
Mobilplukk

Utvikle en multiplattform mobilapplikasjon i Cordova med Ionic, for uthenting og verifisering av biologiske prøver i Trionas databasesystem for Biobank1.

Problemstilling

“How can one create a multi-platform mobile application used to verify biological samples, stored in Biobanken, by scanning barcodes?

The application has to be able to retrieve and update necessary information without compromising security.”

Oppgavebeskrivelse

Lage en multi-plattform mobilapplikasjon som kan:

1. Importere plukkliste fra databasesystemet
2. Vise med tekst - eller helst med grafikk - hvor de aktuelle prøvene befinner seg (rom, fryser, stativ, boks, posisjon)
3. Lese strekkode fra beholderen
4. Kontrollere avlest strekkode mot angitt ID-nummer i plukklista
5. Gi tydelig melding hvis strekkoden ikke tilsvarer noe ID-nummer i plukklista
6. Kvittere for at prøve med tilsvarende ID-nummer er tatt ut av lageret dersom strekkoden stemmer
7. Sende melding til databasesystemet om de prøvene som er tatt ut av lageret, slik at databasen oppdateres
8. Sende melding til databasesystemet om eventuelle avvik (feilplassering av prøvebeholdere) som oppdages

Cordova og Ionic

Vi valgte å løse oppgaven ved å ta i bruk Cordova med Ionic. Cordova er et rammeverk som lar deg bygge multiplattform-applikasjoner ved bruk av HTML og JavaScript. Ionic er et rammeverk som benytter seg av Cordova og lar deg utnytte AngularJS for å enklere lage godt utformede brukergrensesnitt.

Cordova har støtte for iOS, Android og Windows. Ionic er optimalisert for iOS og Android, men fungerer på Windows.

Da vi begge to hadde erfaring med både JavaScript, AngularJS og HTML var dette et naturlig valg.

Ulike Plattformer

For å bygge til de ulike plattformene trenger man ulike utviklingsmiljø på grunn av restriksjoner i Windows og iOS. Etter enighet med oppgavestiller fokuserte vi på Android under utviklingen og migrerte til iOS og Windows etterhvert.

- Restriksjoner
- Bygging
- Versjonskontroll

Sikkerheten?

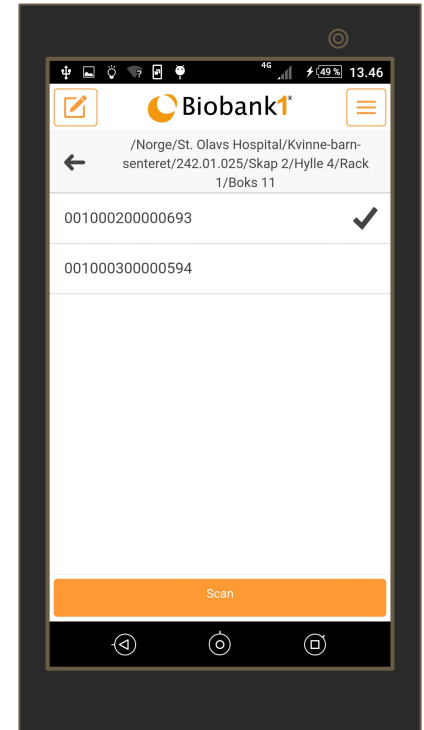
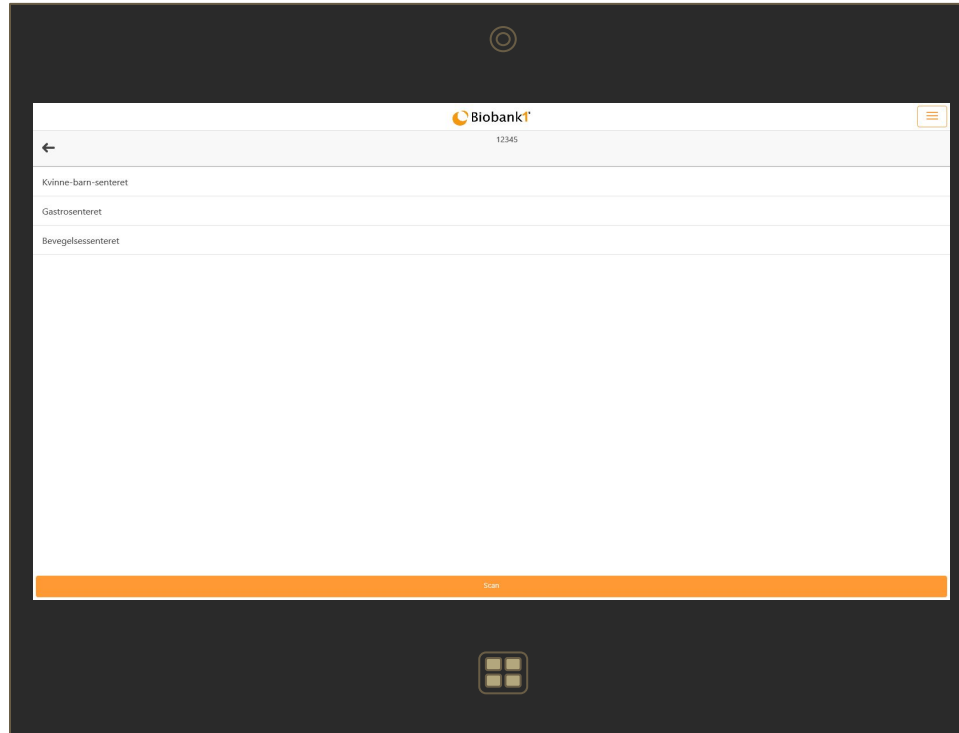
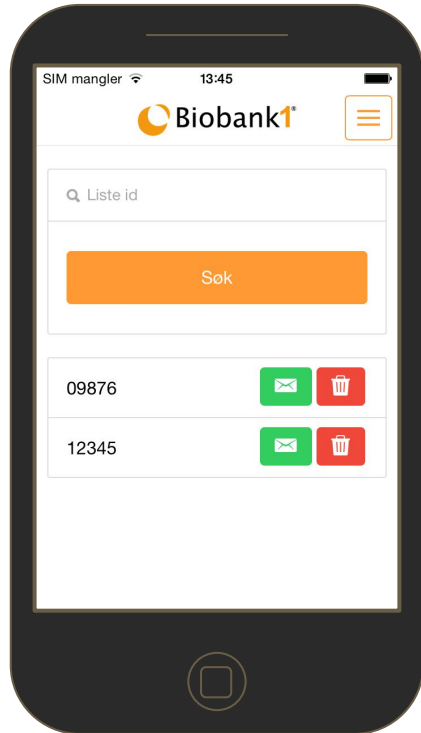
For å unngå problemer med HEMIT sine restriksjoner rundt sensitive data, valgte vi å unngå direkte kontakt med databasen hvor alt av informasjon om prøvene er lagret.

Løsningen ble at applikasjonen kommuniserer med en REST-server som inneholder kun den informasjonen vi trenger:

- Prøve-ID, Liste-ID og Lokasjons-ID (for fremtidig bruk).
- Lokasjon
- Teksten lagret i informasjonsfeltet
- Hvorvidt prøven er hentet/ikke hentet

Dermed håndterer applikasjonen biologiske prøver uten å involvere sensitive data.

Resultatet



Resultatet

