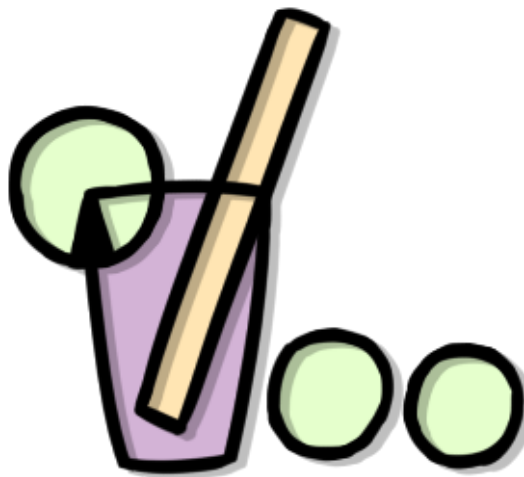


Lille Promille

Detaljert Systemdesign

Gruppe 18:
Joakim Spjutøy Aarskog
Lars Erik Faber



Contents

1	Begrepsliste	1
2	Introduksjon	1
2.1	Hvordan fungerer appen?	1
2.1.1	Et eksempel	1
3	Design	1
3.1	Utseende	1
3.2	Sider i Appen	2
3.2.1	Oversiktsside	2
3.2.2	Enhetskatalog	3
3.2.3	Innstillinger	4
3.2.4	Venneside	4
3.2.5	Kartside	4
3.2.6	Innloggingsside	4
3.2.7	Økthistorikk og Individuelle Økter	5
4	Systemdesign	5
4.1	Navigasjon	5
4.1.1	Struktur og oppdeling	5
4.1.2	Navigation Pattern	5
4.1.3	Navigasjonsflyt	6
4.2	Teknisk oppbygging	7
4.2.1	Services	7
4.2.2	Fragmenter	7
4.2.3	Lagring og uthenting av data	7
4.2.4	Brukerhåndtering	7
4.3	Ressurser	8
4.3.1	Klasser	8
4.3.2	Bilder	8
4.3.3	Lyd	8
4.3.4	Biblioteker	8
4.3.5	Hva vi er usikre på	9
5	Krav	9

1 Begrepsliste

Begrep	Beskrivelse
Økt	Perioden fra man drikker sin første enhet til man er kjørbar
Kjørbar	At man er i stand til å kjøre bil (Man holder promillegrensen)
Enhet	Beskriver en drikke basert på alkoholinhold og volum

2 Introduksjon

Det er ingen hemmelighet at det festes mye gjennom studietiden og som konsekvens fører dette til at noen velger å fyllekjøre, enten om det er for å komme seg hjemm eller for å ta med festen på veien. Samtidig er det ingen god løsning til dette problemet, selv om loven straffer hardt mot fyllekjøring hindrer det ikke noen prøver seg. Så hvordan kan dette begrenses?

Lille Promille er en app som takler dette problemet ved å fokusere på selve drikkingen. Hvis man kan kontrollere seg selv og sine venner, vil man kunne ta bedre valg og unngå å kjøre i påvirket tilstand. For å gjøre dette, tilbyr appen en oversikt over kjøretilstanden sin. Man får innsikt til alkohol i blodet, antall drikker man har drukket, og hvor lenge det er til man kan kjøre bil. Det vil også være et vennesystem der man kan se hvor mye venner har drukket, som bidrar i å engasjere studenter til å bruke appen.

Appen vil ikke kun være for studenter, men alle som ønsker å ha kontroll mens man fester. Visjonen er at man kommer til å bruke Lille Promille appen for å få oversikt over hvor man drikker, samt hvor mye venner drikker. Det vil være en app som gjør festingen bedre men samtidig minsker sjansen for fyllekjøring, siden dette er av erfaring noe mange lurer på mens de er påvirket.

2.1 Hvordan fungerer appen?

2.1.1 Et eksempel

Ta utgangspunkt i at det er en fest på en søndagskveld, og dagen etter må man være klar til å kjøre til høgskolen tidlig ettermiddagen. Når man tar sin første drikk, åpner man appen og legger til en enhet som passer drikken, for eksempel en rusbrus. Man velger å legge til disse for hver slurk eller for hver flaske man har drukket. Appen vil med jevne mellomrom beregne promillen, slik at man har full oversikt gjennom hele festen. Appen viser rett før midnatt at det er tolv timer til man er kjørbar, og man velger å stoppe å drikke.

Dagen derpå sjekker man appen og ser at det fremdeles er noen venner som er påvirket, og til og med noen som enda drikker. Men siden man fulgte appens anvisninger, er man sikrere på at man har beholdt promillegrensen og er klar til å kjøre bil. I tillegg har man muligheten til å se hvordan økten gikk på festen, som hvor man befant seg til diverse tider, hvor mye man drakk og hvor ofte. Disse dataene kan man igjen bruke til senere fester slik at man blir sikrere på sine egne grenser og hva man tåler av alkohol.

3 Design

3.1 Utseende

Kodeordet for grafikken i Lille Promille er ”oversiktlig”. Derfor går vi for et rent utseende med få komponenter i hver side, slik at det ikke blir for mye rot. Hensikten er at man ønsker gjerne kun den relevante informasjonen i en slik app, spesielt når man er påvirket av alkohol. Nedtellingsklokken på

forsiden er ganske stor for å fange oppmerksomheten, og videre er informasjon vist i rekkefølge nedover siden basert på viktighet. Dette design-prinsippet følger stort sett alle sider i appen for å gjøre opplevelsen mest behagelig som mulig for brukeren. Skjermbildene ovenfor demonstrerer dette.

Fargemessig har vi gått for nøytrale farger, med noe rødt og blått for å signalisere funksjonaliteten til brukeren. For eksempel en rød knapp for å signalisere at noe skal slettes, eller en blå tekst for å markere at den er interaktiv.

Videre har vi valgt å gruppere og abstrahere informasjon vekk fra første-sidene, slik at man ikke overvelder brukeren med informasjon. Et godt eksempel på dette er økt-historikken, da man kan velge å inspisere en spesifikk økt for å se mer informasjon om den økten. Dette er også et prinsipp som går gjennom hele appen. Appen benytter seg også av ikoner for å dirigere bruker i riktig retning og gi mer kontekst bak de interaktive elementene.

3.2 Sider i Appen

3.2.1 Oversiktsside

Et av de viktigste stedene i appen er oversiktssiden der all informasjon relatert til en *økt* står oppsummert. Siden består av en nedtellingsklokke, en liste over hva man sist har drukket, og en knapp for å legge til en ny drukket enhet.

Tanken her er at når brukeren begynner å drikke, legger de til en nydruknet enhet til listen vist på skjermen. I dette øyeblikket starter en ny økt, der nedtellingsklokken viser tiden det vil ta før man er *kjørbar* basert på kjønn, vekt og høyde på brukeren. Dersom brukeren drikker enda en enhet, legges denne til i økten ved hjelp av knappen, og nedtellingen kalkuleres på nytt. Etter hvert som tiden går vil listen på oversiktssiden fylles opp med type enhet som ble konsumert, når den ble konsumert og hvor mye alkohol den inneholder. Under nedtellingsklokken vil det stå en tilnærmet verdi av den nåværende promillen i blodet til brukeren.

Når nedtellingsklokken har nådd null sekunder, vil økten avsluttes og bruker vil være *kjørbar*. Denne økten vil deretter bli lagret, slik at man kan inspisere denne økten senere i en egen side med tidligere økter.

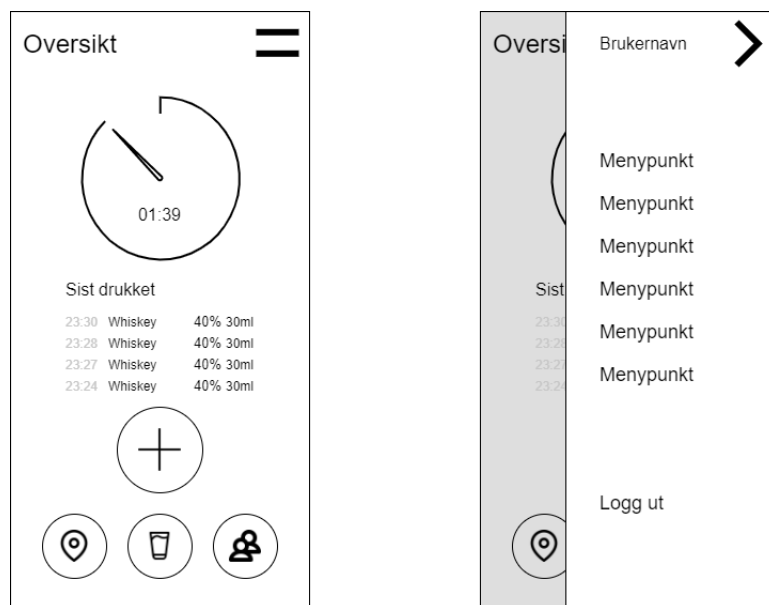


Figure 1: Til venstre er oversiktssiden med følgende elementer: Nedtellingsklokken, liste av ny-drukkede enheter, en pluss-knapp og et navigasjonselement på bunnen. Når man trykker pluss knappen dirigeres man til enhetskatalogen, kan man velge en enhet man vil legge til i økten. Til høyre er det skissert en meny (navigation-drawer i Android studio) med menypunkter som man kan bruke til å navigere appen.

3.2.2 Enhetskatalog

Enhetskatalogen er en side med opplistning av alle definerte enheter i Appen, og brukes til å legge til enheter i løpet av en økt. Som ny bruker, vil det allerede ligge predefinerte enheter i listen som de kan bruke og gjøre endringer på. Ved behov kan nye enheter bli laget av brukeren gjennom pluss-knappen, da vil det komme opp en modal med noen tekst-felter. Her må brukeren beskrive navn på enheten, alkoholprosent og volum. For eksempel kan man lage en enhet som man kaller for "Whiskey Shot" med 40% alkohol og 30ml volum. For å slette enheter trykker man på i-symbolet ved siden av navnet, og man blir sendt til en ny side med informasjon om den spesifikke enheten der man kan velge å slette enheten.

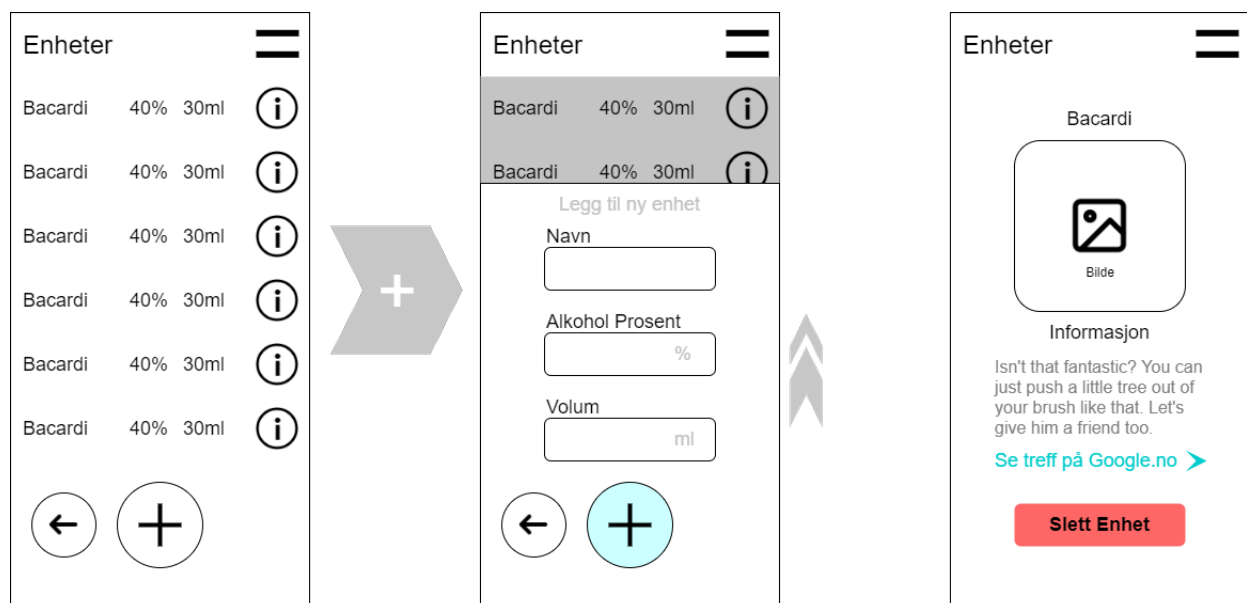


Figure 2: Enhetskatalogen: Listen til venstre består av tidligere definerte enheter. Det er denne siden man benytter når man skal legge til enheter i en økt, og det gjør man ved å trykke på navnet. Med pluss-knappen kan man legge til en ny enhet i listen, og med i-knappen kan man se mer informasjon om enheten. På informasjonssiden har man også valget å slette enheten fra listen.

3.2.3 Innstillinger

For å endre på noe med kontoen eller med appen går man til siden som heter ”innstillinger”. Her vil det være listet en rekke innstillinger gruppert etter konto, apputseende og generelle innstillinger, som man kan endre på. Disse vil være knyttet opp til den spesifikke brukerkontoen som er logget inn. Når man er fornøyd med endringene sine må man trykke ”Lagre endringer” for at de skal ta effekt.

3.2.4 Venneside

Vennesiden er der man kan legge til venner og se statusen på venner, som baserer seg på den nåværende økten. Man kan også sende emoji-meldinger til en venn i et chatrom. Dette er for å øke engasjementet ved å la brukerne kommunisere med hverandre, men de må være kreative i prosessen. Det kan bli interessant å se bruken av emojis når man er påvirket av alkoholen.

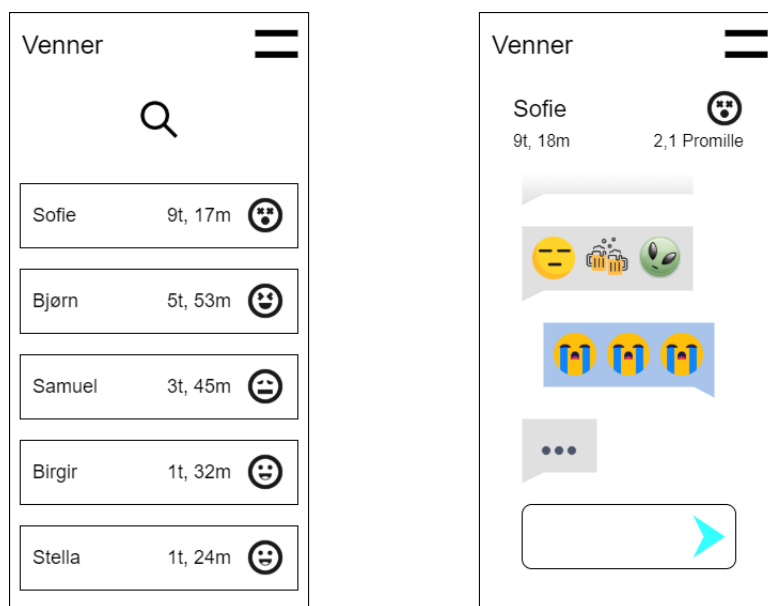


Figure 3: Til venstre er listen av venner. Det er tenkt at brukeren kan søke etter venner ved hjelp av forstørrelsesglasset og kunne sende de en venneforespørsel. Til høyre er en skisse av chat-funksjonen, hvor man kan sende emoji-meldinger til en venn. De kan også svare tilbake, og meldingene vil bli vist på siden sortert etter tid.

3.2.5 Kartside

Appen kommer med en kartside som bruker Google maps API-et. Dersom man er påvirket (En økt har begynt), vil kartet oppdatere hvor man har besøkt i perioden man er påvirket av alkohol. Hensikten med dette er å gi en geografisk oversikt over økten, slik at man vet hvor man har vært mens man drakk. Dette er en funksjon som mange studenter kan ha behov for, siden det ofte hender at de beveger seg rundt når de drikker.

3.2.6 Innloggingsside

Dersom man ikke er innlogget, eller man åpner Lille Promille appen for første gang, møter man en innloggingsside. Appen krever enten en e-mail og passord, eller en google-konto for å logge inn. Har man ikke noen av disse, kan man opprette en ny bruker ved å klikke ”Opprett bruker” lenken på denne siden, da vil man bli ført til en ny side der man skriver ned en e-mail og et passord. Første gang man logger inn vil Appen spørre om navn, kjønn, vekt og høyde. Navnet er viktig i framvisningen av vennesiden, og de tre siste dataene brukes til beregning av promillen i blodet.

3.2.7 Økthistorikk og Individuelle Økter

Ettersom man gjennomgår flere og flere økter, vil de lagres ettersom i en egen historikkside for økter. Her vil man kunne se alle økter sortert etter kronologisk rekkefølge, og man kan inspisere hver individuelle økt ved å klikke på de. Når man gjør dette, åpnes en ny side med informasjon om den spesifikke økten.

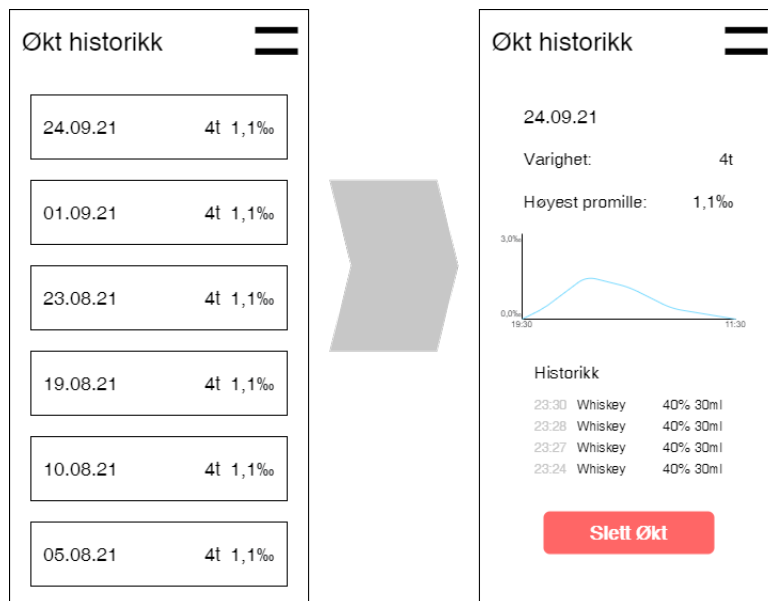


Figure 4: Skisse som viser hvordan økt-historikksiden kan se ut. Her kan man inspisere hver individuelle økt og få utdypet data om hver økt. Det er også mulig å slette en økt.

4 Systemdesign

4.1 Navigasjon

4.1.1 Struktur og oppdeling

Appen består av to hovedelementer for navigasjon, et bottom-navigation element, og et navigation-drawer element. I tillegg til disse har den en swipe-funksjon som gjør det mulig å åpne navigation-drawer. Det vil være mulig å navigere mellom disse tre sidene med bottom-navigation elementet. Det vil være en navigation-drawer tilgjengelig fra alle sider bortsett fra innloggingssiden som brukes til å navigere til resten av appen.

Grunnen til denne oppdelingen er for å skille mellom sider som blir brukt mye og sider som blir brukt lite. Vi har valgt å bruke bottom-navigation og swipe-funksjonen til forsiden, kartsiden og vennesiden, fordi de er sentrale og brukes nesten hver gang appen er i bruk. Forsiden er det samme som oversiktssiden, så dette vil være landingssiden når brukeren logger inn. I navigation-drawer elementet kan man blant annet navigere seg til innstillinger og enhetskatalogen.

4.1.2 Navigation Pattern

Som hurtig navigasjon mellom de viktigste sidene bruker vi bottom-navigation og navigation-drawer. Disse er to gode navigasjons-verktøy fra navigation-pattern som kalles "Literal navigation" siden man velger direkte hvor man ønsker å navigere. Vi har også inkludert forwards navigation i form av knappene i oversiktssiden og vennesiden. For eksempel vil pluss-knappen i oversiktssiden navigere brukeren til drikke-katalogen. Til slutt bruker vi backwards-navigation i form av en pil tilbake som fører brukeren til sist besøkte side.

4.1.3 Navigasjonsflyt

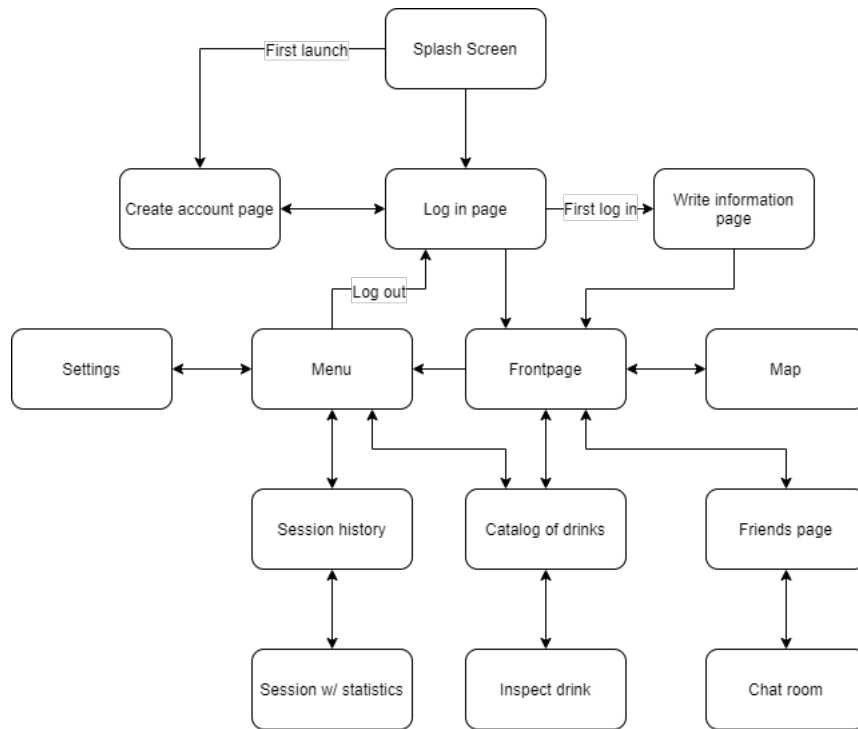


Figure 5: Diagrammet viser hvor man kan navigere seg til fra hvilke sider i appen.

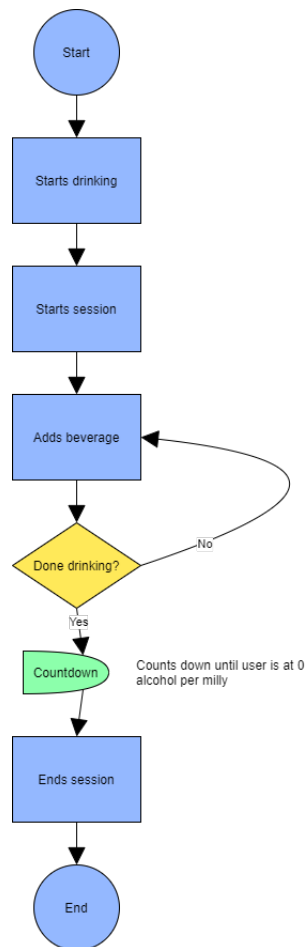


Figure 6: Diagrammet viser hvordan bruker vanligvis vil samhandle med appen.

4.2 Teknisk oppbygging

4.2.1 Services

Det kan være behov for services. Blant annet skal vi implementere gps-sporing og push-notifikasjoner, i tillegg til bør nedtellingsklokken i en økt telle ned mens appen ikke er i bruk. Foreløpig tenker vi at mye kan løses via threads og asynce funksjoner, derfor er vi usikre på om det er behov for services i det hele tatt.

4.2.2 Fragmenter

I vår app er fragmenter et behov. Siden vi bruker Navigation Component biblioteket, er det best praksise å bytte ut sidene via fragmenter. Vi har derfor delt opp appen i tre aktiviteter, hvor vi bytter ut fragmenter i hver av aktivitetene. Dette er både for å gjøre koden mer lesbar ved å abstrahere i logiske inndelinger og for å skille på de forskjellige navigasjonsmetodene, siden bottom-navigation og navigation-drawer kun finnes i hovedaktiviteten.

4.2.3 Lagring og uthenting av data

For å lagre brukerdata benyttes Firestore. Dette gjøres via Firebase, her lagres brukere i en "user" collection. Hver bruker lagres med en UUID som autogeneres fra Firebase.

Brukerinformasjonen lagres rett i dokumentet til brukeren, mens enhetskatalogen og økthistorikken lagres i collections på brukeren.

Når en bruker logger inn hentes dataen til brukeren ved å bruke Firebase UUID'en, da oppdateres først dokumentreferansen så brukes denne til å hente riktig data, brukerdaten blir da lagret in User-DatabaseHandler, denne dataen hentes da senere i settingsFragment.

Når brukeren logger inn gjøres det en sjekk på om det er en ny bruker, hvis det er en ny bruker hentes ikke dataen. Da legges det til en bruker med tomme verdier til database.

Når brukeren navigerer ut av settingsFragment hentes den oppdaterte dataen fra Firebase.

4.2.4 Brukerhåndtering

Håndtering av brukere gjøres gjennom Firebase.

Brukeren får via appen muligheten til å lage en ny bruker, oppdatere og slette brukeren.

En bruker består av feltene:

- age
- height
- weight
- gender
- username
- currentSession
- weight

En sessionHistory collection og en alcoholUnitCollection.

4.3 Ressurser

4.3.1 Klasser

Ekskluderer klasser relatert til aktiviteter og fragmenter, består appen av modell-klasser og referanse-klasser. Modellen består av følgende klasser:

Klassenavn	Beskrivelse
AlcoholUnit	Representerer en enhet som blir lagt til i en økt, arver fra beverage
Beverage	Representerer en drikke
Settings	Håndterer app-innstillinger
Session	Representerer en økt, og har relaterte metoder
SessionLimit	Definerer en grense på hvor mye man ønsker å drikke i en økt (session)
User	Definerer en bruker og inkluderer relaterte variabler

I tillegg har vi klasser for å data fra firestore. Disse ligger i en mappe kalt Datahandler, og inkluderer disse klassene:

Klassenavn	Beskrivelse
BeverageCatalogHandler	Håndterer data relatert til drikke-katalogen i appen
Beverage	Hånterer brukerdata

4.3.2 Bilder

Bilder i appen vil bestå av ikoner, logoer, emoji'er og eventuelt profilbilder. Ikonene brukes til å veilede brukeren der det er raskere å oppfatte ikonet enn å lese teksten, for eksempel til navigasjonspunktene i bottom-navigation og navigation-drawer. Ikonene er en blanding av lånte og egenskapte ikoner som følger design prinsippene til Material Design dokumentasjonen(1). For eksempel at ikonet skal være uthevet og tydelig.

Lille promille logoen har vi designet og laget selv. Det vil også være flere versjoner av logoen til de forskjellige seksjonene i appen der det er passende. For eksempel i sprutskjermen viser den hele logoen, som inkluderer teksten, mens i appikonet bruker den en minimalisert versjon av logoen.

I vennesiden brukes emoji'er til å visualisere hvilken alkoholstatus vennene har. Vi tror dette er en smart løsning fordi mange brukere er allerede familiære med systemet som Snapchat bruker, hvor de forklarer forholdet man har til en venn. Til denne funksjonen bruker Lille promille standardemojiene til android.

Hvis ønskelig kan brukeren legge til sitt eget profilbilde i innstillingene. Dette profilbilde vises både i navigation-drawer ved brukernavnet og i vennesiden ved siden av navnet sitt. Bildet kan hentes fra kamerarullen eller kan tas direkte fra kameraet.

4.3.3 Lyd

Dette har vi ikke tenkt på hittil.

4.3.4 Biblioteker

- Firestore (Firebase) - Dokumentdatabase for lagring av data
- Google dependency - Gir mulighet for innlogging Google-bruker
- Navigation Component - Brukes til all navigasjon innad i appen
- JUnit - Bibliotek for enhets-testing
- Google Play Location - For mobil-sporing

4.3.5 Hva vi er usikre på

Vi må implementere threads på henting av data, dette er vi ganske usikre på hvordan vi skal gjøre. Vi har enda ikke sett noe på vennefunksjonene og kommer nok til å bruke ganske mye tid på dette. Vi har ikke helt funnet ut av hvordan vi skal regne ut den nåværende promillen dynamisk. AlcoholUnit arver også fra Beverage, dette er fordi vi trenger et sted å lagre når en enhet ble lagret i en session. Vi er litt usikre på om dette er den rette måten å implementere dette på. Vi må også finne ut av hvilke services vi skal bruke og hvordan. Vi har ikke fastslått hvordan vi skal implemetere kartet.

5 Krav

- [FR-1] Bruker skal kunne oppgi informasjon om sin vekt, kjønn og høyde**
- [FR-2] Bruker skal kunne opprette en brukerkonto**
- [FR-3] Bruker skal kunne logge inn med eksisterende brukerkonto**
- [FR-4] Bruker skal kunne starte og stoppe en økt**
- [FR-5] Brukeren skal kunne legge til en enhet til nåværende økt**
- [FR-6] Appen skal kunne kalkulere nåværende promille basert på vekt, kjønn og høyde**
- [FR-7] Appen skal kontinuerlig vise nåværende promille**
- [FR-8] Appen skal kunne vise når man kan kjøre basert på nåværende promille**
- [NFR-1] Appen skal opprettholde kontraststandard i forhold til WCAG 2.0**
- [FR-9] Bruker skal kunne slette tilhørende brukerkonto**
- [FR-10] Bruker skal kunne inspisere tidligere økter med vedlagt statistikk**
- [FR-11] Bruker skal kunne slette tidligere økter**
- [FR-12] Brukeren skal kunne legge til sin egendefinerte enhet til den eksisterende listen av enheter**
- [FR-12.1] Brukeren skal kunne bestemme volum, prosentandel og navn på den nye enheten**
- [FR-13] Brukeren skal kunne fjerne enheter fra den eksisterende listen av enheter**
- [FR-14] Appen skal kunne arkivere økter (dvs. antall enheter)**
- [FR-15] Appen skal kunne spore mobilen for hver økt**
- [FR-16] Appen skal kunne loggføre statistikk**
- [FR-17] Appen skal vise nedtellingen på lock-screen på telefonen**
- [FR-18] Bruker skal kunne slå av og på sporing av mobilenheten**
- [FR-19] Appen skal vise en venneliste**
- [FR-20] Bruker skal kunne legge til nye venner i vennelisten**
- [FR-21] Bruker skal kunne sende en emoji-melding til en spesifikk venn**
- [FR-22] Appen skal vise promille-status for hver venn ved siden av navnet deres**
- [NFR-2] Brukeren skal kunne velge mellom darkmode eller lightmode**
- [NFR-3] Brukerene skal varlses dersom drikken de søker etter ikke er i listen**
- [FR-23] Bruker skal valgfritt kunne oppgi alder**
- [FR-24] Bruker skal kunne velge “glemt passord” etter behov**
- [FR-25] Appen skal kunne tilbakestille passord til bruker etter forespørsel**
- [FR-26] Bruker skal kunne endre på alle opplysninger på tilhørende brukerkonto**
- [FR-27] Bruker skal kunne oppgi klokkeslett for når man ønsker å være kjørbar**
- [FR-28] Bruker skal kunne oppgi maksimal promille som ikke skal overstiges**
- [FR-28.1] Appen skal kunne varsle at bruker har nådd maksimal promille**
- [FR-29] Bruker skal kunne oppgi maksimal antall enheter som skal konsumeres**
- [FR-30] Brukeren må samtykke til vilkår for bruk før de kan bruke appen**
- [FR-31] Appen skal kunne detektere om brukeren kjører mens de er påvirket**
- [FR-31.1] Appen skal kunne varsle at bruker kjører**
- [FR-32] Appen skal kunne vise en statusbar som viser brukerens status**
- [FR-33] Appen skal kunne vise hvordan promillen endret gjennom økten seg basert på når en enhet ble konsumert**
- [FR-34] Appen skal kunne varsle at bruker bør drikke vann**

- [FR-35] Bruker skal kunne bestemme om appen skal varsle at de bør drikke vann**
- [FR-36] Bruker skal kunne velge å bli varslet etter hver drukket enhet, eller ved en terskel**
- [FR-37] Appen skal kunne vise åpningstider til barer**
- [FR-37.1] Appen skal kunne finne nærmeste bar**
- [FR-37.2] Brukeren skal kunne ha mulighet til å filtrerer barer på aldersgrenser**
- [FR-38] Appen skal gjøre det mulig å ta bilde av etikken på flasken for å legge til en enhet**
- [FR-39] Brukeren skal kunne se oppskrifter på drinker**
- [FR-40] Brukeren skal kunne sjekke koordinasjonen basert på sensorer**