

**AVDELING FOR TEKNOLOGI
WESTERDALS OSLO ACT**

**KRYSSPLATTFORM (TDS200)
VÅR 2018**

EKSAMENSOPPGAVE

Skrevet av Anh Le

Dato: 11.05.2018

Innhold

1. Teoretisk kompetanse
 - 1.1 Native apps
 - 1.1.1 Fordeler
 - 1.1.2 Ulemper
 - 1.2 Hybrid apps
 - 1.2.1 Fordeler
 - 1.2.2 Ulemper
 - 1.3 Konklusjon
2. Refleksjon over teknisk kompetanse
3. Hjelpemidler

1 Teoretisk kompetanse

Et viktig spørsmål som dukker opp når du bestemmer deg for å lage en mobil app er hva slags teknologi du har tenkt til å bruke. Alle fremgangsmåter har sine sterke sider og svakheter, alt etter hva slags behov du har som bestemmer hvilken fremgangsmåte som er optimal.

I denne oppgaven setter jeg fokus på hva vi har lært gjennom dette semesteret i fagene kryssplattform(TDS200) og Android Programmering(PG4600) for å prøve å svare på når det er bedre å utvikle via en fremgangsmåte framfor den andre.

I Android Programmering lærer vi å skrive en android native applikasjon skrevet i Java, mens i Kryssplattform lærte vi å skrive en hybrid app med ionic rammeverket som tar i bruk TypeScript og web-teknologi(JavaScript, CSS og HTML).

Vi starter først med å dra inn de tre største fordelene og ulempene etter min mening til hver tilnærming.

1.1 Native Apps

1.1.1 Fordeler:

- Native kode er raskere enn HTML og JavaScript, så ressurskrevende apper som grafiske applikasjoner, HD spill og intensive animasjons-applikasjoner kan være best utviklet som en Native App.
- Native Apps har tilgang til eksklusiv native APIer i enhetens operativsystem som for eksempel kamera, lokasjon og push notifikasjoner. Dette er også mulig på hybrid apper, men på en nokså mer tungvint måte fordi det krever en mellommann som fungerer som en bru til en plugin som fungerer som en Native-API(du må også lage pluginen selv med native kode hvis ingen andre har laget den før deg).
- En generell bedre brukeropplevelse siden appen er bygd opp etter en retningslinje som alle andre native apper i samme platform. Dette fører til at brukeren lærer å bruke appen lettere siden den er lett gjenkjennelig med hva brukeren er vant til fra før.

1.1.2 Ulemper:

- Hvis det er ønskelig å nå ut til flere er du nødt til å kode en app for både iOS og Android. du er nødt til å skrive om koden der du teknisk sett lager en ny native app til et annet operativsystem, som er en veldig kostbar prosess.
- Hver plattform vil ha sin egen utgivelsessyklus + oppdateringer, dette blir fort vanskelig å oppnå på begge plattform, som da bygger kostnadene enda høyere.
- Ikke overførbar kunnskap, om du har en bakgrunn innen android eller ios så vil ikke det gjøre læringen av et nytt utviklingsmiljø mye enklere da de forskjellige disiplinene bygger på sin egen struktur og retningslinje.

1.2 Hybrid Apps

1.2.1 Fordeler:

- WORA(Write once, run anywhere), skriv koden en gang, og det vil fungere til både iOS og Android. Dette vil kutte ned utviklingstiden og kostnader drastisk, siden du ikke trenger å lære noen spesifikke programmeringsspråk for iOS eller Android.
- Hybrid Apper er basert på web-teknologi, så den samme appen kan kjøres på en nettleser akkurat som hvilken som helst nettside.
- Siden Hybrid Apper kjøres fra samme kode, så vil brukeropplevelsen være lik på tvers av plattformer, uavhengig av om bruker endrer på enheten eller nettleser.

1.2.2 Ulemper:

- Hybrid Apps vil ikke kunne tilby den samme ytelsen som en Native App, siden en Hybrid App vil ha vanskeligheter med å oppnå gode overganger gjennom appen. Dette kan føre til ytelse problemer hvis det brukes til å lage ressurskrevende apper og spill.
- Android og iOS tilbyr en rekke egenskaper gjennom nativ-API som ikke er alltid mulig for en Hybrid App å ta i bruk, da dette krever ofte en plugin som ikke alltid er tilgjengelig eller lett å implementere.

- Hvis vi ser for oss en scenario hvor du er et skritt inn i framtiden der du allerede har lagd en Hybrid App men du ønsker å nå et ytelsesnivå lik en Native App. Dette er nå en vanskelig situasjon hvor det vil koste masse ressurser, bare for å komme nærmere ytelsesnivået til en Native App. Dette går da imot prinsippet med å forkorte utviklingstiden, og det hadde kanskje vært bedre å bare lage en Native App.

1.3 Konklusjon

Basert på egen erfaring på alt jeg har lært ved å undersøke forskjellene på Native Apps og Hybrid Apper har jeg kommet fram til at det er et formål for begge deler. Mens Native Apps har en klar ytelsefordel i forhold til Hybrid Apps så er det en drastisk mer krevende tidsprosess å utvikle en løsning til enten android eller iOS, og det vil være enda mer tidskrevende hvis du skulle gå for begge.

Spørsmålet om hvilken fremgangsmåte som er best for deg kan bli svart ved å stille deg selv et spørsmål om hvor mye verdi du ser i denne applikasjonen for deg/selskapet ditt. Hvis du for eksempel har planer om å starte en nettbutikk og applikasjonen er en stor komponent i hele din markedsstrategi for å tiltrekke kunder og beholde på dem, kan det være lurt å ta kostnaden og tida som kreves til å utvikle appen din som en Native App.

Du vil skape en bedre brukeropplevelse, og du har tilgang til flere egenskaper for å gi kunden den forbedrede opplevelsen, som vil føre til et større sannsynlighet for at de forblir en kunde. På den andre siden av skalaen, kan det være gunstig å gå for en Hybrid App hvis du er på utkikk etter å få ut en App i markedet så fort som mulig, enten det er for å treffe en trend eller målgruppe som er veldig attraktiv nå.

Det er også slik at jo mer en Hybrid App ønsker å være som en Native App, jo oftere vil du stille spørsmålet på hvorfor du ikke bare utviklet en Native App fra starten.

Mens det vil ta mye tid å sette seg inn i et miljø for å lære seg å kode en Native App, så vil reisen ikke nødvendigvis være mye lettere hvis du går for en Hybrid App som for eksempel ønsker å ta i bruk så mange Native-APIer som mulig.

2 Refleksjon av teknisk kompetanse

2.1 Designvalg

Ved starten av denne oppgaven tenkte jeg en god stund på hva slags apper som allerede ligger ute i markedet som minner noenlunde om hva jeg prøver å få til med min egen løsning på oppgaven. Apper som jeg følte hjalp meg veldig med å komme i gang var Finn sin Android app, og AliExpress. Disse to ga meg et bilde i hodet mitt på mulighetene for hvordan jeg kunne designe min app. Det ble også spontant lagt inn forskjellige egenskaper etter å ha studert og lekt litt rundt med ionic rammeverket, blant annet en searchbar og en tab meny.

2.2 Ekstrafunksjoner

I tillegg til minstekravet av oppgaven har jeg lagt inn noen ekstra funksjoner:

- En søkefunksjon som lar bruker søke etter bøker som har blitt lagt ut.
- En mer detaljert brukerprofil der bruker har mulighet til å legge inn et telefonnummer som dukker opp i annonsene til bruker, og et bilde av bruker.
- Fillagring av bilder tatt med kamera via firebase.
- Henting av geolocation.
- Google maps som visualiser en lokasjon ved hjelp av geolocation.
- Halvveis implementering av google login, men den fungerer desverre bare via nettleser.

2.3 Veiledning gjennom Appen

Appen starter opp der bruker blir henvendt til en loginside med mulighet for å registrere, logge inn med en vanlig bruker eller ta i bruk google login(dette fungerer desverre kun i nettleser, da jeg oppdaget for sent at min implementasjon av metoden ikke støttes i android, en annen måte å gjøre det på krever deep linking via firebase sin dynamic linking). Via registreringssiden får brukeren mulighet til å registrere seg med å taste inn email, passord, brukernavn, telefonnummer, og et bilde av seg selv.

hvis brukeren allerede er logget på så vil de bli sendt videre til en hjemmeside, der de kan søke eller vise alle bok annonser som ligger ute. Ved å trykke på en av bøkene som dukker opp vil appen sende brukeren videre til en detalj-side av boka, hvor brukeren får se flere detaljer om boka, hvem som er selgeren og telefonnummeret til selger og en google maps location av ca hvor selgeren befinner seg.

Via tab meny får bruker tilgang til å gå over til book-overview siden, hvor de har mulighet til å legge ut bøker ved å trykke på + knappen øverst til høyre. Det var planer om å implementere en funksjon for bruker å lagre annonser i denne siden, men dette ble ikke oppnådd i tide så siden ser ganske tomt ut. Hvis brukeren trykker på + knappen for å legge ut en ny bok, får de tilgang til en ny side der de kan legge inn tittelen på boka, pris, beskrivelse, tilstanden boka er i ved hjelp av tre forskjellige tilstander, og et bilde av boka.

Den siste tab knappen i menyen leder brukeren til brukerprofilen sin, hvor de kan se informasjonen om seg selv og mulighet for å logge ut.

Jeg tar i bruk to modeller i min løsning, en for book og en for user. Disse symboliserer dokumententobjektene jeg legger inn i databasen i min books collection og min user collection.

2.4 Potensielle fremtidige funksjoner

Den endelig løsningen kreves litt mer finpussing for å ende opp som det jeg så for meg, hvis jeg hadde mer tid hadde jeg sørget for at book overview siden min hadde en meningsfull oppgave som for eksempel vise frem en liste av bøker brukeren har lagt ut eller en liste med bøker brukeren har satt på overvåking. Google login fungerer dessverre akkurat nå bare via en nettleser, men hvis jeg hadde hatt bedre tid hadde jeg lært hvordan deep linking fungerer og tatt i bruk dynamic deep linking fra firebase til å gjøre det mulig å bruke denne funksjonen via telefonen også. Jeg hadde også likt å kunne vise en enkel implementasjon hvor du bruker pipe til å transformere telefonnummeret til brukeren. På grunn av litt dårlig planlegging ble det satt av alt for lite tid til å kommentere koden, og derfor ble denne oppgaven gjort halvferdig.

Ellers er jeg fornøyd med hva jeg klarte å oppnå basert på tiden som ble lagt inn i løsningen.

3 Hjelpemidler

Forelesningene i TDS200 ble fulgt nøye, og det ble gjenbrukt kode fra livecoding sessionene.

Josh Morony sin guide på google maps:

<https://www.joshmorony.com/getting-started-with-google-maps-in-ionic-2/>

Ionic sin dokumentasjon: <https://ionicframework.com/docs/>

Material design for å finne farge:

<https://material.io/design/color/#color-usage-palettes>

Hvordan lage en google login:

<https://angularfirebase.com/lessons/google-user-auth-with-firestore-custom-data/>

<https://javebratt.com/ionic-social-login-firebase/>

Lesestoff for å lære mer om pros and cons med native og hybrid apper:

<https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/>

<https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/>

<https://www.joshmorony.com/whats-the-difference-between-native-hybrid-and-web-mobile-app-development/>

<https://www.quora.com/What-are-the-pros-and-cons-of-hybrid-vs-native-mobile-development>

<https://www.ready4s.com/blog/key-pros-and-cons-of-native-and-hybrid-apps/>

<https://codeburst.io/native-app-or-hybrid-app-ca08e460df9>