**Mappe 2: Rapport - s358979**

# Innledning

AvtaleManager er en app som er skapt for å forenkle prosessen med å sende SMS-påminnelser til kontakter angående kommende avtaler. Med en praktisk tilnærming til funksjonalitet lar appen brukeren og deres kontakter holde oversikt over avtalte møter.

Utviklingen av appen begynte med etableringen av et grunnleggende brukergrensesnitt. Bruken av fragmenter ble foretrukket for å sikre enkel navigering mellom ulike deler av appen, med faste knapper for konsistent tilgang. Designet ble holdt minimalistisk, med et mål om å opprettholde brukervennlighet fremfor estetisk appell.

# Et bilde som inneholder tekst, skjermbilde, programvare Automatisk generert beskrivelseAvtaleFragment

Ved oppstart, presenterer AvtaleManager brukeren med AvtaleFragmentet som gir en oversiktlig visning av alle planlagte avtaler. Informasjon som dato, tid, sted og deltakere er synlig og gir en umiddelbar forståelse av hver enkelt avtale. Brukergrensesnittet tillater enkel fjerning av avtaler direkte fra listen for å holde oversikten oppdatert.

Et bilde som inneholder tekst, programvare, Font, Operativsystem

Automatisk generert beskrivelseLegging til nye avtaler er intuitivt plassert i toppseksjonen av fragmentet, hvor man kan inkludere detaljer for hver hendelse. En særegen funksjon er implementeringen av en tilpasset MultiSpinner-komponent, som muliggjør valg av flere kontakter til en avtale uten å begrense brukeren til et enkelt valg. Dette gir fleksibilitet og effektivitet i planleggingsprosessen, og sikrer at alle relevante parter enkelt kan inkluderes i en hver avtale.

# Et bilde som inneholder tekst, skjermbilde, programvare Automatisk generert beskrivelseKontakterFragment

I KontakterFragmentet kan brukere enkelt legge til nye kontakter ved å oppgi navn og telefonnummer. En egen knapp aktiverer visningen av alle lagrede kontakter i appens database, fremvist i et ListView. Dette gir en praktisk oversikt og enkel tilgang til kontaktdetaljer.

Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelseBrukerne kan administrere sine kontakter direkte fra ListViewet. Foruten å slette kontakter, tilbyr appen også en funksjon for å redigere kontaktinformasjon via en dialogboks som kan åpnes ved å velge en kontakt. Dette sikrer at brukeren kan holde kontaktlisten oppdatert og relevant uten å forlate appens kontekst.

# SettingsFragment

Et bilde som inneholder tekst, Font, skjermbilde, hvit

Automatisk generert beskrivelseEt bilde som inneholder tekst, skjermbilde

Automatisk generert beskrivelseNår vi trykker på Preferanser så sendes vi til en SettingsActivity, som starter en ny aktivitet, som igjen oppretter et nytt SettingsFragment. Her har vi tre ulike instillinger for SMS-tjenesten. Den første, og viktigste, er den som slår tjenesten av eller på. Når denne slås på så blir AlarmManageren inni MinPeriodisk aktivert. Deretter aktiveres selve SMS-tjenesten, som vi finner i MinSendService. Her gjør vi en daglig sjekk for om det er noen avtaler som er registrert for den gitte dagen. Vi henter ut informasjon fra Avtale- og Deltakelse-tabellene i databasen, og sender SMS til de kontaktene som er registrert med en avtaledato for dagens dato. I tillegg blir det gitt en notifikasjon på telefonen når dette skjer.

Som standard så blir denne sjekken og utsendingen av SMSer gjort kl. 06:00, men i preferansene er det mulig å endre dette til et annet tidspunkt. Dessuten er det mulig å endre på standard-meldingen som sendes til kontaktene. På dette bildet ser vi et eksempel av logging av hvordan en SMS kan se ut.

I dette tilfellet har vi i preferanser satt standardmeldingen vår til å være «Husk den viktige avtalen vi har!»

Vi restarter tjenesten hver gang noen av disse tjenestene blir endret på. På denne måten sørger vi for at appen alltid mottar de siste endringene, og agerer deretter.

# Bakgrunnstjeneste og BootReceiver

I tillegg til at SMS-tjenesten slås på når man slår den på, så kan appen også kjøre i bakgrunnen. Hvis vi trykker på den lille prikken nede i midten på telefonen, tas vi ut tilbake til Hjem-skjermen. Tidligere, før vi fikk implementert bakgrunnstjenesten, så ville SMS-tjenesten slå seg av etter 10-20 sekunder på grunn av inaktivitet. Ettersom at bakgrunnstjenesten ble lagt til, så kan man gå ut av appen på denne måten, og den vil fortsatt forbli på. Det kan for eksempel observeres gjennom varslene og loggingen.

I tillegg har jeg implementert en BootReceiver, noe som gjør at tjenesten starter når vi rebooter appen.

# Et bilde som inneholder tekst, skjermbilde, Font, nummer Automatisk generert beskrivelseDatabase-struktur

Databasen er designet slik at man kan ha flere kontakter per avtale. Dermed vil hver gang man registrerer en ny avtale, så vil det legges 1 ny rad i Avtale-tabellen, og x antall nye rader i Deltakelse-tabellen, alt ettersom hvor mange deltakere den gitte avtalen har. Dette tok det litt tid å implementere, men det var svært spennende å kunne observere hvordan de forskjellige delene av databasen vår samarbeidet for å holde styr på detaljene rundt hver avtale og dens deltakere.

I dette prosjektet har jeg brukt ROOM-biblioteket, som har tilført et moderne og robust databasehåndtering i appen. Med ROOM fikk vi forenklet databasemanipulasjon og spørringer, noe som ga en mer intuitiv forståelse av hvordan dataene strømmet og endret seg i appens livssyklus. Spesielt var det en befrielse å benytte ORM (Object-Relational Mapping) til å representere databasetabeller som Java-objekter, noe som gjorde koden renere og mer vedlikeholdbar.

Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse

Et bilde som inneholder skjermbilde, tekst, programvare, Multimedieprogramvare

Automatisk generert beskrivelseEt bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse

**Deltakelse-tabell**

**Avtale-tabell**

**Kontakt-tabell**

# Avslutning

Gjennom utviklingen av AvtaleManager har jeg fått en bedre forståelse for hvordan man kan bygge en funksjonell og brukervennlig app fra bunnen av. Dette har gitt meg verdifull innsikt i hvordan man kan utnytte Androids ulike funksjoner, som databaser, fragmenter og servicer, for å skape en helhetlig brukeropplevelse.

Det er noen begrensninger i appen som er verdt å merke seg. For eksempel er dato og klokkeslett implementert som strenger uten noen form for validering. Dette er en forenkling som fungerer for dette prosjektet, men som ville kreve en mer robust løsning i en fullskala app.

Designet er også noe som kunne vært forbedret for å gjøre appen mer tiltalende for brukeren. Jeg har ikke hatt mulighet til å teste SMS-funksjonaliteten fullt ut, da jeg ikke har en Android-telefon, men basert på varsler og logging, ser det ut til at denne delen av appen vil fungere som forventet.

Til tross for disse begrensningene, har arbeidet med AvtaleManager vært en lærerik prosess. Det har styrket min forståelse for Android-utvikling og gitt meg praktisk erfaring som jeg kan ta med meg i fremtidige prosjekter.