

Applikasjonsutvikling (DAVE3600)

Mappe 2 – SMS app

Rudi André Dahle – s300373

29.10.2017

Høgskolen i Oslo og Akershus

Appbeskrivelse

Når man starter appen kommer man rett inn til listevietet med menyen øverst i toolbaren (fig. 1).

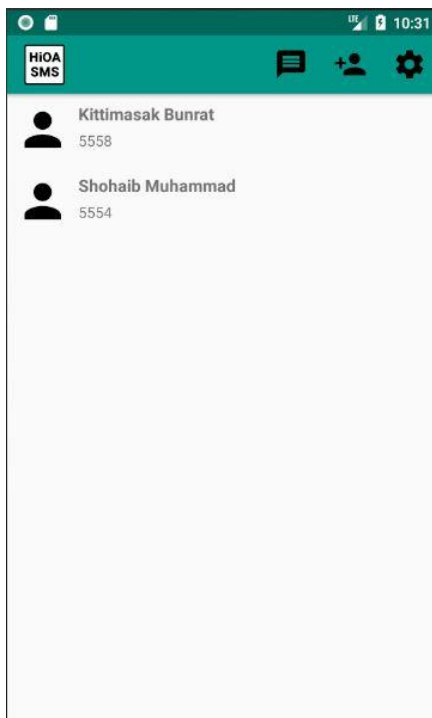


Fig. 1



Fig. 2

I toolbar har jeg plassert 3 knapper representert ved hvert sitt ikon, ikonene er hentet i fra Google sitt material design ikon sett. Trykker man på det venstre ikonet, skriv ny melding, kommer man til skjermbildet for å registrere en melding som skal sendes (fig. 2).

Her velger man først en dato man vil at meldingen skal sendes ut på, så velger man tidspunktet på den aktuelle datoen før man til slutt skriver inn meldingsteksten. Når man har satt opp meldingen slik man ønsker (fig. 3) trykker man på lagre melding for å legge meldingen inn i databasen (fig. 4), klar til å sendes.

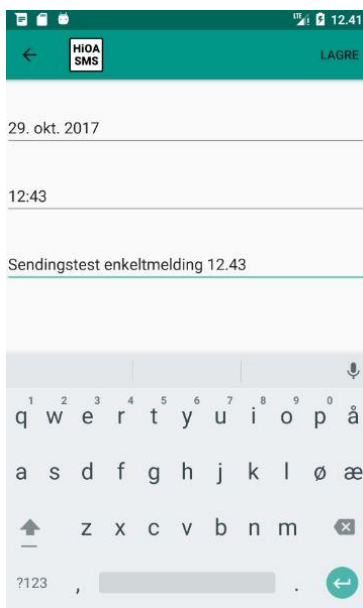


Fig. 3

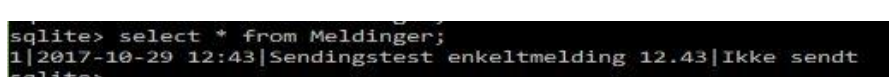


Fig. 4

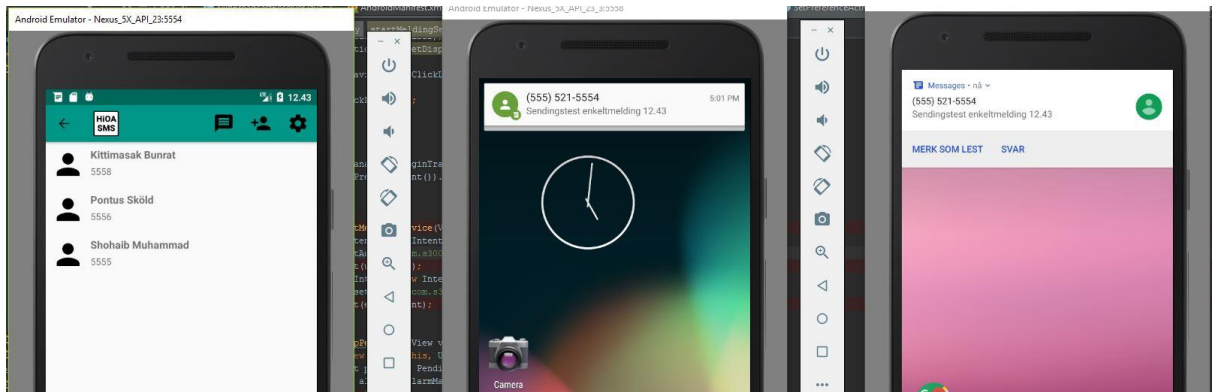


Fig. 5

Vi ser så at meldingen sendes til gitt tidspunkt (fig. 5) til alle nummer som er registrert i databasen, og meldingen i databasen vil oppdateres og få statusen endret i fra "Ikke sendt" (fig. 4) til "Sendt" (fig. 6).

```
sqlite> select * from Meldinger;
1|2017-10-29 12:43|Sendingstest enkeltmelding 12.43|Sendt
sqlite>
```

(fig. 6)

Trykker vi på det midterste ikonet i hovedskjermbildet (fig. 1) så kommer vi til skjermbildet for å registrere en student (fig. 7). Her skriver man først inn fornavn, etternavn og telefonnummer til sist (fig. 8).

Fig. 7

Fig. 8

Når man så trykker på lagreknappen så opprettes det en ny student i databasen, og han/hun dukker opp i listevietet (fig. 9).

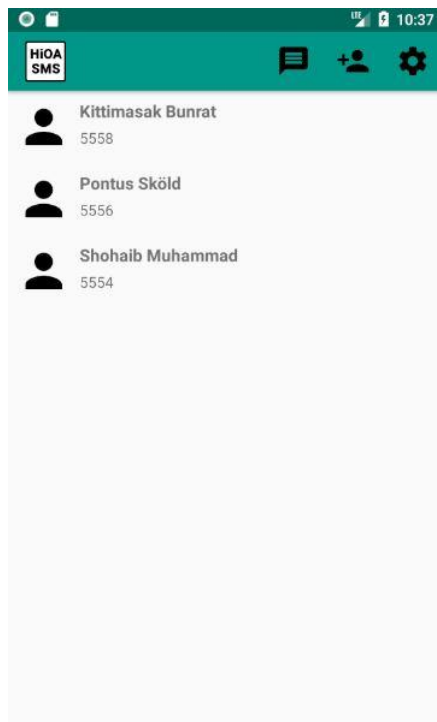


Fig. 9

Videre legger jeg inn et par teststudenter for å vise hvordan sletting foregår (fig. 10).

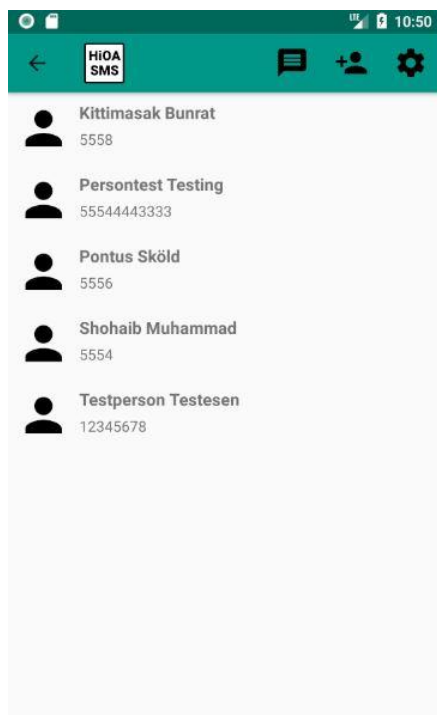


Fig. 10

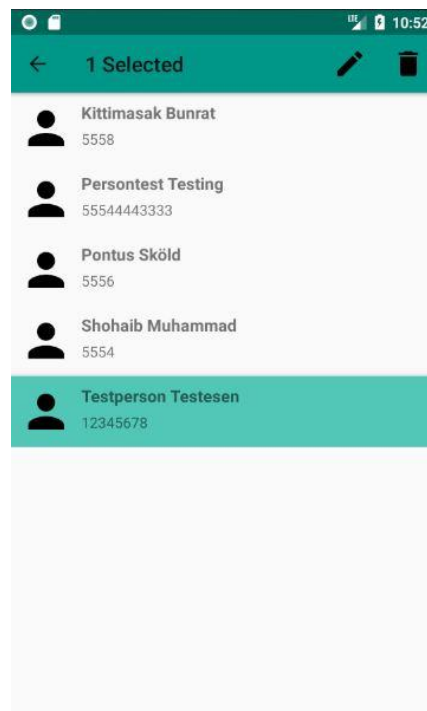


Fig. 11

Om man trykker på en student og holder et sekund på den så vil studenten bli markert og er dermed valgt (fig. 11). Ikonene i toolbar byttes også ut, og man får valget om å slette eller redigere studenten. Trykker man på slett så fjernes studenten i fra listevietet og i fra databasen (fig. 12).

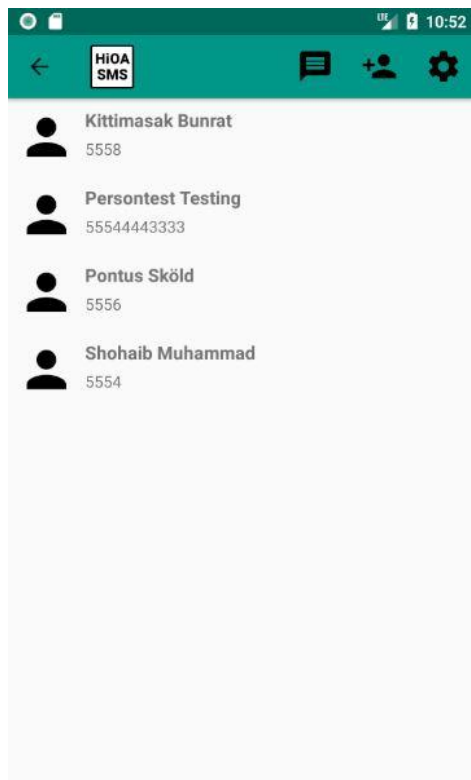


Fig. 12

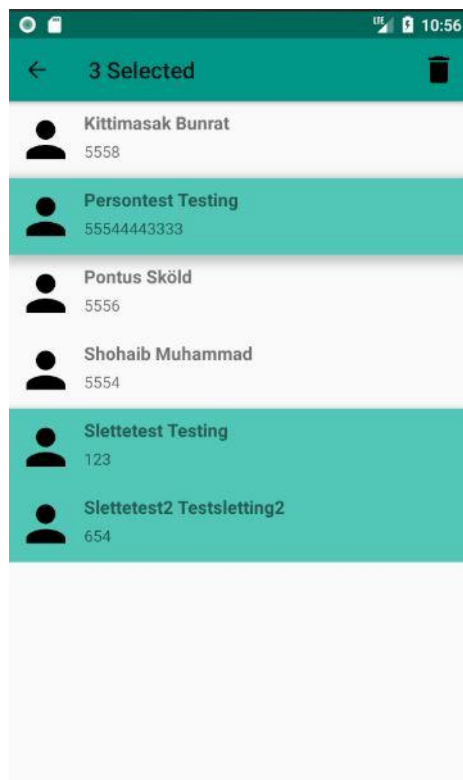


Fig. 13

Man kan også velge flere studenter (fig. 13), da forsvinner også knappen for redigering. Trykker man på slett med flere studenter markert så vil alle fjernes både fra listevue og i fra databasen (fig. 14).

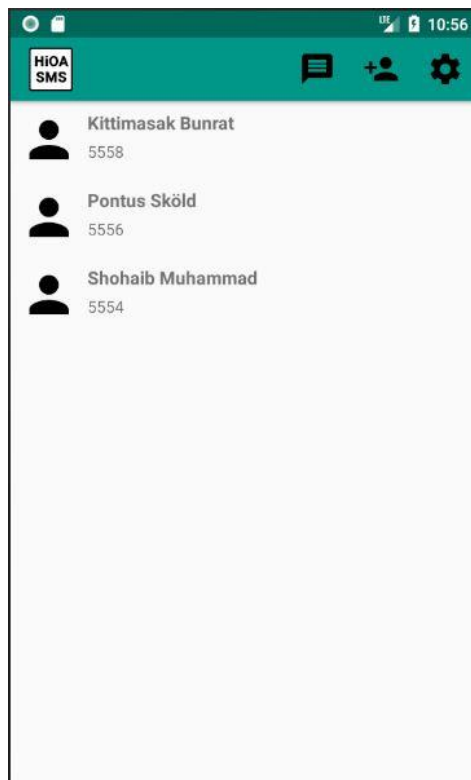


Fig. 14

Skal man redigere en student så trykker man som sagt og holder inne på den studenten man vil endre (fig. 15), og trykker så på redigeringsikonet slik at skjermbildet for redigering vises (fig. 16).

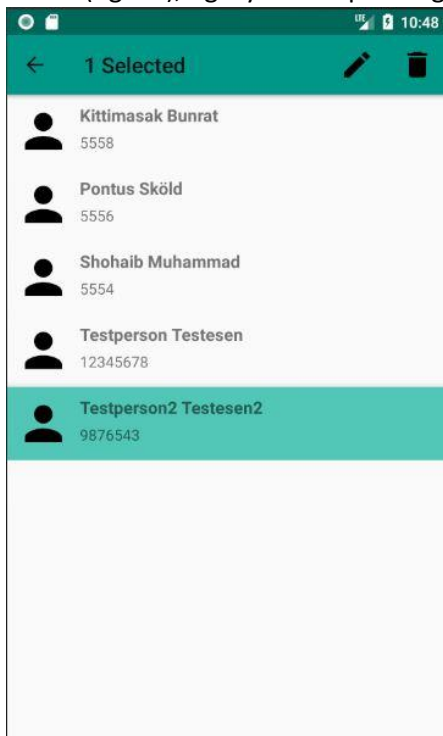


Fig. 15



Fig. 16

Nå kan man endre på de feltene man vil (fig. 17), trykke på lagre og se i listevignet at den nye infoen har blitt registrert og endret (fig. 18).

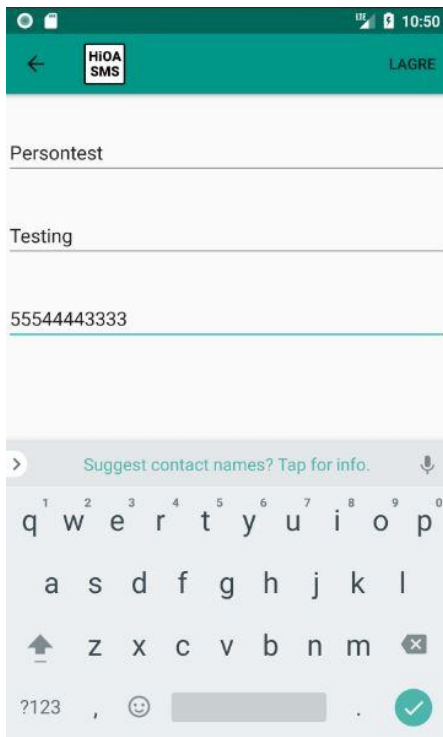


Fig. 17

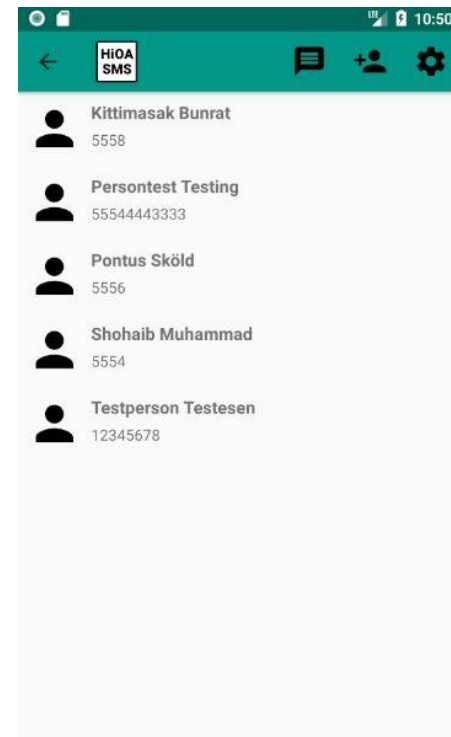


Fig. 18

Det siste ikonet i fra hovedskjermbildet er for instillinger. Trykker man der kommer man til innstillingsskjermbildet (fig. 19).

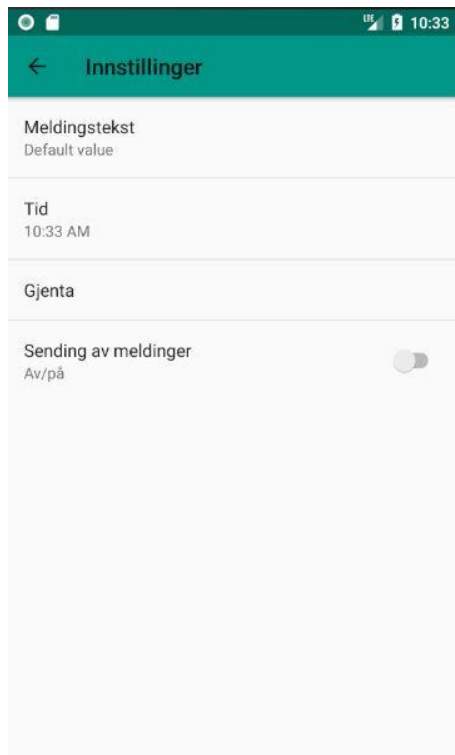


Fig. 19

Her kan man velge en standard meldingstekst som skal sendes ut med et ukentlig intervall. Man velger tid, og hvilken dag i uken man ønsker at meldingen skal sendes på, og kan skru utsending av meldinger av og på (dette gjelder alle typer meldinger, både ukentlige og enkeltmeldinger).

Vi setter en testmelding, velger dag og klokkeslett (har endret koden til å sende med 1 minutters intervaller) (fig. 20).

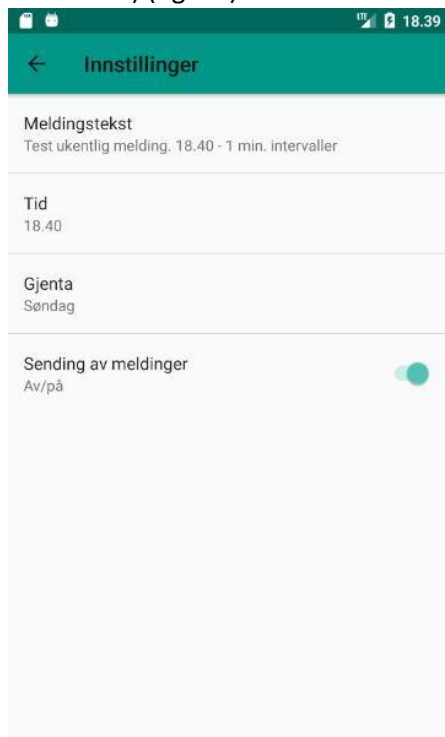


Fig. 20

Vi ser så at meldingen sendes ut på gitt tidspunkt (fig. 21).

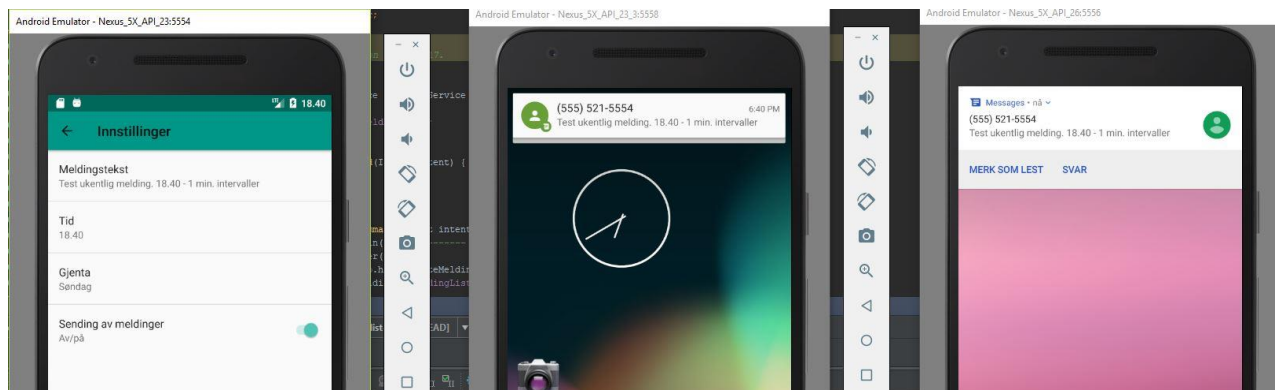


Fig. 21

Vi venter ett minutt og ser at meldingen sendes på nytt (fig. 22).

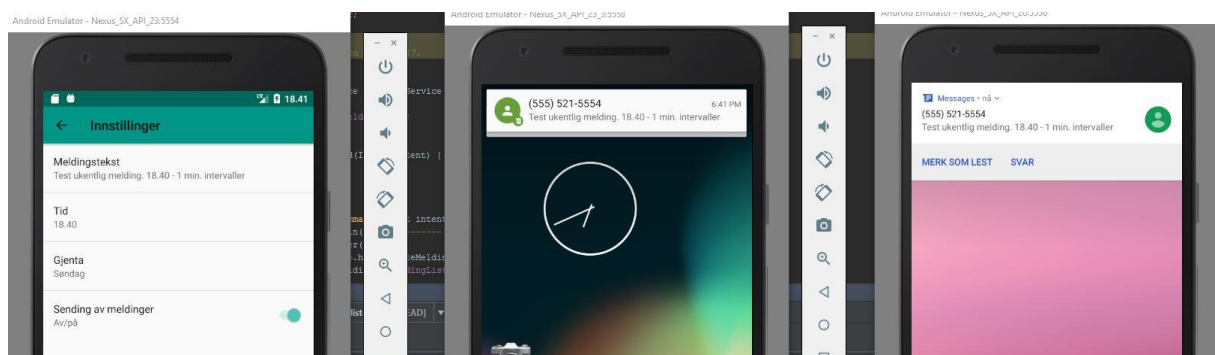


Fig. 22

For å demonstrere contentprovider så har jeg lagt inn en liten kodesnutt i onCreate i MainActivity (fig. 23). Vi ser så hvilke studenter som ligger i listevietet (fig. 24), og starter appen på nytt.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.
        ActivityCompat.requestPermissions(this, new String[]{android.Manifes
    })

    // Contentprovider insert test
    ContentValues values = new ContentValues();
    values.put("Fornavn", "Jonny");
    values.put("Etternavn", "Bravo");
    values.put("Telefon", "19283747");
    getResolver().insert(CONTENT_URI, values);
}
```

Fig. 23

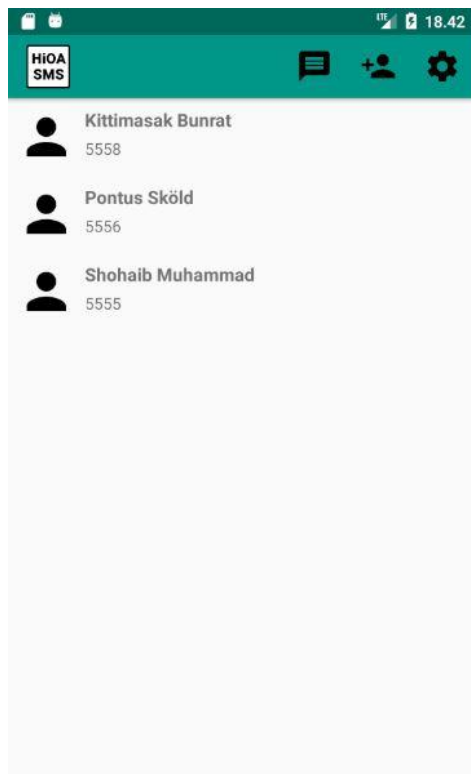


Fig. 24

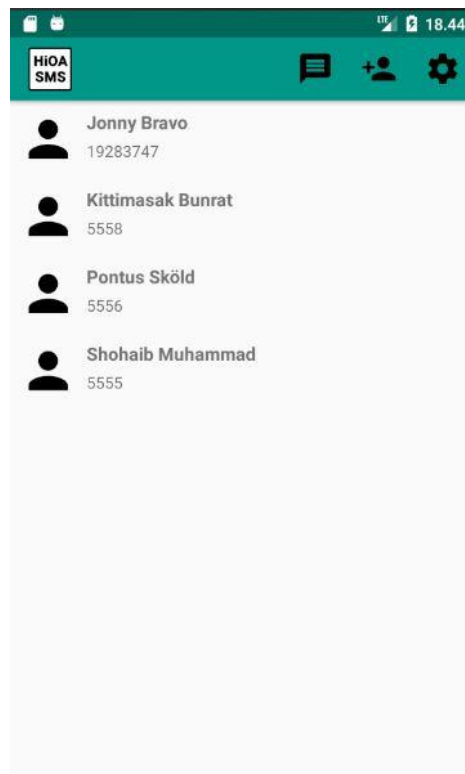


Fig. 25

Når appen har startet opp igjen ser vi at studenten som ble spesifisert i kodesnutten er lagt til i listevietet (fig. 25).

Det er også lagt ved en kodesnutt for å demonstrere sletting via contentprovider, denne koden fjerner alle studenter i databasen (fig. 26 og fig. 27).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{android.Manifest.permission.READ_CONTACTS}, 1);
    }

    /*
    // Contentprovider insert test
    ContentValues values = new ContentValues();
    values.put("Fornavn", "Jonny");
    values.put("Etternavn", "Bravo");
    values.put("Telefon", "19283747");
    getContentResolver().insert(CONTENT_URI, values);

    // Contentprovider delete test (sletter alle)
    getContentResolver().delete(CONTENT_URI, null, null);
    */
}
```

Fig. 26

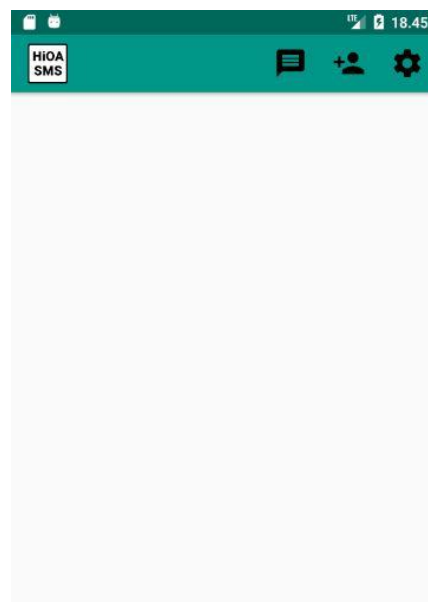


Fig. 27

Testkodene til contentprovider blir liggende i programmet, men blir kommentert ut.

Design og navigasjonsvalg

Jeg har prøvd så godt som mulig å følge material design ved navigasjon, farge- og ikonvalg.