

19. MAJ 2019

KATRINEBJERG FOR DUMMIES

GRUPPE 13

NAVN:	STUDIENUMMER:	STUDIERETNING
ARMINE SANJARIZADEH	201607125	IKT
EMIL HOLST PETERSEN	201607719	IKT
EMMA ROLSTED	201507335	IKT
FREDERIK KASTRUP MORTENSEN	201607221	IKT

1 INDHOLDSFORTEGNELSE

1	Indholdsfortegnelse.....	2
2	Termliste.....	3
3	Ansvarsområder og Tidsplan	3
4	App Vision.....	3
5	Personlig vision.....	3
6	Kontekst.....	4
7	Krav (User Stories).....	4
8	Design Overblik og implementering.....	5
8.1	Room Database	5
8.2	Google Maps.....	5
8.2.1	Implementering	6
8.3	Vejret	6
8.3.1	Implementering	6
8.4	WIFI.....	7
8.4.1	Implementering	7
8.5	Login	7
8.5.1	Implementering	8
8.6	Checkliste.....	8
8.6.1	Implementering	8
8.7	FOOD og Buildings.....	8
8.7.1	Implementering	8
8.8	FAQ	8
8.8.1	Implementering	9
9	Diskussion	9
9.1	Kendte bug.....	9
10	Konklusion	9

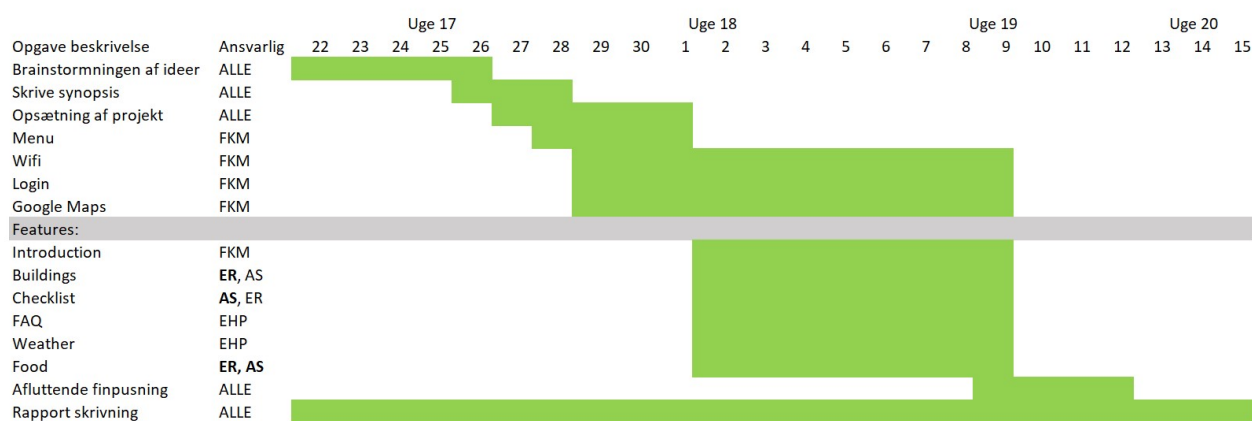
2 TERMLISTE

Term	Beskrivelse
Omgivende faciliteter	Bygninger, parkeringspladser, spisesteder
Captive Portal	Login portal til Wi-Fi
Bygninger	Edison, Shannon, Nyggard, Kahn

3 ANSVARSOMRÅDER OG TIDSPLAN

Hvert gruppemedlem har haft forskellige feature som skulle laves, således arbejdet kunne foregå parallelt.

Navn:	Område:
Armine Sanjarizadeh (AS)	Checkliste feature
Emil Holst Petersen (EHP)	FAQ, Weater feature
Emma Rolsted (ER)	Building- og food-festure
Frederik Kastrup Mortensen (FKM)	Login, WiFi, Menu, Google Maps, Introduction



4 APP VISION

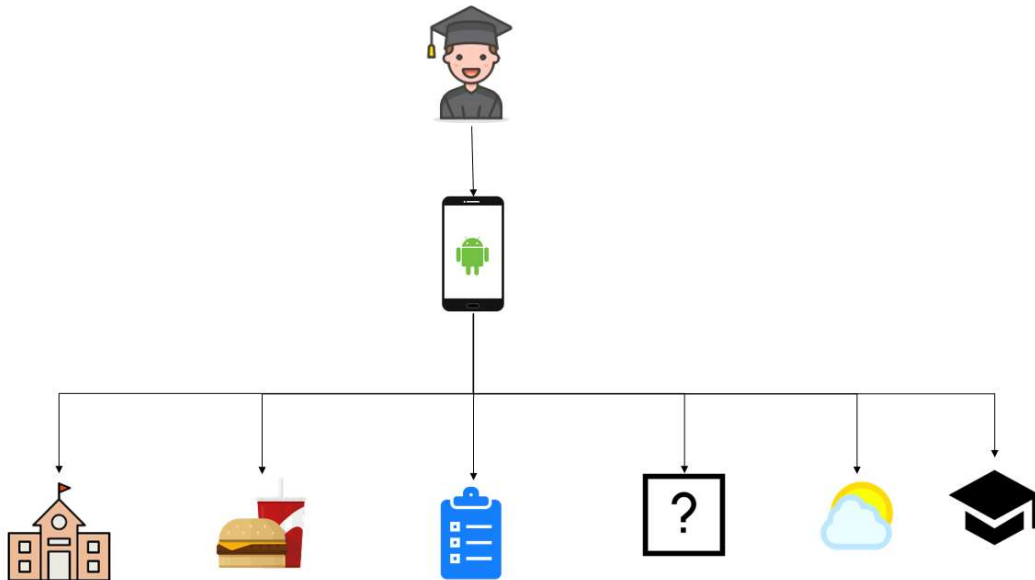
Appen Katrinebjerg For Dummies, løser problemstillingen alle nye studerende på Aarhus Universitet Ingeniørhøjskolen for stået med i de første uger af studietiden. Hvem er studievejlederen? Hvor er de nærmeste spisesteder? Appen skal svare på de mest almindelige spørgsmål, nye studerende i Katrinebjerg kunne have. Det er nemt at glemme de mange ting man skal have klar til studiestart, derfor er det godt med en huskeliste, der minder dig om de ting du mangler at have klar.

5 PERSONLIG VISION

Vi håber på at blive bedre til at strukturere en større app, med mere end bare en feature. Her vil det være en øvelse i at få tildelt en feature man skal arbejde på, parallelt med de andre gruppemedlemmer. Samtidig forventer vi at få ny kendskab til teknologier, vi ikke har så meget erfaring med.

6 KONTEKST

På nedenstående figur kan et rigt billede af appen og brugeren ses.



Figur 1 - Rigt billede

Brugeren har via sin telefon mulighed for at vælge forskellige menu muligheder og dermed få vist alt nødvendig information omkring studiestart.

7 KRAV (USER STORIES)

Som bruger skal jeg kunne bruge Google Maps til at finde information omkring de omgivende faciliteter.

Som bruger skal jeg gennem Google Maps kunne finde yderligere information, ved at blive henvist til andre skærme i appen.

Som bruger skal jeg kunne logge ind med mit eget brugernavn og password.

Som bruger skal jeg kunne logge ud.

Som bruger skal jeg kunne tilkobles skolens Wi-Fi med en knap i appen.

Som bruger skal jeg kunne se en skærm med informationer omkring spisemuligheder på Katrinebjerg.

Som bruger skal jeg kunne tilgå en FAQ-side.

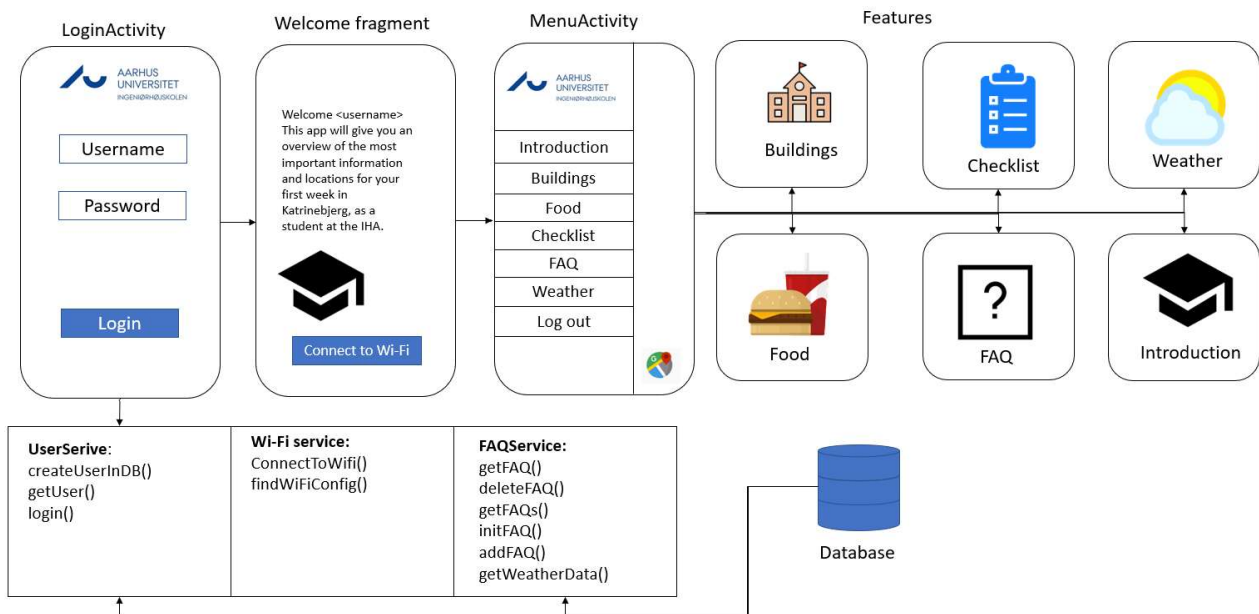
Som bruger skal jeg kunne tilgå en vejruddsigtside, hvor den nyeste vejruddsigst skal være tilgængelig.

Som bruger skal jeg kunne tilgå en huskeliste som minder brugeren om at huske vigtige ting til første dag.

Som bruger skal jeg kunne tilgå en bygningsoversigt, hvor relevante bygninger beskrives.

8 DESIGN OVERBLIK OG IMPLEMENTERING

Diagram over de største komponenter og hvorledes disse kommunikerer. Hver features skal implementeres som et fragment, således at menulisten altid er tilgængelig for brugeren. Derfor er det nødvendigt med en menu som kan navigere mellem disse.



Figur 2 - App oversigt

UserService og FAQService skal være en bound service, så den aktivitet som skal anvende servicen skal blot binde til den. I denne service skal der ligeledes køre en asynkron Task som skal sende notifikationer ud til brugeren.

8.1 ROOM DATABASE

Det som skal gemmes i databasen, er brugerens navn, password samt checklisten. Checklisten indeholder listen fra Checklist fragmentet som brugeren kan tjekke af på alt efter som disse ting er ordnet. Det betyder derfor at listen skal være opdateret og gemmes alt efter brugerens ønske.

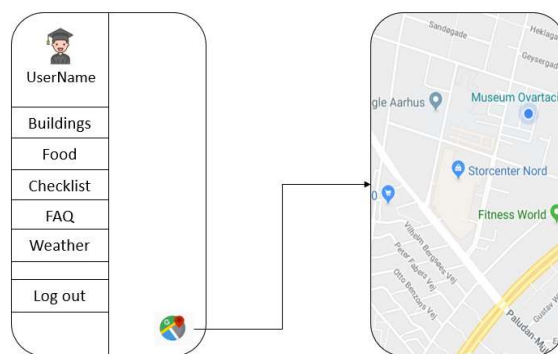
Derudover skal brugeren have mulighed for at slå notifikationerne fra, denne værdi skal derfor også kunne persisteres til databasen.

Der kunne argumenteres for at brugerens password skulle hashes og saltes, dette er nedprioriteret da brugeren udelukkende er lokalt på telefonen og at vi ikke behandler noget følsomt data.

FAQ-indslag er også gemt i databasen. Disse persisteres ved hjælp af FAQ-servicen, og servicen bruges til hele tiden at få vist de nyeste informationer i FAQ-fragmentet.

8.2 GOOGLE MAPS

Google Maps knappen vil altid være tilgængelig for burgeren og hvis denne trykkes vil et kort over de omgivende faciliteter vises.



Figur 3 - Google Maps activity

I Maps aktiviteten, starter kortet ved Katrinebjerg, hvor brugeren straks kan se de mest vigtige bygninger, spisesteder, fredagsbarer og p-pladser. Her har brugeren mulighed for at klikke på et vilkårligt ikon, hvor der hermed vises et custom view i bunden af skærmen, med yderligere information. Hvis en bruger har markeret et bygnings- eller madikon og der trykkes på dette custom view, vil brugeren deep linkes til henholdsvis, bygnings og mad fragments.

8.2.1 Implementering

Google Maps skærmen er en aktivitet der indlæser en Map fragment. Her er der brugt Googles Api. For at gøre brug af Googles Api skal vi generere og registrere en gratis API nøgle. Aktiviteten venter på et map ready callback, som fortæller os hvornår kortet er indlæst. Herefter indlæses marker data fra en csv fil kaldet mapdata, der kan findes i assets. Her laves der model objekter der bindes til map markers.

Selve layoutet består af map fragmenten og et custom view i bunden af skærmen. Dette custom view vises og skjules når brugeren henholdsvis trykker på et ikon på kortet og trykker et andet sted på kortet. Viewet fylder ud med informationer fra modellen bundet til en marker.

8.3 VEJRET

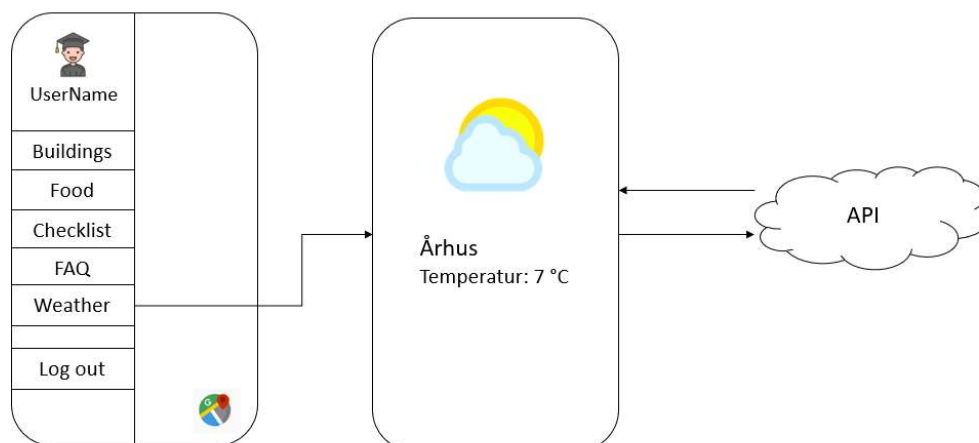
Brugeren skal have mulighed for at tjekke den lokale vejrudsigt. Dette skal være muligt igennem et API.¹

Eftersom denne app kun er aktuel i Aarhus, og mere specifikt Katrinebjerg, har vi valgt at vise data for Aarhus. Vi har valgt at vise hvilket vejr det er, hvad den nuværende temperatur er, samt dagens højeste og laveste temperatur.

8.3.1 Implementering

For at få information om vejret har vi brugt API'et fra OpenWeatherMap. Dette API kan vise en lang række information om vejret i den by man specificerer. Kaldet til API'et er lavet som en del af FAQ-servicen. Der forespørges data om Aarhus vejr, og det vises i Weather-fragmentet. Når der er svar fra servicen bliver et weather-objekt som ligger servicen opdateret, og dette kan hentes fra weather-fragmentet. Servicen broadcaster at objektet er blevet opdateret, og herefter henter weather-fragmentet det nyligt opdaterede objekt.

¹ <https://openweathermap.org/>



Figur 4 - Weather Fragment

8.4 WIFI

Når brugeren er logget ind, skal det være muligt for introduktions siden at gå på skolens WI-FI. Her er der valgt at tilslutte til netværket "AU-Guest". Dette er grundet den kryptering og login credentials der skal til for at tilslutte Eduroam. Brugeren får her en meddelelse i form af en snackbar når der er tilsluttet netværket. Da AU-Guest er et åbent netværk og offentligt netværk, er der sat en captive portal på som brugeren skal interagere med når der er tilsluttet. Dette er ikke noget der åbnes automatisk og brugeren skal selv gøre det, når browseren åbnes. Ideelt skete dette automatisk, men grundet tidspres, blev det ned prioriteret.

8.4.1 Implementering

Wifi tilslutningen er implementeret med brug af WifiManager, Wificonfiguration og network callback. Servicen startes med et SSID, sendt med som intent. Der tjekkes herefter om der er konfigureret WIFI med matchende SSID. Er dette tilfældet, forsøges der at skabe forbindelse til netværket, ved at tilslutte det netværksid der er bundet til konfigurationen. Hvis brugeren ikke har været på netværket før, skal der her laves en konfiguration. Der laves her et nyt WifiConfiguration objekt, som SSID'et assignes til. Det er også vigtigt at fortælle WifiManageren at den ikke skal håndtere noget kodeord på netværket. Herefter tilføjes konfigurationen til WifiManageren og der prøves at tilslutte netværket.

For at give brugeren respons angående en tilsluttet, bruges der network callback. Dette er kun tilgængeligt for API level 24 og opefter. Der tjekkes her for en wifi forbindelse samt om SSID'et stemmer overens med det forventede. Grundet tidspres, er der ikke implementeret et alternativ for API levels lavere end 24. Resultatet broadcastes til introduktions fragment.

8.5 LOGIN

I loginaktiviteten skal brugeren have mulighed for at logge på med sit ønskede brugernavn og password. Der er undladt at lave et sign up view og der oprettes blot en ny bruger, hvis brugernavnet er frit. Grunden til dette er, at loginfeaturen ikke er prioriteret højt nok, i forhold til de andre features og det er vurderet at det ikke vil påvirke brugerens oplevelse af appen tilstrækkeligt nok, til at skære andre features fra. Brugeren føres videre til hovedmenuen, hvor han har adgang til de forskellige features. Brugeren bliver meddelt at han har skrevet et forkert kodeord til et eksisterende brugernavn, hvis dette er tilfældet.

8.5.1 Implementering

Loginaktivitetens funktionalitet er opbygget af en baggrunds service, der startes når aktiviteten startes. Herefter bindes der til den. Når brugeren har udfyldt kodeord og password feltet, kan der herefter logges ind. Her laves der et kald til databasen, som tjekker om brugeren allerede eksisterer og logger herefter brugeren ind, hvis dette er tilfældet. Hvis brugeren ikke eksisterer, oprettes der et nyt User objekt i databasen og der logges herefter ind. Resultaterne fra disse kald, broadcastes til loginaktiviteten. Alle database er lavet asynkron med tråde og runnables.

8.6 CHECKLISTE

I checkliste fragmentet, som tilgås via menuen, vises en liste over relevante punkter som en ny studerende skal huske inden studiestart. Her har den studerende mulighed for huske at have en huskeliste, som gør det overskueligt at starte på en ny uddannelse.

8.6.1 Implementering

Checkliste fragmentet er opbygget af et listview med et textview og en checkbox, så brugeren kan krydse af når genstandene er husket. Disse bliver udfyldt via en adapter, som bruges til at sætte data op. De genstande der ikke er krydset af, vil blive oplyst til brugeren via notifikationer, som servicen opretter. Fragmentet er bundet til en baggrunds service, userService, der udover at notificere brugeren også opdaterer brugerens checklisteitems i databasen.

8.7 FOOD OG BUILDINGS

I food og buildings fragmenterne, som tilgås via menuen, har den studerende mulighed for at orientere sig og få en oversigt over omkringliggende spisesteder og bygninger. Herfra er det muligt at tilgå disse steders hjemmeside for mere information.

8.7.1 Implementering

Disse to fragmenter er implementeret på samme måde, med en viewpager som giver brugeren mulighed for at swipe igennem de forskellige bygninger og spisesteder. Viewpager er opbygget af Cards, hvor disse cards indeholder de forskellige felter, som vises til brugeren. Der anvendes en viewPager adapter for at udfylde siderne (pages) inde i viewpageren. Hvert card indeholder en knap som tager brugeren videre til en hjemmeside, via intents, som starter en web aktivitet. I toppen af disse fragmenter er der implementeret et Tablayout, som derefter sættes op med viewpageren, på denne måde kommer titlerne fra de forskellige kort ind som overskrift i fanen.

For at gemme staten efter brugeren rotere eller forlader fragmentet gemmes viewpager positionen med Shared Preferences.

8.8 FAQ

I FAQ-fragmentet skal brugeren kunne se række ofte stillede spørgsmål. Vi har valgt at der er 7 til at starte med, når man første gang åbner appen. Det er muligt at tilføje nye spørgsmål, og man er krævet et svar på spørgsmålet også. Tanken her er, at hvis appen skulle releases ville alle der bruger appen kunne tilføje ny information. FAQ-elementer kan slettes ved at swipe fra højre side. Slettes alle elementer i fragmentet, vil de 7 startsspørgsmål blive hentet igen.

8.8.1 Implementering

FAQ-servicen er implementeret så den kan snakke sammen med room-databasen, hvor alle FAQ-elementer gemmes. Det er altså gennem denne at brugeren kan tilføje og slette elementer. Servicen broadcaster resultater hvis der er sket noget, som applikationen skal opdatere sit view baseret på.

FAQ-fragmentet er implementeret som et recyclerview, hvor elementer tilføjes og slettes løbende ved hjælp af en adapter. Det er også her der er implementeret funktionaliteten til at kunne slette og tilføje nye elementer ligger. Tilføjelse af et nyt element gøres ved hjælp af en knap. Ved tryk på knapper vises et nyt fragment, hvor brugeren kan indtaste et spørgsmål og et svar. Hvis enten spørgsmålet eller svaret ikke er indtastet, vil elementet ikke blive tilføjet.

9 DISKUSSION

På nuværende tidspunkt logger appen på AU-guest wi-fi, men optimalt ville være at man som studerende kunne anvende sine AU-loginoplysninger og dermed logge på eduroam.

FAQ universel liste således de studerende kan have en samlet liste for alle, så man kan dele erfaringer og problemstillinger. Derfor skal det være muligt når man logger ind at tilmelde sig sit semester. På samme skal en tutor have muligt for at svare på nogle af de problemstillinger som de studerende kan tilføje til listen.

I Checklisten ville det ligeledes være optimalt hvis den studerende havde mulighed for selv at tilføje til denne liste, så man kunne samle hele sin huskeliste før skolestart, både personlige ting og vigtige ting fra skolens side.

Potentielle udvidelser der havde været ideelle, men som ikke er en del af dette kursus, vil være en rigtig backend. Meget af den information der findes i appen er enten gemt i csv filer eller blot som localized strings.

En anden som havde været smart ville være at den studerende kunne have adgang til en profil feature så det er muligt for den studerende at tilpasse sin profil med informationer.

9.1 KENDTE BUG

Her er en liste over kendte bugs i appen:

- Når appen dør, kommer en notifikation op med det samme, dette burde ikke ske.
- Mangler loaders til billeder (placeholder), så brugeren kan se at der burde komme et billede, men at dette er i gang med at load.
- Det er først når huskelisten åbnes at brugeren begynder at få notifikationer, dette brude ske ligeså snart brugeren logger ind.

10 KONKLUSION

Det er lykket os at implementere en app, til at hjælpe de nye ingeniørstuderende med at få en god studiestart. Alle planlagte og ønskede features er implementeret efter hensigten. Derudover er det lykket os at implementere og strukturere en større app hvor alle gruppe har bidraget til opgaven.