



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

OPPIMISPÄIVÄKIRJA  
TAMPEREEN TEKNILLISEN YLIOPISTON OPINNÄYTEPOHJA

PLA-32820 2018 Mobiiliohjelmointi

**Joel Alanko, 246479**

25. maaliskuuta 2018

## SISÄLLYSLUETTELO

1.	JOHDANTO .....	1
2.	HARJOITUSTEHTÄVÄT.....	2
2.1	Harjoitus 1 .....	2
2.2	Harjoitus 2 – Git ja versionhallinta .....	3
2.3	Harjoitus 3 .....	3
2.4	Harjoitus 4 .....	4
2.5	Harjoitus 5-6.....	5
2.6	Harjoitus 7-8.....	7
3.	ANDROID MOOC .....	12
4.	13 .....	
5.	YHTEENVETO .....	14
	LÄHTEET .....	15

### LIITE A: MS WORDIN TEKSTITYYLIEN KÄYTTÖ

## LYHENTEET JA MERKINNÄT

LYHENNE	Selitys
---------	---------

# 1. JOHDANTO

Johdanto?

## 2. HARJOITUSTEHTÄVÄT

Tässä luvussa käydään läpi PLA-32820 2018 Mobiiliohjelmointi Moodlesta löytyvät harjoitustehtävät.

### 2.1 Harjoitus 1

Tavoitteena on selvittää jonkin mobiililaitteen ohjelmointiin vaadittavat asiat [1]. Valittu mobiililaitte on käytössäni oleva Huawei Honor 8 Lite [2].

Puhelimen käyttöjärjestelmä on Android 7.0 ja graafinen käyttöliittymä on Huaweiin kehittämä EMUI 5.0 (Eng. Emotion UI) [2-3].

Tuettu ohjelmointikieli, jota myös Android Studio käyttää, on Java [4]. Näkymien muodostamiseen käytössä on XML-merkintäkieli. Android Studiossa voi ohjelmoida myös C++:lla ja C Sharpilla käyttäen Android NDK:ta. Lukuisilla kolmannen osapuolen työkaluilla voi myös luoda Android sovelluksia. Esimerkiksi RubyMotion työkalulla voi sovelluksia luoda käyttäen Ruby-kieltä ja Adoben tarjoamalla työkalulla, PhoneGapillä, voi sovelluksia luoda verkkosovellusten ohjelmointiin tarkoitetuilla kielillä kuten HTML, CSS ja JavaScript [5-6].

Androidille voi ohjelmoida edellä mainituilla työkaluilla, joista tuettu on Android Studio [4]. Android Studio voidaan asentaa Windows 7/8/10 (32- or 64-bit), Mac OS X 10.10-10.13 sekä Linux GNOME/KDE käyttöjärjestelmiin. Itsellä on käytössä 64-bit Windows 7. Android Studiossa tulee mukana tarvittavat IDE sekä Android SDK. Muita vaatimuksia ei ole.

Kännykällä voi ainakin käyttää kahta kameraa videoiden ja kuvien ottamiseen, toistaa ääntä ja videota, ja siinä on GPS, Wifi, kompassi, Bluetooth, liiketunnistimia, sormenjälkitunnistin sekä mikrofooni vain muutama mainitakseni. Androidin dokumentaatiossa löytyy kaikille näille rajapinnat sekä lyhyet ohjelmapätkät demonstroimaan, miten komponentteihin pääsee käsiksi Java:lla [7].

Harjoituksen tekemiseen käytetty aika: 2h, 8.1.2018

## 2.2 Harjoitus 2 – Git ja versionhallinta

Harjoituksen tavoitteena on tutustua Git-Työkaluun ja github palveluun [1].

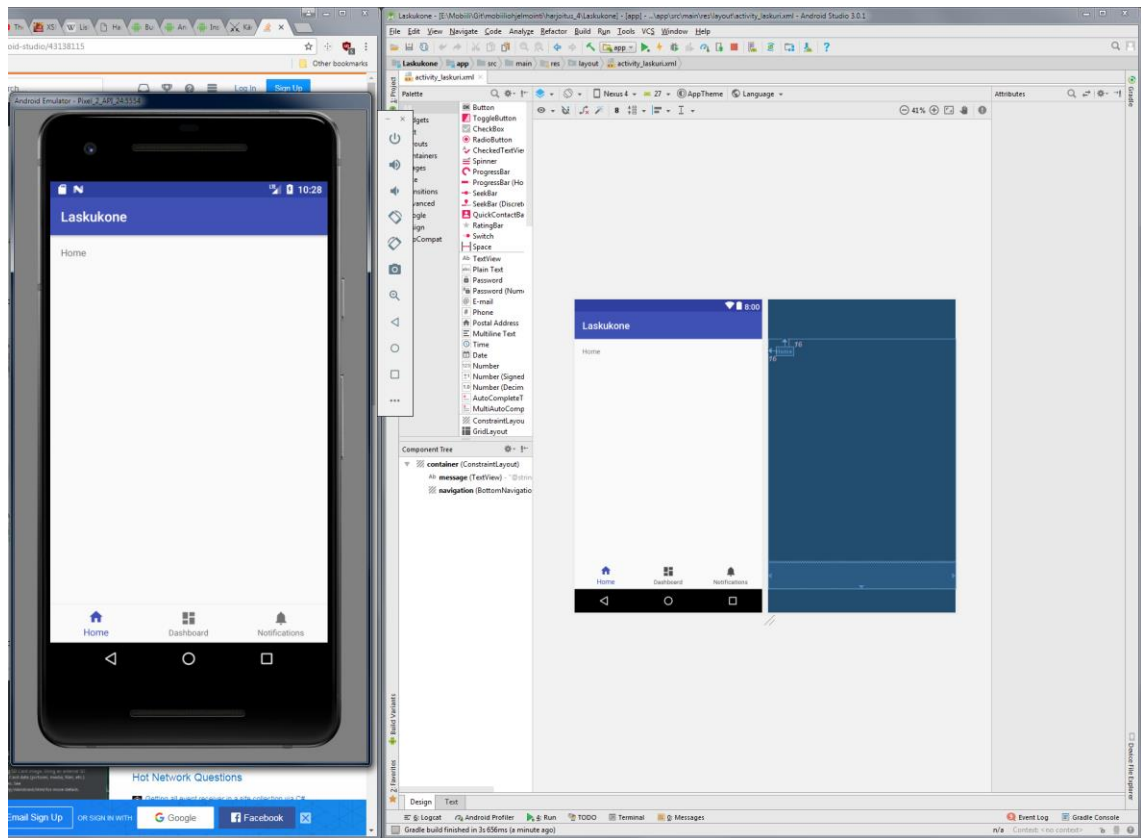
Olen aikaisemmin käyttänyt Git:iä ja githubia, joten siirryin suoraan vaiheeseen 2. Alustin projektin Githubissa ja generoin siihen valmiiksi Androidin .gitignore sekä Readme.md tiedostot. Tämän jälkeen clonasin projektin koneelleni, lisäsin kansion ”harjoitus\_2” ja sinne tein lyhyen ohjelman pythonilla. Tämän jälkeen lisäsin lokaaliin repositorioon muutokset, committasin ja työnsin Githubiin.

Github osoitteeni: [https://github.com/WT26/PLA-32820\\_2018](https://github.com/WT26/PLA-32820_2018)

Harjoituksen tekemiseen käytetty aika: 30min, 8.1.2018

## 2.3 Harjoitus 3

Tehtävänä on asentaa Android Studio tietokoneelle. Minulla on jo valmiiksi asennettu Android Studio versio 1.3.2. Tyhjän applikaation kääntö ei onnistunut, joten päivitin Android SDK:t. Ei vielä kukaan kääntynyt, joten asensin uusimman Android Studion 3.0. Lisäongelmia ja omituisia virheitä ilmaantui levytilan vähyyden vuoksi. Vieläkin valitti jonkin pakkauksen puuttumista, joten pienen googlailun jälkeen sain oikeat asiat asennettua ja applikaation käännettyä ja emulaattori toimi.



Harjoituksen tekemiseen käytetty aika: 2h, 8.1.2018

## 2.4 Harjoitus 4

Tehtävän tarkoituksena on toteuttaa laskukone, ja tallentaa laskutoimituksia logiin. Aikaa kului runsaasti nappula ja tekstielementtien järkevään muotoiluun ja hierarkiaan ja lopulta päädyin kovakoodaamaan joidenkin pituudet kohdilleen, koska visuaalinen aktiviteetti editori ei suostunut muokkaamaan tiettyjen elementtien kokoja. Kaikki UI elementit nimesin alusta asti fiksusti, jotta myöhemmin olisi helppo päästä käsiksi. Tämän jälkeen perehdyin logiikan koodaamiseen. Alustin ja hain tarvittavat nappulat ja tekstikentät myös java koodin puolella. Tämän jälkeen tein onClick eventit nappuloille, jonka jälkeen lisäsin metodit laskutoimitusten sekä tekstikenttien päivittämistä varten. Laskutoimituksia en rajoittanut vielä sen enempää. Muutamia virheitä tuli laskutoimitusfunktioissa varsinkin tekstikentän parsimisen ja näyttämisen yhteydessä. Dokumentaatiota tutkimalla löytyi vihjeitä Stringien ja Integerien välillä parsimiseen [7].

Tämän jälkeen muodostin listan kaikista TextView-elementeistä ja loin niiden arvon tyhjentävän metodin. Kun sekin toimi halutulla tavalla, siirryin tutkimaan, miten aktiviteettien välillä siirtyminen tapahtuu. Tähän apua sain Moodlen kalvoissa mainitusta tutoriaalista [8]. Aktiviteettien vaihtelun toimiessa loin listan logia varten, ja paria StackOverflow kyselyä hyväksi käyttäen tein SharedPreferences lataajan ja tallentajan [9-10]. Varsinkin ”Androiduser” käyttäjän vastaus osoittautui hyödylliseksi [10]. Tämän jälkeen logi toimi toivotulla tavalla.

Lisäksi päätin vielä toteuttaa laskimen kaatumisia estäviä toimia, siistiä koodia sekä kommentoida epäselviä kohtia. Laskimen kaatumiset sain kätevästi kuntoon lisäämällä operaatioiden ympärille try-catch -lausekkeen. Errorin noustessa vastaukseksi tulee vain arvo NaN.



Harjoituksen tekemiseen käytetty aika: 5h, 10.1.2018

## 2.5 Harjoitus 5-6

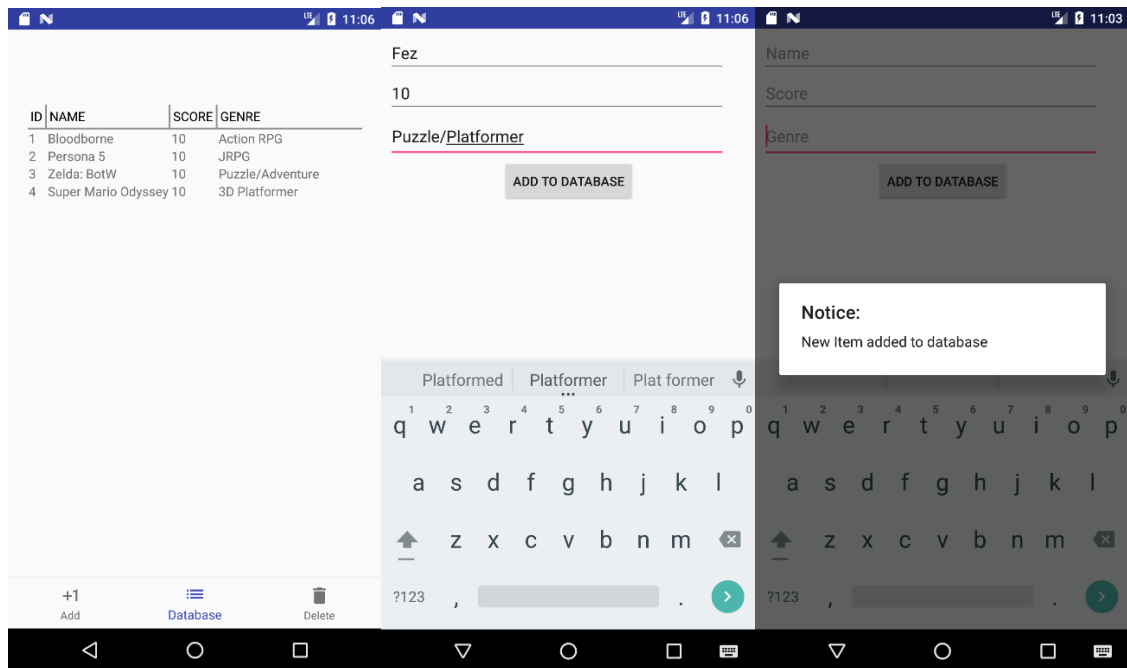
Alkutekijöiksi uppouduin taas UI:n tekemiseen. Tehtävänä oli luoda kolme eri näkymää, joten päätin ottaa selvää, miten NavigationBar on Androidissa toteutettu. Sain sen toimimaan kolmella aktiviteetilla (Activity), mutta vaihtelu ruutujen välillä pätki. Googlailun jälkeen selvisi, että järkevämpää on luoda sama toiminto käyttäen Fragments-widgettiä [11]. Fragmenttien toteutuksessa hankaluutta tuotti varsinkin fragmenttien välillä vaihtelu sekä fragmenttinäkymän Widgetteihin käsiksi pääsy. Lopulta kuitenkin sain toimimaan alustavasti Add- ja Database-näkymät.

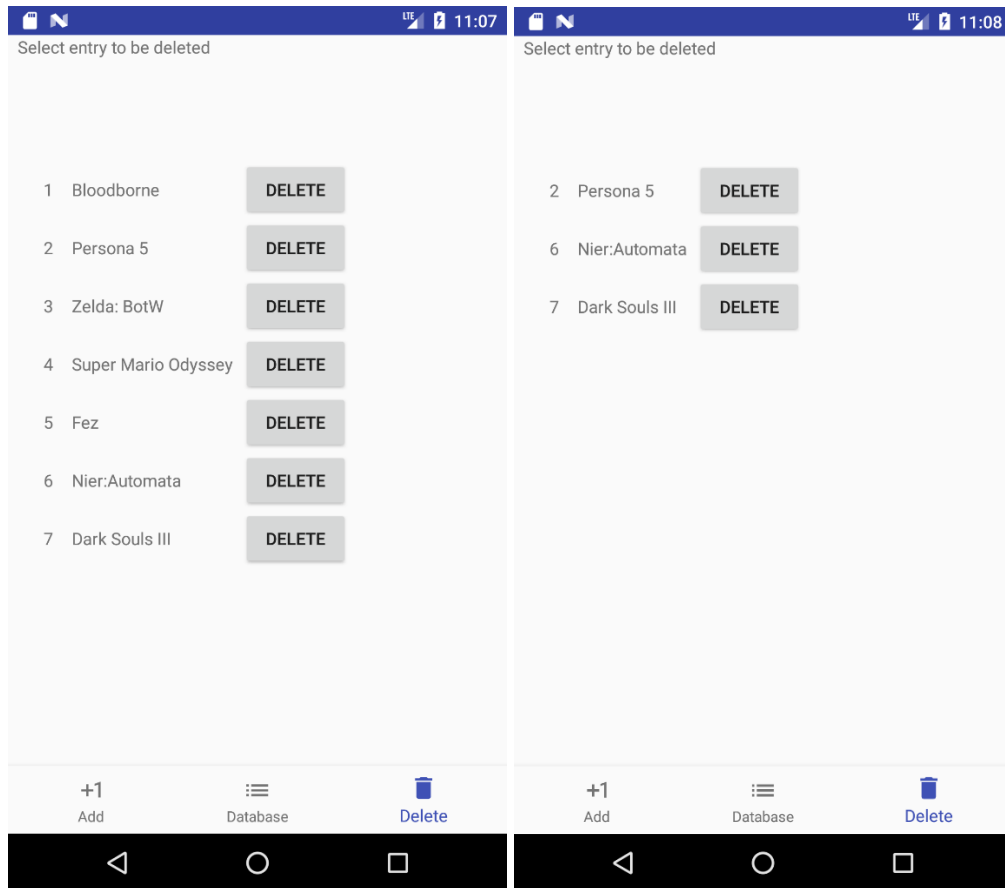
SQLite:n tulostamisen Database-näkymässä sain toimimaan kovakoodauksella nopeasti, mutta näkymän View-komponenttien dynaaminen luominen ja näyttäminen vei oman aikansa [12]. Lisäksi turhia ongelmia tuotti, kun SQL Queryä en päivittänyt SELECT ehtoihin kuin kaksi ensimmäistä kolumnia. Hetken debuggauksen jälkeen ongelmakohdat löytyivät ja sain ne toimimaan.



Tämän jälkeen lisäsin Add näkymälle toiminnot View-elementeille. Tässä ei mennyt kauan. Lisäksi lisäsin AlertDialogin, jos yritetään luoda uutta tietoa tietokantaan, mutta jokin tekstikentistä on vielä tyhjä.

Seuraavaksi siirryin muotoilemaan Delete-näkymän ulkomuotoa. Ulkomuodon päätin toteuttaa dynaamisesti koodissa kuten Database-näkymässä. Pystyin uusiokäyttämään Databasen koodia hyvin, ja tietokannasta poistaminen oltiin jo toteutettu Moodlessa ehdotetun SQLiten tutoriaalissa [12]. Nyt halutut toiminnot ovat kunnossa. Tämän jälkeen oli enää siistimistä ja kommentointia.





Harjoituksen tekemiseen käytetty aika: 3h, 11.1.2018

3h, 13.1.2018

3h, 15.1.2018

## 2.6 Harjoitus 7-8

Työn aloitin tutustumalla Firebaseen [13]. Loin sinne uuden projektin nimeltään firebase. Project ID on fire-78. Tämän jälkeen Firebase piti liittää Android Projektiini. Tein uuden projektin, johon kopioin viime projektin sellaisenaan. Sen jälkeen tein debug sertifikaatin autentikointiohjeiden mukaan [14]. Alla komento jolla Sha1 avain luotiin.

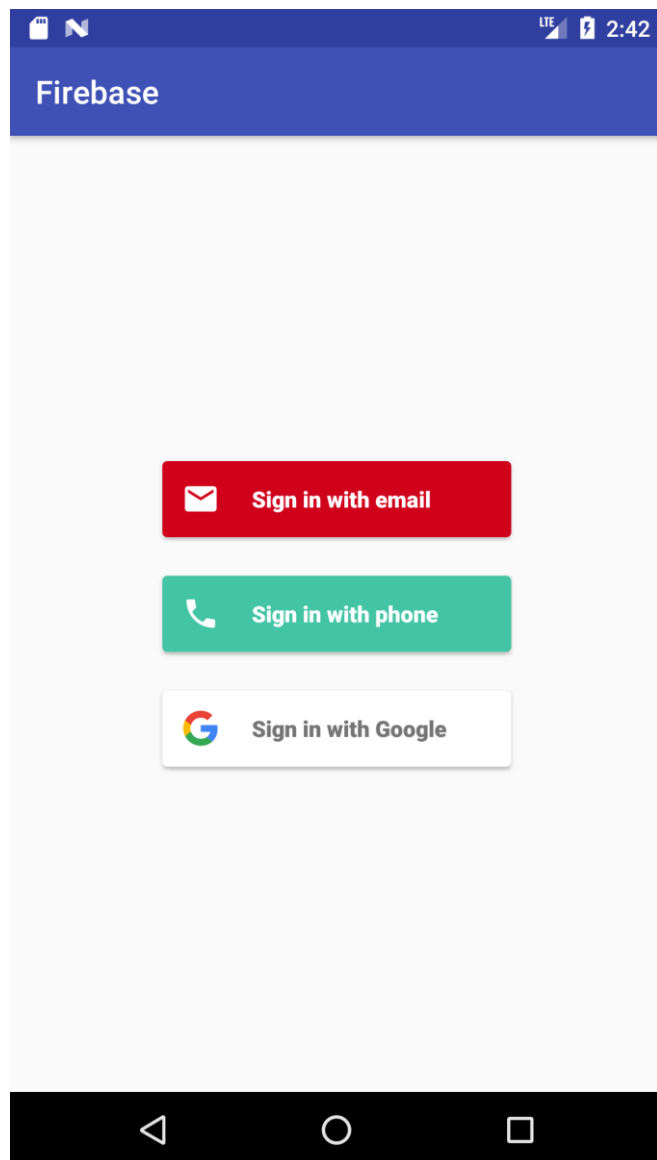
```
keytool -exportcert -list -v -alias androiddebugkey -keystore %USERPROFILE%\android\debug.keystore
```

Myöhemmin release versio voisi autentikoida käyttäen alla olevaa komentoa.

```
keytool -exportcert -list -v -alias <your-key-name> -keystore <path-to-production-keystore>
```

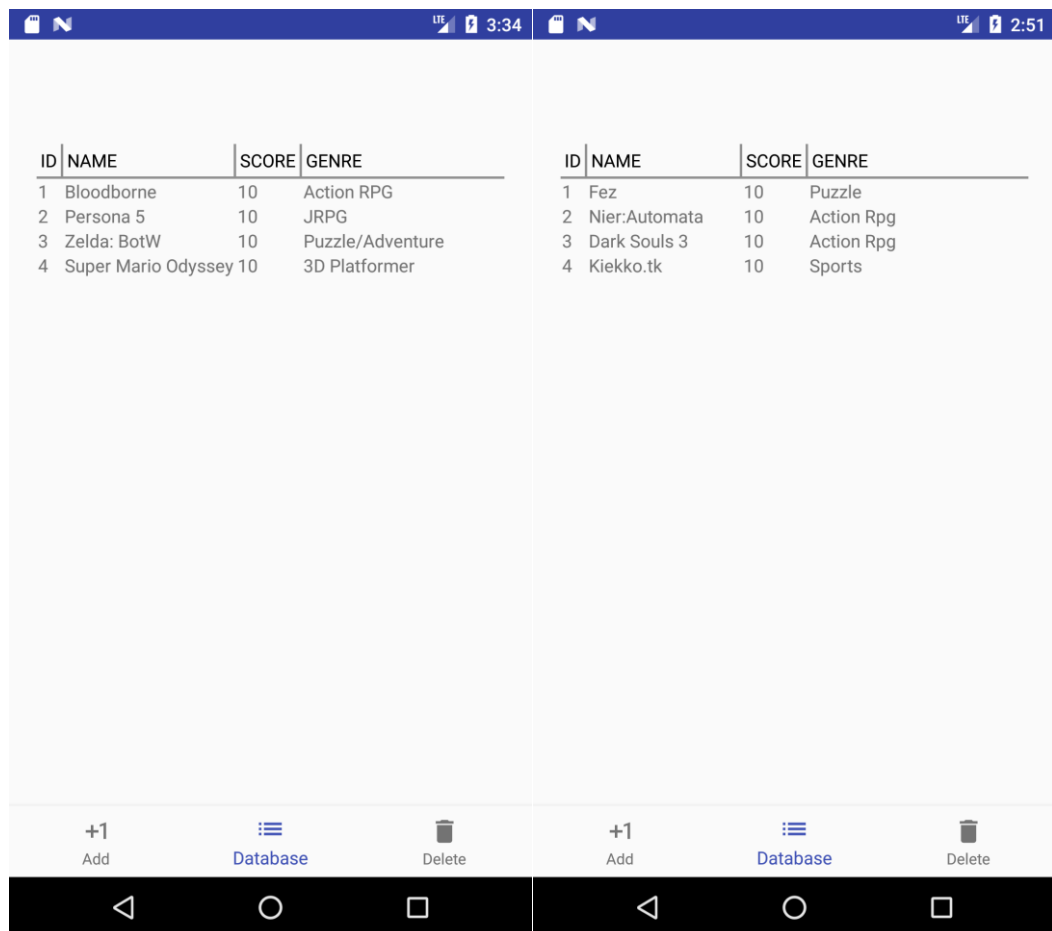
Tämän jälkeen ladattiin google-services.json tiedosto ja liitettiin se projektiin. Synkronoinnin jälkeen Firebase on valmiina sovelluksessa. Tämän jälkeen tutkein Firebasen reaaliaikaista tietokantaa. Estin tietokannan käytön ilman että käyttäjä identifikoit itsensä. Sen jälkeen Android Studioissa siirryin Tools->Firebase ja liitin tunnukseni projektiin. Identifikoinniksi ja autentikoinniksi Firebase suositteli kattavinta FirebaseUI Auth moduulia. Komponentti lisää sovellukselle UI:n autentikointia varten [15].

FirebaseUI:n lisäämisessä seurasin Googlen ohjetta [16]. UI:n liittäminen projektiin aloitin riippuvuuksien lisäämisellä. Tämän jälkeen lisäsin sovellukseeni sign-in autentikoinnin Firebase konsolissa. Autentikointitavoista valitsin sähköpostin, puhelimen sekä Googlen. Ohjeita seuraamalla lisäsin MainActivityyn SignIn Intentin. Autentikointi näyttäisi nyt toimivan. Emulaattorilla tuli jokin varoitus puhelinautentikoinnin kanssa, mutta kun latasin kännykälle ja testasin kaikkia kolmea autentikointia toimivasti.



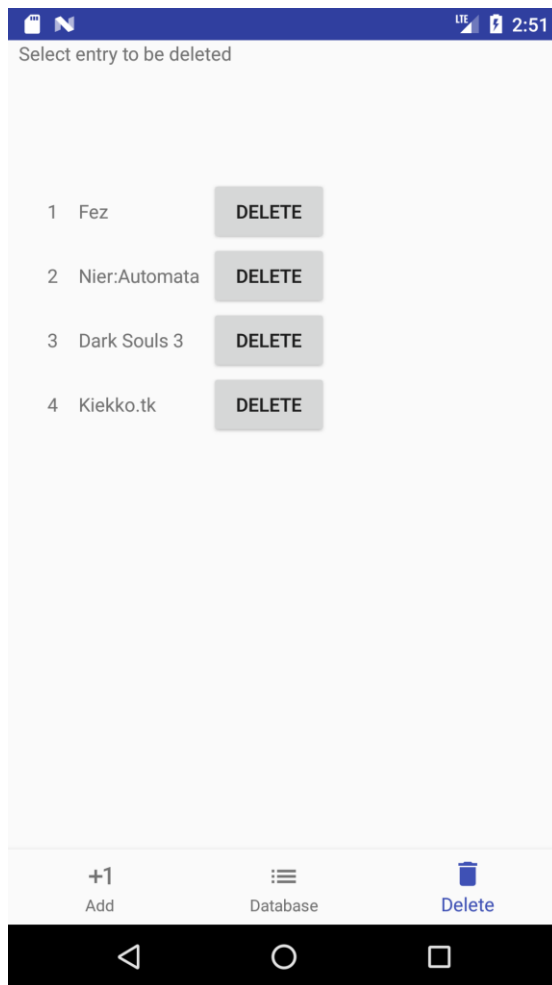
Seuraavaksi perehdyin Firebase Real-time Databasen käyttöön. Suunnittelin Json muodossa rakennettavan tietokannan, ja asetin säännöt luku ja kirjoitusoikeuksille Firebase konsolissa.

Ensin varmistin, että tiedot latautuvat oikein Firebaseesta. Ongelmia muodostui melko paljon Firebaseen ymmärtämisessä. DataSnapshotin avaaminen omassa luokassaan varsinkin oli hieman ongelmallista. Lopulta kuitenkin sain lukemisen toimimaan. Tietojen lisäys Firebaseee ominaisuus onnistui melko nopeasti, ainoana pikku ongelmana oli, kun toinen näkymä kuunteli Firebaseea jatkuvasti, ja koitti päivityksen seurauksena päivittää omaan näkymään uusia rivejä tietokannasta. Koska näkymä ei ollut aktiivinen, heitti se `nullPointerExceptionia`. Tein kaksi käyttäjää ja lisäilin molemmille tietokantoihin tiedot. Vain omat tiedot näkyvät käyttäjälle. Myös lisäys ja poisto toimivat vain omille tiedoille. Tämä varmistetaan vertaamalla `UserID`:ta tietokannan autentikointitietoihin.





Poistamisessa oikean tiedon löytämisessä myös hieman ongelmia, mutta lopulta senkin sain toimimaan. Poistamisessa etsitään käyttäjän peli-id:n perusteella (joka saadaan napin painalluksesta) tietokannasta oikea child data ja kutsutaan sille `removeValue()`.



Harjoituksen tekemiseen käytetty aika:

2h, 15.1.2018

2h, 16.1.2018

3h, 20.1.2018

3h, 24.3.2018

1.5h, 25.3.2018

### 3. ANDROID MOOC

#### Project Sunshine

##### Mitä aiheita käsiteltiin?

Androidin opettelu askeleet, ensimmäisen projektin luominen, emulaattorin käynnistäminen. Lyhyesti Android API versiot, Gradle builds, Android Debug Bridge, Activities, Android Framework, Layouts, The Android Manifest.

Ensimmäisessä harjoituksessa tutustuttiin Layout xml tiedostoon ja muokattiin sitä hieman.

26.1.2018 1h

Toisessa harjoituksessa tutustuttiin Java koodiin ja etsittiin ja alustettiin Layoutissa olevat Viewit. Tämän jälkeen kerättiin lista leluista ja laitettiin ne näkyviin Layoutin TextViewiin.

31.1.2018 15min

Kolmannessa harjoituksessa lisättiin edellisen tehtävän ympärille ScrollView.

31.1.2018 10min

Seuraavaksi tutustuttiin siihen, miten sovellusta voi luoda erikokoisille näytöille. Responsive Designia voi toteuttaa Androidilla esim. Linear, Constraint tai Grid Layouteilla.

Tämän jälkeen edellisten harjoitusten avulla luotiin Project Sunshineen vastaavanlainen skrollaava näkymä.

**4.**



## **5. YHTEENVETO**

## LÄHTEET

- [1] Harjoitusohjeet, Tampereen teknillisen yliopiston Moodlesivut, verkkosivu Saatavissa (viitattu 8.1.2018): [https://moodle2.tut.fi/pluginfile.php/400478/mod\\_resource/content/3/Harjoitus\\_01.pdf](https://moodle2.tut.fi/pluginfile.php/400478/mod_resource/content/3/Harjoitus_01.pdf)
- [2] Huawei Honor, Puhelimen Huawei Honor 8 Lite tuotesivu, verkkosivu Saatavissa (viitattu 8.1.2018): <http://www.hihonor.com/fi/products/mobile-phones/honor8lite/index.html>
- [3] Android 7.0 Nougat, Google LLC, verkkosivu Saatavissa (viitattu 8.1.2018): <https://developer.android.com/about/versions/nougat/index.html>
- [4] Meet Android Studio, Google LLC, verkkosivu Saatavissa (viitattu 8.1.2018): <https://developer.android.com/studio/intro/index.html>
- [5] Adobe PhoneGap, Adobe Systems Inc., verkkosivu Saatavissa (viitattu 8.1.2018): <https://phonegap.com/products/>
- [6] RubyMotion, Scratchwork Development LLC, verkkosivu Saatavissa (viitattu 8.1.2018): <http://www.rubymotion.com/tour/how-it-works/>
- [7] Android Platform Documentation Reference, Google LLC, verkkosivu Saatavissa (viitattu 8.1.2018): <https://developer.android.com/reference/packages.html>
- [8] Build a simple User Interface, Google LLC, verkkosivu Saatavissa (viitattu 10.1.2018): <https://developer.android.com/training/basics/firstapp/building-ui.html>
- [9] Dynamically add elements to a listview Android, Stack Exchange Inc., verkkosivu Saatavissa (viitattu 10.1.2018): <https://stackoverflow.com/questions/4540754/dynamically-add-elements-to-a-listview-android>
- [10] Save ArrayList to SharedPreferences, Stack Exchange Inc., verkkosivu Saatavissa (viitattu 10.1.2018): <https://stackoverflow.com/questions/7057845/save-arraylist-to-sharedpreferences>
- [11] Building a Flexible UI, Google LLC, verkkosivu Saatavissa (viitattu 11.1.2018): <https://developer.android.com/training/basics/fragments/fragment-ui.html>
- [12] Tutorial: Using SQLite, vogella GmbH, verkkosivu Saatavissa (viitattu 11.1.2018): <http://www.vogella.com/tutorials/AndroidSQLite/article.html#tutorial-using-sqlite>

- [13] Firebase helps you build better mobile apps and grow your business., Google LLC, verkkosivu Saatavissa (viitattu 15.1.2018): <https://firebase.google.com/>
- [14] Authenticating Your Client, Google LLC, verkkosivu Saatavissa (viitattu 15.1.2018): <https://developers.google.com/android/guides/client-auth>
- [15] Firebase Authentication, Google LLC, verkkosivu Saatavissa (viitattu 15.1.2018): <https://firebase.google.com/docs/auth/>
- [16] Easily add sign-in to your Android app With FirebaseUI, Google LLC, verkkosivu Saatavissa (viitattu 15.1.2018): <https://firebase.google.com/docs/auth/android/firebaseui>
- [17]

## **LIITE A: MS WORDIN TEKSTITYYLIEN KÄYTTÖ**