Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og realfag

Tidlig oppdagelse og varsling: Utvikling av en mobilapplikasjon for brannsikkerhet i sårbare befolkninger

Early detection and notification: Developing a mobile app for fire safety in vulnerable populations

Systemdokumentasjon

Versjon 1.0

REVISJONSHISTORIE

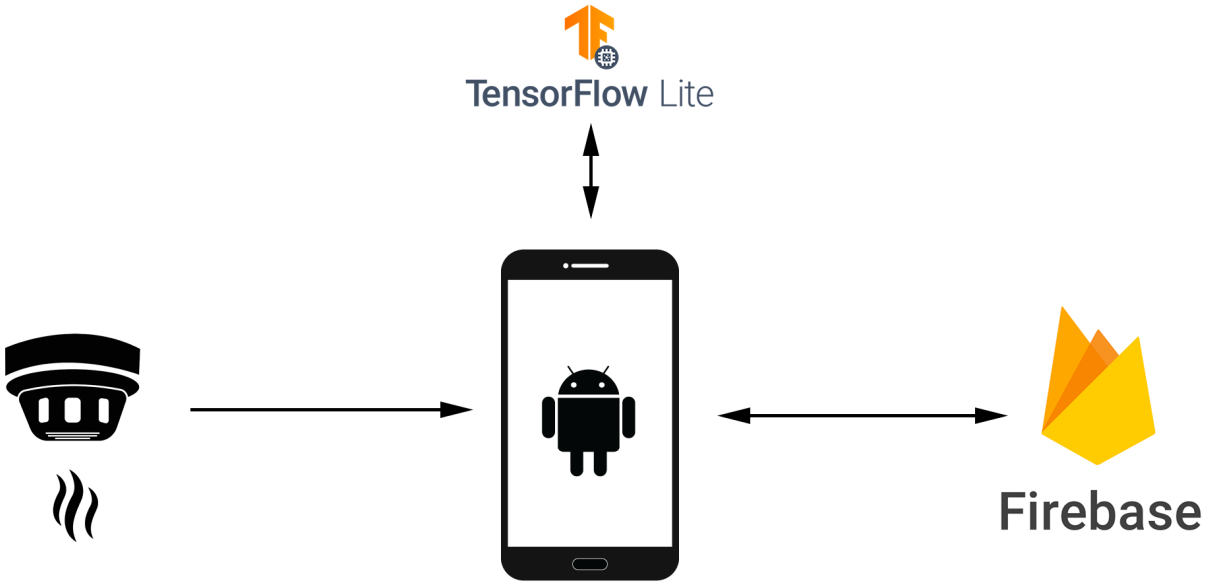
Dato	Versjon	Beskrivelse	Forfatter
21. Mai 2023	1.0	Endelig utgave ferdig	Torstein

INNHOLDSFORTEGNELSE

1 INNLEDNING

Denne systemdokumentasjonen presenterer resultatet av utviklingen. Applikasjonen benytter mikrofonen og lydgjenkjenning på brukerens enhet til å oppdage lyden av en røykvarsler, og sender deretter varsler til brukerens kontakter. Firebase-plattformen er integrert i applikasjonen for autentisering, databasefunksjonalitet og varsler. Dokumentasjonen gir en oversikt over systemets arkitektur, prosjektstruktur og viktige aspekter ved implementasjonen. Den er ment å være en verdifull ressurs for utviklere og interesserte parter som ønsker å utforske og bidra til prosjektets fremtidige utvikling.

2 ARKITEKTUR



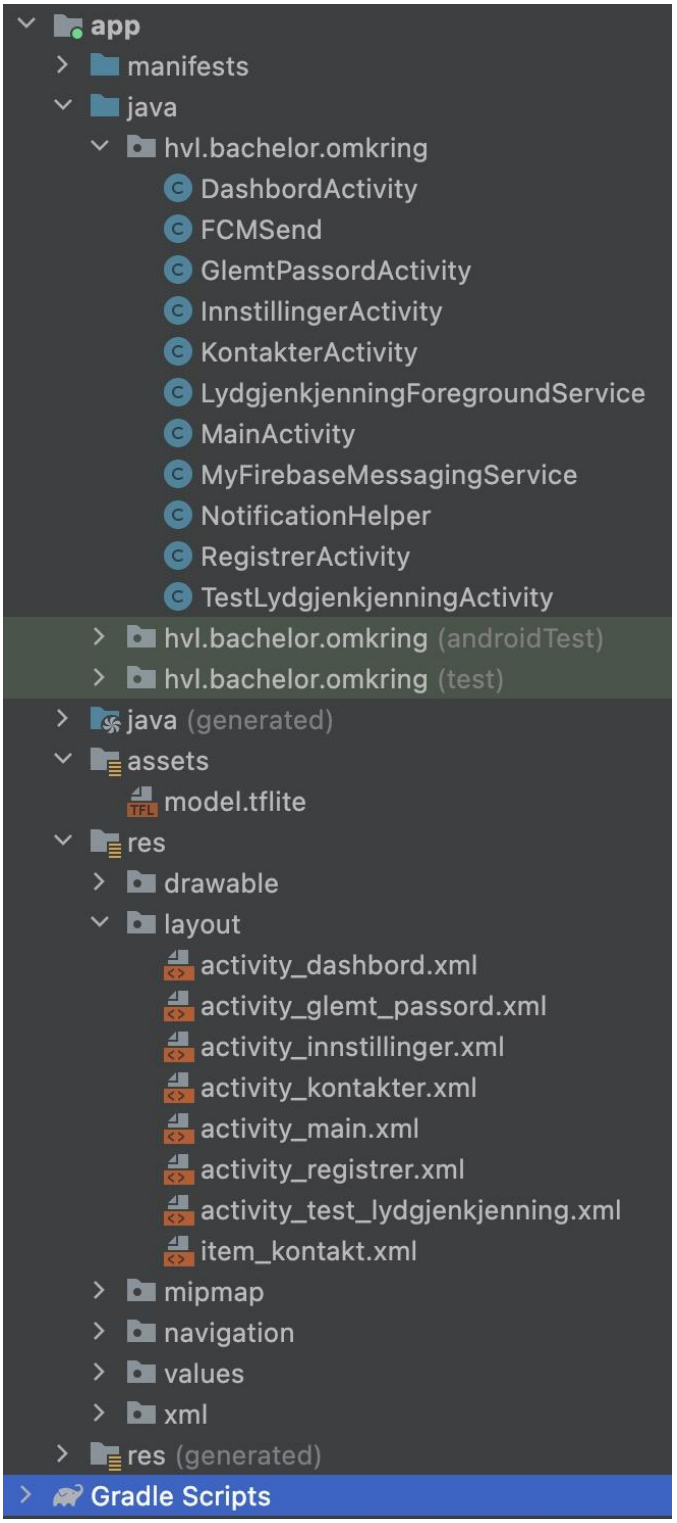
Figur 1: Arkitekturskisse

Arkitekturskissen representerer de viktigste komponentene i systemet. TensorFlow Lite-modellen benyttes til å utføre gjenkjenning av røykvarslere ved hjelp av maskinlæringsteknikker, som er optimalisert for mobile enheter. Når det gjelder autentisering, lagring av brukerdata og kommunikasjon mellom brukerne, blir Firebase benyttet som et integrert rammeverk. Disse komponentene blir brukt sammen til å utvikle mobilappen på Android.

Brukeren starter å opprette en brukerkonto ved å registrere seg i mobilapplikasjonen. Ved registrering blir brukerdataene som e-postaddresse og device token lagret i Firebase sin database. Når en bruker legger til en kontakt i mobilapplikasjonen vil det også bli lagret i databasen.
Når applikasjonen kjører lydgjenkjenningen aktivt i bakgrunnen, benytter den mobiltelefonens mikrofon. Applikasjonen tar lydklipp fra mikrofonen med små tidsintervaller. Disse klippene blir deretter prosessert av vår optimaliserte TensorFlow Lite-modell. Modellen tilordner en kategori til hvert lydklipp, enten som en røykvarsler eller som noe annet enn en røykvarsler.

Hvis en røykvarsler blir gjenkjent, hentes først alle brukerens kontakter ut fra databasen og etterfulgt av henting av enhets token for hver kontakt fra databasen. Deretter benyttes Firebase Cloud Messaging for å sende push-varsler til alle kontaktene, og varsle dem om at en røykvarsler er blitt aktivert.

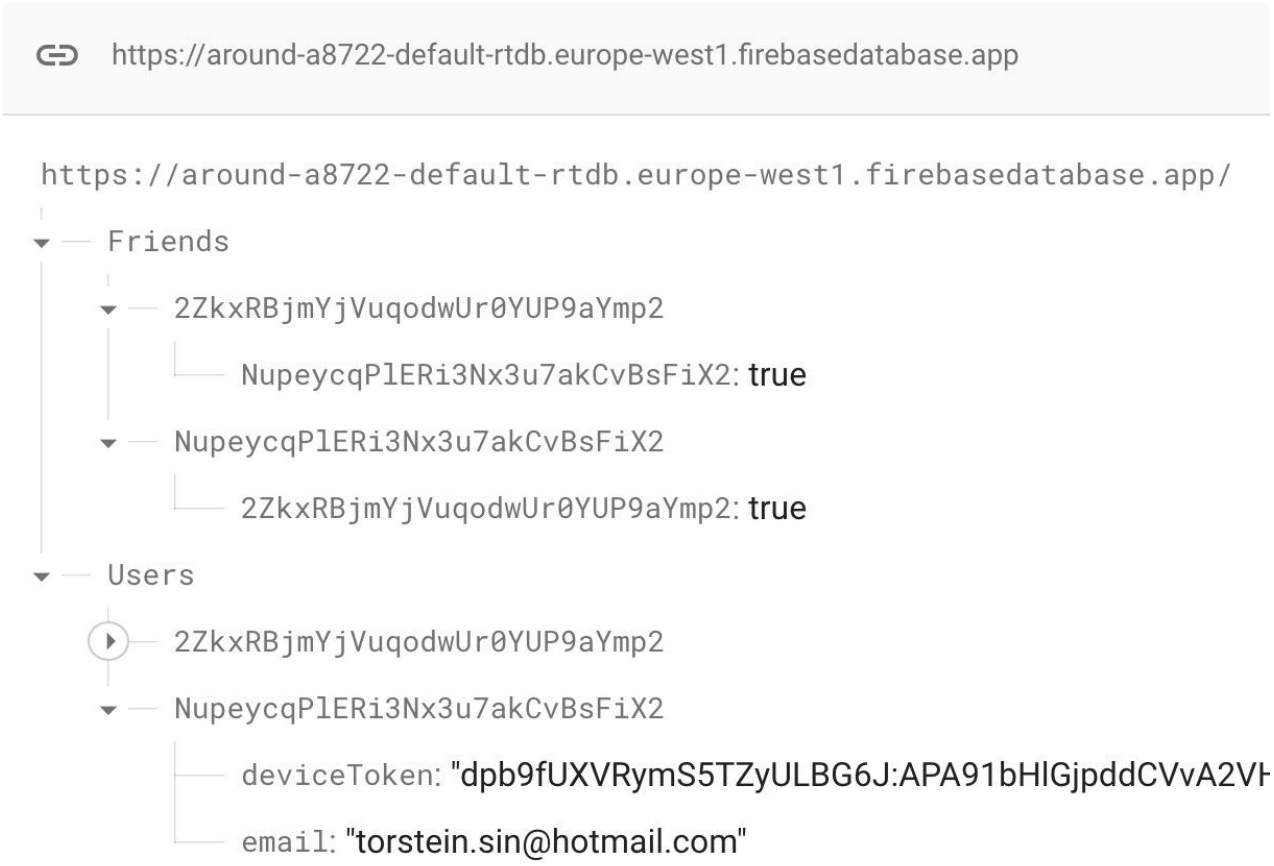
3 PROSJEKTSTRUKTUR



Figur 2: Prosjektstruktur

Under app/java/hvl/bachelor/omkring er java-klassene applikasjonen tatt i bruk. Under app/res/layout er xml filene som bestemmer utseende på applikasjon. Under app/assets finner vi ML-modellen (maskinlæringsmodellen).

4 DATABASEMODELL



Figur 3: Skjermbilde av Firebase Realtime Database

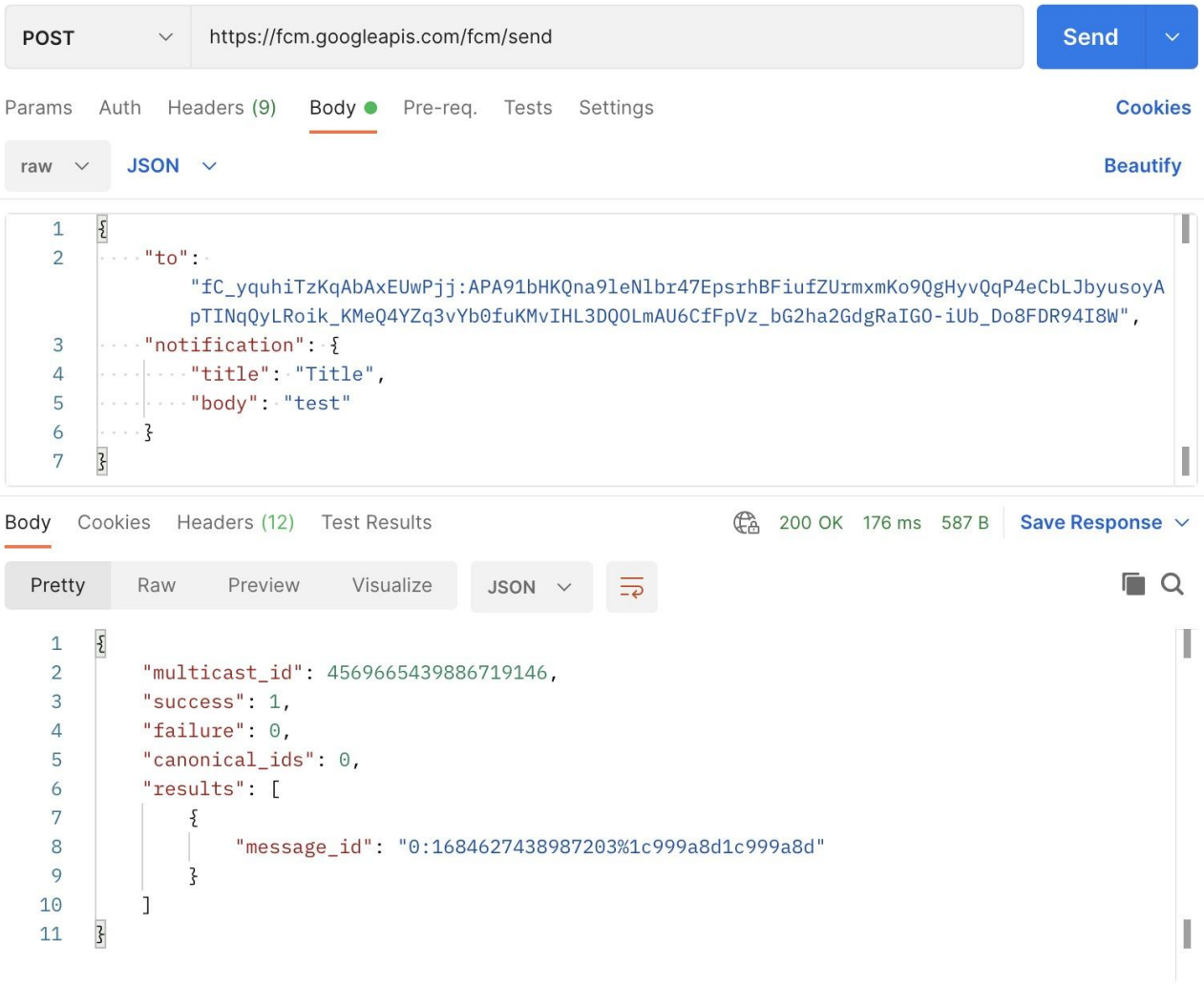
Databasen som applikasjonen tar i bruk er Firebase sin Realtime Database, et skjermbilde tatt under testing er vist i figur 3. Brukeren sin data er lagret under “Users”, hver bruker blir identifisert med sitt unike brukerid, informasjon som e-post, som brukes for å legge til kontakter, og device token, som brukes for å sende varsler mellom enheter, blir lagret her. Under friends blir forholdene mellom brukerne lagret, her er alle en bruker sine kontakter lagret som bruker id under brukerens eget brukerid.

5 Server-tjenester

Løsningen tar i bruk Firebase sin Cloud Messaging tjeneste. Det er ikke direkte koding av REST-servere, men Google sin REST-API er blitt tatt i bruk for å sende varsler mellom brukerne.

Dette blir programmert inn i applikasjonen ved å først hente alle kontaktene sitt bruker id fra "Friends", og deretter finne device-token tilhørende hver enkelt bruker i "Users". Denne blir da brukt for å sende et varsel til andre brukere.

Under testing ble dette også gjort manuelt, ved å ta i bruk postman, som vist i figur 5.1.



Figur 5.1: Postman brukt for testing av varsling.

6 SIKKERHET

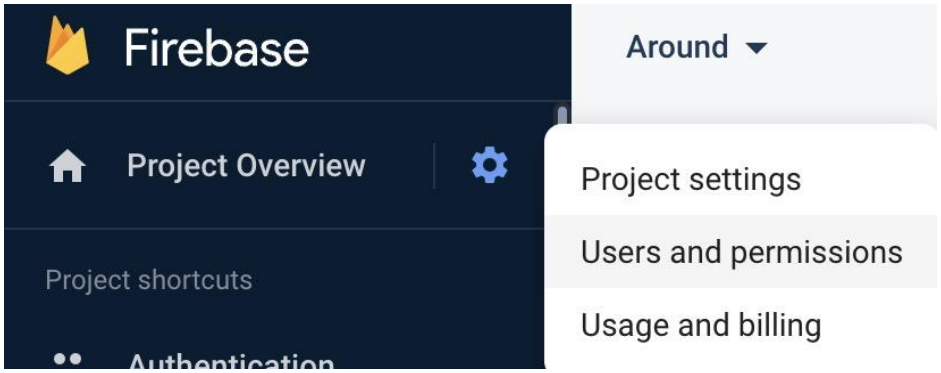
For å sikre brukerens personvern og sikkerhet har vi implementert lydgjenkjenningen lokalt i vår mobilapplikasjon. Dette betyr at lydopptakene som tas med brukerens mikrofon behandles og analyseres direkte på mobiltelefonen. Lyd-dataene forblir dermed lokalt og blir ikke sendt videre til eksterne servere for analyse. Dette reduserer potensielle sikkerhetsrisikoer knyttet til overføring og lagring av personlig informasjon.

Når det gjelder lagring og sikkerhet av brukerens data, bruker vi Firebase som vår plattform. All data som lagres i Firebase-databasen er kryptert både under overføring og i hvile (Firebase, 2019). Dette bidrar til å forhindre uautorisert tilgang til brukerens data. Firebase tilbyr også sikker autentisering.

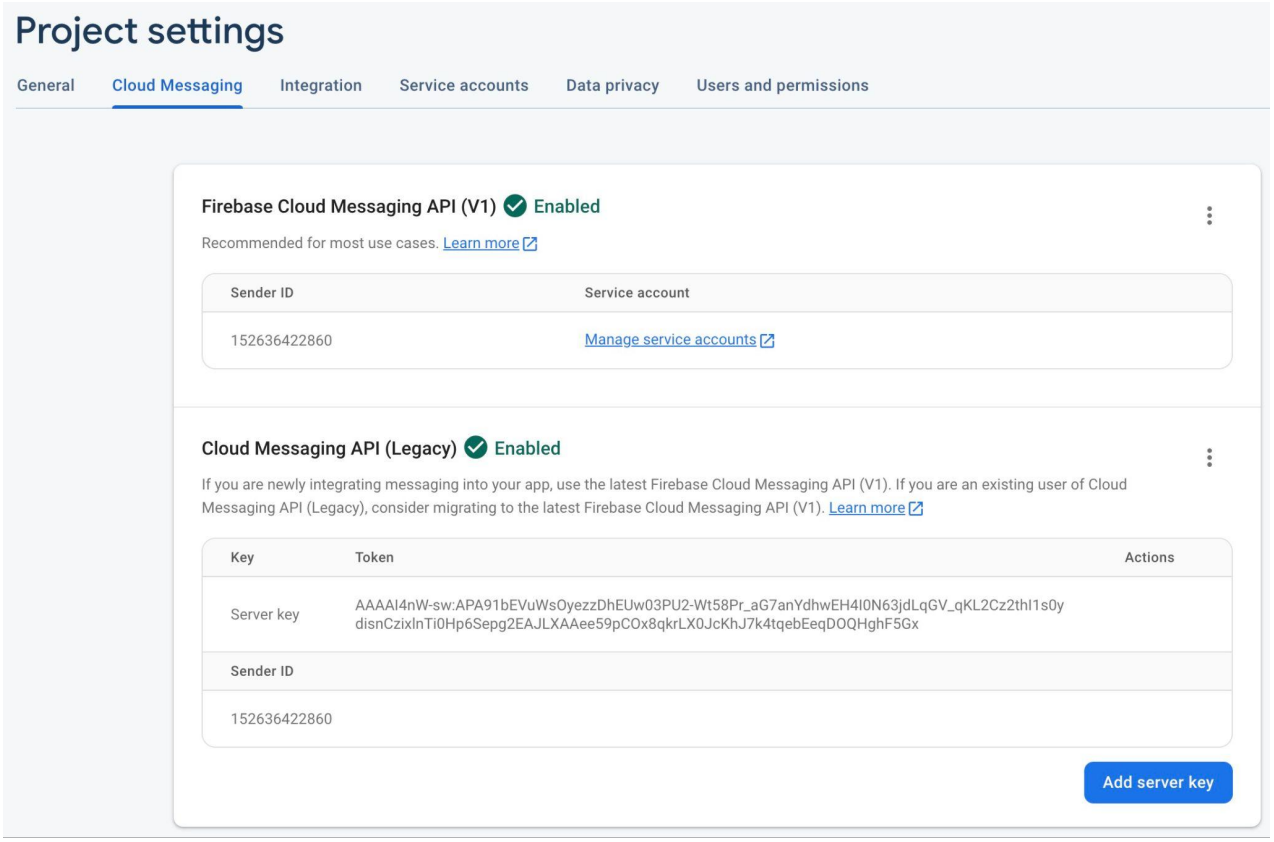
Ved å benytte oss av Firebase som vår backend-plattform, kan vi dra nytte av deres dedikerte sikkerhetsteam og deres kontinuerlige arbeid med å oppdatere og styrke sikkerheten til plattformen. Dette gir oss tillit til at brukerens data blir behandlet på en sikker og pålitelig måte.

7 INSTALLASJON OG KJØRING

For at dette prosjektet skal kjøres må det settes opp et prosjekt i Firebase. Autentisering må bli satt opp i prosjektet, det må også Realtime Databasen. Firebase cloud messaging må også bli satt opp. Deretter kan en Firebase SDK-fil bli lastet ned og bli lagt til i prosjektet. For å ta i bruk REST-APIen tilhørende Firebase Cloud Messaging må server nøkkelen oppdateres i prosjektet. Denne kan bli funnet ved å trykke på tannhjulet, og deretter trykke på "Project settings" eller "Users and permissions" vist i figur 7.1.



Figur 7.1: Tannhjul



Figur 7.2: Server nøkkel

Deretter gå til "Cloud Messaging"-siden, som vist på figur 7.2, nøkkelen blir funnet ved "Server Key". Denne blir lagt til i LydgjenkjenningForegroundService.java klassen.

8 DOKUMENTASJON AV KILDEKODE

Kildekoden er lagt ved i bacheloroppgaven. Koden inkluderer kommentarer der det er relevant.

9 REFERANSER

Firebase. (2019). Privacy and Security in Firebase | Firebase. [online]
Tilgjengelig hos: https://firebase.google.com/support/privacy