OULUN AMMATTIKORKEAKOULU

Samuli Rönkkö

HUAWEI MOBILE SERVICES -VERSIO POLAR FLOW -MOBIILISOVELLUKSESTA JA SEN JULKAISU

HUAWEI MOBILE SERVICES -VERSIO POLAR FLOW -MOBIILISOVELLUKSESTA JA SEN JjulkaisU

Samuli Rönkkö

Opinnäytetyö
Kevät 2021
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Samuli Rönkkö
Opinnäytetyön nimi suomeksi: Huawei Mobile Services -versio Polar Flow -mobiilisovelluksesta ja sen julkaisu
Opinnäytetyön nimi englanniksi: Huawei Mobile Services version of Polar Flow mobile application and it's release
Työn ohjaajat: Pekka Alaluukas, Mika Mustonen
Työn valmistumislukukausi ja -vuosi: Kevät 2021
Sivumäärä: 25

Tämän työn tavoitteena oli tehdä Polar Flow -mobiilisovelluksesta versio, joka käyttää Huawein mobiilipalveluita Googlen sijasta, ja julkaista se Huawei AppGallery-sovelluskaupassa. Tarve työlle tuli, kun Yhdysvaltojen kauppakiellon takia Huawei ei voinut käyttää Googlen mobiilipalveluita laitteissaan. Tällöin Googlen mobiilipalveluita käyttävät sovellukset eivät toimi Huawein laitteissa. Tämä versio tulisi olemaan normaaliversion rinnalla, joka käyttää Googlen palveluita.

Versio toteutettiin riisumalla Googlen riippuvuudet sovelluksesta ja korvaamalla ne Huawein vastaavilla. Jotkut ominaisuudet kuten YouTube-videoiden toisto, ohjattiin verkkosivulla näytettäväksi. Sovelluksen keräämään analytiikkaan lisättiin myös tunnistus, jolla voidaan erotella eri sovellusversiot.

Aluksi opinnäytetyössä käydään läpi taustaa Polar Flow -sovelluksesta ja työssä tarvittavat työkalut ja käsitteet. Sen jälkeen tarkastellaan eri vaihtoehtoja työn toteutukselle ja lopuksi sen vaiheet ja tulokset.

Työn lopputuloksena oli toimiva versio sovelluksesta, jota voidaan käyttää Huawein puhelimissa. Se saatiin toteutettua ja julkaistua aikataulun sisällä.

Asiasanat: ohjelmistokehitys, mobiilisovellukset, Huawei, Android

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Software Development

---

Authors: Samuli Rönkkö
Title of thesis: Huawei Mobile Services version of Polar Flow mobile application and it's release
Supervisors: Pekka Alaluukas, Mika Mustonen
Term and year when the thesis was submitted: Spring 2021
Pages: 25

The goal of this thesis work was to create a version of Polar Flow mobile app that uses Huawei Mobile Services instead of Googles and to release it on Huawei AppGallery application store. The need for this work came when United States of America put Huawei on trade ban and Huawei could not use Googles mobile services as a result of this. Because of this, applications which use Googles mobile services don't work on Huawei devices This version would co-exist with normal version which uses Google Mobile Services.

The work was done by stripping Google dependencies from the application and replacing them with Huaweis. Some features such as YouTube video playback had to be directed to website rather that in-app. Also, property was added to analytics for easy separation on app versions.

This thesis is started by going to through some background of Polar Flow app and the needed tools and concepts for the work. This is followed by examination of different options how the work could have been done and finally its steps and results.

The result of the work was working version of the application which can be used in Huawei smartphones. It was developed and released in schedule.

Keywords: software development, mobile applications, Huawei, Android

ALKULAUSE

Haluan kiittää Polar Electro Oy:tä ja linjaesimies Mika Mustosta mahdollisuudesta saada tehdä tämä opinnäytetyö. Haluan myös kiittää OAMK:n lehtori Pekka Alaluukasta työn ohjaamisesta ja Polarin mobiilitiimiä teknisestä tuesta.

Oulussa 16.3.2021

Samuli Rönkkö

SISÄLLYS

SANASTO

API	Application Programming Interface, ohjelmointirajapinta.
APK	Android Application Package, asennuspaketti, jonka avulla sovellus asennetaan Android-laitteeseen.
Flow	Polar Flow, tämän työn yhteydessä Flow'lla tarkoitetaan Polar Flow'n Android-sovellusta.
GMS	Google Mobile Service, Googlen tarjoama mobiilipalvelukirjasto, joka voidaan lisätä sovellukseen. Vaaditaan, jos halutaan käyttää Googlen palveluita, kuten karttapalvelua.
HMS	Huawei Mobile Service, Huawein tarjoama mobiilipalvelukirjasto, joka voidaan lisätä sovellukseen. Vaaditaan, jos halutaan käyttää Huawein palveluita, kuten karttapalvelua.
SHA-256	Secure Hash Algorithm, salausalgoritmi, joka sisältää 256-bittisen tiivistearvon. Käytetään monipuolisesti erilaisissa turvallisuussovelluksissa ja -protokollissa.

1 JOHDANTO

Työn tarkoituksena oli tehdä Polar Flow Android -sovelluksesta versio, joka käyttää Huawein mobiilipalveluita Googlen sijasta, ja julkaista sovellus Huawei AppGallery -sovelluskaupassa. Työn tilaaja Polar Electro Oy on kempeleläinen langattomaan sykkeenmittaukseen liittyviä laitteita, teknologioita ja ohjelmistoja valmistava sekä kehittävä yritys (1).

Tarve työlle tuli, kun Yhdysvallat laittoi kiinalaisen teknologiayhtiö Huawein kieltolistalleen. Tämän takia uudet Huawein puhelimet eivät pystyneet enää käyttämään Googlen mobiilipalveluita (Google Mobile Services).

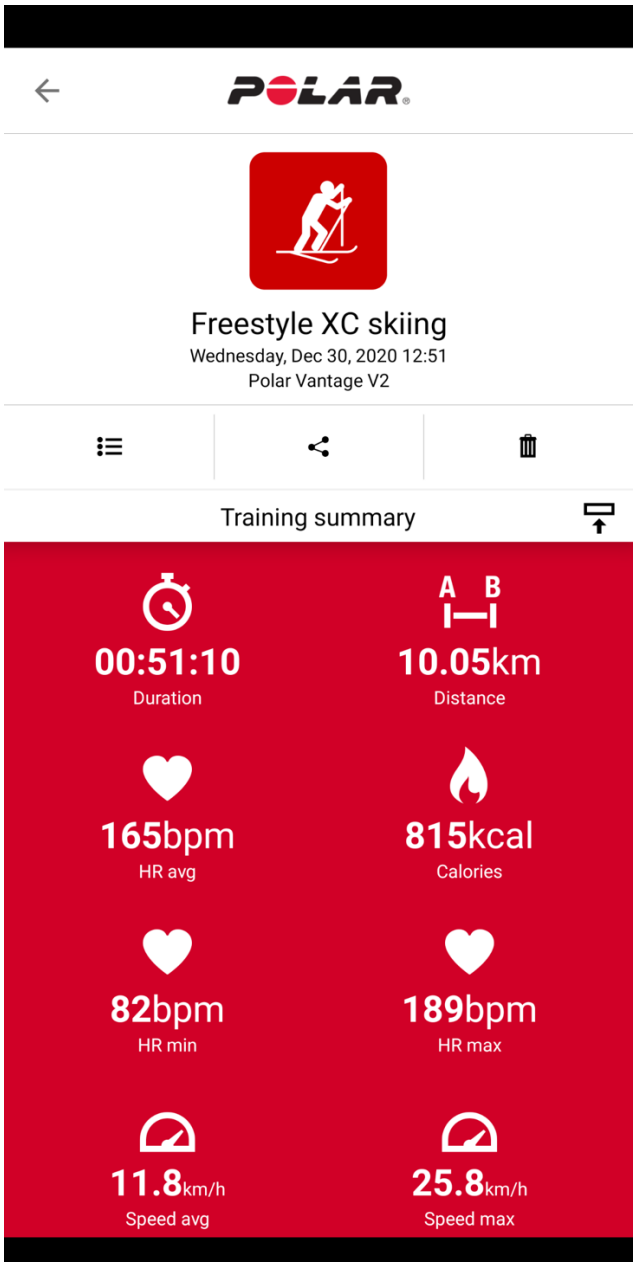
Jotta sovellusta voidaan käyttää Huawein puhelimissa, sen lähdekoodia täytyy muuttaa käyttämään Huawein omia mobiilipalveluita (Huawei Mobile Services) ja se täytyy julkaista Huawein AppGallery-sovelluskauppaan, josta se voidaan ladata.

2 POLAR FLOW -SOVELLUS

Polar Flow -sovellus on Polar Electro Oy:n kehittämä ja julkaisema mobiilisovellus Android- ja iOS -laitteille. Sitä käytetään Polarin fitness- ja urheilukellojen rinnalla ja se antaa mahdollisuuden analysoida rannelaitteen tuottamaa dataa sekä hallinnoida asetuksia ja harjoitusprofiileja.

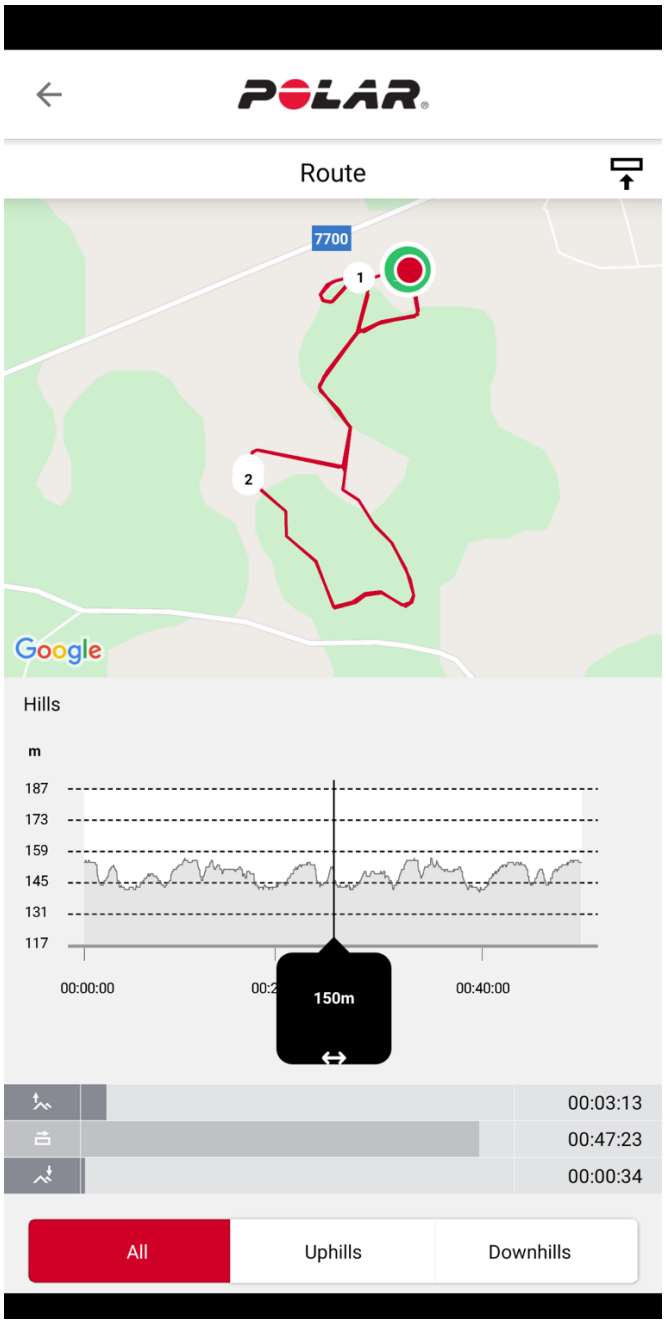
Ostaessaan uuden rannelaitteen käyttäjä voi rekisteröidä sen sovelluksen kautta henkilökohtaiselle Flow-tilillensä. Tällä tavoin useankin laitteen dataa pystytään hallinnoimaan samasta paikasta ja se säilyy laitetta vaihtaessakin. Rannelaitteen tuottama data synkronoituu automaattisesti puhelimeen Bluetooth-tekniikan avulla.

Sovelluksen avulla pystytään analysoimaan käyttäjän omaa aktiivisuutta, unta, harjoittelua ja palautumista. Sovellus piirtää helposti tulkittavia graafeja käyttäjän ympärivuorokautisesta sykkeestä sekä aktiivisuustasoista. Se antaa palautetta nukutuista öistä ja analysoi sitä jakamalla sen eri osa-alueisiin, kuten unen määrään, jatkuvuuteen ja eri univaiheiden kestoon. Unidatan perusteella pystytään myös tarjoamaan käyttäjän palautumisen tila, joka ilmaistaan pistemäärällä. Tämä pistemäärä saadaan vertaamalla viimeisintä unidataa käyttäjän unihistoriaan. Harjoitusten aikaista dataa voidaan myös tarkastella yksityiskohtaisesti (kuva 1). Analyysinäkymä pilkkoo harjoituksen aikaiset nopeudet, sykkeet, korkeudet, energialähteet ja tehot graafeihin, joista niitä on helppo tulkita.



KUVA 1. Harjoituksen analyysisivu

Jos harjoitus on ollut ulkolaji ja saatavilla on ollut GPS-tietoa, voidaan myös piirtää harjoituksen reitti kartalle (kuva 2). Siitä reittiä voidaan kelata ja nähdä, mikä on ollut nopeus ja syke milläkin hetkellä. Yksi tämän työn osa-alueista olikin muuttaa tämä kartta käyttämään eri palveluntarjoajan karttapalvelua.



KUVA 2. Harjoituksen analyysisivun kartta

3 TYÖKALUT JA KÄSITTEET

Projektissa käytettiin apuna työkaluja, jotka helpottivat työn tekemistä, sen julkaisua ja versionhallintaa. Itse sovelluksen lähdekoodin muuttamiseen käytettiin Android Studiota. Versionhallinnassa käytettiin Git-versionhallintajärjestelmää, jolla voitiin pitää Flow-sovelluksen Huawei-versiota erillään normaalista Google-versiosta. Jenkins-automaatiopalvelinta käytettiin APK-pakettien rakentamiseen kun Gitin Huawei-version julkaisuhaaraan tuli uusia muutoksia. Eri valmistajien puhelimien käyttäjämäärien seuraamiseen käytettiin Firebase-alustaa. Sovelluksen julkaisuun käytettiin Huawei AppGallery Connectia.

3.1 Google Mobile Services -palvelut

Google Mobile Sevices (GMS) on valikoima Googlen tarjoamia sovelluksia ja ohjelmistokehityspaketteja, joilla pystytään tarjoamaan käyttäjälle monipuolisia ominaisuuksia. Sovelluksiin kuuluu muun muassa Google Search, YouTube, Google Maps ja monia muita. GMS-sovelluksista puhutaan yleisesti Android-käyttöjärjestelmiä kehittäessä, kun niitä halutaan toimittaa julkaisun yhteydessä. Tämän työn yhteydessä puhutaan GMS-ohjelmistokehityspaketista. Näihin lukeutuu muun muassa Maps, YouTube, Cast ja monia muita. Erona sovelluksiin on, että näiden ohjelmistokehityspakettien avulla pystytään laajentamaan omaa sovellusta näyttämään esimerkiksi karttanäkymää tai YouTube-soitinta. (2.)

3.2 Huawei Mobile Services -palvelut

Huawei Mobile Services (HMS) on Huawein vastaus Yhdysvaltojen kieltolistalle laitolle. Kieltolistalle laiton takia Huawei ei voinut enää käydä kauppaa Googlen kanssa, joten sen täytyi jättää Googlen palvelut pois uusista laitteista. Tämän takia Huawei päätti rakentaa ja julkaista omat palvelut. Niiden perusperiaate on sama kuin Googlen palveluissa. Huawei on pyrkinyt tarjoamaan samat palvelut ja työn tekohetkellä tarjonta olikin melko lähellä Googlea. Tämän avulla kehittäjät pystyvät tarjoamaan samankaltaisia ominaisuuksia myös Huawein uusille laitteille, jotka on julkaistu kieltolistalle laiton jälkeen.

3.3 Android Studio -ohjelmointiympäristö

Android-sovellusten virallinen ohjelmointiympäristö on Android Studio. Se perustuu IntelliJ IDEA -ympäristöön. Tämän ympäristön ominaisuuksien lisäksi Android Studio tarjoaa lukuisia ominaisuuksia, jotka mahdollistavat helpomman ja nopeamman kehitystyön Android-sovelluksille. Näihin ominaisuuksiin kuuluu joustava Gradle-koontijärjestelmä, nopean emulaattori, kattavat testaustyökalut sekä yhtenäinen ympäristö, jolla kehittää sovelluksia kaikille Android-laitteille. (3.)

3.4 Git-versionhallintajärjestelmä

Git on ilmainen avointa lähdekoodia käyttävä hajautettu versionhallintajärjestelmä. Sen avulla lähdekoodi voidaan haarauttaa eri haaroihin kehityksen aikana, mikä mahdollistaa suurenkin yhteistyön projekteissa. Jokainen haara on erillinen versio lähdekoodista. Tällä tavoin eri ominaisuuksia voidaan kehittää samanaikaisesti omissa haaroissaan ja kun ominaisuus on valmis, sen haara yhdistetään pääharaan, jolloin sen muutokset siirtyvät pääharaan. Erityisesti tässä työssä se mahdollistaa HMS- ja GMS-sovellusversioiden pitämisen eri haaroissa sekä kehitystyön julkaisuhaarasta irrallaan. (4.)

3.5 Jenkins-automaatiopalvelin

Jenkins on avoimeen lähdekoodiin perustuva automaatiopalvelin, joka on rakennettu Java Virtual Machinen päälle. Siihen on tarjolla satoja lisäosia, joilla pystytään laajentamaan Jenkinsiä automatisoimaan lähes mitä vain teknologiaa, jota ohjelmistotiimi saattaa tarvita. Näiden avulla projektien koontia, julkaisua ja jatkuvaa integraatiota voidaan automatisoida. Sen pystyy esimerkiksi määrittämään siten, että se ajaa projektin yksikkötestit ja rakentaa uudet APK-paketit, kun se tunnistaa, että Git-haraaan on tullut muutoksia. Tämän avulla nähdään jatkuvasti, rikkooko jokin muutos alkuperäistä toiminnallisuutta, ja saadaan uusimmat APK-paketit testattavaksi. (5.)

3.6 Huawei AppGallery Connect -alusta

AppGallery Connect on alusta, jonka avulla pystytään hallitsemaan sovelluksen käyttämiä Huawein mobiilipalveluita ja julkaisemaan sovellus AppGallery-sovelluskauppaan. Siellä hallitaan kaupan tuotesivun sisältöä, kuten tuotekuvia ja tuote-esittelytekstejä. Sen kautta otetaan käyttöön mobiilipalveluiden käyttämät API:t, kuten Map Kit -kirjasto. Sen kautta tapahtuu myös sovelluksen julkaisu ja päivittäminen.

3.7 Firebase-analytiikka-alusta

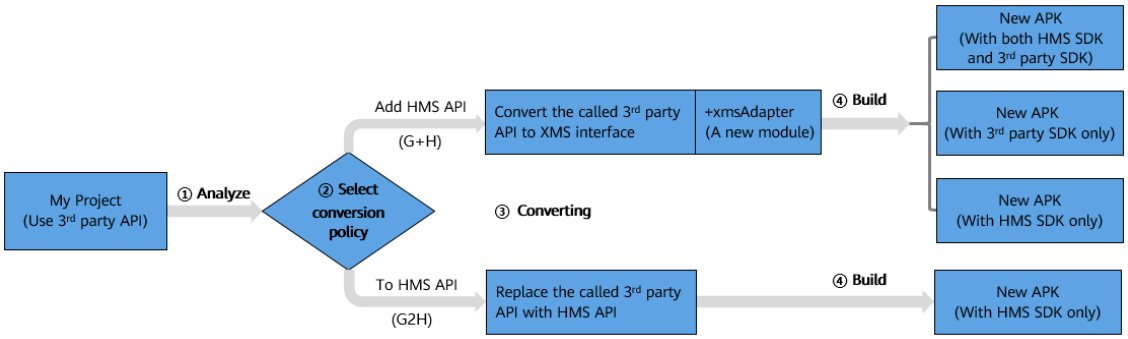
Firebase on alusta, joka mahdollistaa analytiikan ja kaatumisraporttien keräämisen sovelluksesta, tietokannan rakentamisen ja sovelluksen sisäisen viestinnän käyttäjälle. Firebase toimii niin Android- kuin iOS -sovellustenkin kanssa, joten se mahdollistaa yhtenäisen paikan tarkastella sovellukseen liittyvää analytiikkaa. Huaweilta löytyy myös samankaltainen palvelu, Huawei Analytics Kit. (6.)

4 MOBIILIPALVELUIDEN VAIHDON VAIHTOEHDOT

Työssä täytyi ensin vaihtaa Flow-sovelluksen käyttämät Googlen mobiilipalvelut Huawein mobiilipalveluihin, jotta tärkeät sovelluksen ominaisuudet pysyisivät toiminnassa, kuten karttaominaisuus. Huawei Mobile Service antaa kaksi eri vaihtoehtoa, miten sovelluksen kehittäjä voi ottaa sen käyttöön.

4.1 Vaihtoehdot

Sovelluksen kehittäjä voi joko tehdä adapterikerroksen ja lisätä Huawein mobiilipalvelut vanhan koodin päälle tai Googlen mobiilipalvelut voidaan poistaa kokonaan ja korvata Huawein omilla (kuva 3). Molemmissa vaihtoehdoissa on omat hyvät ja huonot puolensa.



KUVA 3. Vaihtoehdot miten sovelluksen muutos voidaan toteuttaa (7)

4.1.1 HMS API:n lisääminen

Lisäämällä HMS API voidaan käyttää Googlen mobiilipalveluita niissä laitteissa, jossa ne ovat saatavilla, ja Huawein mobiilipalveluita muulloin. Tämä toteutetaan lisäämällä XMS API, joka määrittää tukeeko laite pelkästään HMS:ää vai molempia palveluita (kuva 4). Jos laite tukee molempia palveluita, sovellus käyttää niitä palveluita, jotka on määritetty prioriteetiksi. Tämän tavan hyvinä puolina on vain yksi projekti, jota pitää ylläpitää ja kehittää. Huonoina puolina on se, että sovelluksen koko kasvaa ja muunnostyöhön kuluu enemmän aikaa. Tällä tavoin sovelluksen lähdekoodista tulee myös vaikeammin luettavaa, kun se sisältää molempien palveluiden toteutukset.

```
1. DetectedActivity detectedActivity = ...;
2. Intent intent = new Intent(BROADCAST_INTENT_ACTION);
3. if (org.xms.g.utils.GlobalEnvSetting.isHms) {
4.     intent.putExtra(DETECTED_ACTIVITY_EXTRA_ID, (com.huawei.hms.location.ActivityIdentificationData)detectedActivity.getHInst
   ance());
5. } else {
6. intent.putExtra(DETECTED_ACTIVITY_EXTRA_ID, (com.google.android.gms.location.DetectedActivity)detectedActivity.getGInstance
   ());
7. }
```

KUVA 4. Palvelun määrittävä koodi (7)

4.1.2 HMS API:lla korvaaminen

Jos GMS API:t päätetään korvata HMS API:llä, kaikki Googlen mobiilipalveluita käyttävät ominaisuudet vaihdetaan käyttämään HMS:ää. Tämä toteutetaan käymällä lähdekoodissa kaikki GMS:ää käyttävä koodi läpi ja korvaamalla se HMS API:n koodilla. Tämän tavan hyvinä puolina on muunnostyön nopeus ja se, että sovelluksen koko ei kasva. Lähdekoodi pysyy myös siistimpänä, kun ei tarvitse pitää kahta toteutusta samassa projektissa. Huonoina puolina on kaksi eri projektia, joita pitää yllä ja kehittää. Mahdollisena haasteena tässä lähestymistavassa on myös tulevaisuudessa tulevat konfliktit, kun uusia ominaisuuksia yhdistetään päähaarasta HMS-haaraan, koska koodia joudutaan vaihtamaan useassa tiedostossa projektin sisällä. Tämä saattaa lisätä työmäärää, kun tehdään uusia julkaisuja AppGalleryyn.

4.2 Valinta

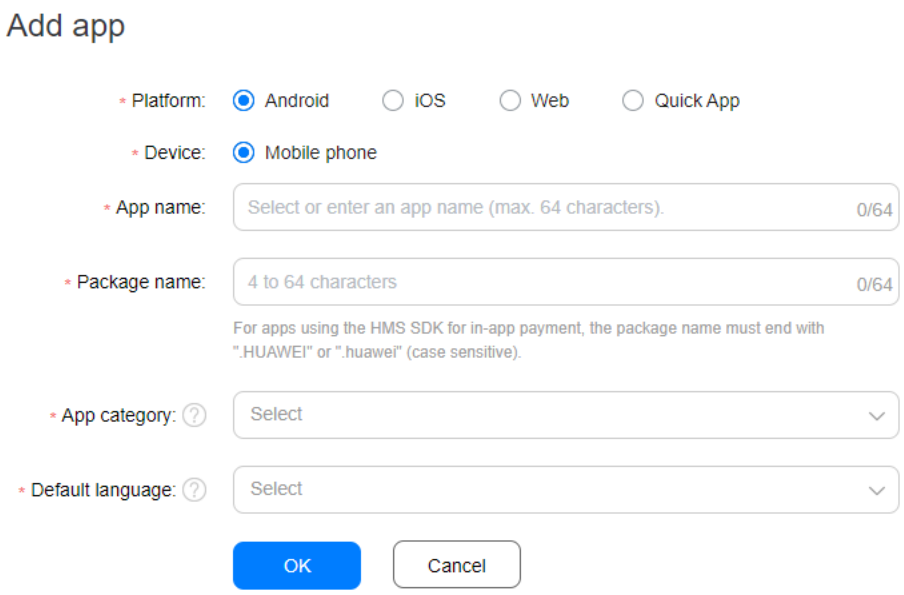
Valinnassa päädyttiin siihen, että tehdään rinnakkainen sovellus, johon Googlen palvelut korvataan Huawein palveluilla. Tällä tavoin pääsovellus voidaan pitää puhtaana Huawein palveluista ja erillisellä sovelluksella päästään näkemään, kuinka paljon kysyntää sovellukselle on AppGalleryssä. HMS-sovellus pidetään omassa haarassaan Git-versionhallintajärjestelmässä, johon voidaan uutta päivitystä tehdessä ottaa uusimmat muutokset Gitin päähaarasta. Tämä toteutustapa tuo hieman lisää työtä päivitysten julkaisuun Huawein alustalle, mutta ei merkittävää kuitenkaan, ja aluksi HMS-version päivityssykliä voidaan pidentää, kunnes nähdään todellinen kysyntä.

5 TOTEUTUS

Työ alkoi tutustumalla Huawein kehittäjäsivuston dokumentteihin mobiilipalveluiden käyttöönotosta. Sivustolla on laaja kattaus tietoa ja oppaita Huawein palveluiden käytöstä. Näihin dokumentteihin oli hyvä palata projektin aikana, jos jokin asia oli epäselvä.

5.1 AppGallery Connect

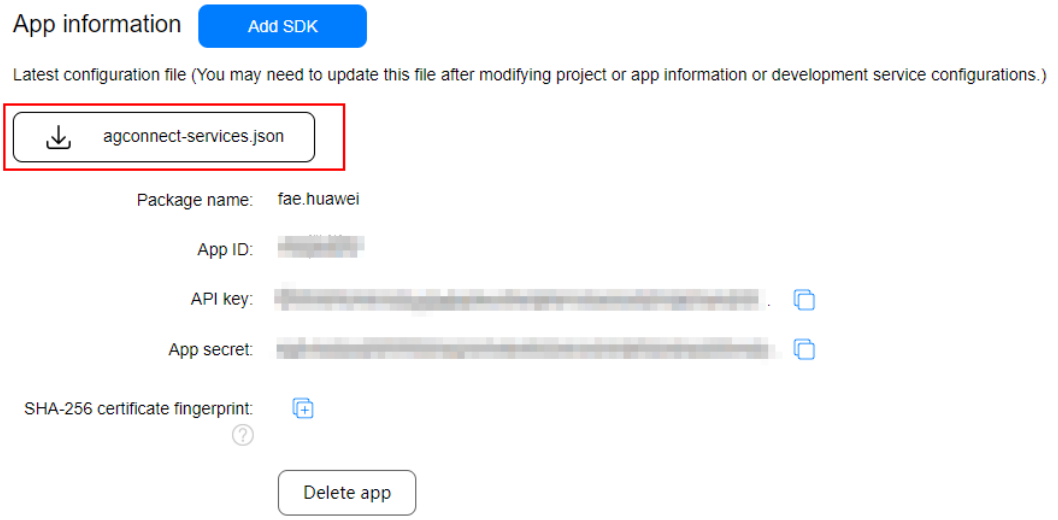
Ensimmäisenä täytyi luoda projekti AppGallery Connectiin ja lisätä sovellus siihen (kuva 5). Lisättäessä sovellusta projektiin sille annetaan tuetut laitemuodot, tässä tapauksessa puhelimet ja tabletit, ja sen pakettinimi, joka on määritetty sovelluksen Gradle-tiedostossa.



KUVA 5. Sovelluksen lisäys (8)

Kun tiedot on täytetty ja sovellus on lisätty onnistuneesti, voidaan ladata määritystiedosto mobiilipalveluiden käyttöä varten (kuva 6). Tämä tiedosto täytyy lisätä Android Studiossa sovelluksen projektin juureen.

Tässä kohdassa täytyy myös lisätä SHA-256-sormenjälki. Tämän avulla voidaan todentaa, että mobiilipalveluiden rajapintoja käyttää sille rekisteröity sovellus.



KUVA 6. Määritystiedoston latauspainike (8)

5.2 Gradle-määrittely

Gradle-koontityökalun avulla voidaan projektiin lisätä ulkopuolisia kirjastoja. Kun projekti on valmisteltu AppGallery Connectissa, voidaan tarvittavat kirjastot lisätä sovellukseen Gradle-liitännäisinä.

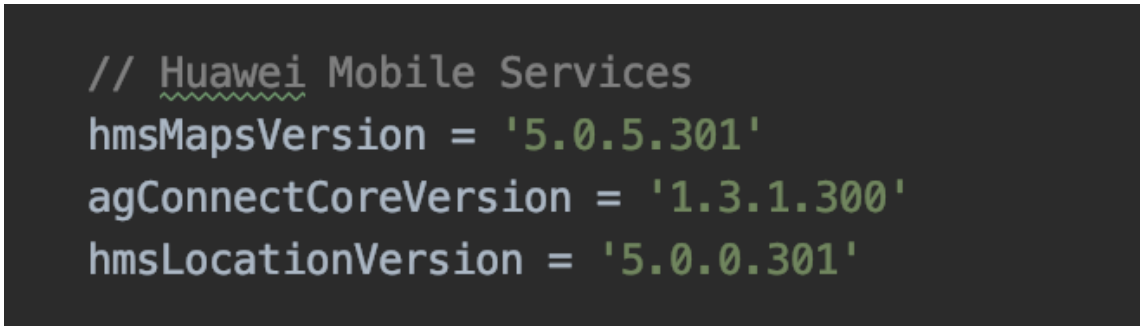
Ensimmäisenä lisätään tarvittava Maven-tietovaraston osoite, josta tarvittavat kirjastot löytyvät (kuva 7). Tämä lisätään projektin build.gradle-tiedostoon.

```
maven {
    url 'https://developer.huawei.com/repo/'
}
```

KUVA 7. Huawein Maven-tietovaraston osoite

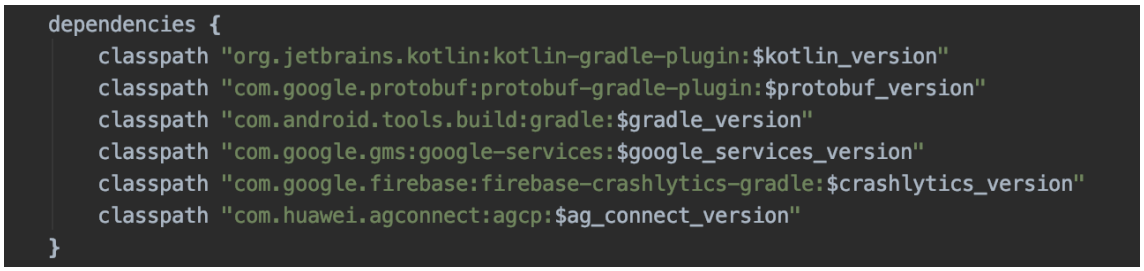
Sitten määritellään versiot AppGallery Connectille (kuva 8) ja tarvittaville kirjastoille, joihin kuuluu kartta- ja sijaintipalvelu (kuva 9).

KUVA 8. AppGallery Connectin version määrittely



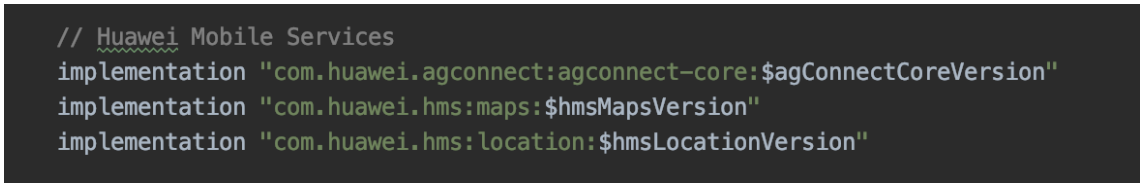
KUVA 9. Tarvittavien kirjastojen versioiden määrittely

Viimeisenä projektitason buil.gradlessa lisätään varsinainen riippuvuus AppGallery Connectiin (kuva 10).



KUVA 10. AppGallery Connectin riippuvuuden määrittely

Tämän jälkeen voidaan siirtyä moduulitason build.gradle-tiedostoon ja määritellä kirjastojen riippuvuudet (kuva 11).



KUVA 11. Tarvittavien kirjastojen riippuvuuksien määrittely

Lopuksi vielä otetaan käyttöön AppGallery Connect, jolloin päästään käyttämään kirjastoja sovelluksessa (kuva 12).

```
apply plugin: 'com.huawei.agconnect'
```

KUVA 12. AppGallery Connectin käyttöönoton määrittely

5.3 Lähdekoodin muutos

Alustavasti tarvittavat muutokset sovellukseen olivat kartat, Youtube-videoiden aukaisu ja Firebasen korvaus, mutta vähän ennen työn aloittamista Firebaseen oli tullut päivitys, joka mahdollisti sen käytön myös HMS-laitteilla.

Muutos aloitettiin karttaominaisuudesta. Huawei Map Kit on rakenteeltaan hyvin samanlainen kuin Googlen, joten muutostyö oli melko suoraviivainen. Sovelluksen karttakomponentit muutettiin käyttämään HMS-riippuvuuksia, muutamia poikkeuksia lukuun ottamatta, joissa toiminnallisuutta piti hieman hioa. GMS-versiossa karttanäkymää voi vaihtaa tavallisen ja satelliittinäkymän välillä, mutta Map Kit ei tukenut satelliittia vielä työn teon aikana. Tämän takia sovelluksesta poistettiin mahdollisuus vaihtaa näkymää.

Seuraavaksi siirryttiin YouTube-videoiden aukaisuun. Koska HMS-laitteissa ei pysty näyttämään YouTube-videoita sovelluksen sisällä, täytyi videot saada aukeamaan verkkoselaimessa. Videosoittimen alustuksessa tarkistetaan, tuleeko palvelun puuttumiseen liittyvä virhe. Jos oikea virhe tulee, silloin video aukaistaan verkkoselaimessa (kuva 13).

```
@Override
public void onInitializationFailure(YouTubePlayer.Provider provider, YouTubeInitializationResult youTubeInitializationResult) {
    if (youTubeInitializationResult == YouTubeInitializationResult.SERVICE_MISSING) {
        finish();
        startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse(YOUTUBE_URL + videoId)));
```

KUVA 13. YouTube-soittimen alustuksen onnistumisen tarkistaminen

Alun perin oli tarkoitus ottaa käyttöön myös Huawein analytiikka-kirjasto, mutta hieman ennen työn aloitusta Firebasen kirjastoon tuli päivitys, joka mahdollisti

sen käytön myös ilman Googlen mobiilipalveluita. Tämän vuoksi oli yksinkertaisempaa käyttää Firebasea molemmissa paketeissa.

Jotta nämä kaksi pakettia voitaisiin erottaa Firebase Consolessa, täytyi seuraavaksi lisätä analytiikkaan käyttäjätieto, joka kertoo, kumpi mobiilipalvelu käyttäjällä on käytössä. Tämä koodipätkä nimeää user propertyn, kun käyttäjä aukaisee sovelluksen ensimmäistä kertaa (kuva 14). Sama lisättiin myös sovelluksen GMS-versioon, mutta vain eri nimellä. Tämän jälkeen kaikki analytiikka on sidottu siihen user propertyyn ja sitä voidaan suodattaa halutulla mobiilipalvelulla.

```
/**
 * Sends mobile service user property when app is started for the first time
 */
public void sendUserPropertyMobileService() {
    SharedPreferences prefs = mContext.getSharedPreferences(PREFS_FILE, i: 0);
    String mobileService = prefs.getString(PREFS_KEY_ANALYTICS_USER_PROP_MOBILE_SERVICE, s1: null);

    if (mobileService == null) {
        sendUserProperty(UserPropertyKeys.ANALYTICS_USER_PROP_MOBILE_SERVICE,
                MOBILE_SERVICE);
        prefs.edit().putString(PREFS_KEY_ANALYTICS_USER_PROP_MOBILE_SERVICE, MOBILE_SERVICE).apply();
    }

}
```

KUVA 14. Firebase-analytiikan user propertyn lähettäminen.

Vielä lopuksi user property täytyy käydä rekisteröimässä Firebase Consolessa, jotta palvelu alkaa keräämään siihen liitettyä tietoa. Rekisteröinnin jälkeen voi mennä joitain tunteja, ennen kuin se ilmestyy analytiikkaan.

6 JULKAISU

Kun HMS-pakettiin tarvittavat muutokset oli tehty, koodi katselmoitu ja testattu, oli aika julkaista sovellus Huawei AppGalleryssä.

Ensimmäisenä sovellukselle täytyy luoda kauppasivu, jonka käyttäjä näkee selatessaan sovelluksia kaupassa. Tänne lisätään sovelluksen ikoni, markkinointikuvat ja kuvaus, millainen sovellus on kyseessä. Tämän osan julkaisusta hoiti Polarin markkinointiosasto.

Seuraavana lisätään itse sovelluspaketti. Kun paketti on lisätty, voidaan valita sille maat ja alueet, johon sovellus halutaan julkaista. Palvelu tunnistaa sovelluksen käyttämät oikeudet ja niiden käyttötarkoitus täytyy selittää.

Kun nämä kaikki on tehty, voidaan sovellus lähettää katselmoitavaksi AppGalleryyn. Hyväksytyn katselmoinnin jälkeen on kaksi vaihtoehtoa julkaisulle. Voidaan joko julkaista kaikille kerralla tai voidaan tehdä ”phased release”, joka tarkoittaa, että voidaan valita prosenttimäärä käyttäjistä, joille julkaisua tarjotaan. Tämä rajoitettu julkaisu kestää sille määritettyyn päivämäärään asti, jonka jälkeen kattavuus nostetaan sataan prosenttiin.

Kaiken tämän jälkeen sovellus on ladattavissa käyttäjille AppGallerystä.

7 PÄIVITTÄMINEN

HMS-version ylläpito toteutettiin siten, että kun on aika julkaista uusi versio, HMS-paketin Git-haaraan yhdistetään sen hetkinen develop-haara, jossa on kaikki uusimmat muutokset. Kun haarat on yhdistetty, Jenkins-palvelin kokoaa uudet APK-paketit sovelluksesta.

AppGallery Connectissa sovelluksen päivittäminen toimii samoin kuin sovellusta julkaistaessa. Kauppasivulle voidaan tehdä muutoksia, kuten lisätä, mitä muutoksia uusimmassa päivityksessä on. Sivulta valitaan ”Upgrade”, joka luo uudelle versiolle oman sivun, jossa vaiheet ovat samat kuin julkaistaessa eli paketin lisäys, oikeuksien tarkoitukset ja julkaisun kattavuus. Kun paketti on mennyt taas läpi katselmoinnista, on se julkaistu käyttäjille ladattavaksi.

8 YHTEENVETO

Opinnäytetyön tavoitteena oli tehdä Polar Flow -sovelluksesta vaihtoehtoinen versio, joka käyttää Huawein mobiilipalveluita Googlen sijasta, ja julkaista se Huawei AppGallery -mobiilisovelluskauppaan. Muutos vaadittiin, koska uudet Huawein laitteet eivät pysty enää käyttämään Googlen palveluita

Lopputulos oli onnistunut. Palvelut saatiin vaihdettua ja sovellus julkaistua ajallaan. Ennen työn alkua tulleiden kirjastopäivitysten avulla työ oli suoraviivaisempi ja sovelluksen ylläpitäminen ja analysointi on helpompaa, kun kaikki sovellusversiot löytyvät samasta paikasta.

Jatkokehitykseen jäi kartan toimiminen Kiinassa, sekä julkaisujen automatisointi. Huawei Map Kit ei työn teon ajankohtana tukenut Kiinan aluetta, joten tulevaisuudessa tulisi miettiä, onko muita vaihtoehtoja kartalle vai odotetaanko, että Huawei tukee sitä. Yksi vaihtoehdoista olisi Mapbox-karttapalvelu, joka tukee myös Kiinan aluetta. Julkaisujen automatisointi on toinen asia, jota olisi mahdollista jatkokehittää. Työn tekohetkellä AppGallery Connect ei tukenut samanlaista automatisointia kuten Google Play, johon Jenkins pystyi automaattisesti lähettämään uudet julkaistavat APK-paketit. Nyt ne täytyy käydä käsin lisäämässä. Tulevaisuuteen jää myös nähtäväksi, kuinka paljon Git-konfliktien selvittämiseen menee aikaa, kun uusia ominaisuuksia yhdistetään HMS-haaraan.

Työ oli erittäin opettavainen kokemus. Siinä pääsi käymään läpi uuden sovelluksen julkaisuprosessin sovelluskauppaan ja sai ottaa paljon vastuuta sen tekemisestä. Uuden projektin luominen sovelluskauppaan ja julkaisuprosessi olivatkin uusia kokemuksia minulle ja työn avulla ne tulivat hyvin tutuiksi. Työtä helpottivat valmiit ratkaisut, joita oli käytetty jo aiemmassa kehitystyössä, kuten Jenkins, jonka määrittelyjä ei tarvinnut alkaa muuttamaan.

LÄHTEET

1. Keitä olemme. 2021. Polar Electro Oy. Saatavissa: https://www.polar.com/fi/tietoa_polarista/keita_olemme. Hakupäivä 14.11.2020.

2. The best of Google, right on your devices. Android. Saatavissa: https://www.android.com/gms/. Hakupäivä 5.2.2021.

3. Meet Android Studio. 2020. Android. Saatavissa: https://developer.android.-com/studio/intro. Hakupäivä 14.11.2020.

4. About. Git. Saatavissa: https://git-scm.com/about. Hakupäivä 14.11.2020.

5. Jenkins Press Information. Jenkins. Saatavissa: https://www.jenkins.io/press/. Hakupäivä 1.2.2021.

6. Firebase helps you build and run successful apps. Firebase. Saatavissa: https://firebase.google.com/. Hakupäivä 15.11.2020.

7. Convertor. 2021. Huawei. Saatavissa: https://developer.huawei.com/consumer/en/doc/development/Tools-Guides/convertor-0000001050147221. Hakupäivä 15.11.2020.

8. Getting Started with Android. 2021. Huawei. Saatavissa: https://developer.huawei.com/consumer/en/doc/development/AppGallery-connect-Guides/agc-get-started. Hakupäivä 15.11.2020.