Oblig 2 - Avtaler

## Introduksjon

I denne applikasjon kan man lagre nye kontakter og opprette avtaler med dem. Det vil vises en oversikt over avtalene, og en notifikasjon vil bli sendt om dagens avtaler, og en SMS vil bli sendt til deltakerne om denne funksjonaliteten er skrudd på.

Jeg har valgt å bruke mye ikoner, som er hentet fra den innebygde Asset biblioteket i Android Studio.  
Ønsket var å lage en app som føles ut som en som kunne vært innebygd på en telefon.

## Applikasjonens gang

Når applikasjonen startes åpnes en «Home screen», som viser nåværende tidspunkt og dato, samt den første kommende avtalen fra avtaleboken. Fra denne skjermen kan man slette eller endre denne avtalen ved å klikke på tannhjulet. Oppe i høyre hjørne er det også mulighet til å skru av utsendelse av SMS.



*«Home screen»*

Denne skjermen gir en brukeren en kjapp oversikt over dagen sin, uten å måtte klikke inn på noen flere menyer.

Jeg har valgt å gå for en «Bottom Navigation Menu» for navigering, som etter min erfaring er mye brukt, og derfor noe de fleste brukere er kjent med.

Oppsettet for å vise avtalen er identisk med oppsettet på avtalersiden, og blir forklart i den delen av rapporten. Eneste forskjell er at jeg her henter kun ut en avtale som er etter nåværende tidspunkt.

Et bilde som inneholder tekst, elektronikk, skjerm, skjermbilde

Automatisk generert beskrivelse

*«Kontakter»*

Ved å klikke på «Kontakter» på navigeringsmenyen blir vi tatt til oversikten over alle kontakter, og muligheten til å legge til nye. Jeg har gått for en «Floating Action Button» (FAB) som knappen for å legge til en ny kontakt. Denne er også mye brukt, og er gjenkjennelig. Den inneholder et tydelig ikon som viser hva knappen gjør. Ved å klikke på denne knappen åpnes det en ny skjerm for å legge til en kontakt.

Et bilde som inneholder tekst, elektronikk, skjermbilde, natthimmel

Automatisk generert beskrivelse

*«Ny Kontakt»*

Dette er en enkel skjerm bestående av to felter for å skrive navn og nummer, og en knapp for å legge til kontakten.

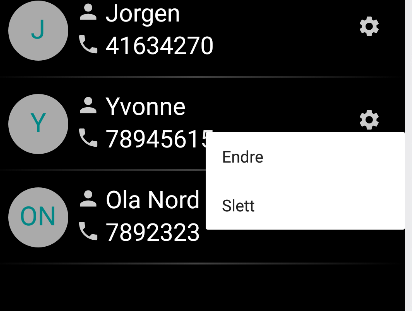
Jeg har valgt å gå for RecycleView i stedenfor ListView, da dette var sterkt anbefalt i dokumentasjonen til Android. De gjør det samme, men RecycleView er en mer moderne, fleksibel og utøvende løsning.

Elementene i listen er satt opp av et eget layout, som hentes inn i en Adapter, som så legges til RecycleView. Dette gir en god mulighet til å lage sitt eget design og funksjonalitet til elementene.

Til venstre er det en profillogo, som består av initialene til personen. Denne generes automatisk.

Så viser den navnet og nummeret med tydelige ikoner. Til høyre er det en knapp med et tannhjulikon, som forteller at det er mulighet for å gjøre endringer på denne kontakten ved å klikke på knappen.

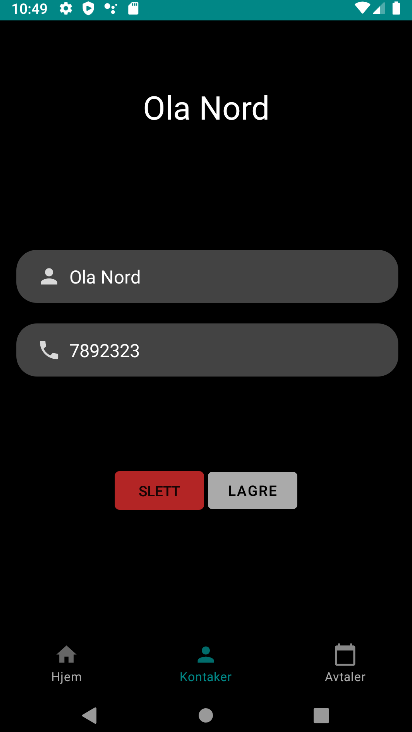
Når denne knappen klikkes, kommer det opp følgende kontekstmeny.



*«Kontekstmeny»*

Brukeren får to muligheter, enten endre kontakten eller slette den.

Ved å klikke på endre, vil den samme «Legg til kontakt»-skjermen åpnes, men med feltene fylt ut med lagrede kontaktinfoen.



*«Endre kontakt»*

Det dukker også opp en slettknapp, hvor brukeren kan enkelt slette kontakten.



*«Avtaler»*

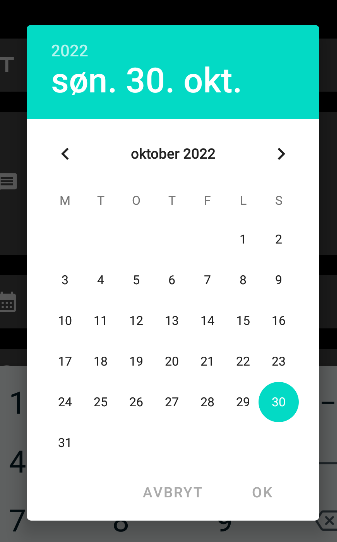
Avtalesiden består av det samme som kontaktsiden. En enkel liste med oversikt over alle avtalene, sortert på dato. Her brukes også samme «FAB»-knapp, med et annet ikon for å vise at det er en ny avtale som lages. Den samme kontekstmenyen vises her også ved å klikke på tannhjulet på en avtale.

Et bilde som inneholder tekst, elektronikk, skjermbilde

Automatisk generert beskrivelse

*«Ny avtale»*

Siden for å legge til en avtale er litt mer kompleks. Det er et felt for å legge til tittel og en melding som kan bli sendt til deltakerne som en SMS.



*«DatePicker»*

Ved å klikke på datofeltet, vil det åpnes en «DatePicker», som er et innebygd fragment for å velge dato.

Et bilde som inneholder tekst, klokke

Automatisk generert beskrivelse

*«TimePicker»*

Det samme gjelder også for klokkslett. Det er også mulighet til å skrive de inn manuelt med tastaturet.

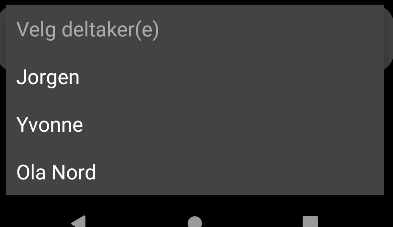
For å legge til deltakere har jeg satt opp en «Spinner», som åpner en liste når den klikkes på.  
Den viser da alle kontaktene brukeren har, og ved å klikke på en, legges den til i en liste.

Når en kontakt er lagt til listen, kan den fjernes ved å klikke på den røde knappen ved siden av navnet. En kontakt som er lagt til, vil gråes ut i listen og kan ikke velges igjen. For å få til denne funksjonaliteten måtte jeg overskrive en del av standardfunksjonaliteten til «Spinner».

For å få plass til alle feltene, er alt lagt til i en ScrollView, slik at brukeren kan «scrolle» feltene opp og ned.

Under er 4 bilder som viser denne prosessen.

Et bilde som inneholder tekst

Automatisk generert beskrivelse 

Et bilde som inneholder tekst

Automatisk generert beskrivelse Et bilde som inneholder tekst

Automatisk generert beskrivelse

*«Prosessen av å legge til en deltaker»*

Det er også samme funksjonalitet her som på kontakter, ved å klikke på «endre» i kontekstmenyen, vil denne siden åpne seg med all infoen fylt ut.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

*«Endre avtale»*

## Arkitektur

Siden jeg ønsket å bruke en «Bottom Navigation Menu» ble jeg nødt til å bruke fragmenter.   
Derfor har jeg bare en aktivitet, som består av menyen og en «FragmentViewer».

Hver side er sitt eget fragment som vises i denne «Vieweren». Dette oppsettet har bydd på en del utfordringer, spesielt ved å skulle sende data mellom aktiviteten og andre fragmenter.

Ved å lese en del i dokumentasjonen, kom jeg frem til en løsning hvor jeg satt opp et interface, «IOnClick», som hovedaktiviteten implementerer, og som blir satt som et medlem i fragmentene.

Denne interfacen består av metoder som kjøres i MainActivity, men kalles på fra fragmentene. Det brukes for eksempel til å åpne en fragment som «Legg til Kontakt» fra «Kontakt»-fragmentet, siden hele oppsettet for å bytte fragmenter ligger i MainActivity. Slik ser interfacet ut.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

*«IOnClick interface»*

For å sende data inn til et fragment, som for eksempel når en kontakt skal endres, så sender jeg data inn med en «Bundle» når jeg oppretter fragmentet.   
Adapterene som brukes til RecycleView har også IOnClick som et medlem, slik at de kan åpne fragmenter fra sin kontekstmeny.

Når applikasjonen startes, sendes det ut et signal til en «BroadCastReciever» som så starter opp «PeriodicService». Denne servicen vil en gang om døgnet sende ut en notifikasjon om dagens avtaler, og en SMS til alle kontakter om SMS er aktivert på hovedskjermen.

Databasen er satt opp ved hjelp av Room. Dette gjorde det veldig enkelt å jobbe mot databasen.   
De fleste feltene er bare strenger og skrives enkelt til databasen, men for at listen av kontakter i avtalerklassen kunne skrives til databasen måtte det settes opp «Converters», siden Room ikke visste hva den skulle gjøre med en ArrayList. For å få dette til å fungere måtte jeg sette opp en «Contacts»-klasse, som inneholder listen av kontakter. Videre satt jeg opp en «Converters»-klasse, med to metoder som konverterer fra en streng av ID’er til en kontaktliste, og omvendt. Ved å annotere i «AppDatabase» med en «TypeConverters» annotasjon som peker på «Converter»-klassen min, gjør Room denne konverteringen automatisk.

For å behandle dato riktig i avtaler, har jeg en del logikk i avtalerklassen som håndterer dette.

Jeg lagrer en variabel i SharedPreferences som styrer om en SMS skal sendes ut eller ikke. Denne blir satt av SMS-«ToggleButton» på hovedskjermen.

Det er lagt til støtte for norsk og engelsk språk.

# Konklusjon

Dette prosjektet har vært svært lærerikt. Jeg har møtt på mange utfordringer for å lage appen slik jeg ønsket, som har vært givende å løse. Det er mye å læres seg, spesielt med bruk av fragmenter. Mye av kunnskapen jeg sitter med nå hadde endret hvordan jeg hadde implementert ting fra starten av. Dette tar jeg med meg videre til neste oppgave.

Jeg ønsket å få implementere «ContentProvider», men det viste seg å være litt utfordrende å oversette implementasjonen til Room, og til slutt gikk jeg tom for tid.

I sin helhet har det vært utrolig gøy å utvikle denne appen, og jeg sitter igjen med mye ny kunnskap. Jeg er veldig fornøyd med hvordan designet ble til slutt, som er veldig nærme det jeg så for meg fra start.

# Referanser

* Android Documentation  
  <https://developer.android.com/docs>
* Kompendiet