

PracticApp

**APPLIKASJONSUTVIKLING
DAVE3600**

Mappe 3

**Sander F. Sandøy
S236305**

Forord

Denne dokumentasjonen beskriver gjennomføringen av siste obligatorisk oppgave i faget Applikasjonsutvikling. Dette omhandler prosjektet ”PracticApp” skrevet i programmeringsspråket Java og utviklet for Android-plattformen.

Produktdokumentasjon

Innledning

Her vil jeg gjennomgå oppbyggingen av programmet. Jeg vil gå igjennom hva som er implementert i programmet, hvilke klasser applikasjonen har og hvordan disse vekselvirker.

Applikasjonens funksjonaliteter

Applikasjonen jeg har laget fungerer som en *planlegger*. Funksjonaliteten applikasjonen tilbyr er at man kan legge til diverse planer man har i løpet av hverdagen, og når det er tid for å gjøre disse planene, så kommer det en varsel om dette. Målet med appen er at den skal gjøre hverdagen enklere og mer praktisk for brukeren, derav navnet *PracticApp*.

Applikasjonens egendefinerte klasser

Applikasjonsklasse

PracticApp – Applikasjonsklasse hvor globale variable deklarerer. Dette er for eksempel dato-formatet til applikasjonen.

Aktiviteter

PlanOverviewActivity – Aktivitet som viser planene til brukeren. Brukeren kan sortere på dag, uke eller måned. Benytter seg av et ListView som plasserer data på en oversiktlig og pen måte. ListViewet populeres av en LoaderManager, som kjører en AsyncTask i bakgrunnen, slik at UI-tråden ikke blir overbelastet.

PlanActivity – Aktivitet hvor bruker kan registrere en ny plan. Kan sette starttid og sluttid på planen og kan bestemme når han vil ha varsel, når hendelsen starter eller før.

EditPlanActivity – Aktivitet hvor bruker kan endre eller slette en valgt plan.

Services

AlarmService – IntentService som mottar ID fra et database-objekt og sender dette videre til min BroadcastReceiver via et Intent. Denne klassen setter også en alarm til det gitte tidspunktet for database-objektet. Benytter seg av et

IntentFilter som filtrerer på hvilken handling som skal utføres. Dette er enten Create, Cancel eller Update.

BroadcastReceiver

AlarmReceiver – BroadcastReceiver som mottar et Intent fra AlarmService. Denne inneholder ID til planen som skal utføres, og receiveren skaper en Notification som skal komme opp på skjermen ved det valgte tidspunktet.

AlarmResetter - BroadcastReceiver som kalles når mobilenheten skrus på og deretter setter alle planlagte alarmer ved å starte opp AlarmService.

Database-aksess

DBHandler - Database-handler som skaper database og gjør database-aksess.

PlannedEvent – POJO-klasse som også aksesserer database. Gjør det enklere å legge til og oppdatere objekter.

ContentProvider

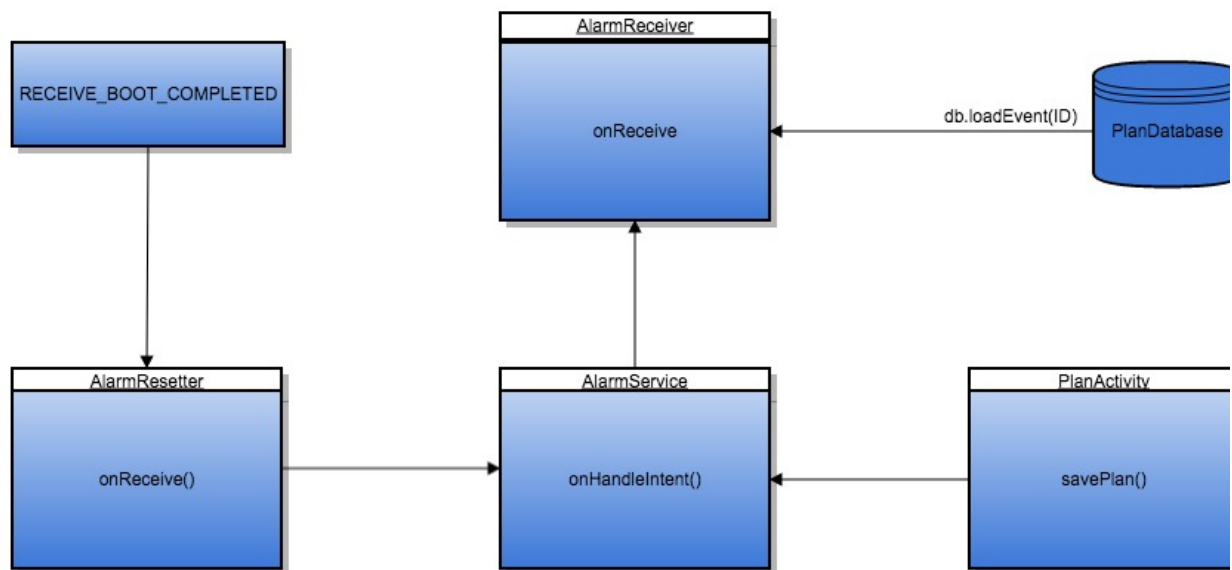
PlanContentProvider – Applikasjonens egen Content Provider. Denne gjør at andre applikasjoner kan hente ut/sette inn data fra applikasjonen ved å benytte seg av en CONTENT_URI og å sette <uses-permission> til ContentProvideren sin <permission> i sitt eget AndroidManifest. ContentProvideren benyttes i prosjektet av en LoaderManager som henter ut data fra denne og legger de til i mitt egendefinerte ListView.

Hjelpeklasser

PlanAdapter - Klasse som benyttes i ListViewet til PlanOverView. Denne setter verdien til hvert ListItem.

Moduler og rolleoversikt

Oversikt for Services og BroadcastReceiver



I *AndroidManifest.xml*-filen settes de to permisjonene:

```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

Samt at inne i receiveren settes:

```
<receiver
    android:name=".AlarmResetter">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

Dette gjør at *onReceive*-metoden til *AlarmResetter* kalles ved oppstart på telefonen, som igjen starter opp *AlarmService*. Dette gjør at alarmene til applikasjonen vil bli satt.

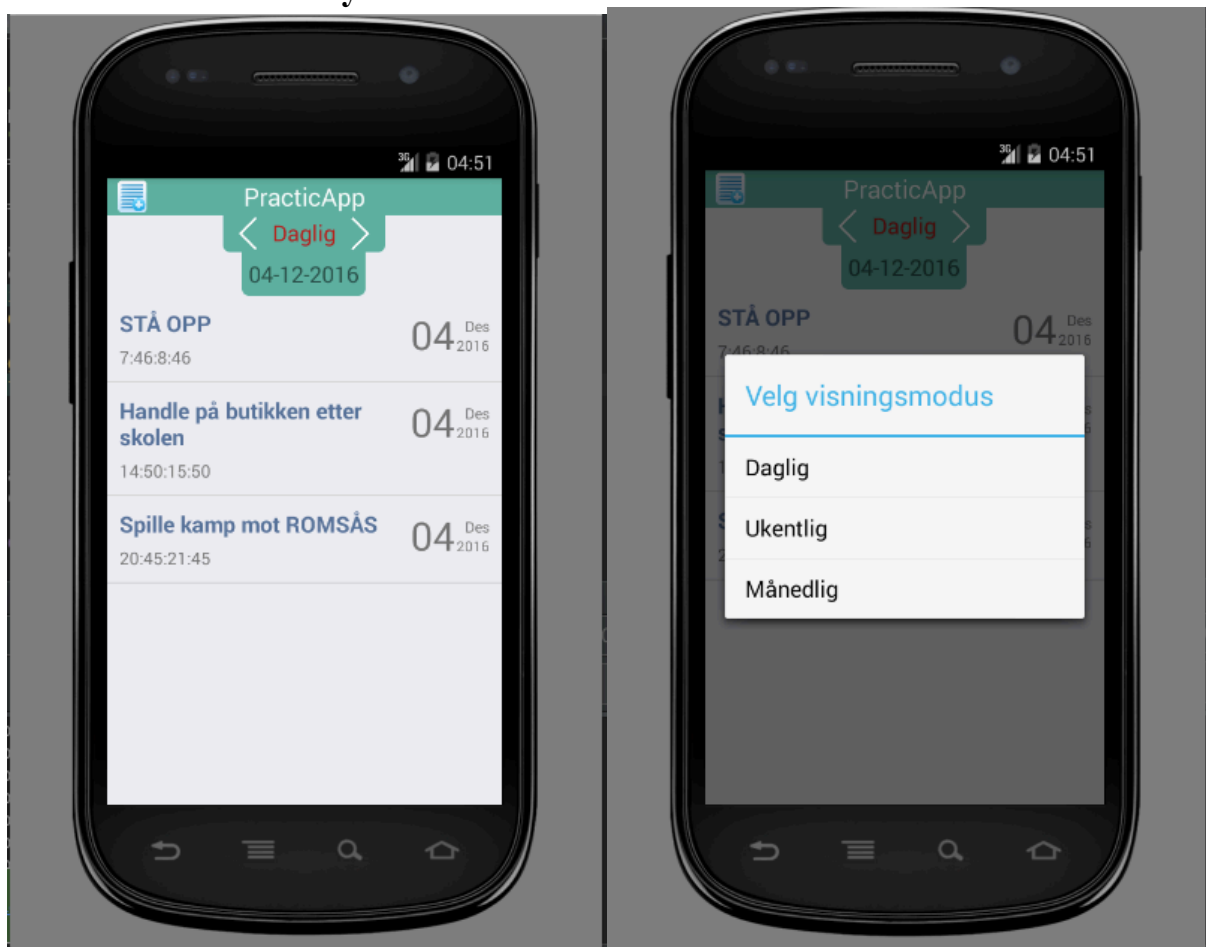
Design og navigasjon

Når det kommer til fargevalg i applikasjonen er målet med applikasjonen å gi en så oversiktlig og ryddig presentasjon som mulig. Derfor er det benyttet en lysegrønn dato-velger, noe som gir en fin overgang til det gråe ListViewet som ligger under. Har benyttet <http://color.adobe.com> for å finne en god kombinasjon mellom grønn og grå.

Prinsippet "Consistency of the design layout" er også benyttet. Det som menes med dette er at alle vinduer i applikasjonen bruker lik bakgrunn, font-type og skriftstørrelse.

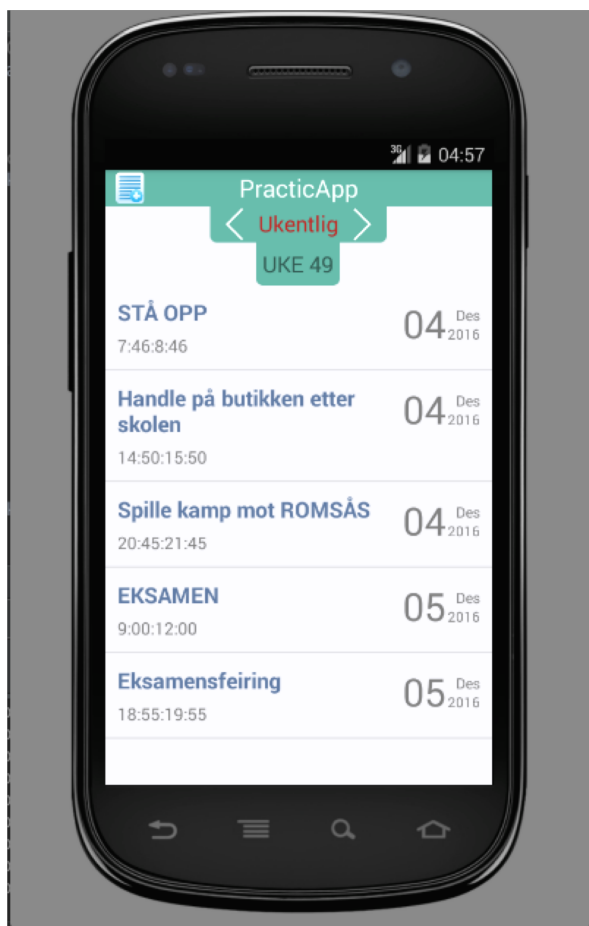
Når det kommer til horisontal layout, så valgte jeg bevisst å ikke implementere dette. Dette fordi at input-feltene når man registrerer en plan er så få at det ikke ville gitt noe forbedring ved å ha en horisontal layout tilgjengelig. Det er også veldig mange kalender-apper som ikke tilbyr dette, så jeg føler valget er berettiget. Jeg har også vist kompetanse på horisontal layout i tidligere mapper, så derfor valgte jeg å ikke benytte dette i denne innleveringen.

PlanOverviewActivity



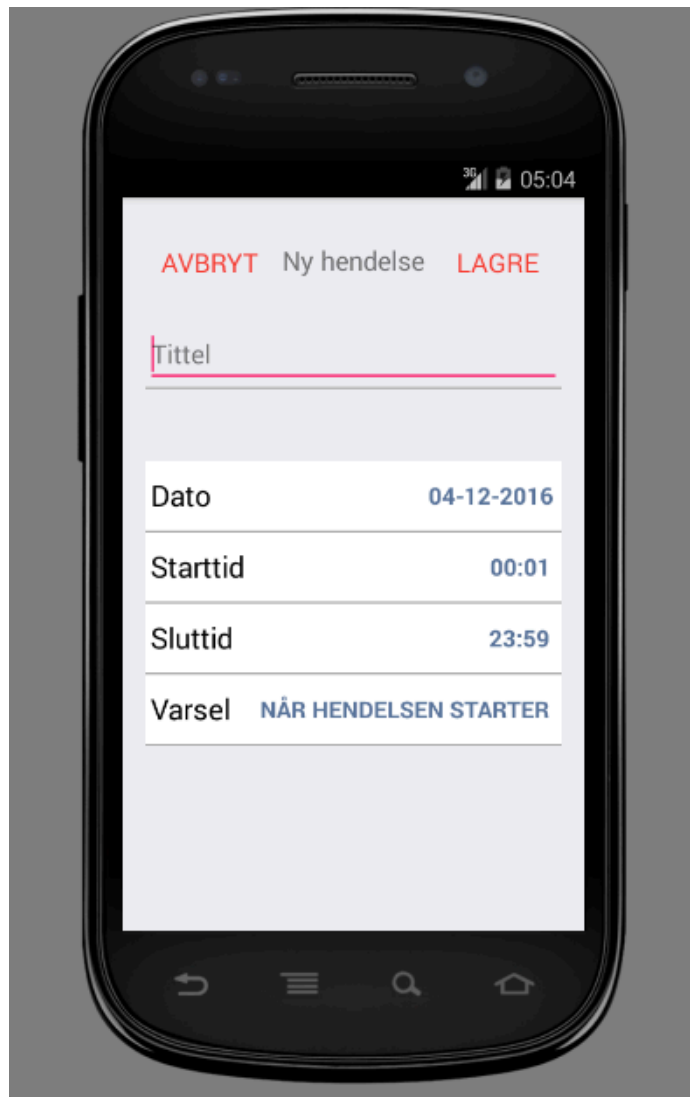
Hovedskjerm ved oppstart og ved klikk på visningsmodus-velger.

I PlanOverviewActivity kan man se dagens, ukens eller månedens planer. Man kan velge hvilket visningsmodus man vil ha ved å trykke på den røde visningsmodus-teksten. Man kan også bla frem og tilbake mellom datoer, uker eller måneder ved hjelp av pilene på siden. Øverst gis det også mulighet til å legge til en ny plan ved å klikke på legg-til-plan-ikonet.



Ukentlig-visningsmodus for hovedskjermen.

PlanActivity



PlanActivity-skjerm bilde

I PlanActivity kan man registrere nye planer. Man kan sette tittel, velge dato, starttid, sluttid og når man vil ha varsel fra telefonen.

Designet er enkelt og oversiktlig med store felter og tydelige navigasjons-valg.

EditPlanActivity

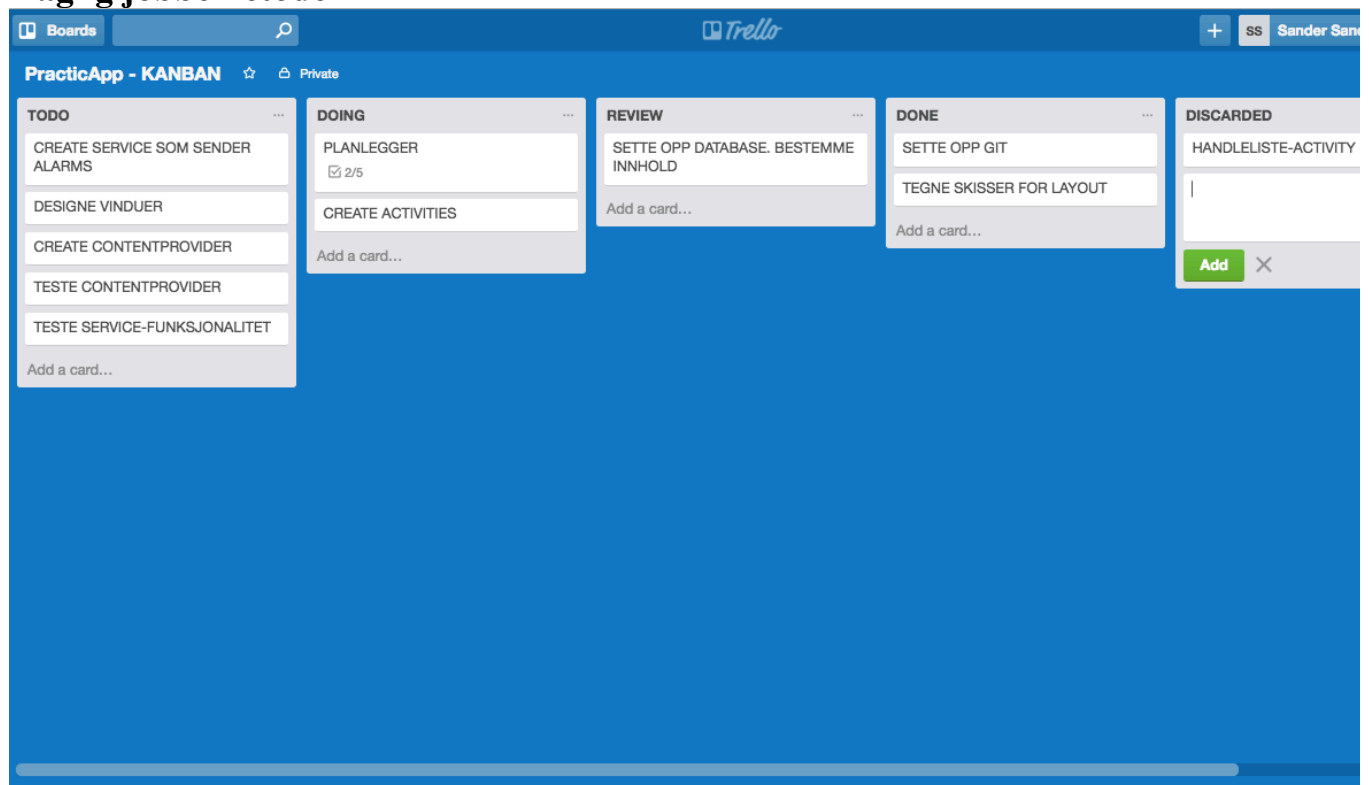


Skjerm bilde for EditPlanActivity.

I EditPlanActivity kan man endre en allerede eksisterende plan. Man kan endre alle feltene til objektet. Bruker samme enkle design som i PlanActivity basert på prinsippet "Consistency of the design layout".

Kort om arbeidsprosessen

Daglig jobbemetode



Kanban-boardet midt i prosjektarbeidet.

Under prosjektet har jeg benyttet meg av et "Kanban-board" via Trello for å enklere holde oversikt over arbeidsoppgaver. Dette gjorde prosessen mye enklere og oversiktlig. Har jobbet strukturert fra dag én og Kanban-boardet har hjulpet veldig med dette.