikk på bekostning av kvalitetskontroll. Ved å ha en prosjektplan og ukentlige sprinter har gruppen hatt en god oversikt over alle oppgavene, en jevn arbeidsfordeling og en fremgang som førte til ferdigstilling av applikasjonen.

Prosjektet er som beskrevet tidligere utviklet med .NET og React Native, samt JWT sikkerhetsimplementering. Disse teknologiene var nye for alle gruppe-medlemmene. Å lære seg teknologiene samtidig som applikasjonen ble utviklet, var derfor noe som ble en sentral del av prosjektgjennomføringen. Dette har resultert i at noen oppgaver har vært mer krevende og tatt lengre tid enn planlagt. Utenom det tekniske har det også forekommet personlige hendelser, som sykdom og over 15 tannlegetimer. Noen av de ukentlige sprintene ble naturligvis preget av dette. Tidsutfordringene har resultert i at valg og prioriteringer ble gjort gjennom utviklingsperioden for å komme i mål, og dette blir diskutert ytterligere i delkapittel [6.1](#).

6 DISKUSJON

6.1 Fremgangsmåte og fremdrift

Den oppsatte planen om å ha ukentlige sprinter og daglige standups ble godt oppfulgt gjennom hele prosjektperioden og førte til et team som samarbeidet tett. I Gantt-fremdriftsplanen ble det satt av god tid til utforming og utvikling av backend, da ASP.NET Core var et nytt rammeverk og språk for utviklingsgruppen. Opplæringen i rammeverket gikk raskere enn antatt, samtidig som EF-Core gjorde at alle databaserelaterte operasjoner ble automatisert og dermed tok lite tid.

Da frontend utviklingen skulle begynne, lå gruppen dermed godt foran skjemaet grunnet de ovennevnte forholdene. I denne fasen av prosjektet ble rapportskrivning et mer aktuelt tema, samtidig som at gruppens opplæring i React Native gikk noe saktere enn antatt. Dette førte til at gruppen falt bak fremdriftsplanen i siste halvdel av perioden. Dette ble også forsterket nærmere sluttfasen da rapporten og andre fag i studiet krevde mer tid. Bruk av verktøyet EAS muliggjorde heldigvis en rask bygging og test-lansering av applikasjonen til brukertesten, og var dermed et nøkkelelement som førte til at prosjektet ble testet i henhold til planen.

Som det også kommer frem ovenfor, samt i kapittel 5, var tidspress det største problemet gruppen hadde underveis i prosjektet. Hvorvidt dette skyldtes dårlig planlegging i oppstartsfasen eller en overvurdering av gruppens evner ifm. å lære seg ny teknologi er vanskelig å si. Det er naturlig å anta at gruppens begrensede erfaring med utvikling av større systemer også kan være en faktor i denne sammenhengen. De negative konsekvensene som kom som et resultat av den begrensede tiden blir videre diskutert nedenfor.

6.2 Negative konsekvenser

Gruppen gjennomførte realistiske brukertester i sluttfasen av prosjektperioden. Til tross for at det ble innhentet viktig data for å evaluere problemstillingen og målene, var det likevel faktorer med brukertesten som kunne vært bedre. Grunnet tidspresset ble det prioritert å utføre utdypende tester på færre objekter, til fordel for å gjennomføre enkelte endringer som ble belyst som nødvendige i testen. Bedre tid til testingen ville bidratt til å innhente data fra en større og mer variert brukergruppe, ikke kun lett tilgjengelige medstudenter. Det er derfor rimelig å til en viss grad tvile på gyldigheten av denne delen av evalueringen.

En annen konsekvens av mangel på tid var at et av målene for oppgaven, i form av et funksjonelt krav, ikke ble oppnådd. Dette kravet, som også er en brukerhistorie, ble ikke implementert delvis grunnet regulatoriske endringer fra Apple om PWA-er, noe gruppen tok høyde for at kunne forekomme i risikoanalysen som ble utformet tidlig i prosjektet. Denne ligger vedlagt i vedlegg C og det nevnes her at tiltaket for denne typen risikoer er å “følge med på regulatoriske endringer og ha planer klare for tilpasning”. Dette gjorde gruppen ved å oppfatte nyhetene, samt velge bort løsningsalternativet som gikk ut på å utvikle en PWA, som potensielt kan bli utfaset på Apple-enheter.

Til tross for at valget gikk på bekostning av prosjektets måloppnåelse, ble dette ansett som et langsiktig valg da det muliggjør videreutvikling av MVP-en i fremtiden uten behov for totale endringer i koden. Det vil også være mulig å legge til funksjonaliteten for invitasjonslenker i løsningen, i tillegg til annen funksjonalitet som for eksempel varslinger. For å integrere invitasjonslenkene på måten det var tiltenkt i den valgte løsningen med en native applikasjon, ville det vært nødvendig med en separat nettside for kun dette, noe som ikke lot seg gjøre i prosjektperioden grunnet de nevnte tidsbegrensningene.

6.3 Mulige forbedringer

Som det kommer frem av diskusjonen, er det enkelte ting gruppen kunne gjort bedre. Majoriteten av arbeidet i prosjektperioden ble gjennomført gjennom gode praksiser, men det er også et forbedringspotensial i måten prosjektet ble gjennomført på. Tidspress og tidsbruk på oppgaver har vært hovedfaktoren til gruppens utfordringer gjennom prosjektperioden. En mulig forbedring for dette til fremtidige prosjekter, ville vært å velge teknologier som gruppen allerede har erfaring med. Dette ville gjort utviklingen noe mer effektiv, slik at det i mindre grad ble nødvendig å lære ny teknologi parallelt med utvikling av systemet.. Dette kunne for eksempel blitt gjort ved bruke Java og Spring Boot som teknologi, noe gruppemedlemmene har erfaring med fra tidligere fag i studieløpet. Valget falt bort fra dette på grunn av ønske om å tilegne seg kunnskap og erfaring innenfor nye teknologier.

Den definerte MVP-en for systemet inneholder, som beskrevet tidligere, de mest essensielle funksjonalitetene, skissert i figur 2.1, til applikasjonen. Et alternativ for å unngå tidspress

ville vært å korte ned eller fjernet noen av funksjonalitetene, slik at systemet ble mindre. Dette alternativet så gruppen imidlertid på som et dårlig valg, ettersom at MVP-modellen allerede var relativt begrenset ut fra den opprinnelige og totale prosjektidéen. Den eneste funksjonaliteten som uteble var invitasjonslenker, som var påvirket av andre faktorer enn kun tidspress, beskrevet i kapittel [6.2](#). Ved å se på de mulige forbedringene mener ikke gruppen at omfanget av disse er veldig stort sett i sammenheng med størrelsen på prosjektet. Det vil likevel være en god erfaring å ta med seg videre inn i andre prosjekter.

7 KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Det endelige prosjektresultatet er et velfungerende system som består av en rekke underliggende komponenter og konfigurasjoner som sørger for den ønskede funksjonaliteten. Systemet muliggjør de essensielle funksjonene som opprettelse og deltakelse på eventer, samt relasjoner mellom brukere. Det er også mulig å utføre en rekke andre operasjoner som er viktige for en sosial plattform, som nevnt flere ganger gjennom rapporten og demonstrert blant annet i delkapittel [4.3.3](#). Det overordnede målet med prosjektet, som også var motivasjonen bak oppgavevalget var å skape en plattform som fremmer sosiale sammenkomster for alle typer personer. Dette ble konkretisert gjennom en problemstilling og mål som gjentas nedenfor.

Problemet ble formulert som: “Hvordan kan det utvikles en plattform dedikert til eventer, som gir en lav terskel både for mennesker som oppsøker nye sosiale kretser, og allerede etablerte grupper som ønsker mer effektiv planlegging.”

Målene ble formulert som: “Å svare på problemstillingen gjennom en fungerende applikasjon som tilfredsstillte teknologiske og ikke-teknologiske krav som er satt av prosjektgruppen” hvor det også ble spesifisert at: “Det mest essensielle for å kunne svare på problemet vil være en lokasjonsbasert event-strøm som gir forslag basert på hvor brukeren befinner seg, invitasjonslenker for enkel deling gjennom andre kanaler, og ulik synlighet for ulike typer arrangementer.”

Fra evalueringen i delkapittel [5.2](#), kommer det tydelig frem at det har blitt utviklet et system som fungerer slik som ønsket, og at de funksjonelle og ikke-funksjonelle kravene er tilfredsstillt utenom ett. Årsaken til mangelen og tankeprosessen rundt dette ble grundig diskutert i delkapittel [6.2](#). Med dette tatt i betraktning mener gruppen at målene har blitt oppfylt så langt det lot seg gjøre innenfor rimelighetens grenser, med en løsning som er skalerbar og fremtidsrettet.

Evalueringen tok også sikte på å gi svar på hvordan prosjektet tilfredsstillte den overordnede problemstillingen. Av resultatene fra brukerintervjuene kom det frem at intervjuobjektene påstod at plattformen var enkel å bruke, samtidig som den tilrettela for oppretting av- og deltakelse på eventer på en intuitiv måte. Spørsmålene og svarene som trekkes frem i figur 5.1 viser at applikasjonen svarer på problemstillingen. Dette fordi de potensielle brukere hevder at de kunne

sett for seg å bruke plattformen til begge brukerscenarioene som er trukket frem i problemstillingen. Det forsterkes også ved at intervjuobjektene på eget initiativ nevner motivasjonen for å benytte plattformen, samt belyser fortrinnene plattformen har sammenliknet med de allerede eksisterende løsningene som gruppen i utgangspunktet mente hadde åpenbare mangler. Disse tydelige manglene var det som lå til grunn for ideen, noe som bekrefter gruppens løsning på problemet ytterligere.

7.2 Videre arbeid

Ettersom prosjektet er basert på en MVP-versjon av et større system er det ganske tydelig hva som er veien videre for selve produktet. Systemet er utviklet med hensyn på videre arbeid og det er dermed enkelt å legge til funksjonalitet fra “should have” og “could have” i MOSCOW diagrammet som er skissert i figur 2.1. I tillegg ville det vært naturlig å begynne med den manglende funksjonaliteten for invitasjonlenker gjennom å etablere en nettside som kan integreres med samme backend-API. En utvidelse av applikasjonen vil føre til et enda mer fullstendig produkt som etter hvert kan lanseres for offentligheten. For å gjennomføre dette ville det også vært nødvendig med en mer omfattende gjennomgang av applikasjonens sikkerhet og personvern. Prosessen for å lansere applikasjonen på en mobilapplikasjonsbutikk vil være lite omfattende, da alle konfigurasjoner for hosting og bygging er satt opp ifm. Testflight som ble brukt til brukertestene.

Hvorvidt gruppen vil gjennomføre de ovennevnte planene for å ta systemet fra MVP til fullversjon er avhengig av en rekke faktorer. Det må tas en vurdering på økonomiske aspekter, samt tilgjengelig tid for videreutvikling av applikasjonen. Fra et forretningsperspektiv vil det være muligheter for å skape inntekter fra plattformen blant annet gjennom annonser, samarbeid med arrangører av offentlige arrangementer, samt betalte medlemskap med utvidet tilgang. Gruppen har ikke valgt å analysere dette opp mot eventuelle kostnader for drift og vedlikehold av plattformen for å beregne lønnsomhet.

Når det gjelder tidsomfanget ifm. med en eventuell videreutvikling anslår gruppen at arbeidet fra MVP til en fullstendig versjon vil være mer tidkrevende enn arbeidet som har blitt gjort hittil i prosjektperioden, som er rundt 950 timer. Dette betyr minimum åtte fulle arbeidsuker for alle gruppe medlemmene, noe som er en betydelig mengde arbeid. Ettersom

at gruppen står bak ideen og har et ønske om å benytte seg av applikasjonen selv, er det likevel en moderat sannsynlighet for at applikasjonen blir å finne på App Store en gang i fremtiden.

8 REFERANSER

Abba, I. V. (2022) *What is an ORM - The meaning of Object Relational Mapping Database Tools*.

Tilgjengelig fra: <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/>. (Hentet 09.04.2024)

Ajcvikers, et al. (2021) *Entity Framework Core*. Tilgjengelig fra: <https://learn.microsoft.com/en-us/ef/core/> (Hentet 09.04.2024)

Apple. (2024) *Update on apps distributed in the European Union*. Tilgjengelig fra: <https://developer.apple.com/support/dma-and-apps-in-the-eu/#dev-qa> (Hentet: 20.02.2024).

Arias, D (2019) *Hashing Passwords: One-Way Road to Security*. Tilgjengelig fra: <https://auth0.com/blog/hashing-passwords-one-way-road-to-security>. (Hentet 11.04.2024).

Atlassian (u.å) *Scrumban: Mastering Two Agile Methodologies*. Tilgjengelig fra: <https://www.atlassian.com/agile/project-management/scrumban> (Hentet 30.04.2024)

AWS (u.å.) *What is an API?* Tilgjengelig fra: <https://aws.amazon.com/what-is/api/>. (Hentet 15.04.2024).

BillWagner, pkulikov. (2023) *Task asynchronous programming model*. Tilgjengelig fra: <https://learn.microsoft.com/en-us/dotnet/csharp/asynchronous-programming/task-asynchronous-programming-model> (Hentet 09.04.2024)

BillWagner, et al. (2023) *Exceptions and Exception Handling*. Tilgjengelig fra: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/> (Hentet 09.04.2024)

Bratbergsengen, K. (2017) *Relasjonsdatabase*. Tilgjengelig fra: <https://snl.no/relasjonsdatabase> (Hentet 11.04.2024)

Bratbergsengen, K., Tøssebro, E. (2023) *Relasjonsmodellen*. Tilgjengelig fra: <https://snl.no/relasjonsmodellen> (Hentet 11.04.2024)

Buck, A. (2024) *Mobile Apps Vs Mobile Websites - What's the Best Choice in 2024?* Tilgjengelig fra: <https://www.mobiloud.com/blog/mobile-apps-vs-mobile-websites> (Hentet: 20.02.2024).

Datatilsynet. (2019) *Risikovurdering*. Tilgjengelig fra: <https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/informasjonsikkerhet-internkontroll/risikovurdering/> (Hentet: 22.02.2024).

Digital.ai. (2024) *17th State of Agile Report*. Tilgjengelig fra: <https://digital.ai/press-releases/17th-state-of-agile-report-71-use-agile-in-their-sdlc-small-organizations-report-strong-business-benefits-medium-and-larger-sized-companies-continue-to-experience-barriers-in-successfully-scaling-a/> (Hentet: 21.02.2024).

Dingsøy, T. (2023) *Smidige utviklingsmetoder*. Store norske leksikon. Tilgjengelig fra: https://snl.no/smidige_utviklingsmetoder (Hentet: 21.02.2024).

Dingsøy, T. (2024) *Scrum (informatikk)*. Store norske leksikon. Tilgjengelig fra: https://snl.no/Scrum_-_informatikk (Hentet: 23.02.2024)

GitHubPang, et al. (2023) *Interface- define behaviour for multiple types*. Tilgjengelig fra: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/interfaces> (Hentet 10.04.2024)

JWT (u.å.) *Introduction to JSON Web Tokens*. Tilgjengelig fra: <https://jwt.io/introduction/>. (Hentet 15.04.2024).

Radigan, D. (2024) *Kanban*. Tilgjengelig fra: <https://www.atlassian.com/agile/kanban> (Hentet: 22.02.2024).

React (2023) *Passing Data Deeply with Context*. Tilgjengelig fra: <https://react.dev/learn/passing-data-deeply-with-context>. (Hentet 1.1.04.2024)

React Native (2023) *Networking*. Tilgjengelig fra: <https://reactnative.dev/docs/network>. (Hentet 11.04.2024)

Swagger (u.å.) *What is Swagger?* Tilgjengelig fra: <https://swagger.io/docs/specification/2-0/what-is-swagger>. (Hentet 09.04.2024).

Visual Paradigm. (2023) *Balancing Data Integrity and Performance: Normalization vs. Denormalization in Database Design*. Visual Paradigm Guides. Tilgjengelig fra: <https://guides.visual-paradigm.com/balancing-data-integrity-and-performance-normalization-vs-denormalization-in-database-design/> (Hentet 06.05.2024)

Wikipedia (2023) *Single responsibility principle*. Tilgjengelig fra: https://en.wikipedia.org/wiki/Single_responsibility_principle (Hentet 15.04.2024)

Wurmser, Y. (2020) *The Majority of Americans' Mobile Time Spent Takes Place in Apps*. Tilgjengelig fra: <https://www.insiderintelligence.com/content/the-majority-of-americans-mobile-time-spent-takes-place-in-apps> (Hentet: 20.02.2024).

Figurer

121WATT. (2023) *Kanban vs. Scrum - was sind die Unterschiede?* Tilgjengelig fra: <https://www.121watt.de/121stunden/agile-methoden/kanban-vs-scrum-was-sind-die-unterschiede/> (Hentet: 23.02.2024).

ONES.com. (2023) *9 Best Scrum Tools For 2023*. Tilgjengelig fra: <https://ones.com/blog/scrum-tools> (Hentet: 23.02.2024).

9 VEDLEGG

VEDLEGG 1 – Prosjekthåndbok og prosessdokumentasjon

VEDLEGG 2 – Kravdokument

VEDLEGG 3 – Visjonsdokument

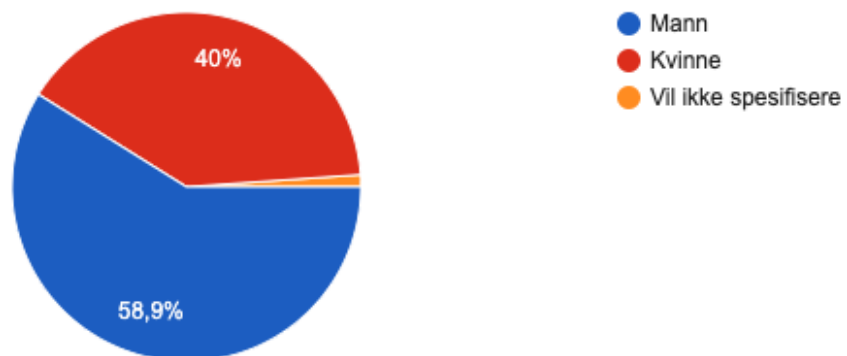
VEDLEGG 4 – Systemdokumentasjon

VEDLEGG 5 – Kildekode

VEDLEGG A – Spørreundersøkelse

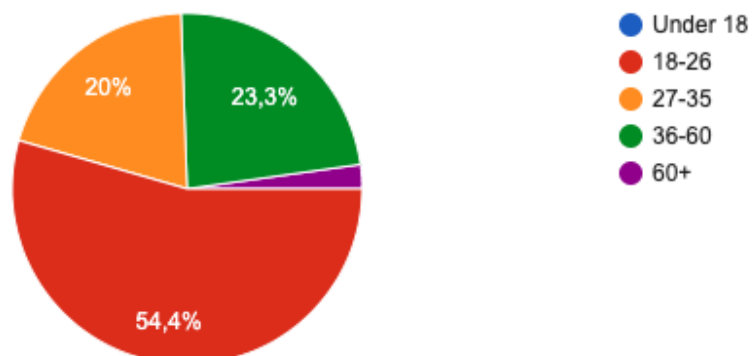
Kjønn

90 svar



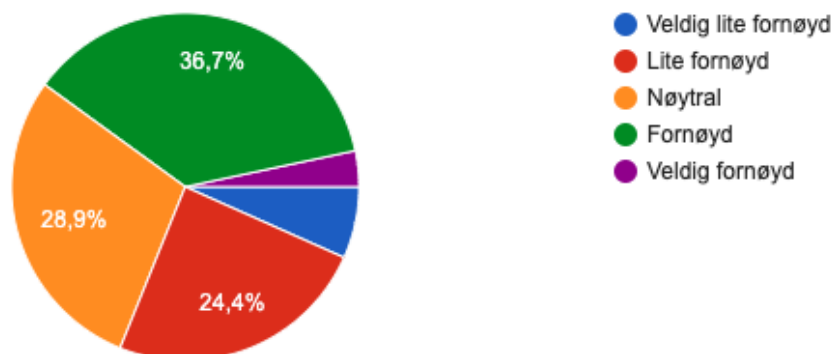
Alder

90 svar



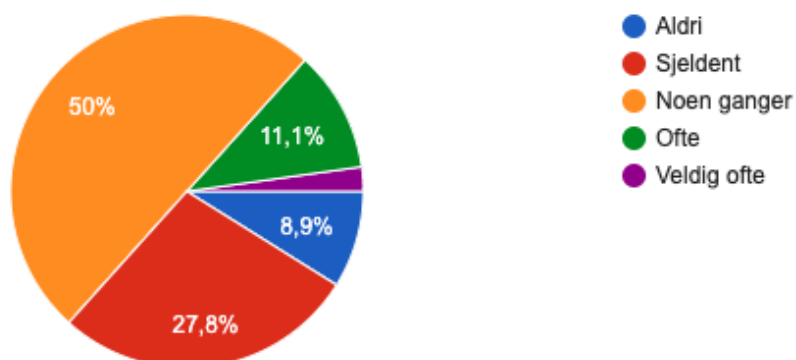
Hvor fornøyd er du med dagens teknologiske løsninger for å møte nye mennesker og bygge sosiale nettverk når du er i nye omgivelser?

90 svar



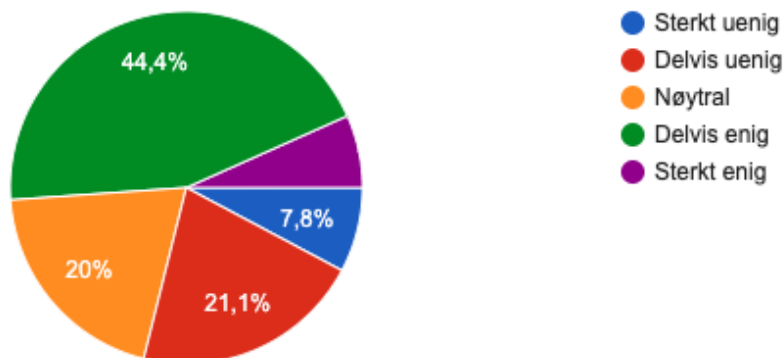
Hvor ofte deltar du i åpne/offentlige arrangementer?

90 svar



I hvilken grad er du enig i at det er vanskelig å finne hverdagslige arrangementer som passer dine interesser?

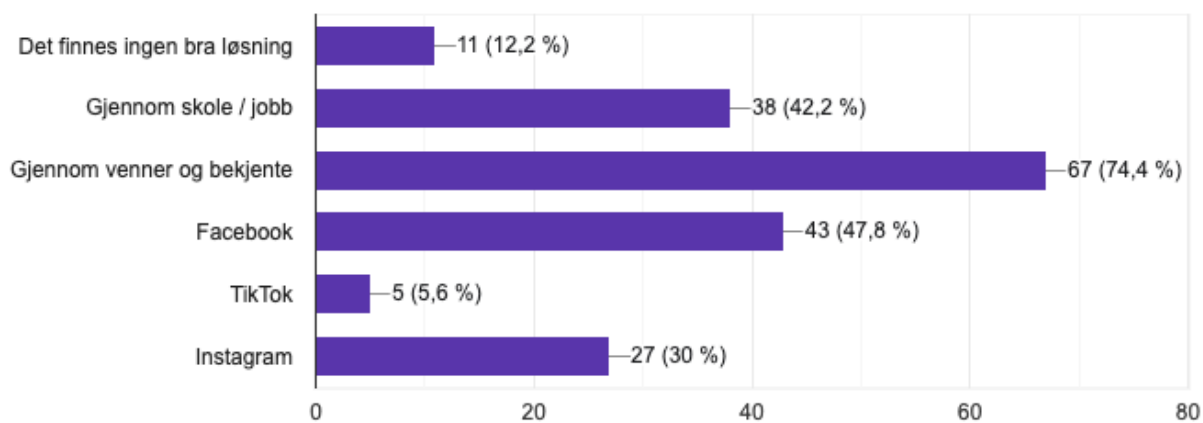
90 svar



Hvis du ønsker å bli med på åpne sammenkomster eller aktiviteter rundt der du bor, hva ville du benyttet deg av for å finne dette?

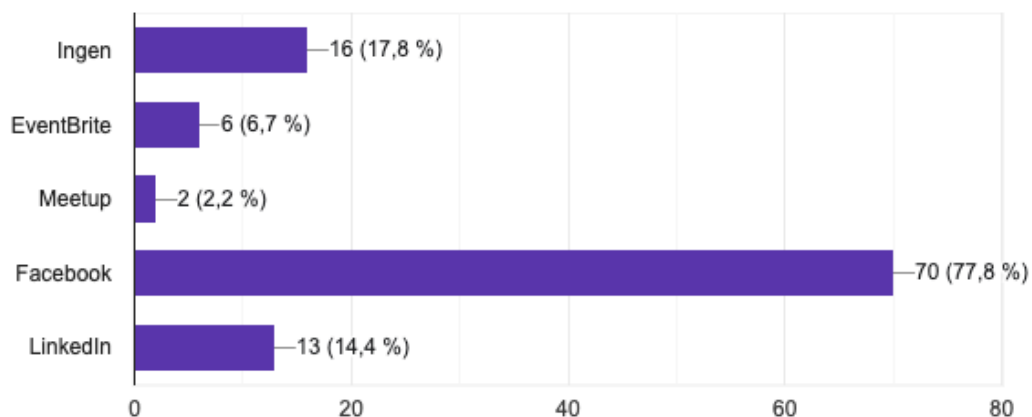


90 svar



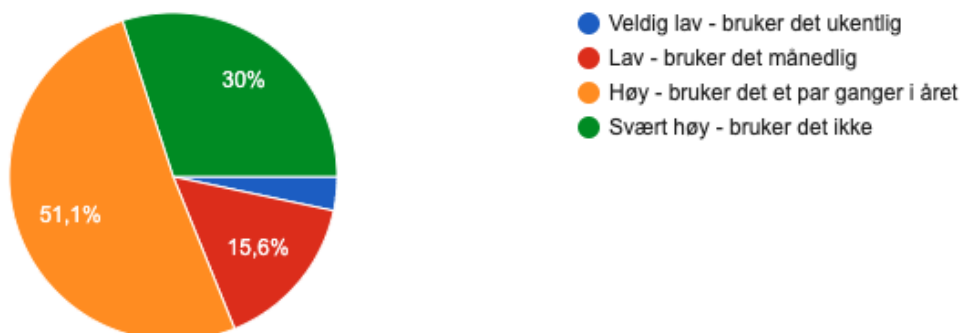
Hvilke(n) av disse applikasjonene har du benyttet til å arrangere/delta i arrangementer tidligere?

90 svar



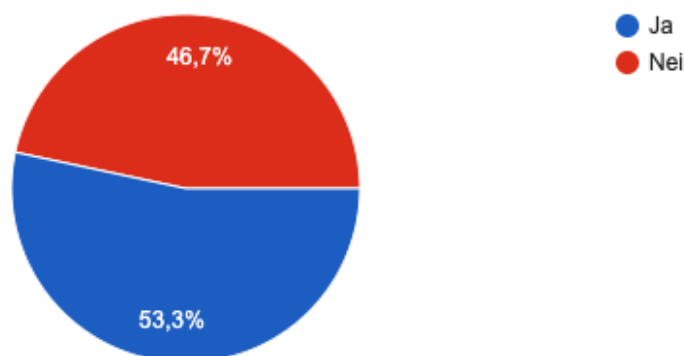
Hvordan opplever du terskelen knyttet til å opprette arrangementer på Facebook eller andre liknende løsninger?

90 svar



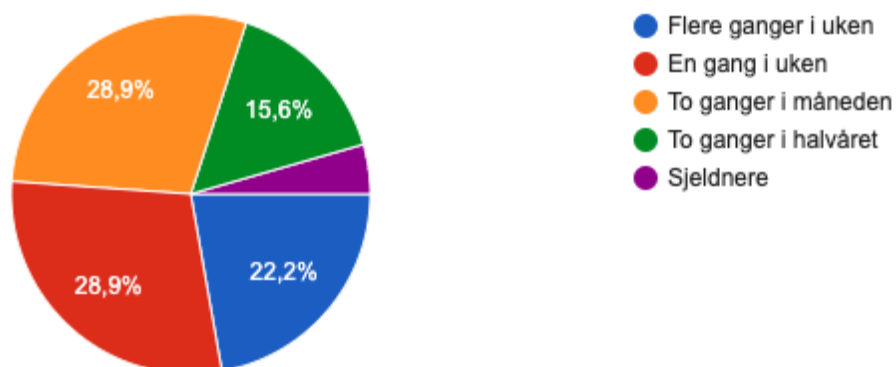
Har du noen gang opprettet et arrangement på en app, og hatt problem med at for få svarer på om de kommer eller ikke?

90 svar



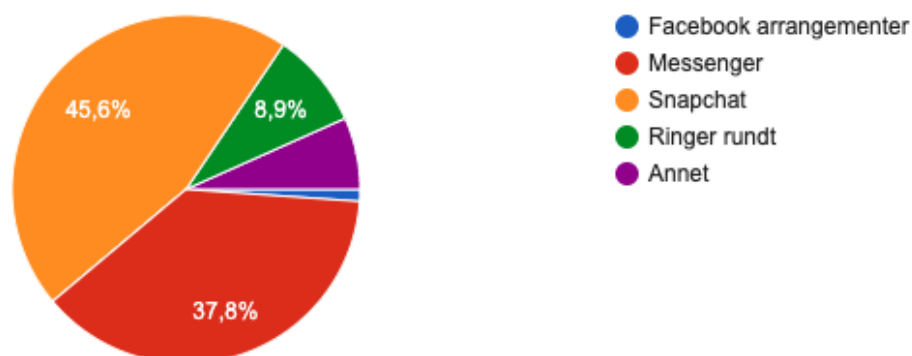
Hvor ofte planlegges/gjennomføres interne sammenkomster i din vennegruppe?

90 svar



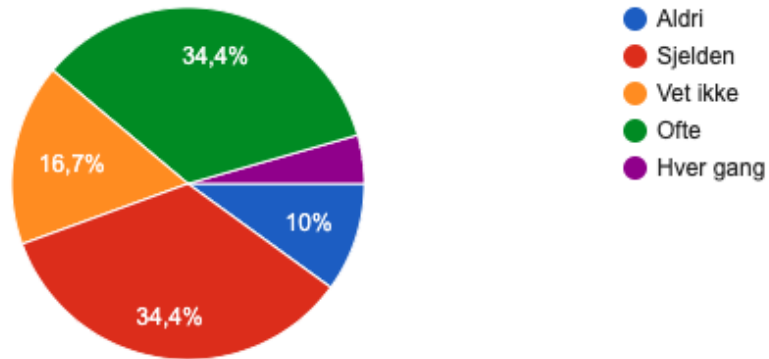
Hvordan løser du planlegging av **mindre** arrangementer blant venner og bekjente?

90 svar



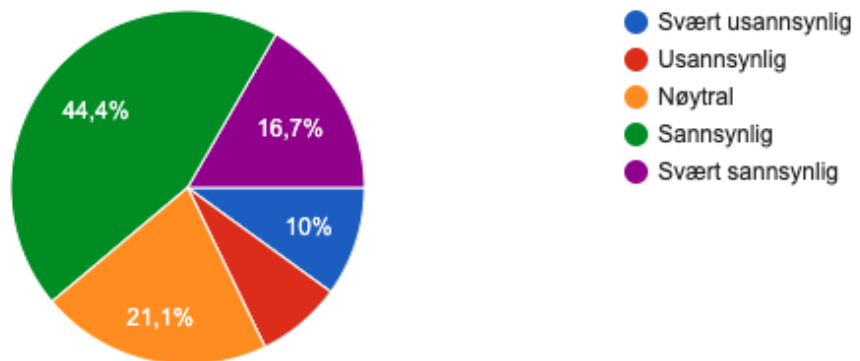
Ved slik planlegging, i hvor mange av tilfellene har du problemer med oversikt, informasjon og kommunikasjon rundt arrangementet?

90 svar

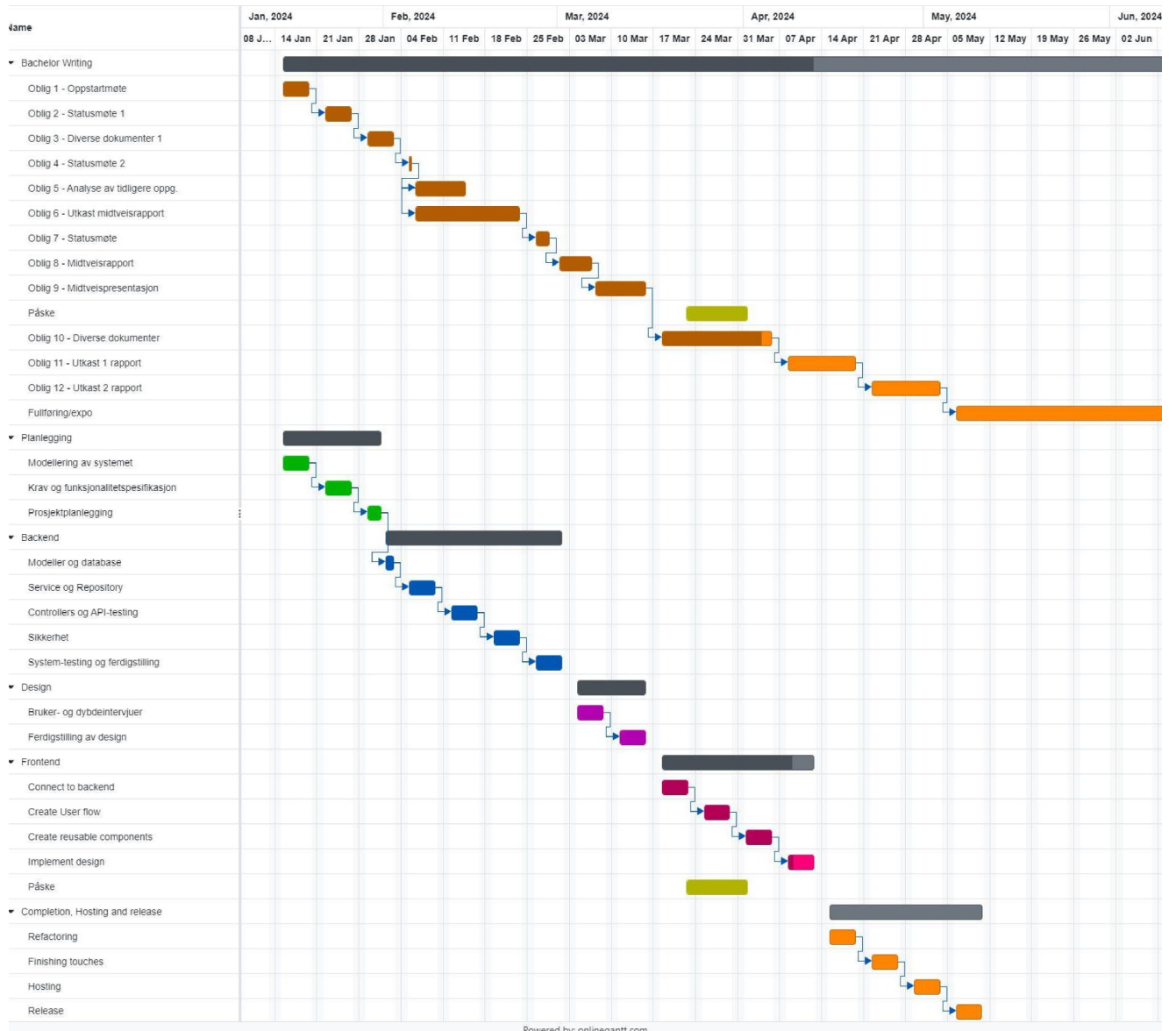


Hvor sannsynlig er det at du vil bruke en ny app designet for å opprette og delta i store eller små arrangementer?

90 svar



VEDLEGG B - Gantt-diagram



VEDLEGG C - Risikoanalyse

	Hedelse/Risiko	Årsak	Sannsynlighet	Konsekvens	Risiko-produkt	Tiltak
1	Applikasjonen blir ikke tatt i bruk.	Lite intuitiv UX, vanskelig for nye brukere å ta i bruk applikasjonen.	2	5	10	Gjennomføre brukertesting.
2	Problemer med høy brukerbelastning som fører til dårlig skalerbarhet.	Underestimert av brukervekst, svak implementasjon for håndtering av mye trafikk og nedetid for server.	2	4	8	Optimalisere systemarkitektur for skalerbarhet. Bruke cloud løsninger for håndtering av tider med høy trafikk. Backups og varslingssystem.
3	Klarer ikke å ferdigstille must-have funksjonalitet.	Mangel på kompetanse og tid, for kompleks funksjonalitet, dårlig planlegging, nye krav og ny informasjon.	2	5	10	Nøyaktig planlegging, godt samarbeid, god opplæring av teknologiene.
4	API-nøkler blir lekket.	Uønsket offentliggjøring av kildekode som inneholder nøkler.	2	4	8	Lagre API nøkler i vaults, lukket utviklingsmiljø, og gode rutiner ved håndtering av API nøkler.
5	Brukere får uautorisert tilgang til data.	Dårlig implementasjon av autorisering og autentisering grunnet mangel på kompetanse, utilstrekkelig testing.	2	4	8	Manuelle- og automatiske tester for sikkerhet, penetration testing og nøye kodegjennomgang av hvilke data som sendes til frontend.
6	Interne samarbeidskonflikter.	Dårlig samarbeid, arbeidsmoral og uenigheter.	1	4	4	Nøyaktig planlegging, god kommunikasjon og konflikthåndtering.
7	Fravær i gruppen.	Sykdom, personlige årsaker og ferie.	2	2	4	Planlegge og legge til rette rundt planlagte forfall. Jevnt arbeid ellers for å minske påvirkning av forfall. God hygiene, samt digitalt samarbeid da dette lar seg gjøre.
8	Regulatoriske endringer som påvirker drift.	Endringer i lovgivning eller forskrifter.	2	2	4	Følge med på regulatoriske endringer og ha planer klare for tilpasning.

VEDLEGG D - Brukertest og Brukerintervju

Brukertesting – feedback fra bruk av applikasjonen:

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

[Navn på side](#) | [funksjonelle krav](#)

Logg inn/Registrer | opprette/logge inn på en bruker

Navigasjonsbar | applikasjonens ulike sider

Feed (Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event
- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner
- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Event-side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

Nytt Event | Opprette event

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere
- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

Hvis du skulle arrangert en sammenkomst for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

MANN - 24 ÅR

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

Navn på side | funksjonelle krav

Logg inn/Registrer | opprette/logge inn på en bruker

Bra greier, må kunne trykke seg ut av tastaturet ved å trykke annet sted på skjermen, foretrekker en felles login/register, vil ha med confirm password

Navigasjonsbar | applikasjonens ulike sider

Enkel og oversiktlig, ikke mye å utsette

Feed (Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Kunne vært litt bedre formatert tid øverst til høyre, hadde vært sexy med dato ikon, eller potensielt både dato og klokke, under hverandre. Ønsker en confirmation før man joiner/leaver eventer, som advarer om bailscore osv. Forventer at kortene er klikkbare, forventer mulighet til å swipe mellom tabs

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Forventer mulighet til å swipe mellom tabs, ellers ser det bra ut, er likt public

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Forventer muligheten til å swipe mellom tabs. Ønsker en confirmation før man joiner/leaver eventer, som advarer om bailscore osv. Skulle ønske at disse kortene skilte seg mer fra public/friends, for eksempel antall dager nedtelling, og sortert i neste først

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Forventer mulighet til å swipe mellom tabs

Event side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

Brukerkortene bør være klikkbare i "folk" oversikten, mulighet for swipe mellom tabs, hadde foretrukket profilbilde av host istedenfor bruker-ikoner

Nytt Event | Opprette event

Ønsker at siden er scrollbar selv om det ikke er mer lengre ned, eventdato-slutt må alltid være etter start. Sted og inviter modalene kan være fort gjort å ignorere

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

Ønsker live update underveis mens man skriver inn, ønsker at det er innhold når man lander på utforsk, men at dette kan feks være fellesvenner eller aktuelle folk.

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Ser bra ut, kan være en mulighet å lande på denne

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Ser veldig bra ut, skulle ønske det gikk an å trykke på statsa/tallene, samt profilbilde

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Skulle gått an å trykke for å forstørre bilder, bør ikke kunne se telefonnummer til ikke-venner, kanskje ikke generelt heller, trykk for å ringe på telefonnummer

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Ja, enkelte tilleggsfunksjonalitet som nevnt ovenfor kan gjøre det enda enklere å navigere*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Ønsket mer funksjonalitet ifm. knappetrykk. Ellers var det meste på plass og enkelt å finne*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?
 - *Ja, det virker enkelt og effektivt. Registreringsprosessen er litt omfattende, samt nedlastning. Kunne vært bedre med 3-parti innlogging. Ville også vært bra med en web-versjon hvor man kan dele eventer med en lenke og melde seg på uten å anskaffe appen og registrere seg.*

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer ,eller en dedikert applikasjon som denne?

- *Ville helst benytte en dedikert plattform, ettersom at dette gir arrangementet en ny vri, og det blir mer spennende samt ryddig å følge med i en slik plattform.*

Hvis du skulle arrangert en sammenkomst for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Dedikert plattform, ettersom at slike arrangementer krever mer oversikt, samt at man i akkurat denne plattformen kan se på bailscores for å sile ut personer som har en tendens til å ikke møte opp.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

- *Til en viss grad, men det ville vært naturlig å legge til rette for å knytte kontakt på selve plattformen etter man har deltatt i åpne eventer, både ved å bli venner på plattform, samt ev. chat-mulighet. Ville også vært enklere om eventer var sortert i større grad på hva det går ut på, og også filtrering i offentlig feed. Sortering gjelder også på tid. Foreslås også mulighet for ulik frivillig tilleggsinformasjon på eventer, som for eksempel "dresscode" eller "ta-med" for å gi mer spesifikke eventer.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *Veldig lav terskel, hadde gjerne gjort dette for å knytte nye kontakter, og håper at dette produktet blir tilgjengelig så fort som mulig.*

MANN - 24 år

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

Navn på side | funksjonelle krav

Logg inn/Registrer | opprette/logge inn på en bruker

Ser bra og clean ut, lett å forstå, ikke for mye informasjon, litt usikker på hva som er mann og hva som er dame.

Navigasjonsbar | applikasjonens ulike sider

Ser proff ut, veldig lett å forstå, bra med ikoner, akkurat passe

Feed(Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Kult å ha feed for lokalområdet, kanskje ha noe feedback på at jeg har blitt med på eventet

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Samme som feed, nice clean, oversiktlig, ikke for mye info.

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Samme som feed, bra enkel og nice

Kult å kunne invitere folk, clean, lett å forstå, veldig enkelt. Så enkelt at kids kunne brukt det, proft design, ser bra ut.

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Samme, veldig lett å forstå, du får den informasjonen du trenger hvertfall, ikke noe mer å si på den egentlig.

Event side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

veldig rett fram, skjønner hvor jeg skal invitere folk, usikker på hvordan jeg redigerer eventet.

Nytt Event | Opprette event

*Veldig enkelt, skjønner akkurat hva jeg skal gjøre. Ingen ting som mangler, akkurat nok informasjon og ikke for mye.
Utrolig enkelt og effektivt.*

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

Clean, men burde fått opp noen forslag fra folk i nærheten eller fra kontaktlisten. Burde kanskje søke på noe annet en enter. Gjerne gjøre det mulig å trykke på profilbildet til folk. ellers nice og enkel å bruke.

burde skje noe på utforsk siden by default ettersom det heter utforsk

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Lett å legge til venner, lett å finne venner forespørsel når de er øverst.

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Fett med scoring mekanisme, burde ha deltatt i 6 om man har opprettet 6. Kul og enkelt å oppdatere informasjon, veldig rett fram og lett.

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Funker bra å slette folk, blokkere folk, kult å kunne gjøre det også.

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Ja, lett å bruke. Alt var sykt bra, bare manglet mulighet for å endre lokasjonen du er.*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Ingenting var vanskelig å finne eller forstå. funksjonaliteten virker ganske bra men merker det er småting som ikke er så bra , men basert på infoen jeg fikk så synes jeg alt er sykt bra.*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

JA

- Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?
 - *Ingenting, kanskje varslinger*
 - *Kanskje hatt en innboks på profilsiden eller på startsiden som en bjelle.*

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Brukt appen, enkel og har alt samlet på en plass.*

Hvis du skulle arrangert en sammenkost for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Denne appen her er lett å bruke og enkel, ikke for mye informasjon eller kommunikasjon.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

- *Jo, det virker lett, ikke noen hindringer, spørs vel bare om folk gidder å melde seg på.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *Relativt lav, så lenge jeg finner noe jeg synes er interessant*

KVINNE - 25 ÅR

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

Navn på side | funksjonelle krav

Logg inn/Registrer | opprette/logge inn på en bruker

Enkelt, lett å forstå, og raskt å opprette en bruker. Irriterende å måtte trykke ned på tastaturet for å få ned tastaturet.

Navigasjonsbar | applikasjonens ulike sider

Veldig enkel å forstå, enkel å bruke, ikke for mange valg.

Feed(Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Fine kort, enkle å se, burde kanskje være en indikasjon på at man ikke kan åpne kort på feed, men gir mening når man får det fortalt.

Smart og lettvis med feed som viser opp alt som er i nærmiljøet. Morsomt å kunne utforske ting som er i egen by

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Kult å kunne se hva venner har opprettet, ryddig å ha en egen fane for venner

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Oversiktlig å se gjennom alle kort, ikke for mye informasjon på en gang. Kanskje sortere slik at evt eventer kommer øverst?

Kult å ha egen fane inne på eventet om hvem som kommer, er moro å kunne se hvem som er invitert eller skal komme.

Enkelt å invitere flere folk. Deilig at det ikke er så utrolig mye innstilling å måtte se igjennom eller forstå.

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Behagelig at man kan dra ned så siden oppdaterer seg, enkelt å forstå.

Dato burde kanskje ha vært i rekkefølge.

Event side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event
*Lett å invitere nye folk, veldig lite innstillinger som gjør det lett og intuitivt å bruke appen.
Kanskje ha en knapp for å slette brukere fra et event?*

Nytt Event | Opprette event

Liker at det ikke er så mye informasjon å fylle ut, men burde kanskje vært noe mer info om hva venner/private er?

deilig at lokasjon er skjult til man bestemmer seg å eventuelt legge til, deilig at man ikke må legge til så mye, men at man kan.

Nice at man kan legge til folk etter opprettet event.

lett å forstå hvordan man oppretter, og intuitivt at denne skjermen er på midten av nav baren

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

Lett å finne og legge til venner, føles naturlig eller vanlig.

Kanskje ha med forslag på venner (Felles venner)

kult å kunne se profilen til andre folk.

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Bra, vanlig, nice å kunne søke om man har mange venner.

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Kult med litt statistikk på hva man har gjort og forlatt. Lett å oppdaterer informasjon.

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere,

Kult å kunne se om folk har opprettet mye, eller forlatt mye. Bra å kunne blokkere folk.

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Veldig enkel, lite innstillinger og funksjonalitet som gjorde det lett å forstå hva man kan og ikke kan gjøre*
- *Store kapper og lyse og fine farger som gjorde det lett å forstå hvor man kunne trykke osv.*
- *Har du brukt noen apper, er designet lett og enkelt å komme i gang med.*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Lett å forstå, lite funksjonalitet som gjør det lett å bruke, skulle kanskje ønske det var med meldingstjeneste.*
- *Kult med bruk av forskjellige faner, kontra å ha masse vanskelige plasser å komme til eller masse i navigeringen.*

- *Ville kanskje ikke blitt med i event om man ikke kan se hvem som skal.*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- *Veldig lett og enkelt, eneste som kanskje kunne hjulpet hadde vært å hatt mulighet til å legge til i kalender*

Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?

- *Varslinger kanskje*
- *opprette var veldig enkelt*
- *Sortere eventer på datoer, litt kronglete å finne frem til hvilke som er i dag, for eksempel.*

Hvis du skulle arrangert en sammenkost for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Hvis appen ble brukt i vennegjengen så hadde jeg brukt denne, det er passe antall for å bruke denne. Snapchat hadde blitt slitsomt, men hadde det vært mange flere så er jeg usikker.*

Hvis du skulle arrangert en sammenkost for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Hvis flesteparten av vennene mine hadde brukt eller hatt appen hadde jeg brukt den, hvis ikke hadde jeg kanskje benyttet noe annet som facebook.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

- *Jeg liker veldig godt at man kan se etter ting å gjøre i byen du er i*
- *Kunne gjerne blitt med på ting som markeder, eller bading eller ting som ikke er så personlig som å dra til noen.*
- *Hadde vært utrolig bra med en sånn plass som dette, istedenfor å måtte følge masse forskjellige instagramprofiler osv.*
- *Hadde vært utrolig hjelpsomt om det ble brukt til en plattform, som markeder osv.*
- *Hadde vært bra med en sentral plass.*
- *Digg og oversiktlig å kunne se etter ting på ett sted.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *Ganske lav, så lenge det er typiske åpne aktiviteter. Hadde vært veldig lav terskel for å sende eventer til venner.*
- *Lavt til alt egentlig*
- *Spesielt til venner, hadde den vært ekstremt lav.*

EKSTRA

- *Tror det hadde vært morsomt å kunne stalket hva andre gjør eller skal gjøre.*

KVINNE - 23 ÅR

Brukertesting – feedback fra bruk av applikasjonen:

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

Navn på side | funksjonelle krav

Logg inn/Registrer | opprette/logge inn på en bruker

Litt vanskelig å forstå kjønn symbolene, kunne stått "hva de er".

Ellers veldig lett å legge inn riktige detaljer når man registrerer seg, veldig cleant og oversiktlig.

Flytte på skjermen ved at man kan scrolle på skjermen mens tastaturet er oppe

Lett å logge inn

Navigasjonsbar | applikasjonens ulike sider

Tydlig og oversiktlig, skjønner hva symbolene og teksten skal formidle om innholdet på de ulike sidene.

Likte større Nytt Event for å indikere at man kan opprette et event

Feed(Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Kan man få opp eventer andre steder, slik at man kan få med seg eventer som ikke kun er basert på din nåværende lokasjon

Oversiktlig innhold til hvert event, får den informasjon som trengs.

Kanskje en interessert knapp, for å holde oversikt over interessante eventer

Kunne lagre eventer, få en egen liste på profilsiden sin hvor man kan holde oversikt over favoritter/interessante eventer

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Litt tung setning på "empty screen"

"Ingen av dine venner har lagt ut nye eventer"

Oversiktlig å kunne se Venner sine eventer, vil være en tab jeg bruker mye tid på

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Fikse slik at antall personer på kortet matcher antallet på loading-baren inne på event-siden.

Validere dato oppsettet og klokkeslett, slik at det er riktig oppsatt, og endre fra engelsk til norsk på måned

Organisere eventer slik at man ser de eventene man skal på først være øverst og først.

Ellers oversiktlig og bra med informasjon, fint med en forlat knapp og fint å kunne redigere når det er ditt eget event.

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Fint å kunne ha både offentlige, venner og private eventer

Gir oversikt over de ulike, og er lett å navigere seg rundt for å se hvilke type eventer man har blitt invitert til, er for venner eller offentlig og derav hvem man har blitt med på og ikke.

Event side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

Oversiktlig og fint, lett å forstå innholdet og nok informasjon, samtidig som du kan endre innholdet eller ikke.

Fint å kunne se alle inviterte til et event, altså deltakelsen.

Nytt Event | Opprette event

Fikse så Date-Picker nr 2 blir satt til startTime.

- Datoen som ble satt ble endra, ikke riktig etter oppretting

Litt vanskelig å skjønne forskjellen på Venner og Privat → bytte om rekkefølge til "Offentlig, Venner, Privat".

Scrolling når man får opp tastaturet, blokkerer description.

STED:

- Når skrive inn postnummer at sted automatisk kommer opp, altså by og land.

ETTER EVENT LAGET:

- Kommer inn på ferdig laget event etter "opprett event" er trykket.
- Adresse står som ikke definert
- Tilbake knappen gjør at man kommer tilbake til event opprettingen, må være feil, men når man går over til Eventer så er de laget, og adresse er riktig, klokkeslett fortsatt feil

GENERELT

- Må kunne redigere hvor mange som kan komme, capacity
- Stor bokstav på måned, skal være liten bokstav og norsk
- Sted og Inviter over Bio/description, lett å glemme og vanskelig å se når de ligger så tett på opprett.
- Ellers er det veldig enkelt og brukervennlig å opprette et event, legge til sted og invitere venner.

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

God forklarende tekst med meningen til tabben.

Ville ikke endret noe.

Veldig enkelt å søke etter andre personer, og legge til venn.

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Ville ikke endret noe, eneste er alfabetisk rekkefølge når mange brukere kommer opp.

Lett å forstå, og fint å kunne se egne venner og svare på innkommende venneforespørsler.

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Henger seg litt opp, blir treig når man skal redigere, må trykke flere ganger på input for å komme inn og skrive.

Rart med picker for sivilstatus, vanskelig å skjønne at man skal bla og ser ikke hva som står i pickeren.

Enkelt å redigere, lav terskel for å ville redigere egen profil, fint å kunne redigere og legge inn eget bilde fra kamerarull.

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Fint å kunne se andre brukere sine profiler.

Legge inn zoom på bilde, eller få en pop up modal med bilde litt større.

Skjønner ikke BAIL, hva det er til.

Fint å ha mulighet til å endre relasjonen med andre brukere, og kunne blokke om det skulle trengs.

Generelt:

Kunne swipe mer fra tab til tab.

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Ja, jeg kan lett navigere meg rundt, og forstår godt de ulike sidene og meningen og deres funksjonalitet.*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Veldig rett fram, lik andre apper så man skjønner oppbyggingen.*
- *Forstår ikke BAILS og hva det betyr*
- *Forstår ikke picker på sivilstatus.*
- *Venner og Privat på Create er litt uforståelig.*
- *Ellers mye bra funksjonalitet som gjør den enkel å bruke og navigere seg rundt*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- *Ja, det er oversiktlig og enkelt å navigere til de ulike scenarioene.*

- Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?
 - *LAGT TIL:*
 - *stigende rekkefølge på de man deltar*
 - *Velge favoritter eller interessert mulighet*
 - *Se i "For Alle" det eventet man har blitt med på*
 - *DressCode eller tema velger*
 - *Placeholder som hjelper deg med beskrivelse i BIO på create (Dresscode, hva som skal skje, ellers annet viktig info)*
 - *Kunne hatt noe som indikerer hvor mange som kommer på event kortet i DELTAR feed*
 - *Når man deltar så kan man delta anonymt, men det står at man kommer.*
 - *TATT BORT:*
 - *Nei, ingenting, kun det nødvendige som er der nå og det er nok.*

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Ville brukt denne, slippe det å sende melding til ulike personer, eller meddele eventet som man har opprettet.*
- *Veldig oversiktlig å kunne ha alt der.*

Hvis du skulle arrangert en sammenkomst for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Samme som over.*
- *Enda lettere å holde kontroll over flere personer med en slik app.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

- *Ja, fint sted for noen å komme inn å ha mulighet til å delta på åpne eventer. Enkelt og oversiktlig feed som gir deg mulighet til å delta kun ved din egen lokasjon, ryddig og lavere terskel for å bli med.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *For meg er terskelen høy, men om det skulle kommet opp noe veldig interessant ville jeg kunne blitt med.*
- *På Facebook er det mye større, mens her er det mer å gjøre i form av mindre eventer og aktiviteter, som kanskje vil bidra til lavere terskel for å bli med på eventer med bare nye og ukjente personer. Jeg tror dette ville bidratt til å skape et tryggere medium for å bli kjent med nye personer i form av å delta på spennende og åpne eventer i ditt nærrområde.*

MANN - 25 ÅR

Brukertesting – feedback fra bruk av applikasjonen:

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

Navn på side | funksjonelle krav

Logg inn/Registrer | opprette/logge inn på en bruker

Det er lett å opprette en konto og det er enkelt å logge inn med konto. Enkelt å velge fødselsdato, kan kanskje ha en mer forståelig måte å velge kjønn på.

Navigasjonsbar | applikasjonens ulike sider

Veldig enkelt og oversiktlig. Jeg liker bruken av ikonene som forklares godt med teksten.

Feed(Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Hvert event-kort er nok forklarende, og kult med at jeg kan få inn eventer som skjer i mitt nærområde.

Savner å kunne velge eller filtrere andre byer, hvis jeg for eksempel skal til et annet sted og vil se om det skjer noe.

Enkelt og forståelig at det er offentlig eventer med "For alle" tabben og man kan velge å bli med.

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Samme som over, veldig lett å forstå at det er dine egne venners eventer, og fint å kunne se hva dine venner legger ut.

Generelt

- *Gøy å kunne å ha et valg om å kanskje bli med på et offentlig arrangement, eller om jeg vil forholde meg til mine venner sine.*
- *Fint å kunne trykke på eventkortet for å se mer av eventet.*

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Oversiktlig og lett å kunne se hvilke jeg deltar i, og hvilke det er som er mine egne jeg kan redigere.

Synes det kunne vært enda mer oversiktlig hvis de var sortert etter når det starter, slik at jeg enkelt kan vite hva som skjer først og ikke.

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Kunne hatt noe form for varsel-system hvor man får beskjed om at man er blitt invitert.

Enkelt å forstå, og fint skille mellom de jeg har deltatt på og de jeg ikke har svart på enda.

Event-side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

Gøy at man kan se hvem som kommer og ikke, samtidig veldig enkelt å invitere folk.

Fint å få litt mer informasjon, og bra bruk av ikoner for adresse og klokka,

- *Kanskje hatt dette på hvert eventkort også?*

Liker stort bilde, kanskje mulighet for å bla gjennom flere bilder

Nytt Event | Opprette event

Forståelig og brukervennlig å opprette et event, bra mengde detaljer for å lage et bra event.

Enkelt å velge hva slags event man vil ha, og legge til sted, dato og klokkeslett og samtidig kunne velge hvem av mine venner jeg vil invitere.

Litt vanskelig å se alt man skriver i beskrivelsen pga at tastaturet går over, også litt vanskelig å se sted og invitere knappene etter jeg har skrevet bio, eventuelt flytte på disse eller gjøre tydeligere.

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

Forståelig og enkelt å kunne søke opp i andre brukere

Kunne hatt noen brukere som vises ut i fra felles eventer eller felles venner

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Fint å få opp nye venneforespørsler øverst slik at man ser de.

Bra å ha en oversikt over venner, og fint at man kan søke opp når man eventuelt får så mange.

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Fint å kunne se sine egne stats, og veldig enkelt å forstå hva jeg kan redigere.

Burde endre Picker, vanskelig å forstå hva man skal med den og ser ikke valgene.

Fint å få en varsel på om man vil lagre eller ikke, så man er sikker og kan gå over å se at det ikke er noe feil.

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Fint å ha mulighet til å fjerne venn, eller generelt endre på vennskapet, også blokkering.

Burde vært mulig å swipe til venstre til at man kommer tilbake enn å måtte trykke på knappen.

Bra med informasjon om andre brukere.

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Ja, forståelig og enkelt.*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Bra funksjonalitet, kunne lagt til noe form for varslinger for å se når man er invitert og når man får venneforespørsel.*
- *Enkelt å finne alt og navigere seg ut i fra ikonene og tabsene.*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- *Ja.*
- *Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?*
 - *Velge antallet som skal få komme*
 - *Ellers er all nødvendig funksjonalitet og informasjon der til at jeg kan lage et kult og bra event.*

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Ja, fint å slippe å opprette grupper i andre sosiale medier, og bare ha alt i en applikasjon så jeg har oversikt.*

Hvis du skulle arrangert en sammenkomst for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Samme som over, men enda bedre for større sammenkomster, gjør det lettere å holde oversikt.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne arrangementer?

- *Ja, det er en fin mulighet å ha.*
- *Hvis man kan filtrere på områder også vil det kanskje være enda enklere.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *Det spørs litt på eventet, men om det er noe interessant hadde det vært gøy å bli med på noe uten at det er min vanlige vennegjeng.*
- *Tror terskelen min ville kunne endret seg til mye lavere, om jeg hadde hatt muligheten og et valg om å kunne bli med på åpne arrangementer tilgjengelig.*

KVINNE - 24 ÅR

Brukertesting – feedback fra bruk av applikasjonen:

1. Intro - intervjuholder forklarer hva applikasjonen er på et overordnet nivå
2. Generelle tilbakemeldinger, på brukervennlighet, hvor enkel appen er å bruke og navigere på, forståelig:

[Navn på side](#) | [funksjonelle krav](#)

Logg inn/Registrer | opprette/logge inn på en bruker

Veldig enkelt å logge inn og å registrere en ny bruker, uvant å velge kjønn.

Navigasjonsbar | applikasjonens ulike sider

Enkel å bruke og liker at det er ikoner som forklarer sidene

Feed(Hjem)

- **For Alle** | melde seg på eventer, vise frem eventer i lokalområdet, opprette relasjon mellom brukere og event

Lett å forstå, ikke for mye informasjon, enkelt å bli med i et event.

Kunne hatt litt bredere bilder eller mulighet til å forstørre ved å trykke på.

Merkelig at eventer ikke er sortert på dato.

Kult med innhenting av eventer i nærområdet, men hva med å finne eventer i andre byer.

- **Venner** | melde seg på eventer, vise frem eventer kun for venner, opprette relasjon mellom brukere og event

Samme som over, men fint at det er mulighet for kun venner også.

Eventer

- **Deltar** | vise en brukers opprettede eventer, vise eventer bruker deltar i, vise frem brukers eventrelasjoner

Merkelig at eventer ikke er sortert på dato.

Fint og oversiktlig å ha et sted hvor jeg kan se eventer jeg skal på, og eventer jeg har laget.

- **Invitert** | vise frem eventer kun for inviterte, opprette relasjon mellom brukere og event, vise frem brukers eventrelasjoner

Fint og oversiktlig, men hadde vært fint å få en bekreftelse på at man svarer ja på en invitasjon, og ikke bare forsvinner.

Event-side | Slette event - oppdatere event, funksjonalitet for å håndtere brukere i et event, slette relasjon mellom bruker og event, oppdatere relasjon mellom bruker og event

Oversiktlig og bra med informasjon som forklarer det nødvendige til et event.

Fint å kunne se hvem som kommer, fordi det kan gjøre det lettere for meg å bestemme meg for å dra eller ikke.

Hvis det skjer en oppdatering på et event, kanskje ha noe form for varslinger så jeg vet hva som er endret.

Nytt Event | Opprette event

Liker at det er lite informasjon som må fylles ut, samtidig som det er all den nødvendige informasjonen.

Fint å kunne invitere folk og spesifisere lokasjon.

Hadde vært fint å ha mulighet til å legge til flere bilder, som man kan scrolle gjennom.

Personer

- **Utforsk** | Søke etter bruker, opprette relasjon mellom brukere

Greit med tom side, men kunne kanskje fått opp forslag til andre brukere å legge til som venn.

Ellers veldig enkelt og bra å kunne søke på både fornavn og etternavn.

- **Venner** | vise en brukers venner, vise en brukers relasjon til andre brukere

Godt å ha et sted med oversikt over mine venner, samtidig kunne søke spesifikt etter noen.

Profil | oppdatere egen bruker, automatisk oppdatering av scoring mekanisme

Enkelt å oppdatere og lagre det jeg kan og vil endre.

Kunne vært fint å få opp antallet venner jeg har, ellers kult å se stats på hvor aktiv man er.

Annen Profil | Gå inn på andre brukere sine profiler, slette relasjon mellom brukere, oppdatere relasjon mellom brukere

Synes det hadde vært bedre om jeg kunne sett om vi hadde noen felles venner for eksempel.

Utenom det er det bra med informasjon om andre brukere, og bra med bilde, men burde kunne vært mulig å trykke på og få det zoomet inn.

Generelt:

- *Behagelig at det er kun to faner/tabs på de ulike sidene, oversiktlig og brukervennlig.*
- *Fint med ulike typer eventer, altså åpne, for venner og privat.*

3. Brukerintervju – hvordan appen svarer til problemstillingen:

Synes du applikasjonen var enkel/intuitiv å bruke?

- *Ja, det er den.*

Hva synes du om funksjonaliteten, og var noe av det vanskelig å finne/forstå?

- *Alt var veldig oversiktlig og fint.*
- *Første gang var det litt vanskelig å finne alt og forstå alt, men etter å ha gått gjennom applikasjonen var det veldig enkelt og fint fordelt med tabs.*

Virker det effektivt og oversiktlig å opprette/planlegge/delta i et event med denne applikasjonen?

- *Ja, alt virker utrolig enkelt og bra.*
- *Eneste som manglet litt var det å kunne oppdatere/redigere et event etter at jeg hadde opprettet det.*
- Hvilke funksjoner kunne vært lagt til/tatt bort for å gjøre dette mer effektivt og oversiktlig?
 - *Det er veldig effektivt, ville ikke tatt bort noe. Eneste er muligheten til å legge til mer bilder når jeg oppretter et event.*

Hvis du skulle arrangert en sammenkomst for 5-10 personer du allerede kjenner godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Ville brukt denne applikasjonen, om flere av mine venner brukte den også.*
- *Den gjør det lettere å holde oversikt og kontroll på detaljene.*

Hvis du skulle arrangert en sammenkomst for 20+ personer du allerede kjenner nokså godt, ville du brukt meldingstjenester, andre sosiale medier for arrangementer, eller en dedikert applikasjon som denne?

- *Ja, her er en slik applikasjon perfekt, ettersom det er så mange personer og detaljer til et så stort event, er det fint å kunne ha det på et samlet sted.*

Synes du applikasjonen gir enkel mulighet for å bli kjent med andre personer med åpne eventer?

- *Jeg synes det, ettersom det er et valg vi som brukere har om vi vil delta eller ikke. Du bestemmer da selv om du vil utforske og eventuelt bli kjent med andre personer.*

Hvordan tror du terskelen din ville vært for å melde deg på et åpent event som du synes var interessant, om det dukket opp i din feed?

- *Terskelen min ville basert seg på hvor relevant det er for meg å dra på dette eventet. Men jeg tror at ved å ha denne appen over tid kunne jeg dratt på flere ulike åpne eventer.*
- *Vil nevne at jeg synes det er fint at man kan ha en app for det er en valgmulighet til å sosialisere seg.*

VEDLEGG E - Applikasjonens brukergrensesnitt

