

Rombestillings app, Apputvikling Mappe 3

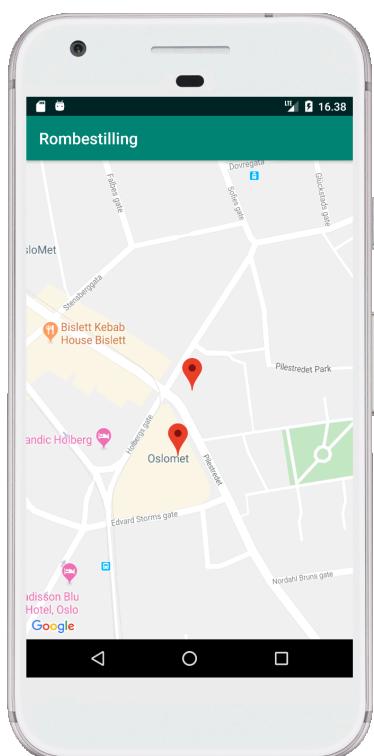
Innledning

Denne oppgaven går ut på å lage en app for rombestilling, hvor brukeren kan selv legge til markører for bygninger, som kan inneholde rom. For hvert rom kan man også registrere bestillinger eller «booke» rom. Appen jeg har utviklet «Rombestilling» tar for seg dette ved å benytte en database og PHP-scripts på studentserveren.

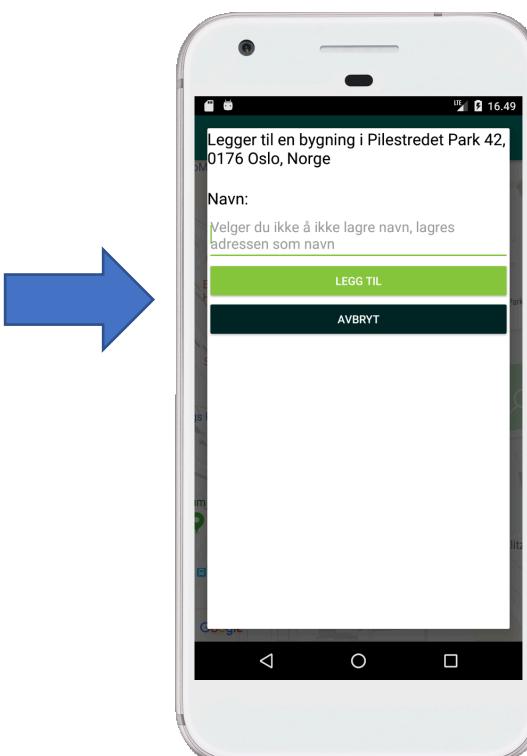
Appens funksjonalitet og diverse klasser

MainActivity

Når appen kjøres starter MainActivity et MapFragment som viser kartet. Her kan brukeren se markører for byggene som er lagret. Ved å trykke på et sted på kartet hvor det ikke er en markør kan man legge til et nytt bygg. Da vises et DialogFragment hvor man kan legge til en bygning. Her hentes addressen fra WebHandler klasses som har en Task som henter adressen fra GeoCoder APIet. Dette skjer asynkront, ettersom appen ikke burde fryse til adressen er hentet.



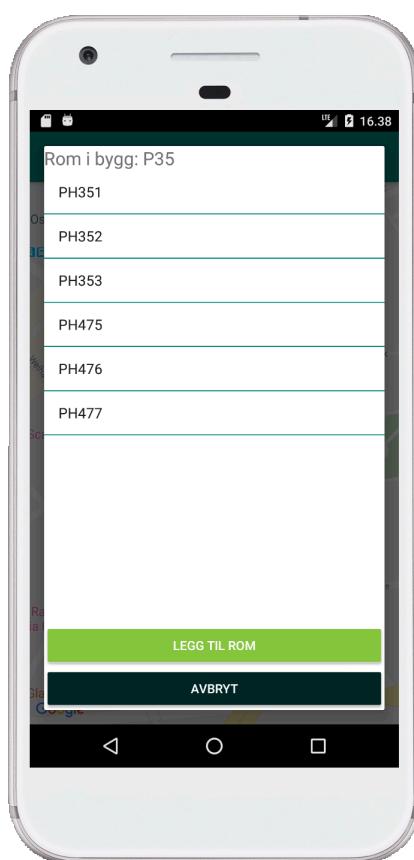
Figur 1 MainActivity med MapFragment



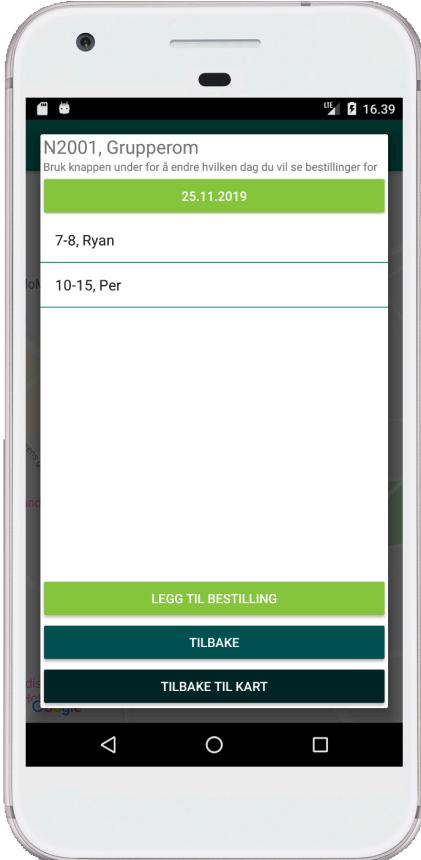
Figur 2 AddBuildingFragment

BuildingInfoFragment, RoomOrdersFragment, AddOrderFragment, OrderInfoFragment og AddRoomFragment

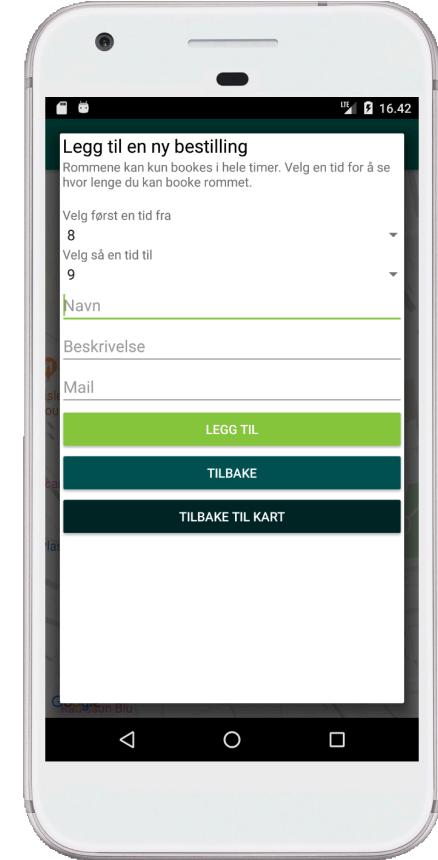
Trykkes det på en markør vises DialogFragmentet BuildingInfoFragment, som viser rommene i bygget. Disse vises i et ListView. Trykker man på et rom vises DialogFragmentet RoomOrdersFragment. Her inne kan man se alle bestillingene for rommet man valgte, sortert etter tid. Ved å benytte knappen øverst kan man endre hvilken dag man vil se bestillingene for. I dette fragmentet har man også muligheten til å lukke alle de åpne fragmentene ved å bruke «tilbake til kart» knappen. Denne «dismisser»/lukker da alle åpne DialogFragments.



Figur 3 BuildingInfoFragment

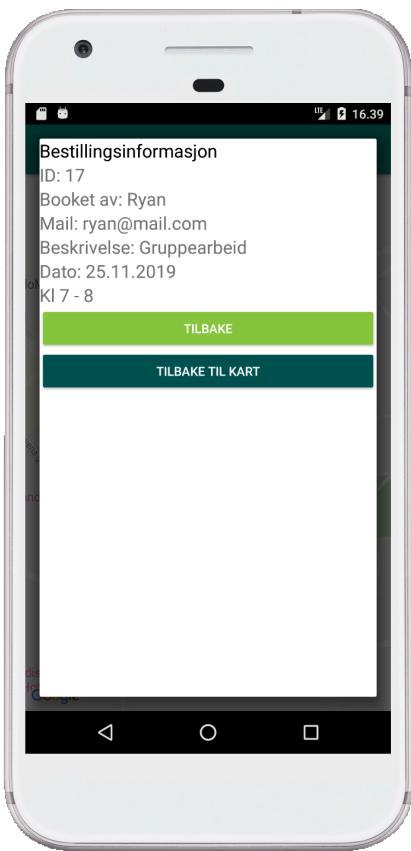


Figur 4 RoomOrdersFragment

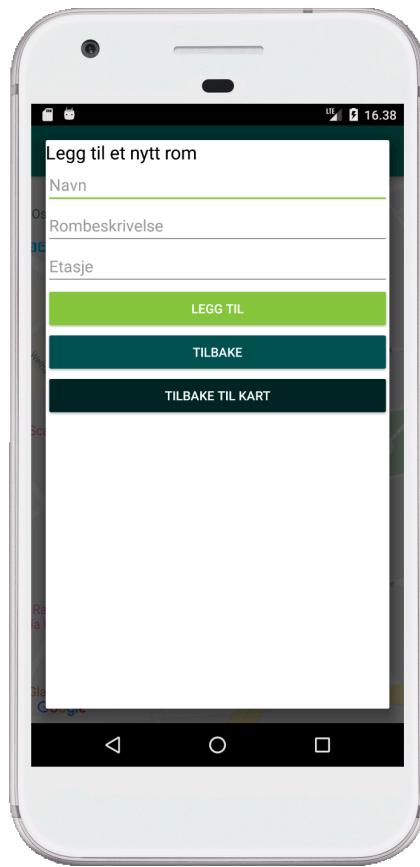


Figur 5 AddOrderFragment

Ved å trykke på «legg til bestilling» knappen i RoomOrdersFragmentet vises fragmentet AddOrderFragment. Her inne vises to «Spinners» hvor den første spinneren er hvor man velger tid fra og den andre tid til. Brukeren kan kun booke rom i hele timer og fra klokken 7 til 22. Tidene i spinneren til «tid til» populeres etter hvor mange timer som er ledige frem til neste bestilling. Trykker man på en av bestillingene i ListViewet i RoomOrdersFragmentet vises alle detaljene om bestillingen i fragmentet OrderInfoFragment som vist i figur 6.



Figur 6 OrderInfoFragment

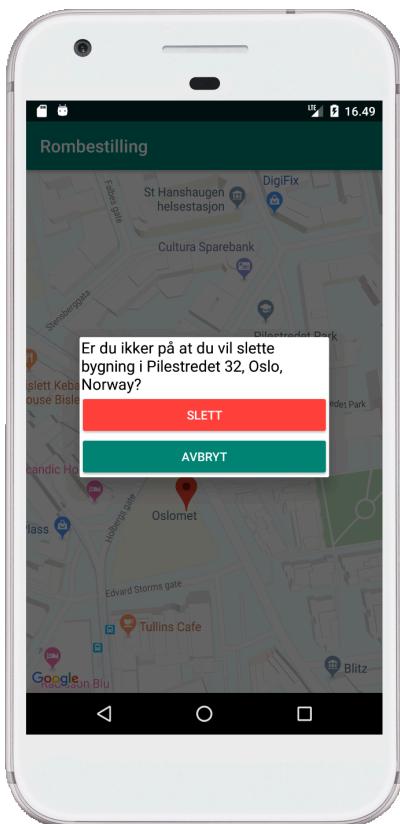


Figur 7 AddRoomFragment

I BuildingInfoFragmentet kan man også legge til et nytt rom ved å trykke på «Legg til rom» knappen. Her dukker da opp fragmentet AddRoomFragment som vist i figur 7.

DeleteBuildingFragment, DeleteRoomFragment og DeleteOrderFragment

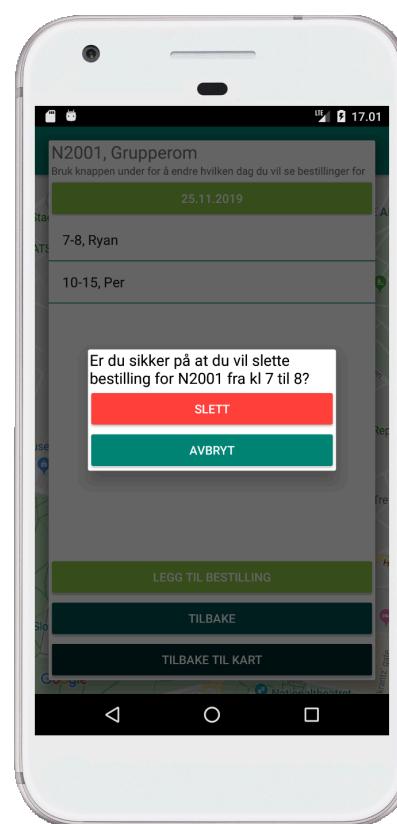
«DeleteFragmentene» ser slik ut og sletter både lokale objekter og radene i databasen på serveren.



Figur 8 DeleteBuildingFragment



Figur 9 DeleteRoomFragment



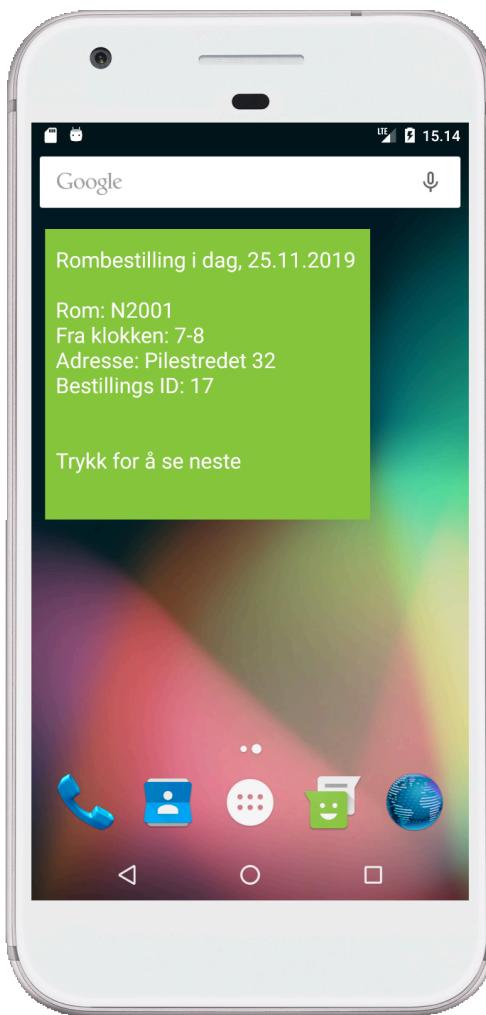
Figur 10 DeleteOrderFragment

WebHandler og Async interfacene

Klassen WebHandler tar for seg alt av funksjonalitet relatert til webtjenestene på serveren. WebHandler inneholder Tasks for å utføre innlegging og sletting av bygninger, rom og ordre. Den inneholder også en Task for å hente data fra serveren, og oppretter da ArrayLists av typen Building, Room og Order som brukes av MainActivity og alle fragmentene. Alt dette skjer asynkront. Derfor har jeg brukt interfacet AsyncResponse, som MainActivity implementerer. Når WebHandler er ferdig med Tasket GetJSON, kalles da funksjonen ProcessFinish. I MainActivity sin implementasjon av dette interfacet, settes da markørene til bygningene på kartet. I tillegg settes alle relasjoner mellom bygning, rom og ordre, slik at jeg slipper å kalle på webtjenestene hver gang jeg skal bruke dataene. AsyncAddressResponse fungerer på samme måte, og blir brukt når GeoCoderen er ferdig med å hente adresse fra LatLng objektet. På denne måten kan man trykke på kartet for å legge til et nytt bygg og få opp AddBuildingFragmentet, uten å fryse UI tråden. Adressen settes da i TextViewet txtAddBuildingAddress asynkront. Jeg bruker også AsyncPostBuildingResponse, AsyncPostRoomResponse og AsyncPostOrderResponse for å lytte etter når post taskene er ferdige og sette riktige ID'er til de nye objektene.

OrdersWidget

OrdersWidget er en widget som viser dagens rombestillingene. Det er et TextView som får verdien sin fra en liste som populeres med detaljer om dagens bestillinger. Jeg prøvde meg på å bruke et ListView istedenfor, men dette viste seg å være mye mer utfordrende enn forventet. Derfor har jeg det slik at hvis man trykker på widgeten, vises neste bestilling som er et String objekt. Dette skjer ved å øke telleren i OrdersWidget og så sette neste bestilling fra lista. Det eneste kjipe ved dette er at jeg ikke fikk til å hente dataene fra serveren asynkront. Dette gjør at appen bruker litt lengre tid ved oppstart for å sette lista med detaljer om dagens bestillinger.



Oppsummering

Jeg er veldig fornøyd med sluttproduktet og måten jeg håndterer kall på webtjenesten på studentserveren. Appen lar deg enkelt legge til nye bygninger, rom og ordre. Alt dette skjer også asynkront, som jeg også er veldig fornøyd med. Likevel skulle jeg ønske at jeg hadde litt mer tid til å gjøre kallene mot serveren fra Widgeten asynkront, og eventuelt bruke et ListView istedenfor et TextView. Bortsett fra det mener jeg at dette er en komplett app som gir brukeren funksjonalitet som beskrevet i oppgaveteksten.