

VAASAN YLIOPISTO

TEKNIIKAN JA INNOVAATIOJOHTAMISEN YKSIKKÖ

OHJELMISTOTEKNIikka

Ville Hanhisuanto (x102755), Matias Talonpoika (x102784)

”HIMAAN” -MOBIILIAPLIKAATIO

Software Specification -raportti

Mobile application development

Vaasassa 30.6.2019

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO	2
1 TIIVISTELMÄ	3
2 SOVELLUS	4
2.1 Sovelluksen periaate	4
2.2 Sovelluksen osat	4
2.2.1 Views	4
2.2.2 ViewModels	5
2.2.3 Models ja Helper	5
2.2.4 Tietokanta	5
2.3 Käytetyt funktiot ja niiden toiminnot	5
2.4 Käytetyt parametrit ja niiden väliset riippuvuudet	6

1 TIIVISTELMÄ

Tämä on raportti ”Himaan”-nimisestä yhteismatkustus, tai kansankielisesti kimppakyyti mobiilisovelluksesta. Sovellus oli harjoitustyö Vaasan yliopistolla järjestetyssä Mobile Application Development -kurssilla, jonka luennot ja harjoitukset piti Sebastian Laitila. Mobiilisovellus on toteutettu kahden henkilön ryhmässä, johon kuuluvat Ville Hanhisuanto ja Matias Talonpoika. Roolijako oli seuraavanlainen:

- Ville, vastaava devaaja. Vastuualueina mm. sovelluksen Back-End ja sen toiminnallisuuden yhdistäminen Front-End:iin.
- Matias, raporttivastaava. Vastuualueina raportointi ja sovelluksen Front-End.

”Himaan” syntyi Villen ajatuksesta nykyaikaisesta kimppakyyti-sovelluksesta. Ryhmä totesi sovelluksen tarpeeksi haastavaksi, mutta samalla toteutuskelpoiseksi. Ryhmä päätyi käyttämään sovelluksen kehittämiseen Xamarinia, sekä opettajan suosittamaa ilmaista tietokantaa Firebasea.

Xamarin oli menetelmä molemmille ryhmäläisille entuudestaan tuntematon, mutta internet tarjosi Microsoftin toimesta erittäin laajat materiaalit tarvittavan osaamisen kerryttämiseksi ja siten sovelluksen kehittämiseksi. Xamarinilla voidaan luoda sovelluksia sekä Android, että iOS käyttöjärjestelmille. Tämä työ on testattu vain androidilla.

2 SOVELLUS

2.1 Sovelluksen periaate

Sovelluksen perusajatuksena on tarjota käyttäjille intuitiivinen käyttöliittymä, jossa käyttäjät voivat pyytää kyytiä, luoda kyydin muille käyttäjille tarkasteltavaksi tai vain tarkastella kyytitarjontaa. Kyytiä varten on ilmoitettava seuraavat tiedot: lähtöpaikka, päämäärä, päivämäärä, vapaiden/tarvittavien paikkojen määrä, sekä kuvaus, johon suositellaan laittamaan tapa, jolla yhteydenotto tapahtuu.

Käyttäjien tulee kirjautua sisään, ennen kuin he voivat tehdä mitään yllämainituista toimenpiteistä. Kirjautumiseen käytetään Googlen tarjoamaa Firebase Authenticationia. Näin mm. lisätään sovelluksen tietoturva sekä ehkäistään mahdollista väärinkäyttöä.

Vain sovelluksen kehittäjillä on oikeudet poistaa ja muokata epämääräiset kyytipyynnöt tai -tarjoukset. Sovellus perustuu käyttäjien väliseen keskinäiseen luottamukseen, joten sen takia rekisteröidessä on kerättävä joku tieto käyttäjästä, jolla voi yksilöidä kunkin käyttäjän, siten että sovelluksen väärinkäyttöä ei tehtäisi liian helpoksi.

Koska kirjautumisen toteuttaminen osoittautui oletettua haastavammaksi, ei sovellukseen keretty tällä aikamääreellä lisäämään rekisteröitymistä. Kirjautumiseen on käytettävä sähköpostina ”hanhi123123123@gmail.com” ja salasanaksi ”salasana”, jota sovellus ehdottaa automaattisesti kirjautumisen yhteydessä. Rekisteröinnin toteuttaminen on seuraava askel, jos sovellus päättyy jatkokehitykseen.

2.2 Sovelluksen osat

Sovellus on rakennettu Model-View-ViewModel, eli MVVM arkkitehtuurilla.

2.2.1 Views

Käyttöliittymän ulkoasu, joka on yksinkertainen, käyttäjäystävällinen ja intuitiivinen.

2.2.2 ViewModels

Sisältää käytetyt funktiot ja niiden välisen toiminnallisuuden.

2.2.3 Models ja Helper

Models sisältää sovelluksen tarvitsemat luokat, eli pyynnön ja tarjouksen. Helper luo luokista olioita ja on tarvittaessa yhteydessä tietokantaan.

2.2.4 Tietokanta

Tietokantana toimii Firebase Database. Se sisältää kaiken sovelluksen käyttämästä tiedosta, eli pyynnot ja tarjoukset.

2.3 Käytetyt funktiot ja niiden toiminnot

Sovelluksen keskeisimpiä funktioita ja niiden käyttöperiaatteita, ovat:

- Login(sähköposti, salasana), joka luo yhteyden Firebaseen, jonka perusteella päästää sisään sovellukseen tai ilmoittaa väärästä salasanasta ja/tai sähköpostiosoitteesta.
- searchOffersAsync(), ottaa yhteyden Firebasen tietokantaan ja palauttaa tarjotut kyydit.
- searchNeedsAsync(), ottaa yhteyden Firebasen tietokantaan ja palauttaa pyydetty kyydit.
- AddNeed (mihin, mistä, päivämäärä, tarvittavien paikkojen lukumäärä, kuvaus, henkilön id), tallentaa pyynnön tietokantaan.

- AddOffer(mihin, mistä, päivämäärä, vapaiden paikkojen lukumäärä, kuvaus, henkilön id), tallentaa tarjouksen tietokantaan.

2.4 Käytetyt parametrit ja niiden väliset riippuvuudet

Sovelluksen keskeisimpiä parametrejä, niiden käyttöperiaatteita ja niiden välisiä riippuvuuksia ovat:

- string AuthToken, johon palautuu kirjautumisen yhteydessä käyttäjän uniikki id, joka asetetaan tarjouksen ja pyynnön luonnin yhteydessä tietokantaan, jotta tapahtuman luoja on tunnistettavissa.
- Need, pyynnön luokka.
- Offer, tarjouksen luokka
- FirebaseHelper, luokka, joka ottaa yhteyden tietokantaan. Tiedon tallentaminen ja hakeminen tapahtuu tämän kautta.