

Mappe 3 Apputvikling 2020

Deltagere:

Tobias Strøm: s331392

Karol Turlej: s344193

Rapport

Applikasjonen

Applikasjonen er et hjelpemiddel for organisering av grupperom reservasjon på Universitetet. Dette gjøres ved at brukeren har mulighet til å reservere en bestemt time, i et bestemt rom på et bestemt hus. Brukeren har også oversikt over hvilke rom som er opptatt eller ikke. Samtidig er applikasjonen en hybrid løsning, hvor den samme applikasjon tilbyr administrator verktøy. Det vil si at det er mulig å opprette hus og rom ved å benytte seg av samme applikasjon. Applikasjonen gjør det enklere å holde rede på grupperom reservasjon og bygninger.

Hoved skjerm

Applikasjonen viser et kart på åpning. Det er et kart som zoomer inn på OsloMets campus med kartmerknader som tilsvarer alle registrerte bygninger inn i databasen. Dersom det ikke er registrert noen bygninger, vises det et tomt kart. For å få mer informasjon om huset, eller se husets rom, kan brukeren trykke på kartmerknaden. Det vil komme opp et boks på skjermet med husets detaljer som navn, adresse, antall etasjer, åpningstider og beskrivelse. Det er en knapp «rom» som åpner et nytt vindu med rommene i det aktuelle huset. For å opprette et hus, må brukeren holde på plassen huset skal ligge. Det vil komme opp en likende boks opp hvor brukeren kan tilføre detaljer. Det er ikke mulig å registrere et hus uten en gyldig adresse.

Vi har også tatt med muligheten får å kunne registrere åpningstid på bygningen. Det gjør at det ikke er mulig å booke er rom på et tidspunkt hvor bygge er stengt og det ikke er noe “ansvarlige i bygge”.

Rom skjerm

Det er en skjerm som viser alle registrerte rom knyttet til et bestemt hus. Dersom det ikke er registrert noen rom, vil listen være tom. Hvert rom presenteres som et boks i en liste med rommens detaljer opp på den. Om rommene vises det informasjon som romnummer, kapasitet, etasje den ligger på og beskrivelse. For å se reservasjoner for rommet, kan brukeren trykke på det aktuelle rommet. Det er også mulig å registrere nye rom. Dette skjer ved at brukeren trykker på «+» ikonet på verktøylinje. Slik vil det vises et registreringsskjema hvor brukeren kan skrive inn rommets detaljer.

Reservasjon skjerm

Reservasjon skjermen gir oversikt over reservasjoner for et bestemt grupperom. Det vises reservasjoner og ledige timer som kan reserveres for en bestemt dato. Reservasjonstimer presenteres i en liste hvor hver enkel time vises i form av et boks. På boksen står det start og slutt tidspunkt knyttet til bestemt reservasjonstimen. Rommene er satt opp i times intervall og man kan booke så mange man ønsker. Reserverte timer markeres med rød skrift og det er ikke mulig å reservere dem flere ganger. Dato for reservasjon endres ved hjelp av pilknappene under verktøylinje. Antall timer som er tilgjengelig for reservasjon hver dag er avhengig av huset åpningstider.

Utsende

Vi har valgt et veldig enkelt og "kjedelig" stil. For hovedfokus til en administrator/student er ikke at det skal være masse farger som gjør at de mister fokus. Vi har fokuser med på brukervennlighet og effektivitet. Istedenfor at man skal måtte gå inn å skrive når man ønsker å ha rommet fra og til så har vi bare bokser som man trykker på. Det gjør det veldig enkelt å reservere rom. Vi har valgt en mørk lilla farge som går igjennom hele appen.

Navigasjon

Navigasjonen foregår i kombinasjon av applikasjonens og Android knapper. Navigasjonen forover håndteres av applikasjonens knapper derimot navigasjonen bakover skjer via Androids nativ tilbake knapp. Et slik tilnærming hjelper å holde brukergrensesnittet enklere, dersom det er ikke nødvendig å implementere en tilbakeknapp inn i applikasjonen. Dette er også et navigasjonssystem som er brukt i andre applikasjoner slik at brukeren behøver ikke å lære seg navigasjon i denne applikasjonen. Det er ikke noe varsel om at informasjon blir borte i input feltene når man oppretter et objekt. Dette er fordi informasjonen man skriver inn er så kort og det går fort å skrive den inn på nytt.

System dokumentasjon

Database struktur

Databasen er sammensatt av 3 tabeller. Det fantes Hus, Rom og Reservasjon tabell. Hus tabell inneholder informasjon om hvert hus samt et hus id. Rom tabell inneholder informasjon om rom i tillegg til en rom id og id til huset den tilhører til. Reservasjon tabell inneholder informasjon om reservasjonen og id til rom reservasjon er knyttet til.

Webtjenester

Webtjenester knyttet til database håndteres av et selv dannet API. Alle mulige spørringer som kan utføres mot databasen er beskrevet i dens dokumentasjon. Dokumentasjonen befinner seg under denne lenken:

<http://student.cs.hioa.no/~s344193/AppApi/documentation.html>

Det fantes også mulighet for debug håndtering av data (debug utskrift + sletting):

<http://student.cs.hioa.no/~s344193/AppApi/debug.php>

Kart

Kartfragmente er et fragment som er inkludert i Android studio. Får at det skal funke må man opprette en nøkkel som er koblet opp mot google sine servere. Dette er en sikkerhets mekanisme i google sine systemer som gjør at de ha kontroll på hvem som bruker systemene deres.

Hente data i applikasjonen

Får at det skal være mulig å kunne hente ut data fra webserveren så bruker vi APIene som vi har laget. APIene sender over JSON fil (ved suksess) som vi kan hente dataen vi trenger. Det første vi gjør er å lage en privat klasse i den klassen vi trenger i. Den private klassen utvider AsyncTask som er en metode som gjør at man kan gjøre oppgaver i bakgrunnen av appen, uten at den krasjer.

Først opprette vi en HttpURLConnection. Det er det objektet som tar seg at hvilke type RequestMethod vi ønsker og hva vi forventer å få tilbake i RequestProperty. Så sjekker det om det ble en vellykket kobling. Så opprettes det et BufferedReader objekt som tar inn informasjonen den har fått inn i fra HttpURLConnection og leser den linje for linje.

Så blir det opprette et JSONArray som tar informasjon fra BufferedReader og konverterer det til et array av JSON Elementer. Så ut ifra hvor mange elementer vi ønsker å få ut så kan vi bruke en for-loop til å hente ut alle elementene eller bare sette at vi ønsker element nummer en. Og nå vi har et JSONObject så kan vi hente ut den informasjon vi ønsker å bruke den der vi trenger den.

Recyclevieweradapter (RVA)

Vi bruker en RVA to ganger i denne oppgaven. Den ene er for å liste ut rom og den andre er for å liste ut møtene. RVA gjør at vi kan håndtere en liste med elementer å få dem til å se ut som vi ønsker. Det blir da også mulig å kunne trykke på elementene å få dem til å gjøre det vi ønsker.

Inputvalidering

Det er input validering på all input feltene som det passer å ha det på. Det gjør at det ikke blir noe problem med sql angrep eller lignende. Og at informasjon som blir lagt inn i databasen kan bli lest og tolket sånn at programmet ikke krasjer eller gjør noe vi ikke ønsker.

Møte listen

Vi hadde hele tiden lyst til å ha en liste hvor man trykker på det tidspunktet man ønsker å reservere uten å måtte skrive inn noe klokkeslett. Det var en av de vanskeligste oppgavene i prosjektet.

Vi bruker API får å hente ut møtene som er booket for et spesifikt rom og et spesifikt tidspunkt. De blir lagt inn i et array som vi har kalt “selectedMeetings”. Det er også lagt til et felt som heter “selected” som forteller om timen er reservert eller ikke. Det felte blir ikke lagret i databasen siden alle timene som blir lagret i databasen er reservert. Alle elementene i “selectedMeetings” har derfor selected som true for de er booket.

Det er i metoden buildList (i MeetingsActivity) vi lager listen som skal vises. Den starter med en for-loop som går fra åpningstid til stengetid. I loopen så blir det opprettet et meetingA objekt hvor selected verdier false. Videre blir dette objektet sammenlignet med alle elementene (meetingB) i selectedMeetings. Hvis meetingA finnes i selectedMeeting så blir meetingA byttet ut med meetingB. Når det er gjort så blir det objektet lagt inn i meetingList som er listen som blir skrevet ut. Dette gjør den så mange ganger den trenger får å få dekket hele åpningstiden til bygningen. Så når RVA går igjennom meetingList så ser den om noen av meeting objektene som har selected som true og da sette fargen på den boksen til rød.

Så er det to piler oppe på siden (figur 10 og 11) som gjør at man kan bytte mellom dagene. Listen blir da oppdatert med den nye informasjonen og bruken av appen kan booke rom på hvilken dag han skulle ønske.

GitHub

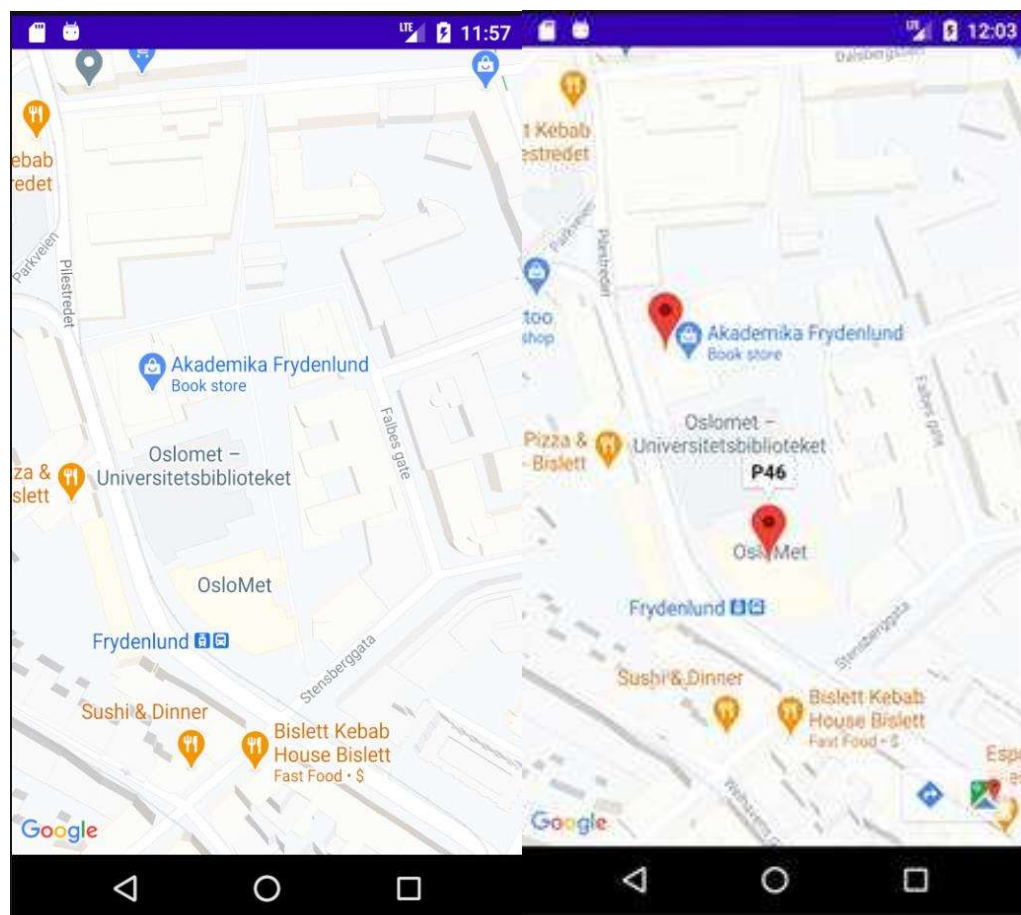
Vi har brukt GitHub jevnt over hele prosjektet. Det gjør av selv om vi sitter hver for oss så ha vi alltid tilgang på den nyeste koden til prosjektet. Vi kan også jobbe i hver vår Branch som gjør av vi kan jobbe med det samme og finne ut hva som er best. Når vi har funnet ut hva som er best så kan den merges inn i master Branchen. Det andre finne med GitHub er versjons kontroll. Det gjør at uansett hvilke endringer du har gjort så har man alltid tilgang på tidligere versjoner. Så hvis man ønsker å hente noe man har gjort før som man ikke trodde man trengte så ligger det klart til bruk.

Kjente feil

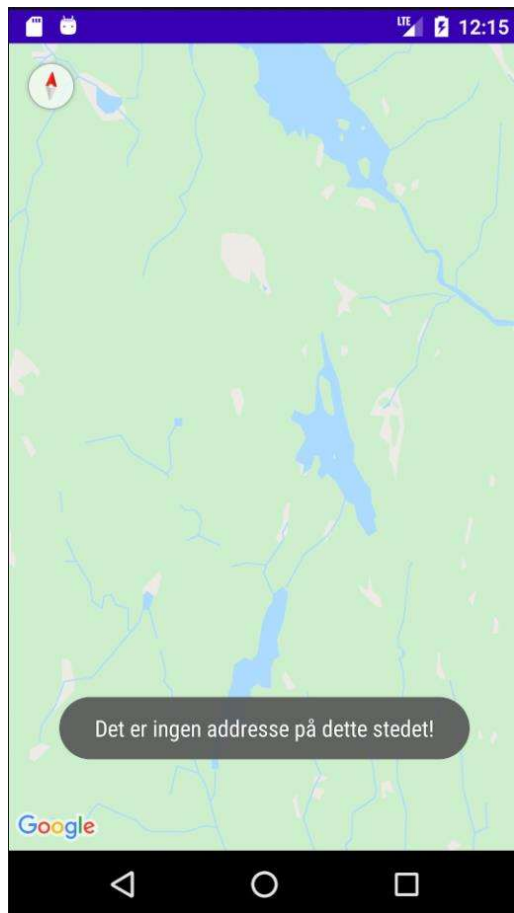
En liten feil er at det ikke er mulig å registrere rom i minus etasjer. For vi tenkte at når man bare har med hvor mange etasjer det er i en bygning er det fra bakkenivå. Får å kunne ha med bygninger med negative etasjer måtte vi ha lagt inn hva den laveste etasjen er og hva den høyeste og beregnet det ut ifra det. Så derfor valgt vi bare å inkludere positive etasjer.

Noen ganger når man har trykket på en bygning og får opp pop-upen så må man trykke flere ganger på kartet bak før at pop-upen skal bli borte. Hvis man ikke ønsker å bruke tilbake knappen. Dette er nok en feil i Android studio siden det bare skjer noen ganger.

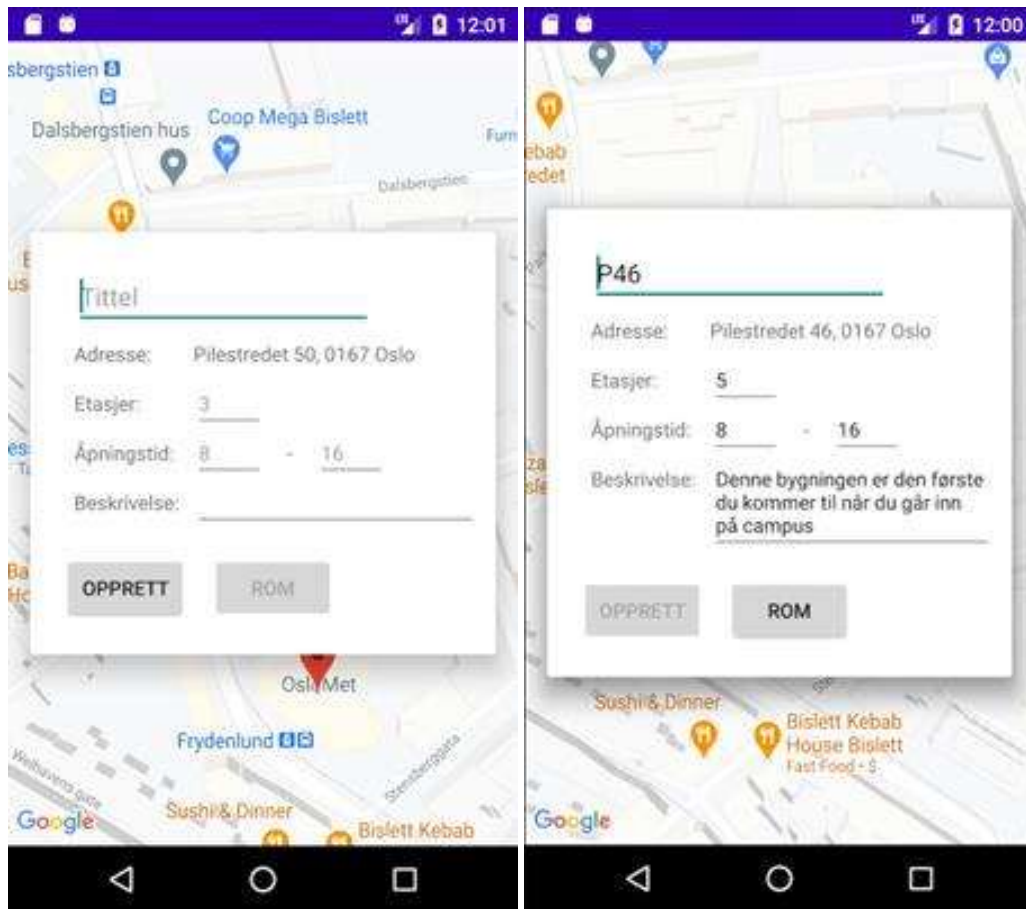
Figurer



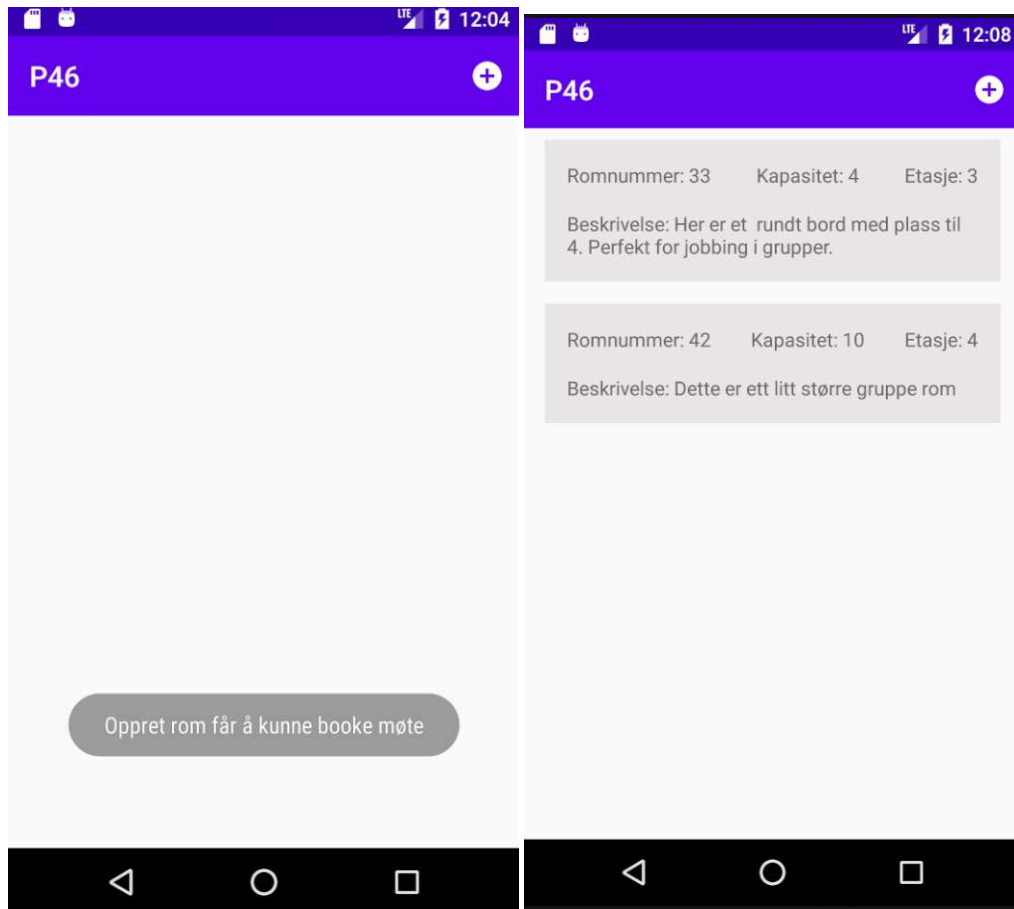
Figur 1 og 2: Her ser vi et bilde av det første du møter når du går inn i appen. Første bilde er hvis det ikke er noen bygninger og den andre er når det er bygninger. Den zoomer inn på OsloMet.



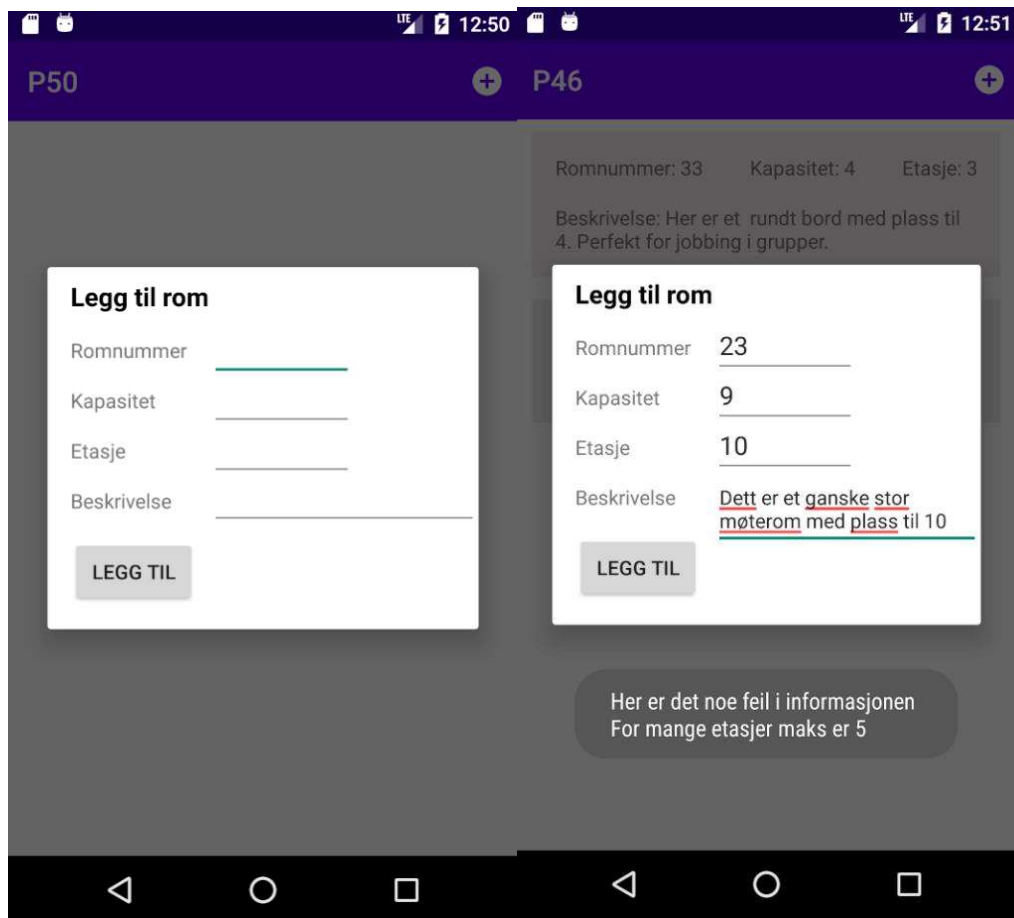
Figur 3: Hvis man trykker på et sted som ikke har noe adresse får man opp en melding.



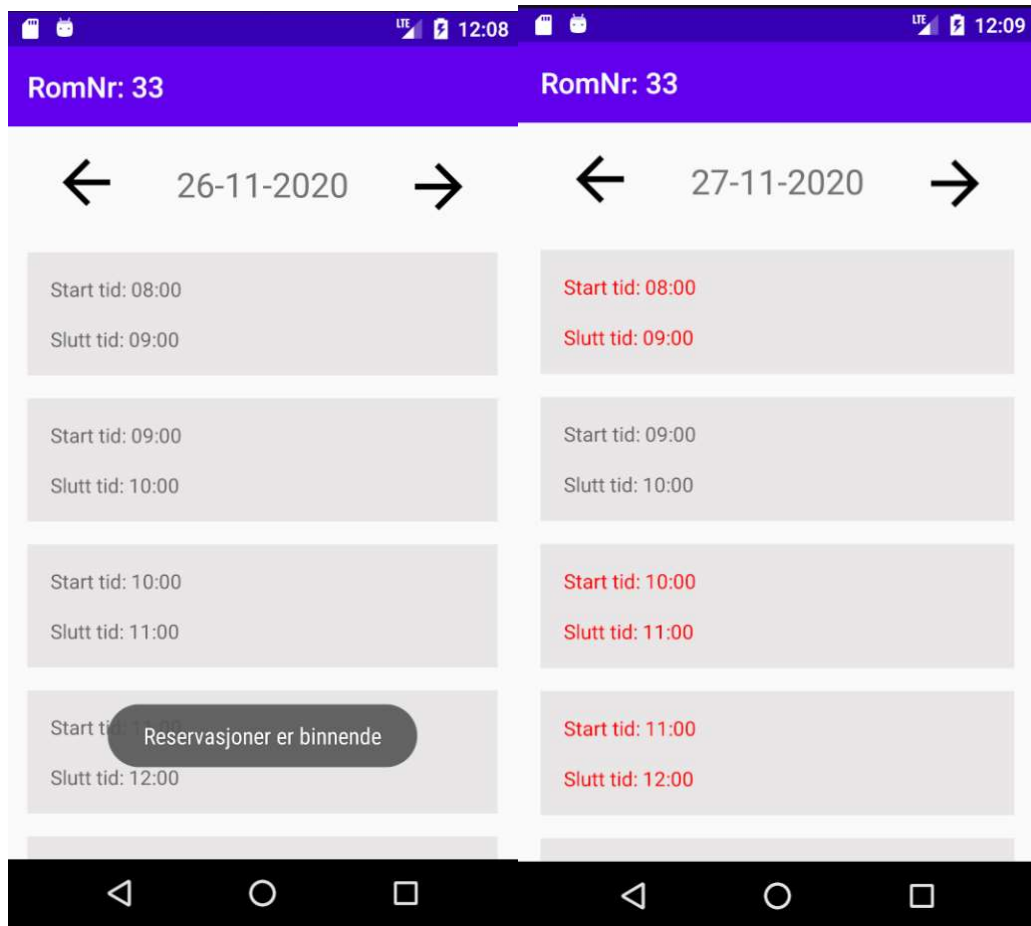
Figur 4 og 5: Når man trykker på en bygning som det ikke er laget noe bygning på får man opp bilde til venstre. Hvis man trykker på en markør, får man opp bilde til venstre.



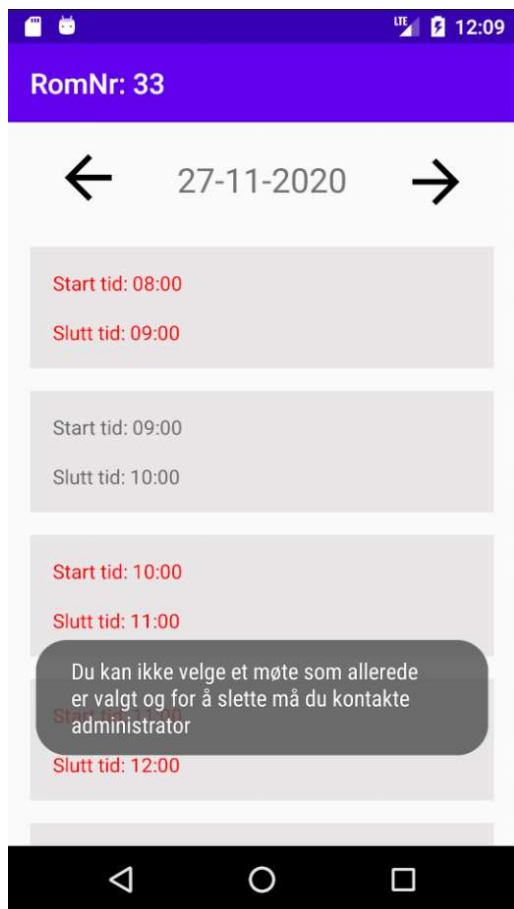
Figur 6 og 7: Til venstre ser man hvordan det ser ut hvis det ikke finnes noe rom. Med legg til knappen oppe i høyre hjørne. Bilde til Høyre ser man hvordan det ser ut hvis det er to rom i bygningen.



Figur 8 og 9: Her er pop-upen som lar deg legge til rom. Får også feilmeldinger hvis noe av informasjonen ikke stemmer.



Figur 10 og 11: Når man trykker på et rom så får man en melding om at reservasjonen er binnende. Det er mulig å bytte mellom dager med pilene og alle møtetidspunktene er på en time, men man kan booke flere. Timene som er booket blir røde.



Figur 12: Når man trykker eller holder lenge på en reservasjon som allerede er valgt. Får man opp en melding.

idHus	tittel	beskrivelse	gate	gateNr	postNummer	postSted	gpsLat	gpsLong	antallEtasjer	aapenTid	stengtTid
24	P46	Denne bygningen er den første du kommer til når du går inn på campus	Pilestredet	46	0167	Oslo	59.921098596198696	10.733308978378775	5	1970-01-01 08:00:00	1970-01-01 16:00:00
25	P50	Dette er p50	Pilestredet	50	0167	Oslo	59.921908528282934	10.732521414756775	7	1970-01-01 08:00:00	1970-01-01 18:00:00

idRom	Hus_idHus	beskrivelse	nummer	kapasitet	etasje
49	24	Her er et rundt bord med plass til 4. Perfekt for jobbing i grupper	33	4	3
50	24	Dette er ett litt større gruppe rom	42	10	4

idReservasjon	Rom_idRom	startDato	sluttDato
5	49	2020-11-27 08:00:00	2020-11-27 09:00:00
6	49	2020-11-27 10:00:00	2020-11-27 11:00:00
7	49	2020-11-27 11:00:00	2020-11-27 12:00:00

Fjern

Velg tabell: id til elementet:

Figurer 13: Her er ett bilde av hvordan debug.php filen ser ut.

AppApi dokumentasjon

Get queries

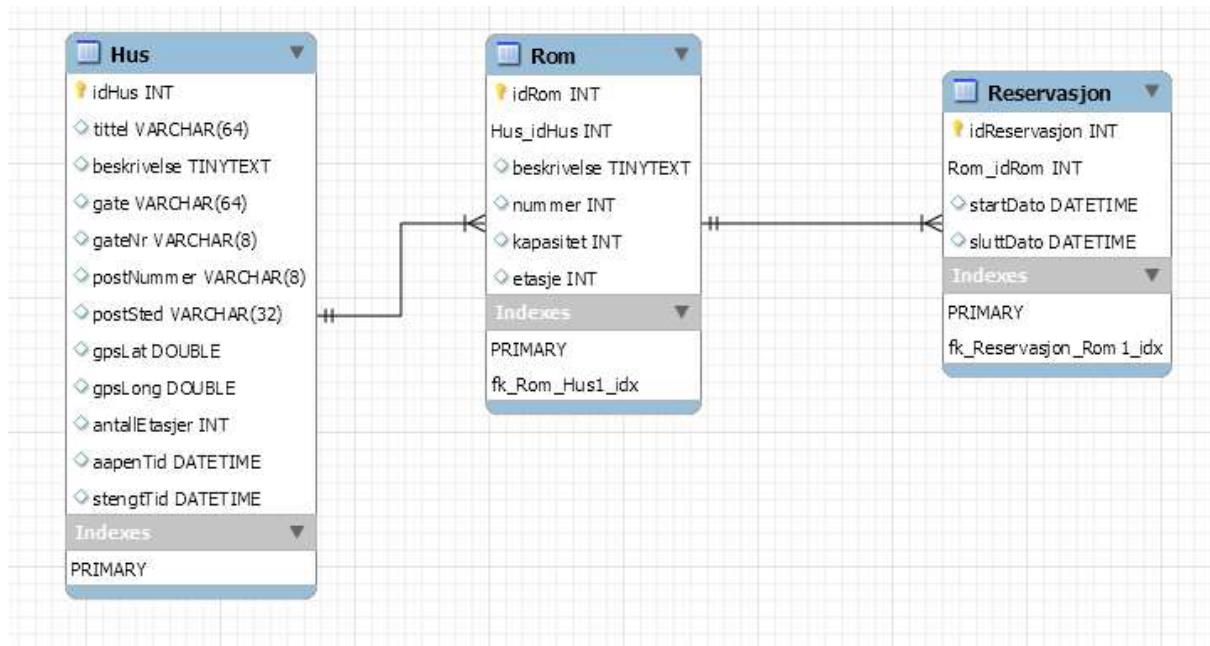
• getHus

Henter Hus lagret inn i databasen. <http://student.cs.hioa.no/~s344193/AppApi/getHus.php>
 Det er mulig å returnere kun hus med gitt id <http://student.cs.hioa.no/~s344193/AppApi/getHus.php?idHus=0>
 Det er mulig å returnere kun hus med gitt gps koordinater. <http://student.cs.hioa.no/~s344193/AppApi/getHus.php?gpsLat=0.0&gpsLong=0.0>

• getRom

Henter Rom lagret inn i databasen. <http://student.cs.hioa.no/~s344193/AppApi/getRom.php>
 Det er mulig å returnere kun rom knyttet til en bestemt hus id. <http://student.cs.hioa.no/~s344193/AppApi/getRom.php?idHus=0>

Figur 14: Bilde av hvordan AppApi ser ut i documentation.html



Figur 15: UML diagram av databasen. PostNummer og postSted lagres direkte i Hus tabell noe som ikke er optimal for store datamengder, men er akseptabel i dette tilfelle. Reservasjoner har sitt egen id, dette er ikke nødvendig dersom rom id + startDato kan brukes som id.