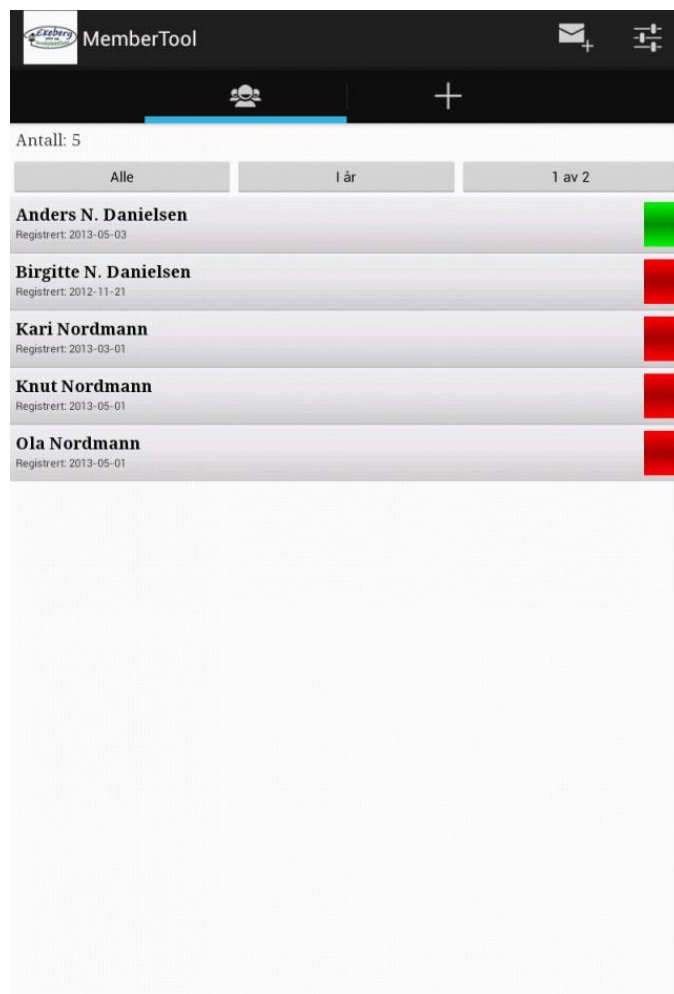


Android apputvikling

Våren 2013



Anders N. Danielsen

2IA

s180475

Innledning

Denne mappeinnleveringen var av det spennende slaget siden vi selv kunne velge hva vi ville utvikle.

MemberTool er resultatet. Denne er en app som holder oversikt over alle medlemmer i en idrettsklubb, om de har betalt, sende ferdiglagde mailer til én eller alle medlemmer, og selvfølgelig redigere dette. Som medlemsansvarlig i en frisbeegolfklubb kommer dette til å bli et meget nyttig verktøy.

Arbeidet har krevd mye tid og tålmodighet, og mange timer har gått med til å lære nye ting, lete etter feil m.m., men jeg sitter igjen med et resultat som jeg er stolt over.

Utstyr

Jeg er eier av en HTC Sensation med API 15(4.0.3), og har brukt denne under utvikling siden det er denne jeg kommer til å bruke også når jeg tar applikasjonen i bruk. Derfor har jeg heller ikke tatt hensyn på å gjøre appen kompatibel med eldre versjoner enn API 14.

Applikasjonen er også testet med en emulator av typen Nexus 7(7.27') med API17. For å kunne sende mail med emulatoren må det legges inn en mailkonto i innstillingene til emulatoren. Gjør man ikke det vil man få opp et dialogvindu om at ingen applikasjoner for sending av mail finnes. Dette fungerer utmerket på min mobile enhet.

Oppbygging

MainActivity er som navnet tilsier hovedaktiviteten. Denne oppretter 2 navigasjons-tab'er, og har ansvaret for kommunikasjon med Fragment'ene «AllMembersFragment», «NewMemberFragment», «DatePickerFragment» og «SettingsFragment» via «SettingsActivity» i tillegg til å kommunisere med aktivitetene «EditMemberActivity», «EditPayments» på vegne av nevnte fragmenter siden disse kun skal forholde seg til aktiviteten som startet den opp. MainActivity implementerer flere interface for å kommunisere med nevnte fragmenter.

Det er kun MainActivity som har en menu-xml-fil knyttet til seg. Det er fordi denne aktiviteten stort sett er aktiv, og aktiviteter utenom denne er kun ment til å endre/slette detaljer, og trenger derfor ikke tilgang til elementer i actionBar. Siden min mobil(og alle Android) har en «tilbake»-knapp har jeg vurdert det slik at jeg ikke trenger dette i denne applikasjonen. Fordi hovedaktiviteten er så stor del av applikasjonen er det aldri mer enn 2 klikk for å avslutte.

Det finnes en annen menu-xml-fil med navn «contextmenu». Denne angir utseende på menyen som dukker opp ved lange trykk på listen over medlemmer, og gir forskjellige muligheter som f.eks. å redigere valgte medlem eller sende en mail til denne personen.

I mappen «xml» under «res» ligger det en fil som heter preferences.xml. Denne angir utseende på innstillings-skjermen. Til elementene under kategorien «Maler» tilhører det en ny skjerm med preferences som angir tekst for emne og melding som brukes i ferdiglagde mailer. Legg merke til at det i begge meldingene er skrevet «Sett inn %1&skr til kontonummer %2&s...». Disse spesielle strengene blir byttet ut med beløp og kontonummer fra preferences før teksten sendes til en mail-applikasjon.

Utseende på hovedaktiviteten består hovedsakelig av en ListView som inneholder alle medlemmene som ligger lagret i databasen. Hver rad i lista er stylet ved hjelp av xml. List_selector angir hvilke

bakgrunner som skal bli brukt i de forskjellige tilstandene, og skifter da mellom `gradient_bg` og `gradient_bg_hover`. Hver rad i lista blir fylt ut ved hjelp av `MemberAdapter` som bruker `layouten` angitt i filen `list_item` i mappa `layout`. `List_item` angir er `ImageView` som bruker to andre xml-filer for å tegne sitt bilde: `gradient_paid` og `gradient_notpaid`. `MemberAdapter` velger hvilke av disse filene som skal bli satt ut i fra om medlemmet har betalt medlemskontingenten for inneværende år. Grønn farge betyr betalt, og rød det motsatte.

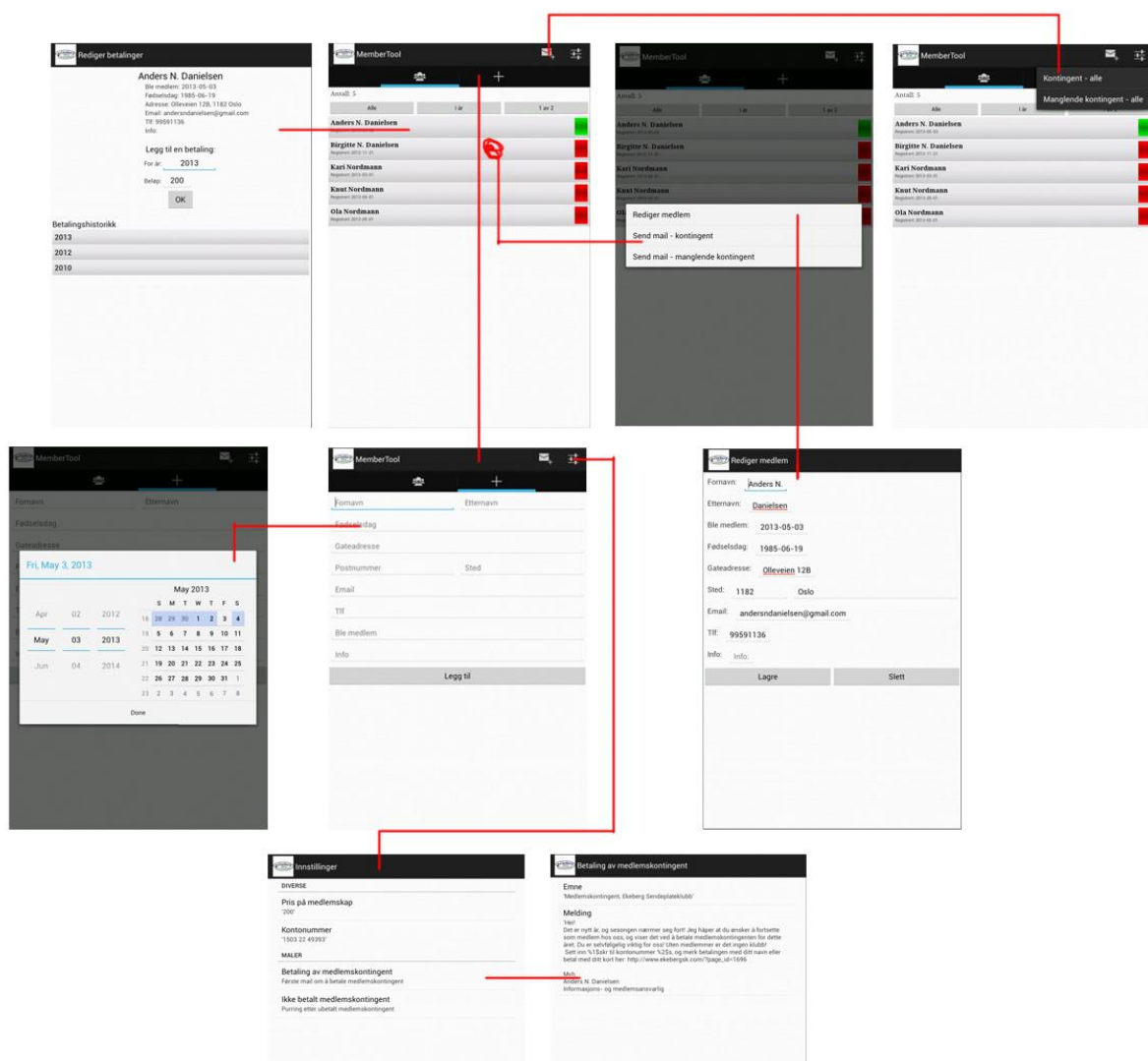
For å lagre all data er det tatt i bruk `MySQL`-databaser i stedet for intern lagring på mobilen. Dette fordi disse dataene kommer til å bli brukt i flere forskjellige sammenhenger, og man har derfor behov for å ha tilgang til samme oppdaterte data. For å få til dette benytter applikasjonen seg av klassen `AsyncTask` som med `HttpRequest` og `JSON` henter eller legger inn data i databasen ved hjelp av php-filer på server.

Databasen består av 3 tabeller:


`esk_members(id, firstname, lastname, start_membership, birth, streetadr, postnr*, email, tlf, info)`



`esk_members_city(postnr, city)`

`esk_members_paid(id, date, year, amount)`



Hovedside

MemberTool



Antall: 5

Alle

I år

1 av 2

Anders N. Danielsen

Registrert: 2013-05-03

Birgitte N. Danielsen

Registrert: 2012-11-21

Kari Nordmann

Registrert: 2013-03-01

Knut Nordmann

Registrert: 2013-05-01

Ola Nordmann

Registrert: 2013-05-01

Denne hovedsiden lar oss se alle medlemmer ut i fra 3 kriterier:

- Alle som har betalt for inneværende år.
- Absolutt alle som ligger inne i databasen
- Alle som har betalt i fjor eller inneværende år eller begge årene.

Dette er kriteriene som vår klubb har behov for å kunne skille mellom.

Rett over listen med medlemmer telles antall elementer/medlemmer i gjeldene visning.

Lista gir rask oversikt over navn på medlem, når personen ble registrert som medlem, og om personen har betalt eller ikke. Om man trykker på «I år» så vises ingen farge. Dette fordi alle som faller under dette kriteriet har betalt, og ville vært grønne alle sammen.

Utgangspunkt

Jeg har lagt inn 5 personer ved innlevering av denne oppgave:

Anders N. Danielsen – Betalt i 2010, 2012, 2013, og vil derfor vises i alle kategoriene.

Birgitte N. Danielsen – Ingen registrert betaling, og vil kun vises i kategorien «alle».

Kari Nordmann – Betalt i 2012. Vil også vises i «1 av 2».

Knut Nordmann – Betalt i 2012. Vil også vises i «1 av 2».

Ola Nordman - Ingen registrert betaling, og vil kun vises i kategorien «alle».

Applikasjonen vil ikke bli tatt i bruk før karakter er gitt ut så følg deg fri til å gjøre som du vil så lenge.

Problemer underveis

Jeg brukte mye tid på å få AsyncTask til å fungere riktig, og få ut det jeg var ute etter. Appen klikket ofte pga. bruk av feil tråder osv samtidig som jeg stadig lærte meg mer og mer.

Til og starte med bestod MainActivity kun av 2 knapper: «Se alle» og «Legg til». Jeg ønsket selvfølgelig et finere utseende og valgte derfor å gå for navigation-tab i actionBar. Da måtte jeg gjøre om allerede store aktiviteter til Fragment, noe som tok mye tid, og mye tenking. Fragment synes jeg er vanskelig å få helt taket på, men har lært meg mye av å slite med dette.

Svakheter/videreutvikling

Appen lagrer til databasen ved enhver forandring noe som tar mer tid enn om jeg hadde lagret innimellom. Jeg tenker at det kunne vært lurt og kun lagt til/slettet elementer fra listene for så å lagre til database ved onPause eller noe. Da burde jeg kanskje lagre dette med en Service som jobber i bakgrunnen.

Det er ikke brukt noen form for «escape» av input, og her må dette tilføres slik at det ikke er mulig å ødelegge/slette databaseinnhold for de med onde hensikter.

Gi muligheter for å sortere listene etter forskjellige valgmuligheter som fornavn, etternavn, dato betalt osv.

Applikasjon i mappe2 skal inn i denne applikasjonen, men denne har jeg ikke prioritert i denne omgang. Det hadde vært bra å implementert bursdags-sms'er i denne!

Lage innlogging.

Konklusjon

Jeg sier meg veldig fornøyd med denne prosessen, og med resultatet som jeg sitter igjen med; enda en applikasjon som jeg vil få bruk for i mitt styreverv som medlemsansvarlig i en frisbeegolfklubb. Nå har jeg full tilgang til det jeg trenger hvor som helst, og gjerne på frisbeegolfbanen.

Jeg kommer til å videreutvikle denne appen ut i fra ovenfor nevnte feil/mangler, og om andre klubber ser nytten av denne kan jeg til og med dele den ut.

Ved å bruke mange timer på denne oppgaven har jeg satt meg inn i mange situasjoner, lest masse på nett, og lært mye mer enn jeg ville gjort uten å ha en spesifikk oppgave.