

## Lé Møte

Vi starta med å fokusere på funksjonaliteten til databasen, og fikset da kontakter først og fremst.

Derfor tenker jeg at vi kan starte med databasen.

### **Database:**

Vår database består av tre tabeller:

- TABLE\_KONTAKTER
- TABLE\_MOETER
- TABLE\_MOETEDELTAGELSE

Her holder naturligvis TABLE\_KONTAKTER oversikten over kontaktene, TABLE\_MOETER oversikten over møtene, og TABLE\_MOETEDELTAGELSE er hjelpetabellen vi har laget siden en kontakt kan være del av flere møter og et møte kan ha flere kontakter.

Attributtene i hver av tabellene er som etterspurt i oppgaven, men vi la til brukernavn for moroskyld i kontakter.

TABLE\_MOETEDELTAGELSE vil bare inneholde Moete\_ID og Kontakt\_ID, da den ikke trenger mer.

### **Design:**

Vi har brukt et BottomNavigationView som base for MainActivity, hvor hvert MenuItem har et frame-layout, som vi bytter ut med layout filer for fragmentene våre med kommandoen:

```
getFragmentManager().beginTransaction().replace(R.id.frame_container,  
selectedFragment).commit();
```

Avhengig av hvilket MenuItem vil vi få ulike view. Fragmentene kjører onCreateView på oppstart som vanlig, og LayoutInflateren vil «blåse opp» layout spesifisert fra res/layout.

Møteoversikt og Kontakter har hver sine listview som lister informasjon fra databasen, og hver sine floatingButtons som starter aktiviteter der vi kan opprette møtte eller en ny kontakt.

For å redigere møter eller kontakter må man holde inne på et listItem. Man vil da få opp en liten meny hvor man kan slette eller endre.

Ved slett: Sletter møte eller kontakt ifra databasen.

Ved endre: Får opp EditText for de ulike feltene man har lov til å endre på. Dette vil da oppdatere hva som ligger i databasen.

I innstillinger fragmentet endte vi opp med å kode vårt eget settingsvindu. Vi hadde opprinnelig tenkt til å bruke et preferencefragment, men fikk det dessverre ikke til å fungere med et BottomNavigationView. Vi føler vi gjorde et godt forsøk, da vi satt i 8 timer for å fikse det, men fant ingen løsning. Alternativet var å gå bort fra BottomNavigationView og implementere noe annet, men det tar vi oss ikke tid til da vi har mye andre fag å følge med i også.

I settingsvinduet har vi en switch for å skru av og på servicen for meldinger/notifikasjoner, en knapp for å velge tidspunkt ved en TimeDialog. Klokkeslettet for meldingen vises i et TextView. Vi har også en EditText man kan bruke for å velge hva som skal stå i meldingen som minner deg på møter, med en knapp for å lagre ønsket melding.

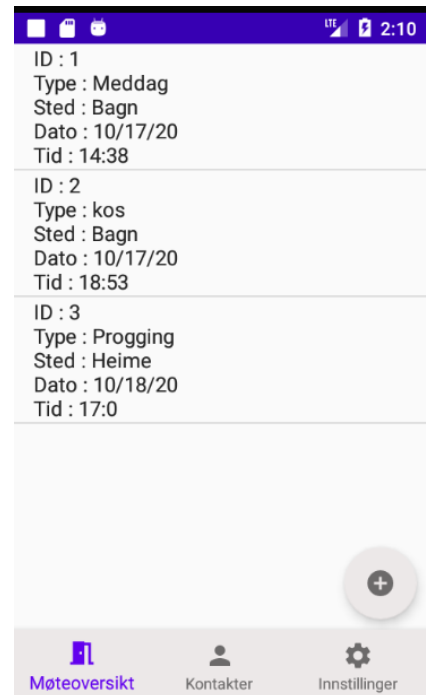
Dette førte til at vi måtte skrive egen kode for å legge til brukerens valg i SharedPreferences, men da kan vi beholde BottomNavigationView'et, så det ble vi egentlig veldig fornøyde med.

Vi har hyppig bruk av ikoner i applikasjonen vår, men har flere plasser hvor vi kunne hatt ikoner ellers også. Vi har ikke hatt fokus på å legge ikoner overalt siden vi ser i egne apper at det ikke alltid gjøres, og vi føler det personlig ikke er nødvendig.

For fargevalg har vi valgt og bare bruke standard fargene man får på applikasjonene i android, og gitt Toolbar og BottomNavigationBar en gråfarge for å skille den bedre fra bakgrunnen til vinduene.

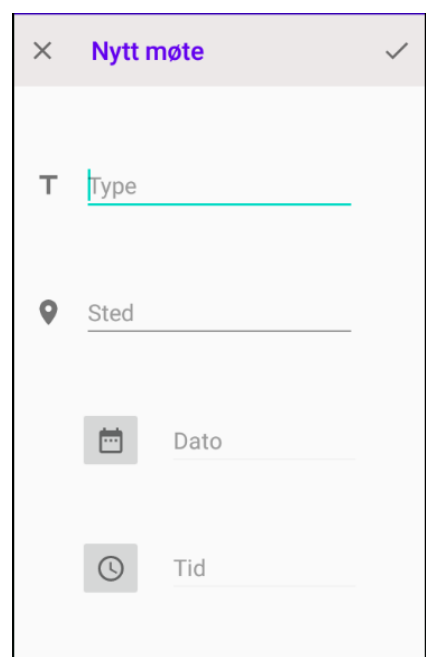
### Møteoversikt:

I møteoversikten har vi et listview som lister informasjonen om møtene. Vi har valgt og beholde ID-feltet her så du kan se at det oppdaterer seg om du legger til et møte.



Vår FloatingActionButton tar deg videre til en aktivitet hvor du kan opprette ett nytt møte.

Trykker man V vil man skrive informasjon til databasen og opprette møte.



Ved trykk på X vil man få opp en alerdialog:

### Avslutt?

Endringene lagres ikke.

IKKE AVSLUTT AVSLUTT

Holder man på et møte ved «long-click» vil man få opp en meny hvor man kan slette eller endre kontakt.

Trykker man slett vil man fjerne det valgte møte fra listen og databasen.

Trykker man endre vil man få opp samme vinduet som ved å trykke ny kontakt, men tittelen i toolbar vil være:

«Endre møte».

Der vil dataen man skriver inn overskrive det som ligger i databasen.

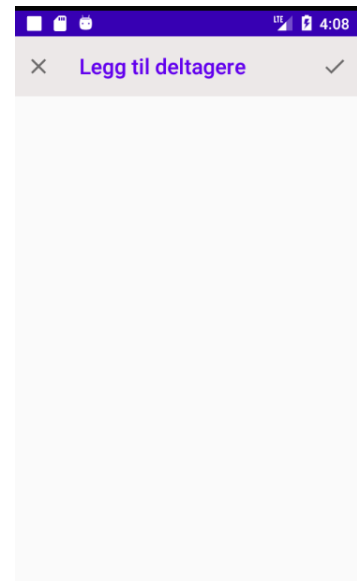


Trykker man på et møte vil man få opp en oversikt over hvilke deltagere det er på møte, og informasjon om dem. Ved å trykke på knappen vil man få en ny activity der man vil kunne legge flere deltagere til møte.

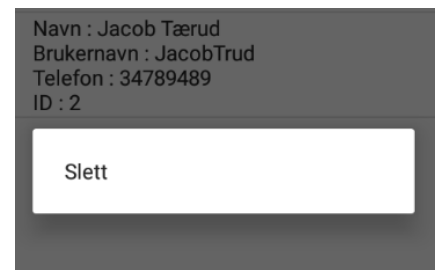


Her vil man kun få opp alle kontakter som ikke er en del av møtet fra før. Når man har valgt kontaktene man ønsker som nye deltagere av møte vil trykker man V for å legge de til i TABELL\_MOETEDELTAGELSE.

Trykker man X vil man få opp samme AlertDialog som over og avslutter man vil ingen endringer skje. Man blir da tatt tilbake til oversikten av deltagere på møte.



Ved «long-click» i oversikten over deltagere kan man slette deltagere ifra møtet.



Hvis man trykker V uten å ha fylt inn informasjon i alle feltene i «Nytt møte» vil man få opp en Toast-melding som sier at man må fylle inn feltene:

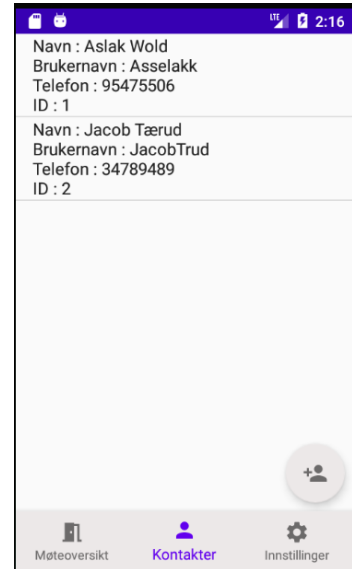


### Kontakter:

Vi har for kontakter holdt oss til samme designvalg og funksjonalitet som i møter, for at det skal være lett for brukeren å forstå appen.

Den viser informasjon som du ser på samme måte som i møteoversikten. Vi har her også valgt å beholde ID så du kan se at den oppdaterer seg automatisk når nye kontakter legges til.

Man kan også her ved «long-click» på en kontakt velge å endre eller slette.



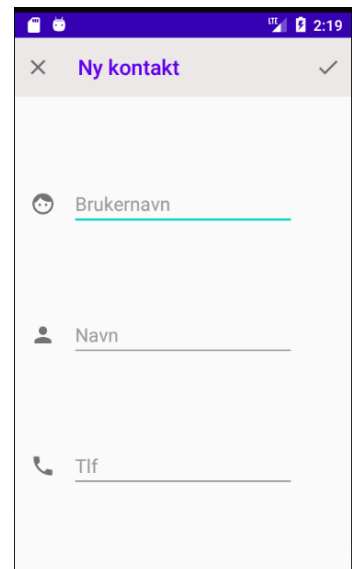
Ved å trykke på knappen nede til høyre får du opp en activity der du kan legge til en ny kontakt:

Den er som du ser lik siden hvor man legger til møter, bare med felter relevante for kontakter i stedet.

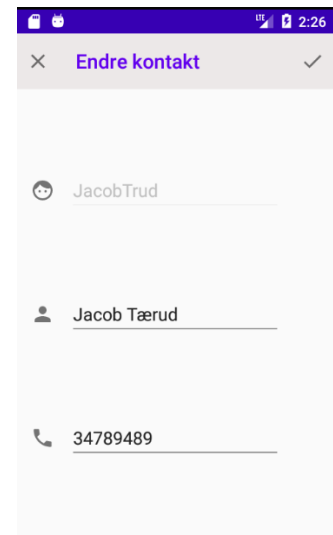
Trykker man på X vil man som i aktiviteten for nytt møte få opp dialogvinduet som spør om du vil avslutte. Ved trykk på V skriver man data til databasen.

Brukernavn la vi til som nevnt for moroskyld.

På samme måte som i møte vil man få opp en Toast-melding som ber deg fylle inn informasjon i alle feltene om man trykker V med tomme felter.

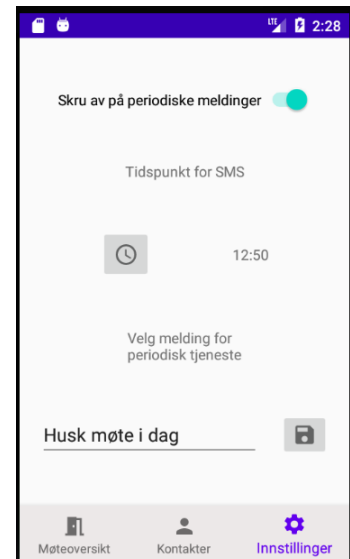


Velger man «endre» etter «long-click» vil man få opp et vindu som er helt likt, bortsett fra at man ikke kan endre på brukernavnet da det er UNIQUE NOT NULL i vår database.



### Innstillinger:

Innstillinger forklarte vi i design på første siden av rapporten.



### Feilhåndtering av input:

Vi har håndtert input av verdier på forskjellige måter.

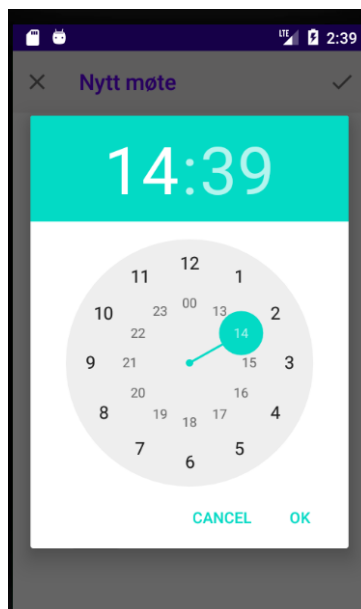
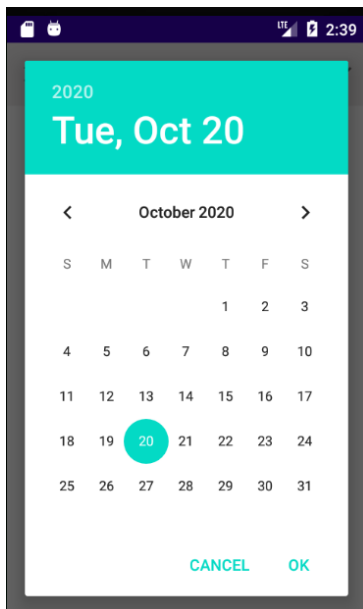
Den enkleste håndteringen av input vi har er bruk av EditText hvor vi har satt InputType i layout/xml-filene til å være spesifikt det vi etterspør i feltene.

Dette er for eksempel `android:inputType="phone"` i EditTexten der vi ønsker telefonnummeret.

Her har vi ikke håndtert noe for lengden på diverse, rett og slett fordi våre egne telefoner ikke håndterer det på noen måte. Vi kunne satt en grense på for eksempel 8 siffer, men vi ser ikke helt vitsen i å gjøre det.

Dette gjelder også for feltene type, navn osv.

For input av dato og tid har vi brukt DatePickerDialog og TimePickerDialog:



For få opp disse trykker man på ImageButton elementene i aktiviteten. Valgt dato/tid legges i EditText-vinduene ved siden av i et kort format. Som du kan se på bildene lenger opp har disse knappene passende ikoner.

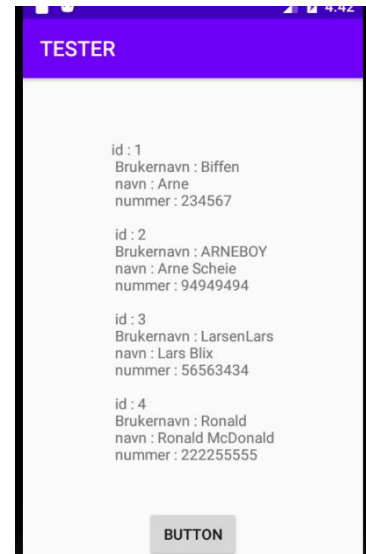
TimePickerDialog bruker vi også i Innstillinger for å velge tidspunkt for utsendelse av melding.

Det skal ikke være noen mulighet for å putte inn feil-verdier i appen nå, så vidt vi vet.

### ContentProvider og Service:

Vi har også laget en contentprovider som sender informasjonen til andre apper. Dette har vi testet ved å lage en egen app som i forelesningen:

Her viser vi informasjon om kontakter, men ved å endre på navn i koden til appen kan vi også få opp lister for møter og møtedeltagelse.



Servicen sender ut notifikasjon og melding:

Som du ser har vi endret på ikonet for notifikasjonen til ikonet til appen.

Når man trykker på notifikasjonen vil man åpne appen vår Lé Møte.

