

Mappe 2 Høst 2022

Av Vilde Kure, s354592

Innhold

1. Innledning	2
2. Design	3
3. Funksjonalitet.....	7
4. Diskusjon	13
5. Konklusjon.....	13
6. Kilder	14

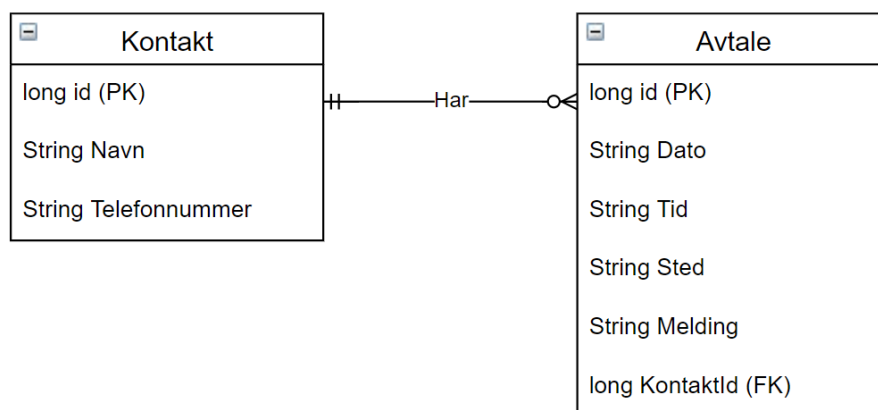
Figurliste

Figur 1-1: ER-Diagram.....	2
Figur 2-1: Skisse av app	3
Figur 2-2: Forside Kontakter og Avtaler	4
Figur 2-3: Legg til og endre kontakt side	5
Figur 2-4: Legg til eller endre avtale side	5
Figur 2-5: Dato sitt input felt når den er aktiv.....	6
Figur 3-1: ListView over Kontakter og Avtaler.....	7
Figur 3-2: Lagre og endre sidene til Avtaler	7
Figur 3-3: oppretting av SQLite database i koden	8
Figur 3-4: Kode som viser at Datoen sammenligner Stringer.	8
Figur 3-5: Input felt til dato for å se hjelpemelding til dato format.....	9
Figur 3-6: Notifikasjon vises	9
Figur 3-7: Standardmelding som sendes	10
Figur 3-8: Melding en selv har skrevet sendes	10
Figur 3-9: Preferanse side.....	11
Figur 3-10: Forespørsel om å sende SMS	11
Figur 3-11: Navigasjon på appen	12

1. Innledning

I denne rapporten skal det legges frem utviklingen en applikasjon eller en app for lagring av kontakter. Først så skal designvalgene bli forklart, den startet som en skisse og ble så videre utviklet. Videre blir det sett på de funksjonelle kravene som er satt og hvordan appen gjennomfører disse. Så diskuteres mulige forbedringer eller ting som kunne vært endret for fremtidig utvikling av produktet.

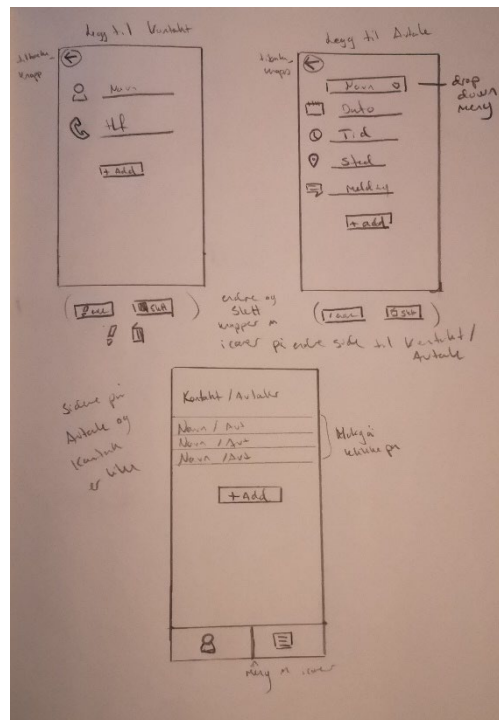
For å få et bedre bilde av databasen som skulle lages ble det utviklet et ER-diagram som visualiserte den (Figur 1.1). Et ER-Diagram er en visuell beskrivelse av en datamodell, som består av entiteter, attributter og forholdet mellom entitetene, som gjerne blir brukt til å grafisk fremstille databasestrukturen i et system. Entiteten vil si objektet som i dette tilfelle er Kontaktene og Avtalene, attributtene vil si egenskapene til entitetene som navn, telefonnummer, dato og lignende. Forholdet mellom de er visualisert ved å trekke linjer mellom entitetene og i denne oppgaven blir det tatt i bruk et en-til-mange relasjon som sier at en avtale må ha en kontakt, men en kontakt kan ha null til mange avtaler (Kristoffersen, 2020, s.173, 177, 179-182, LucidChart, u.å.).



Figur 1-1: ER-Diagram

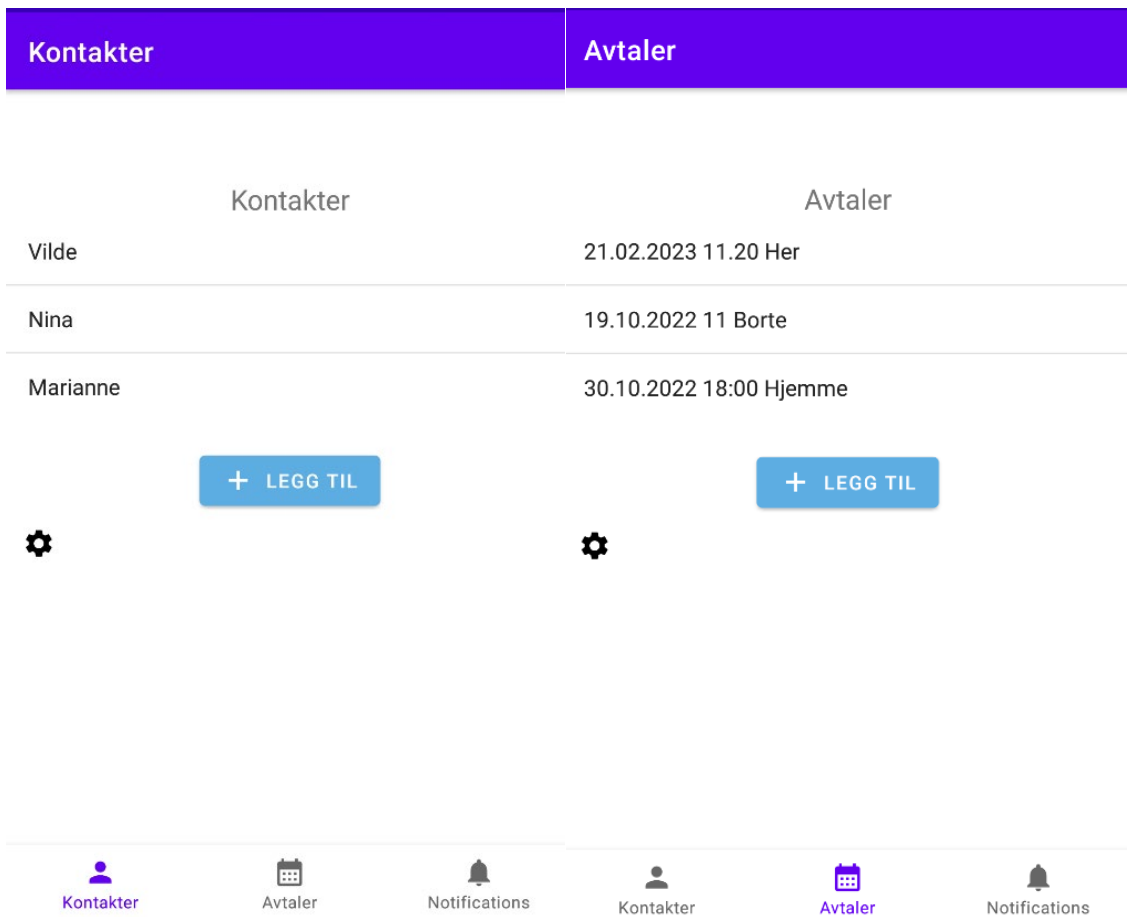
2. Design

Etter ER-diagrammet var klart ble det utviklet en skisse for å ha en ide av hvordan appen skulle se ut (Figur 2.1). Skisser brukes til å uttrykke ideer og er vanligvis enkle frihåndstegninger som sender signaler om uferdig arbeid. De tar kort tid å lage for å fange ideer på papir før de glemmes, men inneholder tilstrekkelig med detaljer som gjør at de som ser den skjønner hva den symboliserer, og hva som er fokus (Sandnes, 2018, s.248 – 250)



Figur 2-1: Skisse av app

Skissen inneholder enkel informasjon om hvordan jeg ønsker å utforme appen, med at det er en hjemmeside med to menyer og to ekstra aktiviteter for å lage, endre eller slette info for både kontakter og avtaler. Før å få det inn i Android studio ble det laget et nytt prosjekt som inneholder 3 bunnmeny elementer eller fragmenter, så hele appen tar da i bruk både fragmenter og aktiviteter, da det er aktiviteter som sender brukeren fra forsidene til lagre, endre eller slette sidene. Figur 2.2 viser designet av forsidene.



Figur 2-2: Forside Kontakter og Avtaler

Hvor elementer er plassert i forhold til hverandre har noe å si, derfor er det viktig at de grupperes sammen eller har forskjellig størrelse eller form, for å vise relasjon til hverandre (Cooper et al., 2014, s. 409, 413, Sandnes, s 64, 67 - 69). Når man kommer inn på forsiden vil man se de lagrede kontaktene bli listet ut under hverandre og har likt utseende og at legg til knappen er for seg selv, det samme med innstillingsknappen. Dette er en metode for å få produktet til å virke mer organisert og systematisk ut, ved at ting er justert vertikalt eller horisontalt, på linje eller ha jevne mellomrom (Cooper et al., 2014, s 414-415, Sandnes, s 72-73). Dette kommer også igjen på legg til og endre siden til kontaktene og avtalene (Figur 2.3 og 2.4).

Figur 2-3: Legg til og endre kontakt side

Figur 2-4: Legg til eller endre avtale side

På disse sidene ser man at input boksene er plassert på linje med hverandre vertikalt og er like store for at det skal være mer organisert, knappene er plassert på midten under inputfeltene for å komme i fokus. På endre-siden er det to knapper, en endre og en slett som er på linje horisontalt. Tilbakeknappen er plassert øverst i venstre hjørnet der de av konvensjon pleier å være plassert. Designet til appen har minimalt med elementer for mer systematikk og organisering da det er en visuelt effektivt og hvert element har en funksjon i designet. Alt som er plassert er der for en grunn og er konsistent, effektivt og funksjonelt (Cooper et al., 2014, s.423).

Når det kommer til fargevalg er det tatt i bruk få farger for å få fokuset på de riktige stedene, dette er bevisst med tanke på brukerens mål som er å lage kontakter og avtaler, så fokuset ikke havner overalt. Farge er brukt for å tydeliggjøre informasjon, og gjøre det enkelt å finne informasjonen (Cooper et al., 2014, 407, Sandnes, s.115). Det er knappene i appen som har tatt i bruk mest farge, og utgangspunktet er i slett knappen sin farge, som er rød, en konvensjon for feil eller fjerning av noe. Legg til og endre er i blå som er splittkomplementært til rødt, da rød er naboen til blå sin komplementærfarge oransje, dette vil oppleves mer harmonisk enn bare en komplementærkontrast samtidig som grønn, som er rød sin komplementære kontrast, kan skape problemer for de med rød/grønn fargeblindhet (Sandnes, s.110, 114). Det er også lagt til blå rammer på inputfeltene når brukeren klikker på de for å vise at de er aktive, på dato input er det også lagt til et hjelpefelt for å vise hvordan brukeren skal skrive dato, for at det ikke skal skje feil i funksjonen som jeg kommer tilbake til senere i oppgaven (Figur 2.5). Ved at farger og font gjennomgående på alle siden vil designet være konsistent og fungere harmonisk med hverandre for å gi brukeren en mer positiv erfaring (NIBusinessInfo, u.å).



Figur 2-5: Dato sitt input felt når den er aktiv

I denne appen er det også tatt i bruk ikoner eller grafiske symboler, fordi de tar mindre plass i brukergrensesnittet og er mye brukt i mobile applikasjoner hvor det er mindre plass på skjermene. Enkelte ikoner har per konvensjon blitt tilordnet spesielle betydninger, men er også avhengig av at brukeren vet eller lærer betydningen av de (Sandnes, 2018, s.92 – 93). I denne appen er det brukt en del ikoner som er ganske selvforklarende, som en telefon for å symbolisere telefonnummer, kalender for dato, klokke for klokkeslett med mer. Samtidig som det er ikoner på inputfeltene og knappene er det også lagt til tekst i tilfelle det ikke er en selvfølge at konvensjonene er forståelige. På inputfeltene er teksten i de som hint, men når man klikker på dem, blir teksten flyttet til rammen på feltet som man kan se på figur 2.5.

3. Funksjonalitet

Appen som er utviklet skal kunne registrere venner med navn og telefon, og det skal være mulig å endre og slette på disse, de skal også listes ut i et ListView. Appen skal også kunne registrere avtaler med dato, sted, klokkeslett og melding som også skal listes ut og kunne slettes. Figur 3.1 viser kontakter og avtaler som er lagret, bli listet ut i et ListView, viktig å merke seg at disse blir listet ved at appen starter, så for å få de opp må man starte appen på nytt, men de er lagret i databasen. Hvis man lager en kontakt og går rett inn på å lage avtale så vil kontakten eksistere, men den blir ikke listet før appen re-startes, usikker på hvordan jeg kan fikse det problemet.

Kontakter	Avtaler
Vilde	21.02.2023 11.20 Her
Nina	19.10.2022 11 Borte
Marianne	30.10.2022 18:00 Hjemme

Figur 3-1: ListView over Kontakter og Avtaler

Figur 3.2 viser eksempel på lagre og endre side til avtaler, den er like kontakter, bare at kontakter har kun navn og telefonnummer felt som vi har sett tidligere i oppgaven. Det er mulig å lage, endre og slette både kontakter og avtaler i appen. For å komme inn på endre-siden så klikker man på den avtalen, eller kontakten i ListView, hvor da feltene er fylt ut med infoen som er registrert om denne i databasen.

←

Vilde ▾

Dato

Klokkeslett

Sted

Melding

+ LEGG TIL

←

Vilde ▾

Dato

Klokkeslett

Sted

Melding

ENDRE SLETT

Figur 3-2: Lagre og endre sidene til Avtaler

Som nevnt er alt lagret på en lokal database på telefonen, og i denne oppgaven ble det tatt i bruk en SQLite database. Figur 3.3 viser opprettingen av databasen i onCreate funksjonen.

```
@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_TABLE_CONTACTS = "CREATE TABLE "
        + TABLE_CONTACTS
        + "(" + CONTACT_KEY_ID + " INTEGER PRIMARY KEY, "
        + CONTACT_KEY_NAME + " TEXT, "
        + CONTACT_PH_NO + " TEXT" + ")";
    db.execSQL(CREATE_TABLE_CONTACTS);

    String CREATE_TABLE_APPOINTMENTS = "CREATE TABLE "
        + TABLE_APPOINTMENTS
        + "(" + APPOINTMENT_KEY_ID + " INTEGER PRIMARY KEY, "
        + APPOINTMENT_KEY_DATE + " TEXT, "
        + APPOINTMENT_KEY_TIME + " TEXT, "
        + APPOINTMENT_KEY_LOCATION + " TEXT, "
        + APPOINTMENT_KEY_MESSAGE + " TEXT, "
        + APPOINTMENT_KEY_CONTACT_ID + " TEXT, "
        + "FOREIGN KEY (" + APPOINTMENT_KEY_CONTACT_ID +
        ") REFERENCES " + TABLE_CONTACTS + "(" + CONTACT_KEY_ID + ")";
    db.execSQL(CREATE_TABLE_APPOINTMENTS);
}
```

Figur 3-3: oppretting av SQLite database i koden

Videre skal appen en gang i døgnet sjekke om det er noen avtaler den dagen, og hvis det er det skal brukeren få opp en notifikasjon om avtalen. I tillegg skal det sendes en SMS med en påminnelse til kontakten som har avtalen.

```
Calendar dato = Calendar.getInstance();

int year = dato.get(Calendar.YEAR);
String yearString = Integer.toString(year);

int month = dato.get(Calendar.MONTH) + 1;
String monthString = Integer.toString(month);

int day = dato.get(Calendar.DAY_OF_MONTH);
String dayString = Integer.toString(day);

String todayDate = dayString + "." + monthString + "." + yearString;

if (todayDate.equals(appointmentDate)) {
```

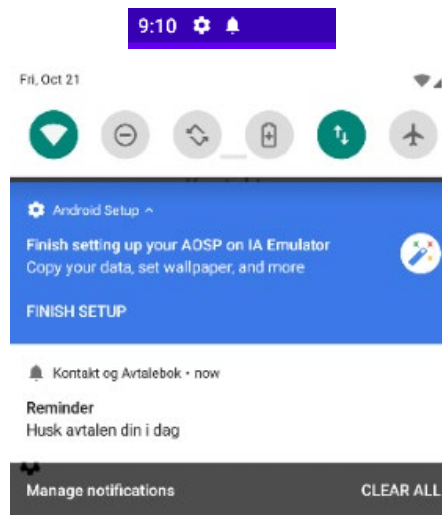
Figur 3-4: Kode som viser at Datoen sammenligner Stringer.

Figur 3.4 viser noe av koden som skal sjekke om datoen matcher, det man kan merke seg er at den skal matche en string og ikke en dato så det er viktig at brukeren setter inn dato i riktig format. Derfor er det en hjelpesetning i input feltet på dato (figur 3.5) som viser akkurat hvordan datoen skal skrives inn.



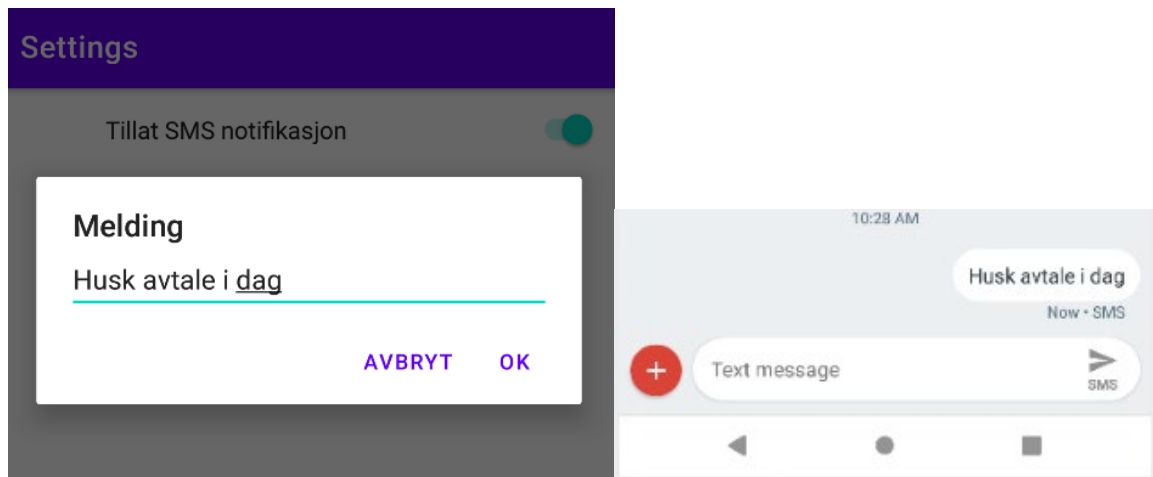
Figur 3-5: Input felt til dato for å se hjelpemelding til dato format

Denne funksjonen fungerer i appen, figur 3.6 viser testen som ble gjort av dette. Det er laget så appen viser et bjelle-ikon som er typisk for notifikasjoner på øverste linje av telefonen. Drar man ned menyen så kan man se notifikasjonen med beskjednen som sendes med.



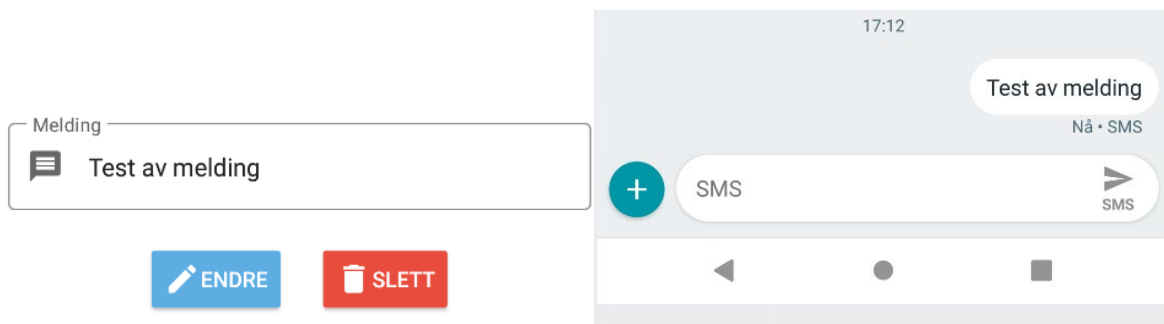
Figur 3-6: Notifikasjon vises

Dersom det er registrert en med melding med avtalen, skal denne meldingen sendes, og hvis ikke det er lagret noe sendes en standard melding fra SharedPreferences. Figur 3.7 viser at man kan sette en standard melding i preferanser og at den ble sendt under en test av appen. Da emulatoren ikke har Sim kort ble det ikke faktisk sendt en melding, men man kunne sjekke meldingsappen for forsøkt utsendte SMS 'er.



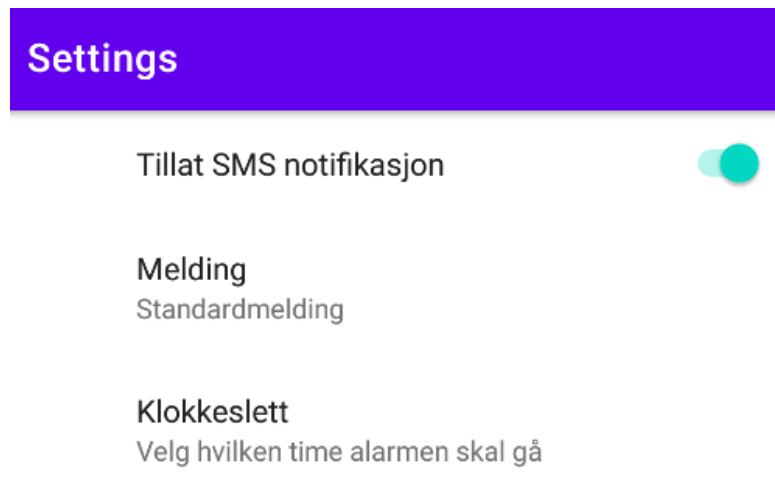
Figur 3-7: Standardmelding som sendes

Skriver man en melding i avtalen man lager så er det den som sendes i stedet (Figur 3.8).



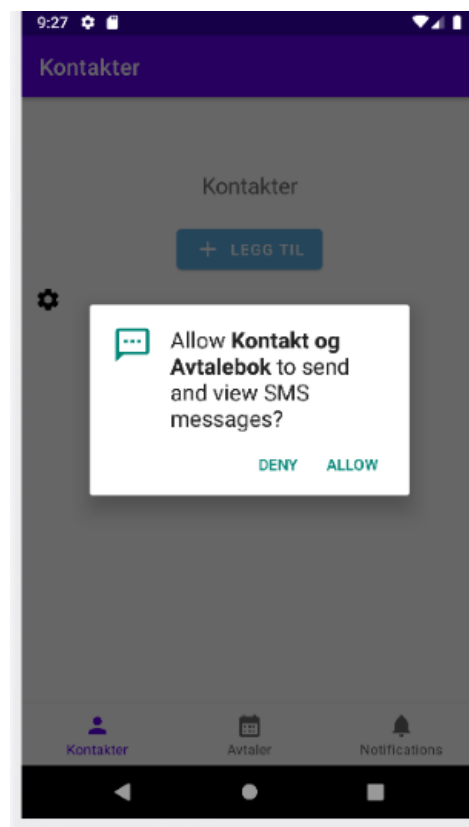
Figur 3-8: Melding en selv har skrevet sendes

Tidspunktet man ønsker at SMS skal sendes skal også lagres i SharedPreferences og det skal være mulig å slå av og SMS-tjenesten. Det er da lagt en settings side hvor man kan slå dette av og på (figur 3.9). Det eneste problemet er at appen er satt kun til å sende på hele timer, da jeg ikke delte opp tiden til å sette timer og minutter. Derfor gir det bekjed om å velge hele timer.



Figur 3-9: Preferanse side

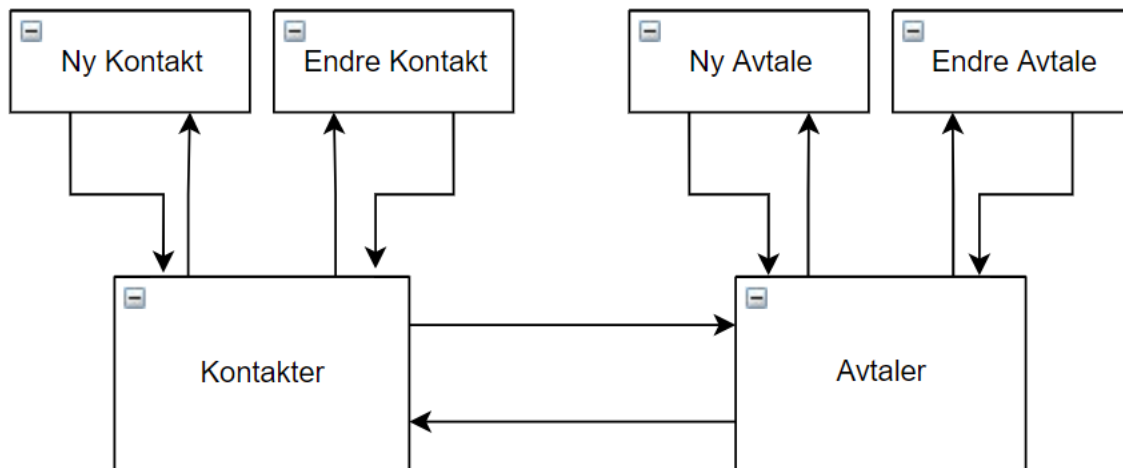
Det er også lagt til en forespørsel om å la appen tillate å sende SMS' er når appen starter opp (figur 3.10).



Figur 3-10: Forespørsel om å sende SMS

Når tjenesten slås på så sendes det et broadcast til MinBroadcastReceiver som kaller på en service i MinPeriodiskService, den setter en AlarmManager til å kjøre i MinSendService som ser etter påminnelser til en tid som er bestemt i preferences.

Et godt datasystem måles ofte fra hvor godt funksjonaliteten samsvarer med brukerens oppgaver og støtter deres behov, eller at man finner det man leter etter, for eksempel ved bruk av Don Normans action cycle. Hvis brukeren klarer å gjøre de tingene appen sier de kan gjøre, er det å oppnå et mål og brukergrensesnittet fungerer som det skal (Sandnes, 2018, s.22, 42-43, Spence, 2014, s.175-183). Figur 3.11 viser hvordan navigasjonen fungerer på appen så man ser hvordan de henger sammen. Appen skal være intuitiv i forhold til hva man ønsker å oppnå og det gir mening å ha legg til og endre kontakt koblet opp til kontakt siden og avtaler der de skal være. Knappene som vi har sett på alle sidene skal også følge konvensjoner så de skal være lette å forstå, trykker man en pil tilbake kommer man tilbake, søppeldunk er slett, pluss er legg til noe og hvis man klikker på en kontakt eller en avtale åpner man den.



Figur 3-11: Navigasjon på appen

4. Diskusjon

I denne appen er det tatt i bruk farger kun på knappene for å gi de mer mening sammen med ikonene på disse, men det er viktig med appell som engasjerer brukeren. Mangelen på eventuell bakgrunnsfarge gjør at det faller litt, og er kanskje ikke like appellerende på et konkurrerende marked (Sandnes, 2018, s.22). Grunnen til at appellen er nedprioritert er for å bruke mer tid på god funksjonalitet, fordi et produkt som ser fint ut, men ikke fungerer, gjør lite for seg.

Jeg hadde også problemer med de tre fragmentene som danner menyen på bunnen. Grunnen til at det er 3 meny elementer er for å ikke ødelegge appen da det skapte mye feil når jeg forsøkte dette, men den siste menyen kan være et godt utgangspunkt for videre utvikling av for eksempel en notifikasjons-setting-side som er det jeg har lagt det opp til å bli.

Selv om et symbol eller et ikon er standardisert, så er det ikke ensbetydende med at brukeren vil forstå hva det betyr, og det er ingen garanti for at ikonet har en konvensjonell betydning for alle brukere (Sandnes, 2018, s.93) Derfor ville en brukertest være viktig for videre utvikling av appen, for å se hvilken forståelse brukeren har til appen og alle betydningene av ikoner og lignende. Nå er det også lagt til ord ved ikonene for å forsikre at betydningen kommer frem, men det er andre aspekter som kanskje ikke er like intuitivt, slik som å trykke på listen over kontakter eller avtaler for å komme til endre-siden.

5. Konklusjon

I denne oppgaven ble det utviklet en app for lagring av kontakter og avtaler som tilhører disse kontaktene. Det ble utviklet skisser og er-diagrammer for bedre forståelse av hvordan appen skulle se ut og fungere rent teknisk, men i designet er det fortsatt rom for mer utvikling for å fremme mer appell for brukeren, men den har alle funksjonelle krav som ble satt.

Da det ikke er gjennomført en brukertest, er det vanskelig å si hva i både design og funksjon som må forbedres eller endres, derfor er det neste steg for en videre utvikling av appen. For eksempel ved å legge innstillinger eller preferanser inn på notifikasjons menyen/fragmentet som ikke ble fjernet i utviklingen, i stedet for hjulet som er på de andre sidene, da det er en bedre og mer intuitiv plassering av dette.

6. Kilder

Cooper, A., Noessel, C., Cronin, D., Reimann, R. (2014). *About Face: The Essentials of Interaction Design* (4.utg.) John Wiley Sons Inc.

Kristoffersen, B. (2020) *Databasesystemer*. (5.utg) Universitetsforlaget.

LucidCharts (u.å) *What is an Entity Relationship Diagram (ERD)?* Hentet 26.10.2022 fra <https://www.lucidchart.com/pages/er-diagrams>

NIBusinessInfo (u.å) *Best practice in web design*. Consistency in web design. Hentet 14.09.2022 fra <https://www.nibusinessinfo.co.uk/content/consistency-web-design>

Sandnes, F. E. (2018) *Universell utforming av IKT-systemer: Brukergrensesnitt for alle*. (2.utg.) Universitetsforlaget.

Spence, R. (2014). *Information Visualization: An Introduction*. (3.utg.) Springer