

1.1 Starten

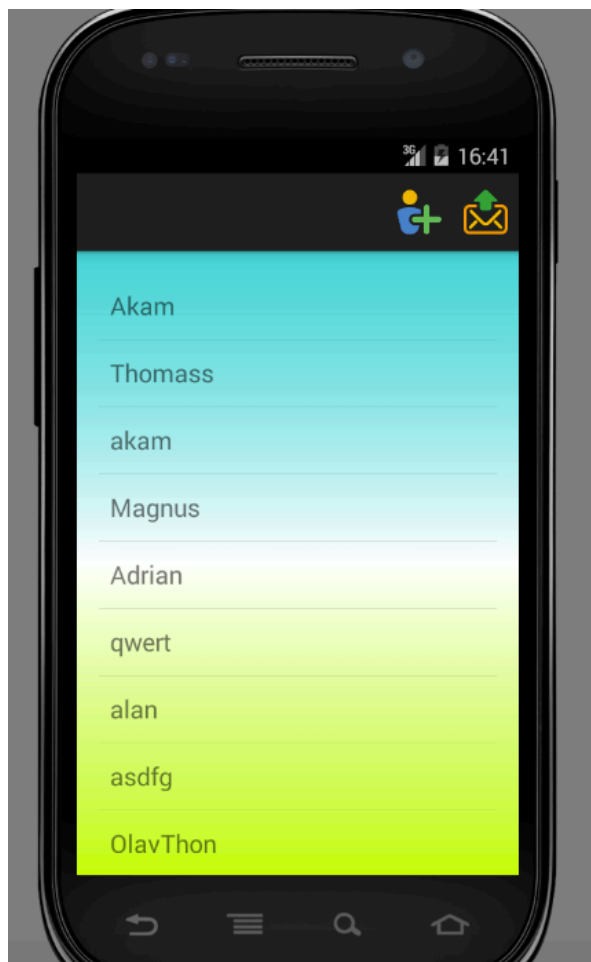
Det var veldig andre prosjektet som vi fikk i dette faget. Prosjektet var helt annerledes med den første. På dette appen skulle vi programmere mot databasen. Man skulle kunne registrere personer med navn, telefon nummer og fødselsdato. Man kunne endre data av personer eller slette det. Når man skulle slo på mobilen programmet skulle sjekke fødselsdatoen av alle personene i listen og det skulle sende en sms i en oppgitt tidspunkt til de som hadde bursdag. Prosjektet en del vanskeligheter og mye nye stoff som hjulpet oss til å lære.

1.2 Antall screenssider

Spillet består av følgende screenssider:

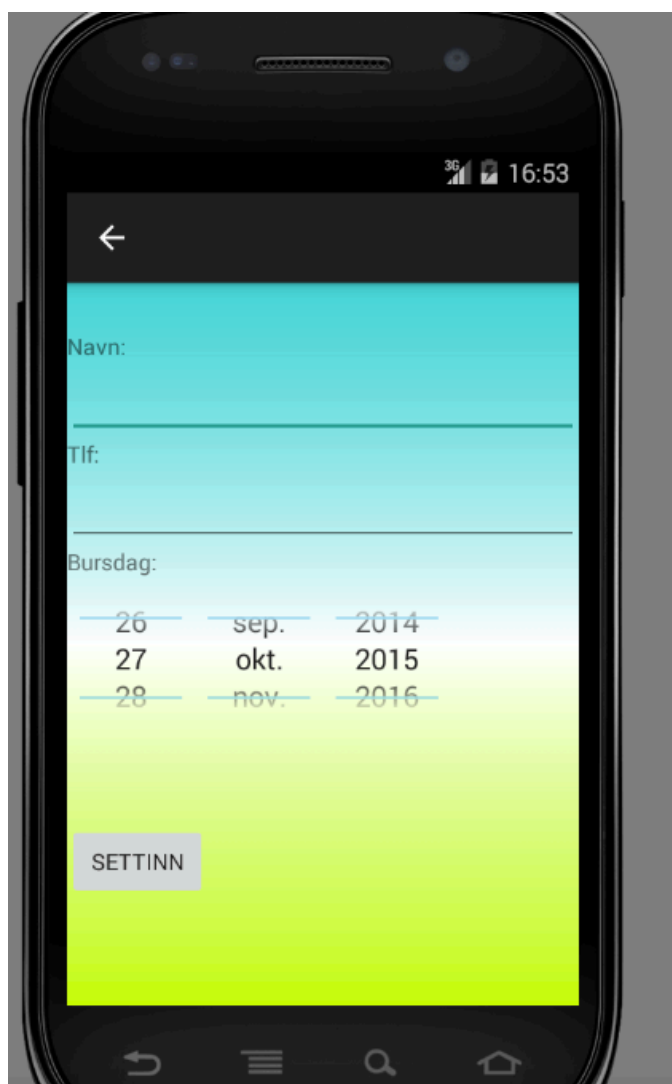
1.2.1

Hovedsiden



Hoved siden er en liste av alle personer som er registeret. Det er samme liste man har i databasen som hentet ut og vises på hovedscreenet. Alle navnene er klikkbare som kan føre brukeren til en annen screenside. Brukeren kan bruke action baren for å navigere mellom sidene. På action baren brukte vi to ikoner. Ikonene er veldig tydelige og brukeren forstår om den første fører til Meldingsiden og den andre fører til register siden.

1.2.2 Registerside



Register siden er den screensiden som brukeren kan registrere personer. Denne siden består av to felter navn og telefon nummer og en datePicker som

inneholder dag, måned, og årstall. Her man kan registrere navn, telefon nummer, og bursdag dato. Etter man fyller skjemaet klikker man på sett Inn knappen for å registrere og går man videre til listen i hovedsiden som er oppdatert med navnet til den personen som er registrert sist.

Vi har bruker RegEx for å validere inputene. Man må skrive navn of telefon nummer for at personen skal registerets. Ingen mellomrom er tillat og telefon nummer må være eksakt 8 nummer.

1.2.3 Endreside



Endre siden er denne siden som brukeren oppdatere dataene for personer i listen. Som sakt er alle navnene klikkbare i listen. Når man klikker på en navnene i listen går man videre til oppdatering siden. Siden inneholder samme felter som registrering siden men hver felt inneholder navn, telefon nummer og fødselsdatoen til personen som man klikker på. Man kan endre dataene til personen og når man klikker på knappen oppdater går man til listen mens alle dataene oppdateres. Man kan endre navnet til personen for å se om navnet endres på listen. Hvis man skal slette personen klikker man på søppelbåten som er i menyen. Når man velger å slette personen slettes i listen og databasen også. RegEx reglene er samme reglr som vi har brukt på registrering.

1.2.3 SendSms

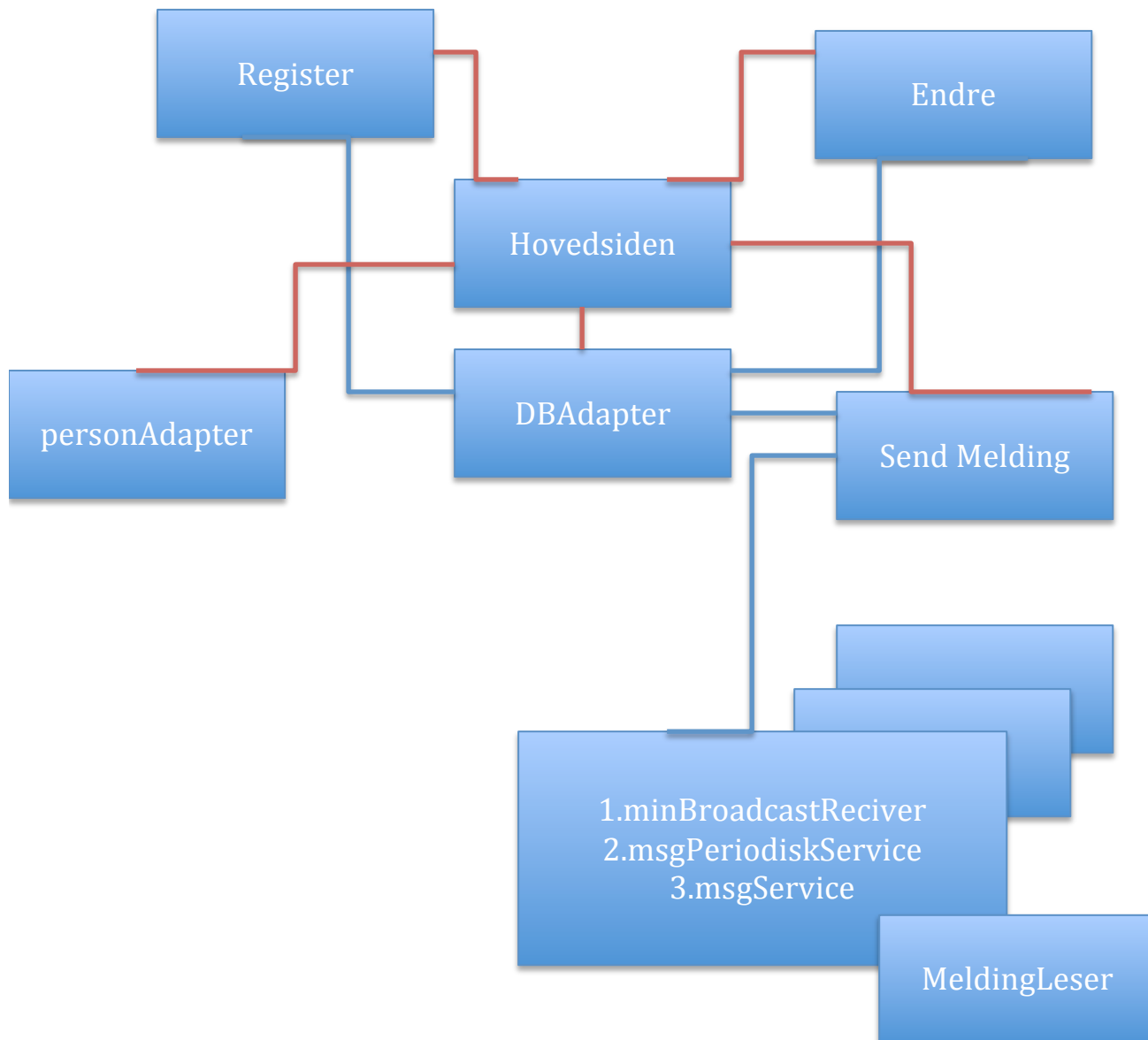


Denne siden inneholder en TimePicker som man bruker for å sette en tidspunkt for å sende smsen og en tekst felt som man skriver smsen inn i. Det står lagre knapp for å lagre smsen i shareperformances. Og det er en AV/På knapp som brukeren kan bruker for å slå på eller slå av sms tjenesten.

NB!

På første inneleringen vi fikk ikke til dette , men i dag etter vi fikk sjanse til å jobbe med det fikset vi denne delen. Nå Alt fungerer OK ☺ . Takk Torunn

1.3 Klassediagram



Klassediagrammet for applikasjonen ser ut slik. Appen har aktivitetene "HovedAktivitet", "Endre", "Register" og "send melding". I tillegg har vi brukt mange hjelpe klasser. DBAdapter klassen brukes for å programmere og lage databasen. PersonAdapter klassen brukes for å liste ut listen på knapper og gjøre navnene klikkbare. Klassene "minBroadcastReciver", "msgReciver", og "msgServic" skulle brukes for å sende smsen. Fra Hoved klassen kan man navigere seg til endre, register og sendMelding. Alle aktivitetene er koblet til DBAdapter fordi for å registrere, endre eller send melding burde vi ha tilgang til databasen. På diagrammet brukte jeg blå linje for å vise tilkoblingen fra DBAdapter til alle aktivitetene og rød linje for å vise tilgangen fra hovedsiden til alle klassene og aktivitetene. Melding Leser klassen er en java klassen som bruker vi til å lese og skrive ut den melding som brukeren skriver i feltet. M.a.o denne klassen lagrer smsen som brukeren skriver og skriver den igjen og hindrer om sms teksten blir borte til selve brukeren endrer eller sletter det.

Viktigste funksjonene som vi har brukt er:

- **onCreat(SQLiteDatabase db):** det er metoden som brukte vi for å lage databasen.
- **oppdater(...):** det er methoden som brukte vi for å oppdatere databasen og listen som vi har på appen etter vi endrer data av personene i listen.
- **slett(id):** er metoden for å slette peroner fra listen.
- **finnPersonMId(id):** er metoden for å finne personer fra databasen.
- **stableArrayAdapter():** er en metode som vi bruker for å sende opprinnelige person Iden til de forskjellige aktivitetene .
- **personAdapter** klassen er en klasse som bruker vi for å gjøre listen på hoved siden klikkbare. I forhold til dagens mobil telefoner er kontakt listen til mobilene er klikkbare. Vi har tatt hensikt til dagens teknologi og brukte vi samme metode på vår app.
- **En del funksjoner og klasser som vi brukte for å sende smsen.**

1.4 Navigasjon

Som har forklart på forrige punkter brukte vi en blanding av knapper og actionbar. Brukeren kan bruke action baren i hoved aktiviteten for å navigere seg til forskjellige aktivitetene. I hvert aktivitet har en PIL i action baren som fører brukeren tilbake. Jeg har brukt knappene settinn og oppdater og lagre for å registrere og oppdatere data av personer. Slett er item som ligger i Endre aktiviteten og den er i actionbaren som brukes for å slette brukeren.

1.5 Design

På designet brukte vi vanlig farger som vi har blandet med hverandre. Bakgrunn fargen ble lagde av en XML file som vi har i drawable mappen.

NB! Vi har ikke brukt noe design for legende versjon og språket også fordi vi rakk ikke å gjøre det og det var ikke krav på obligen.

1.6 Konklusjon

Dette obligen var mye forskjellig fra den forrige. Vi lærte mye om hvordan man lager en applikasjon som har database. Som sakt det var en del vanskelighet på dette prosjektet , spesial sms delen. Det var en utfordring som vi opplevde og vi prøver å gjøre best på neste applikasjon.