



Firestore-tietokannan hyödyntäminen mobiilisovelluksessa

Marika Parttimaa

OPINNÄYTETYÖ
Toukokuu 2020

Tieto- ja viestintäteknikan koulutus
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikan koulutus
Ohjelmistotekniikka

PARTTIMAA, MARIKA:
Firebase-tietokannan hyödyntäminen mobiilisovelluksessa

Opinnäytetyö 21 sivua
Toukokuu 2020

Tässä opinnäytetyössä tarkastellaan Firebase-tietokannan hyödyntämistä mobiilisovelluksessa. Työn tarkoituksena on havainnollistaa Firebase-ympäristön monipuolisia mahdollisuuksia mobiilikehityksen näkökulmasta.

Firestore on Googlen ylläpitämä ilmainen tietokantapalvelu. Palvelu on suunniteltu ensisijaisesti mobiilikehitykseen, mutta soveltuu myös web-sovelluksille. Firestore tukee sekä iOS- että Android-käyttöjärjestelmiä.

Työssä käytetään tarkastelun tukena yksinkertaista, Instagramin kaltaista Photo Feed -mobiilisovellusta. Tarkastelussa käytetty Photo Feed on toteutettu React Nativella, mikä on Facebookin julkaisema, Java-pohjainen ohjelmointikieli. Ohjelmointikielenä React Native on monipuolinen, sillä se on oletuksena natiivi sekä web-, iOS-, että Android-pohjalle. Tarkastellun sovelluksen keskeisimmät ominaisuudet ovat kirjautuminen sovellukseen sähköpostiosoitteella ja salasanalla, sovellukseen lisättyjen kuvien selaaminen sekä omien kuvien lisääminen sovellukseen. Näistä ominaisuuksista jokaisessa on hyödynnetty Firestore-palvelua.

Työssä todetaan Firestore-tietokannan keskeisimpien etujen olevan monipuoliset mahdollisuudet yhden palvelun alla sekä usean eri alustan samanaikainen tukeminen. Mobiilikehityksen kannalta todetaan Firestore-ympäristön ja React Nativen olevan ajankohtainen yhdistelmä.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree in ICT Engineering
Software Engineering

PARTTIMAA, MARIKA:
Utilizing Firebase Database in a Mobile Application

Bachelor's thesis 21 pages
May 2020

This thesis examines the utilization of the Firebase database in a mobile application. The purpose of this work is to illustrate the diverse possibilities of the Firebase environment from the perspective of mobile development.

Firebase is a free database service maintained by Google. The service is primarily designed for mobile development but is also suitable for web applications. Firebase supports both iOS and Android operating systems.

The work uses a simple, Instagram-like mobile app, Pho-to Feed, to support the review. Photo Feed is implemented with React Native, a Java-based programming language published by Facebook. As a programming language, React Native is versatile, as it is the default native for both web, iOS and Android. The most important features of the application are logging in to the application with an email address and password, browsing the images added to the application and adding own images to the application. Each of these features takes advantage of Firebase.

The main advantages of the Firebase database are stated to be the versatile possibilities under one service and the simultaneous support of several different platforms. In terms of mobile development, the Firebase and React Native are said to be a topical combination.

Key words: firebase, mobile development, react native

SISÄLLYS

1	JOHDANTO	6
2	FIREBASE	7
2.1	Kehitysalustan palvelut	7
2.1.1	Google Analytics ja Firebase Cloud Messaging.....	7
2.1.2	Cloud Firestore ja Firebase Storage	7
2.1.3	Firebase Hosting	8
2.1.4	ML Kit	8
2.2	Kehitysalustan teknisiä ominaisuuksia.....	8
2.2.1	Kaatumisraportit ja suorituskyvyn analysointi.....	8
2.2.2	Firebase Test Lab	9
2.2.3	Dynaamiset linkit	9
2.3	Käyttö mobiilisovelluksessa	9
2.3.1	Käyttäjän todentaminen.....	10
2.3.2	Reaaliaikainen tietokanta	11
3	PHOTO FEED	12
3.1	Ominaisuudet ja arkkitehtuuri	12
3.2	Tekninen toteutus	17
3.2.1	React Native.....	17
3.2.2	Firebase osana sovellusta.....	17
4	POHDINTA	20
	LÄHTEET.....	21

LYHENTEET JA TERMIT

Beta	Ohjelmistoversio, joka on julkaistu testikäyttöön ja jota kehitetään testitulosten mukaan ennen lopullista julkaisua.
FMC	<i>Firestore Cloud Messaging</i> , on eri alustojen välinen viestintäratkaisu, jonka avulla voi lähettää viestejä turvallisesti.
Framework	Ohjelmistokehys, joka sisältää valmiita ohjelman osia sovelluksen sisällä.
GCM	<i>Google Cloud Messaging</i> , oli Googlen kehittämä matkaviestinilmoituspalvelu, joka toimi Firestore Cloud Messaging:n edeltäjänä.
HTTPS	<i>Hypertext Transport Protocol Secure</i> , on verkkoviestintäprotokolla, joka suojaaa tietojen eheyttä ja luottamuksellisuutta.
JSON	<i>JavaScript Object Notation</i> , on yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen.
NoSQL	<i>Not only SQL</i> , on tietokanta, joka poikkeaa perinteisestä relaatiotietokannasta.
SKD	<i>Semi Knocked Down</i> , sarja, joka sisältää tuotteen koamiseen tarvittavia osia.
TLS	<i>Transport Layer Security</i> , on salausprotokolla, jolla voidaan suojata sovellusten tietoliikenne IP-verkossa.

1 JOHDANTO

Tässä opinnäytetyössä tarkastellaan Firebase-tietokannan hyödyntämistä mobiilisovelluksessa. Työn tarkoituksena on havainnollistaa Firebase-ympäristön monipuolisia mahdollisuuksia mobiilikehityksen näkökulmasta. Aluksi työssä perehdytään Firebase-tietokantaan sekä sen monipuolisiin ominaisuuksiin.

Firestore on Googlen ylläpitämä ilmainen tietokantapalvelu. Palvelu on suunniteltu ensisijaisesti mobiilikehitykseen, mutta soveltuu myös web-sovelluksille. Firestore tukee suoraan sekä iOS- että Android-käyttöjärjestelmää.

Tarkastelun tukena käytetään yksinkertaista, Instagramin kaltaista Photo Feed -mobiilisovellusta. Photo Feed on toteutettu React Nativella, mikä on Facebookin julkaisema, Java-pohjainen ohjelmointikieli. Ohjelmointikielenä React Native on monipuolinen, sillä se on oletuksena natiivi sekä web-, iOS- että Android-pohjalle.

Mobiilikehityksen kannalta Firestore-tietokannan keskeisimpiä etuja ovat monipuoliset mahdollisuudet yhden palvelun alla sekä usean eri alustan samanaikainen tukeminen.

2 FIREBASE

Firestore on Googlen tarjoama mobiili- ja verkkosovellusten kehitysalusta. Se on reaaliaikainen pilvipalveluna toimiva NoSQL-tietokanta (NoSQL), joka synkronoi sovellustiedot vaivattomasti myös iOS-, Android- ja web-laitteiden välillä, ja tallentaa ne tietokantaan. Firestore on käytettävissä myös Unitylla ja C++:lla, mutta se ei ole oleellista tämän työn kannalta. Maaliskuussa 2020 Firestore-palvelua käytti jo yli 1,5 miljoonaa sovellusta. (Google. n.d.)

2.1 Kehitysalustan palvelut

2.1.1 Google Analytics ja Firebase Cloud Messaging

Google Analytics on maksuton sovellusmittausratkaisu, joka tarjoaa tietoa sovelluksen käytöstä. Se analysoi esimerkiksi käyttäjien tottumuksia sovellusten säännöllisestä käytöstä.

Aiemmin nimellä Google Cloud Messaging (GCM) tunnettu, Firebase Cloud Messaging (FCM) on alustojen välinen ratkaisu viesteille ja ilmoituksille Androidille, iOS:lle ja verkkosovelluksille. Palvelua on voinut käyttää ilmaiseksi vuodesta 2016 alkaen.

2.1.2 Cloud Firestore ja Firebase Storage

Cloud Firestore -ohjelmiston lopullinen versio julkaistiin 31. tammikuuta 2019, jolloin siitä tuli Firestore-palvelun virallinen tuote. Aikaisemmin palvelu oli saatavana vain beetaversiona (Beta). Se on Firestore-tietokannan alkuperäisen reaaliaikaisen tietokannan seuraaja ja sallii sisäkkäiset asiakirjat ja kentät perinteisesti tietokannassa olevan puunäkymän sijasta.

Firestore Storage tarjoaa suojatut tiedostojen lataukset ja lataukset Firestore-sovelluksille verkon laadusta riippumatta. Tätä hyödynnetään kuvien, äänen, videon tai muun käyttäjän luoman sisällön tallentamiseen.

2.1.3 Firebase Hosting

Firebase Hosting on ylläpidollinen verkkopalvelu, joka julkaistiin 13. toukokuuta 2014. Se tukee staattisten tiedostojen, kuten CSS, HTML, JavaScript ylläpitoa, sekä tukea pilvitoimintojen kautta. Palvelu käyttää tiedonsiirtoon HTTP Secure:a (HTTPS) ja Transport Layer Security:a (TLS).

Yrityksen mukaan Firebase Hosting kehitettiin asiakkaiden pyynnöistä; kehittäjät käyttivät Firebase-tietokantaa sen reaaliaikaisen tietokannan takia, mutta tarvitsivat sijainnin sisällön ylläpitämiseen.

2.1.4 ML Kit

ML Kit on mobiili koneoppimisjärjestelmä kehittäjille. Järjestelmän valmis versio julkaistiin 8. toukokuuta 2018. ML Kit -sovellusliittymillä on useita ominaisuuksia, kuten optinen merkkitunnistus, kasvojen havaitseminen, viivakoodien skannaaminen, kuvien merkitseminen ja maamerkkien tunnistaminen.

Se on tällä hetkellä nousevassa suosiossa iOS- tai Android-kehittäjille. Esimerkiksi kasvojen tunnistus on käytössä yhä enemmän uusissa puhelinmalleissa.

2.2 Kehitysalustan teknisiä ominaisuuksia

2.2.1 Kaatumisraportit ja suorituskyvyn analysointi

Kaatumisraportointi luo yksityiskohtaiset raportit sovelluksen virheistä. Virheet on ryhmitelty klustereihin, joissa on samanlaisia pinojälkiä, sekä sovelluksen käyttäjille aiheutuvien vaikutusten vakavuuden perusteella.

Automaattisten raporttien lisäksi kehittäjä voi kirjata mukautettuja tapahtumia havaitakseen kaatumiseen johtavat vaiheet. Ennen erillisiä kaatumisraportteja Firebase käytti omaa Firebase Crash Reporting -sovellusta.

Firestore Performance tarjoaa tietoa sovelluksen suorituskyvystä. Paikallisen sovelluksen suorituskyvyn lisäksi se analysoi, kauanko datansiirrossa kahden käyttäjän välillä kestää.

2.2.2 Firebase Test Lab

Firebase Test Lab tarjoaa pilvipohjaisen toimintaympäristön Android- ja iOS-sovellusten testaamiseen yhdessä operaatiossa. Kehittäjät voivat testata sovelluksiaan useilla laitteilla ja laitekoonpanoilla.

Testitulokset, mukaan lukien lokit, videot ja kuvakaappaukset, ovat saatavana Firebase-konsolissa. Vaikka kehittäjä ei olisi kirjoittanut sovellusohjelmaansa testikoodia, Test Lab voi käyttää sovellusta automaattisesti etsien kaatumisia. Test Lab iOS:lle on tällä hetkellä beetaversio.

2.2.3 Dynaamiset linkit

Dynaamiset Firebase-linkit ovat älykkäitä URL-osoitteita, jotka muuttavat dynaamisesti toimintaansa tarjoamaan parhaan käytettävissä olevan kokemuksen useilla alustoilla, mukaan lukien työpöytäselaimet, iOS ja Android, sekä perusteelliset linkit mobiilisovelluksiin.

Dynaamiset linkit toimivat kaikissa sovelluksen asennuksissa. Jos käyttäjä avaa dynaamisen linkin iOS:ssa tai Androidissa ja sovellusta ei ole asennettu, käyttäjä pyydetään asentamaan sovellus ensin. Asennuksen jälkeen sovellus alkaa toimia ja voi käyttää linkkiä.

2.3 Käyttö mobiilisovelluksessa

Firebase tarjoaa mobiilikehittäjälle mahdollisuuden hyödyntää monia eri ominaisuuksia yhdellä palvelulla. Kun Firebase-tietokantaan on luotu uusi projekti, saa mobiilikehittäjä SDK-avaimen (SKD) (ohjelma 1), jolla tietokanta yhdistetään suoraan sovelluksen kooditasolla. Näin jokainen käyttäjä voi hyödyntää tietokantaa sovelluksessa reaaliaikaisesti, eikä erillistä muutostenajoa tietokantaan tarvita.

```
const firebaseConfig = {  
  apiKey: "AIzaSyBc6vRF6wow3l2EuvRikvQpD_00I2TA-fc",  
  authDomain: "photo-feed-5ff29.firebaseio.com",  
  databaseURL: "https://photofeed-f10a6.firebaseio.com",  
  projectId: "photo-feed-5ff29",  
  storageBucket: "photo-feed-5ff29.appspot.com",  
  messagingSenderId: "168362324581",  
  appId: "1:168362324581:web:966de5125db7338b938c8c",  
  measurementId: "G-RG0HC1JF5M"  
};
```

OHJELMA 1 SKD-avain Firebase-projektin yhdistämiseksi sovellukseen

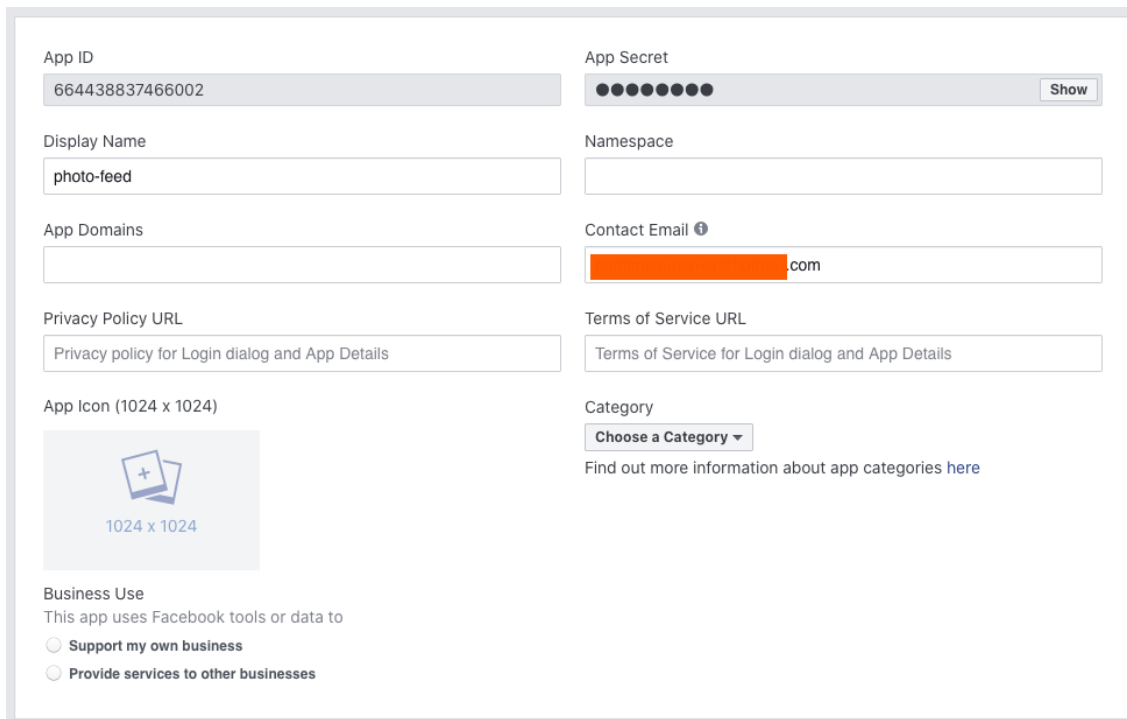
2.3.1 Käyttäjän todentaminen

Firestore sisältää käyttäjän hallintajärjestelmän, jonka avulla kehittäjät voivat sallia käyttäjän todennuksen Firestore-palveluun tallennetulla sähköposti- ja salasanan sisäänkirjautumisella. Käyttäjän todentamista voidaan hyödyntää paitsi tietokannan käyttöoikeuksissa, myös ylipäätään koko sovellukseen kirjautumisessa. Tämän lisäksi Firestore kuitenkin tukee monipuolisesti myös sosiaalisen median sisäänkirjautumisen tarjoajia, kuten

- Google
- Google Play Games
- Facebook
- Twitter
- GitHub
- Yahoo
- Microsoft
- Apple.

Näistä esimerkiksi Facebook tarjoaa Facebook for Developers -palvelun. Palvelussa voidaan luoda Facebook App halutulle sovellusalustalle, kuten web, iOS tai Android. Sovelluksen luoman App ID:n ja App Secret -tunnuksen avulla kyseinen

Facebook App voidaan linkittää luotuun sovellukseen suoraan koodissa. Näin kyseinen App käyttää vain haluttu Firebase:n kantaa (kuva 1). (Facebook. n.d.)



The image shows the Facebook App configuration page. It contains the following fields and sections:

- App ID:** 664438837466002
- App Secret:** A masked field with a "Show" button.
- Display Name:** photo-feed
- Namespace:** An empty text field.
- App Domains:** An empty text field.
- Contact Email:** A masked email address ending in ".com".
- Privacy Policy URL:** Privacy policy for Login dialog and App Details
- Terms of Service URL:** Terms of Service for Login dialog and App Details
- App Icon (1024 x 1024):** A placeholder icon with a plus sign and the text "1024 x 1024".
- Category:** A dropdown menu with the option "Choose a Category". Below it is a link: "Find out more information about app categories here".
- Business Use:** A section with the text "This app uses Facebook tools or data to" and two radio button options:
 - Support my own business
 - Provide services to other businesses

KUVA 1 Facebook App:n asetusten sivu

2.3.2 Reaaliaikainen tietokanta

Reaaliaikainen tietokanta on oikeastaan vain yksi iso JSON-objekti (JSON), jota kehittäjät voivat hallita reaaliajassa eri päätelaitteilta. Reaaliaikainen synkronointi helpottaa käyttäjien pääsyä tietoihinsa mistä tahansa laitteesta, olipa se sitten verkko- tai mobiililaitte.

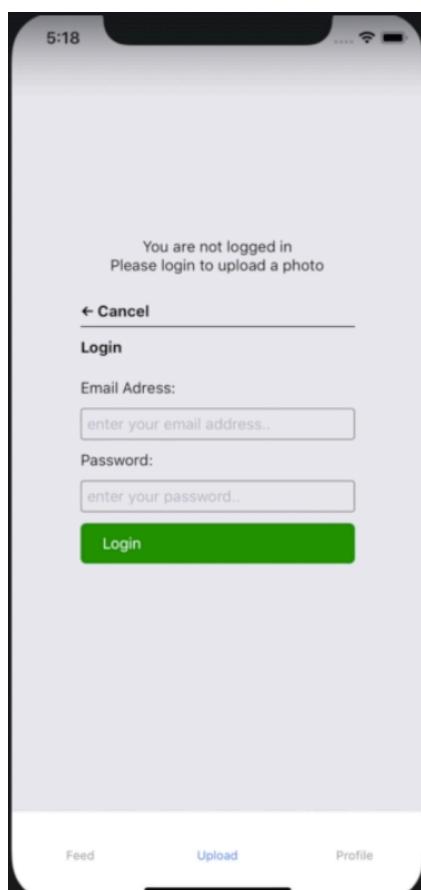
Reaaliaikainen tietokanta myös helpottaa käyttäjien välistä vuorovaikutusta. Toinen reaaliaikaisen tietokannan etu on, että se toimii mobiili- ja web-SDK-tiedostoilla, jolloin on mahdollista rakentaa sovellus ilman erillisiä palvelimia.

3 PHOTO FEED

Photo Feed on yksinkertainen, Instagramin kaltainen mobiilisovellus. Mobiilisovellus on toteutettu React Native:lla. Sovelluksen tietokantana ja käyttäjien tunnistaminen ja rekisteröinti on toteutettu Firebase-palvelun eri ominaisuuksilla.

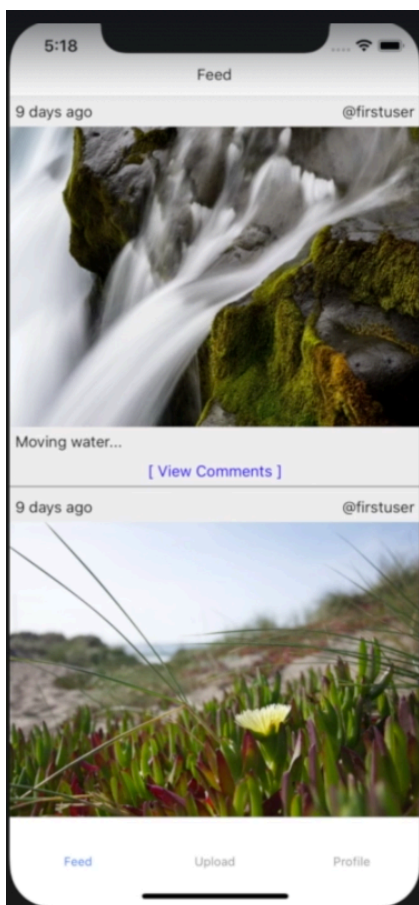
3.1 Ominaisuudet ja arkkitehtuuri

Sovellukseen rekisteröidytään ja kirjaudutaan sähköpostiosoitteella ja salasanalla (kuva 2).



KUVA 2 Sovelluksen näkymä sisäänkirjautuessa

Kirjautumisen jälkeen käyttäjä näkee perusnäkyssä jo sovellukseen lisätyt kuvat (kuva 3).



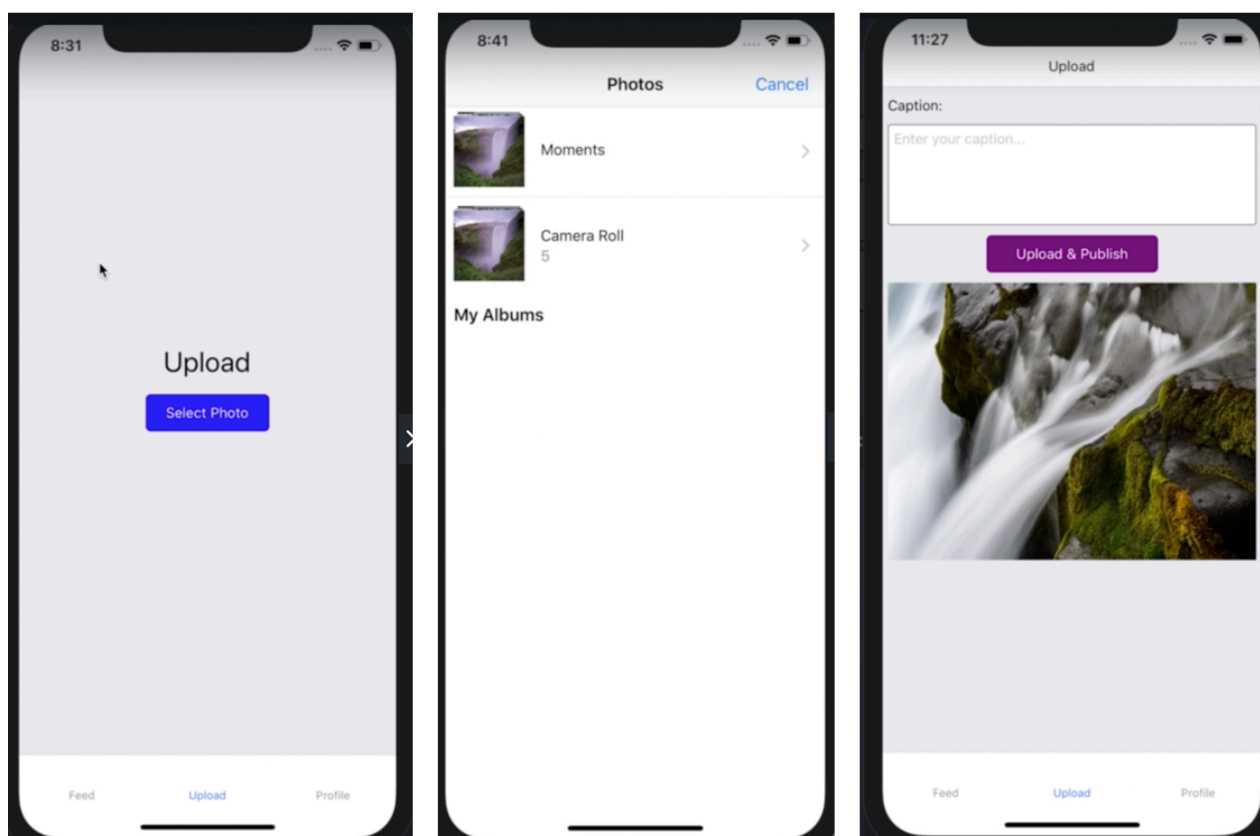
KUVA 3 Sovelluksen perusnäkö

Kuvat ovat järjestyksessä uusimmasta vanhimpaan, ja niitä on mahdollista selata ja kommentoida (kuva 4).



KUVA 4 Sovelluksen näkymä kuvaa kommentoitaessa

Myös kommentit lähetetään sovelluksesta tietokantaan, jolloin muut käyttäjät voivat nähdä kuvien kommentit. Perusnäköymästä on mahdollista siirtyä toiselle välilehdelle, jossa voi lisätä oman kuvan mobiilisovellukseen (kuva 5). Kuva tallentuu Firebase-tietokantaan kirjautuneen käyttäjän tietoihin (kuva 6).

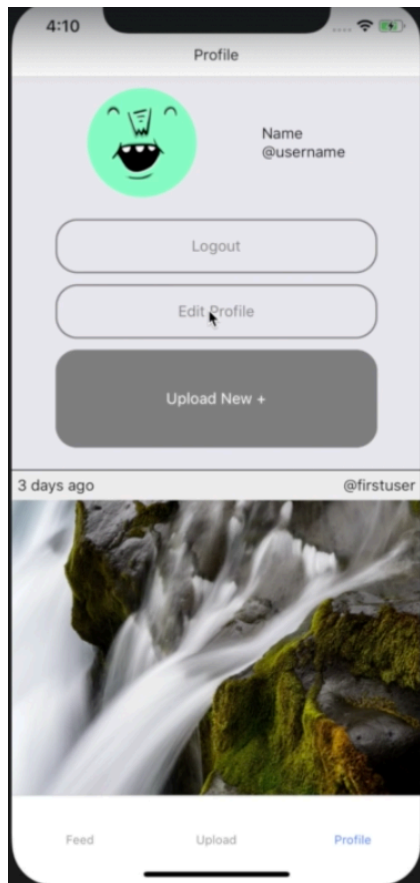


KUVA 5 Sovelluksen näkymät kuvaa lisättäessä

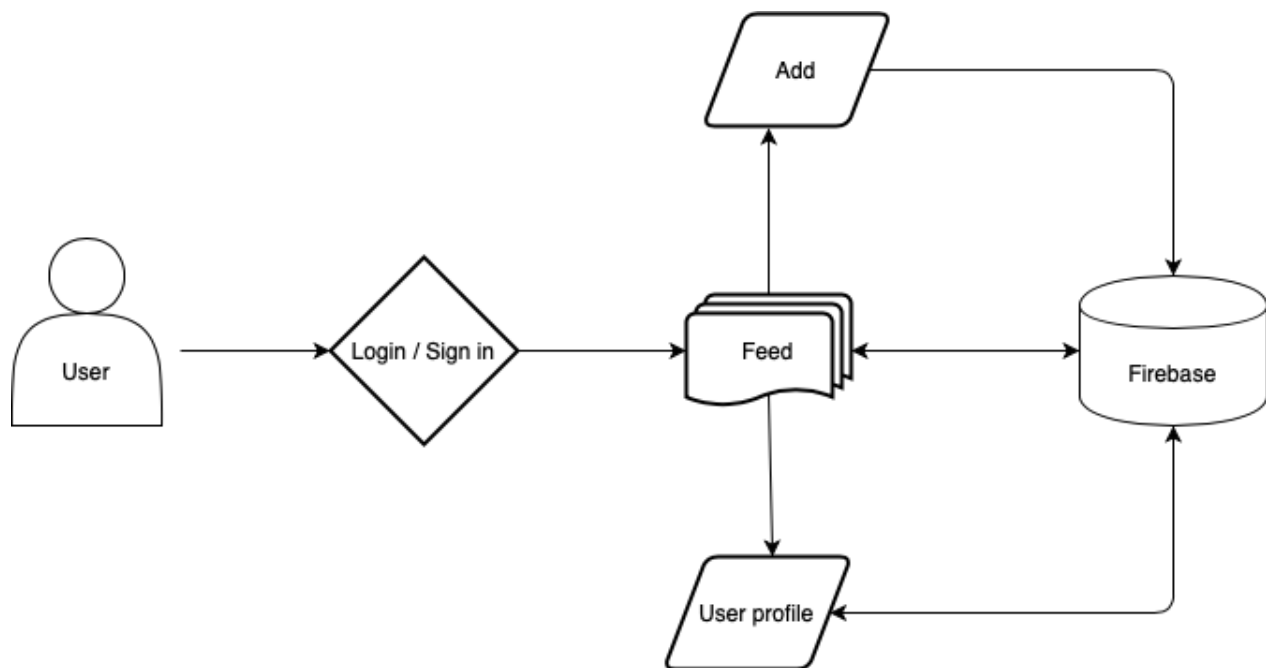


KUVA 6 Tietokannan kansiorakenne

Sovelluksessa on oma välilehtensä myös kirjautuneen käyttäjän profiilille (kuva 7). Profiilissa voi tarkastella omia lisäämiään kuvia sekä editoida omaa profiilinsa, kuten käyttäjänimeään. Myös sovelluksesta ulos kirjaudutaan oman profiilin kautta. Kuviossa 1 on esitetty yksinkertaistettu arkkitehtuurimalli Photo Feed -sovelluksen toiminnasta.



KUVA 7 Sovelluksen näkymä käyttäjän omassa profiilissa



KUVIO 1 Photo Feed -sovelluksen yksinkertaistettu arkkitehtuuri

3.2 Tekninen toteutus

Photo Feed:n frontend on toteutettu React Nativella. Backend:in toteutuksessa on hyödynnetty Firebase-palvelun monipuolisia ominaisuuksia, kuten tietokantaa sekä käyttäjän tunnistusta.

3.2.1 React Native

React Native on mobiilisovellusten framework (Framework), joka mahdollistaa Android- ja iOS-sovellusten kehittämisen natiivien käyttöliittymäelementtien avulla. React Native on Facebookin kehittämä, Java-pohjainen ohjelmointikieli. Sen edeltäjänä oli lähinnä web-ohjelmointiin suunnattu React.

Tämä framework mobiilisovellusten kehittämiseksi alkoi kesällä 2013 Facebookin sisäisenä hackathon-projektina. Sen ensimmäinen julkinen esittely pidettiin tammikuussa 2015 ReactJS-konferenssissa. Maaliskuussa 2015 Facebook teki React Native -ohjelmointipaketista julkisen ja saataville GitHubiin.

Mobiilikehittäjät ja organisaatiot ovat ottaneet sen käyttöön laajasti, koska se kykenee tuottamaan natiivisovelluksia usealle käyttöjärjestelmälle yhdellä koodilla ja tukee monipuolisia käyttöliittymiä. Vain 1,5 vuoden kuluttua julkaisustaan se ohitti käyttäjämäärässä Androidin ja iOS:n omat natiivikielet. (React Native. n.d.)

3.2.2 Firebase osana sovellusta

Keskeisimmät toiminnot sovelluksessa ovat kirjautuminen, jo sovellukseen lisättyjen kuvien selaaminen ja kuvien lisääminen sovellukseen. Jokainen näistä ominaisuuksista hyödyntää oleellisesti Firebase-palvelua.

Sovellukseen rekisteröitymisessä ja kirjautumisessa käytetään sähköpostiosoitetta ja salasanaa. Tämä on yksinkertaisin ratkaisu Firebase-palvelussa kirjautumisen toteuttamiseksi, koska se ei vaadi ulkopuolisten sosiaalisen median palveluiden linkittämistä sovellukseen. Kirjautuessa tarkistetaan, että syötetyt tiedot vastaavat rekisteröidyn käyttäjän tietoja tietokannassa.

Kuvien selaamista varten sovellus hakee tietokannasta jo lisätyt kuvat ja kommentit päätelaitteelle näkyviin. Tietokannan sisältö päivittyy reaaliaikaisesti, joten jos joku lisää tai poistaa kuvan, muutos näkyy muille käyttäjille heti, kun käyttäjä on selannut feedin ylös päivittääkseen sen hetkisen kannan näkyviin. Myös kuvan tai kommentin lisäys päivittyy suoraan tietokantaan.

Firestore-palvelussa voi myös itse määritellä suoraan koodilla luodun sovelluksen kannan turvallisuussäännöt. Esimerkiksi Photo Feed -sovelluksessa on määritetty erikseen, että käyttäjä voi hallinnoida vain itse lisäämiään kuvia (ohjelma 2). Koodipohjaisen määrittelyn etu on, että ne saa määriteltä juuri oman sovelluksen tarpeita vastaavaksi. Tämä antaa huomattavasti tarkemmat mahdollisuudet rajoittaa kannan käyttöä, kuin esimerkiksi valmiina olevat on-off-tyyppiset asetukset.

```

{
  "rules": {
    "photos": {
      //Anyone can read
      ".read": true,
      //Can only write or edit if you are author
      "$photoId": {
        ".write": "root.child('photos').child('author').val()
          == auth.uid || newData.child('author').val()
          == auth.uid"
        ".author": {
          ".validate": "newData.val() == auth.uid"
        }
      }
    },
    "users": {
      "$userId": {
        ".read": "auth.uid == userId",
        ".write": "auth.uid == userId",
        "username": {
          ".read": true
        },
        "name": {
          ".read": true
        },
        "avatar": {
          ".read": true
        },
      },
    }
  }
}

```

OHJELMA 2 Firebase-tietokannan turvallissääntöjä määrittelevä koodi

4 POHDINTA

Työssä esitettiin vain yksi esimerkki, miten Firebase-palvelua voi hyödyntää yksinkertaisessa mobiilisovelluksessa. Isommassa ja monipuolisemmassa sovelluksessa, jolla olisi huomattava käyttäjämäärä, Firebase:n monipuoliset mahdollisuudet tulisivat huomattavasti paremmin esille.

Firestore on hyvä ja joustava vaihtoehto perinteisen tietokannan tilalle. Mobiilikehityksen kannalta Firestore-tietokannan keskeisimpiä etuja ovat monipuoliset mahdollisuudet yhden palvelun alla sekä usean eri alustan samanaikainen tukeminen. Tietokanta on helppo luoda ja helposti käytettävissä, eikä erillistä serveriä tarvita. Tämä tekee tietokannan käytöstä sujuvaa ja kevyttä mobiililaitteelle.

Erilaiset lisäominaisuudet palveluntarjoajalta tuovat lisäarvoa kehittäjän kannalta. Yhdestä paikasta löytyvät palvelut helpottavat ja nopeuttavat mobiilisovelluksen hallintaa ja kehitystä. Eri palvelut myös toimivat huomattavasti paremmin yhdessä, kun ne ovat samalta palveluntarjoajalta.

Google ja Facebook ovat molemmat isoja ja edelleen kasvavia yrityksiä. Molemmat yritykset omistavat useita eniten käytettyjä palveluita niin web- kuin myös mobiilisovellusten saralla, joten React Native ja Firestore ovat erittäin ajankohtainen yhdistelmä mobiilikehityksen kannalta.

LÄHTEET

Expo. n.d. Expo-dokumentaatio. Luettu 15.3.2020. <https://docs.expo.io/guides/using-firebase/>

Facebook. n.d. Facebook for developers -palvelun kotisivut. Luettu 20.3.2020. <https://developers.facebook.com>

Google. n.d. Firebase-palvelun kotisivut. Luettu 28.3.2020. https://firebase.google.com/?gclid=CjwKCAjwwYP2BRBGEiwAkoBpAu0qY-ujt5DDErha5_cHHsPom-qUbBGvIoLVWBKlvyHytGkbdh_QIRGRoCpOQQAvD_BwE

React Native. n.d. React Native -dokumentaatio. Luettu 15.3.2020. <https://react-native.dev/docs/0.60/getting-started>