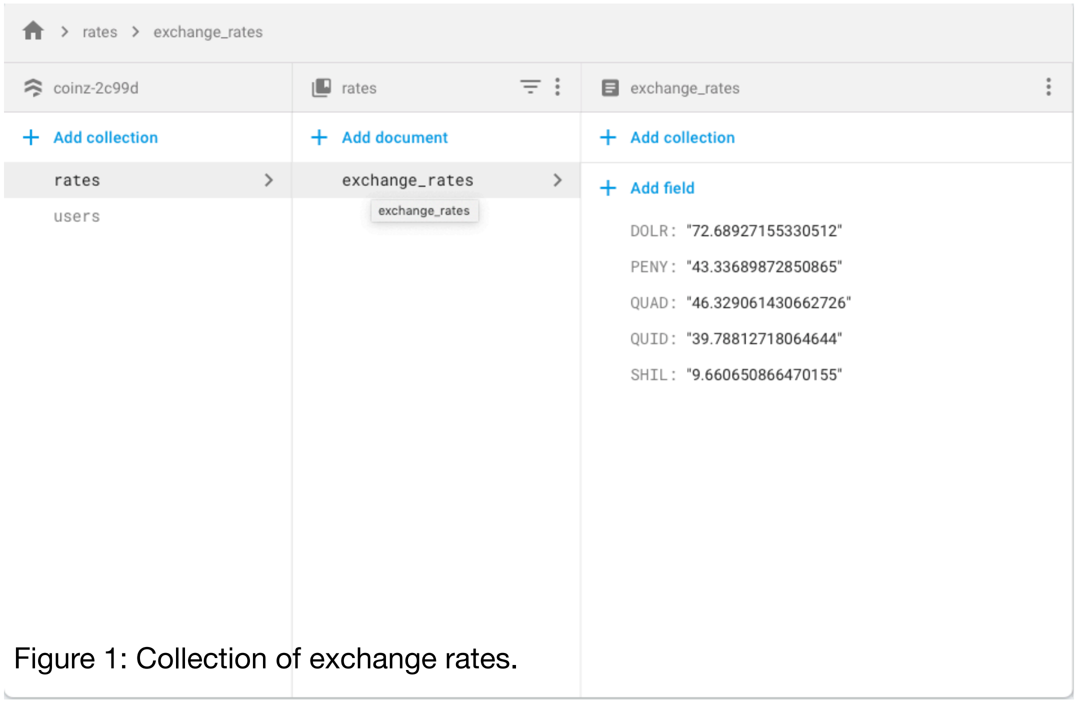
Download and parsing of the Geo-JSON Coinz map from the server

Map needs to be downloaded from the URL the first time user plays the game on every new calendar day. For this reason, the firebase database stores when was the last time the user downloaded the map and compares it with today's date. If the dates are different, Geo-JSON Coinz map needs to be downloaded from the URL. If they are the same, the Coinz map will be rendered from firebase.
The firebase stores the map, in string format and is the same as the map downloaded from the URL, and IDs of coins that have already been picked. When the coins are downloaded and rendered from firebase, it takes this map from firebase and goes through every coin of the map, checking if the coin's ID is not between the coins' IDs that have already been picked. Only coins that have not been picked are then rendered.
The coins' IDs are stored in an ArrayList, the map itself is stored in a string.
Getting today's date runs on the main thread, then it is compared with information retrieved from the firebase. Firebase runs in the background thread but it returns to the main thread with the result, where date comparison is made.
If dates are different, downloading from the URL runs in the background thread using a class defined within MainActivity which extends AsyncTask.
If dates are the same, a chain of methods is run in the main thread, jumping to the background when data from firebase is to be downloaded and then back to the main thread with the retrieved result. The order of execution is important in this case, so that one method executes after another one is finished. This is ensured via onComplete listeners.

Database structure

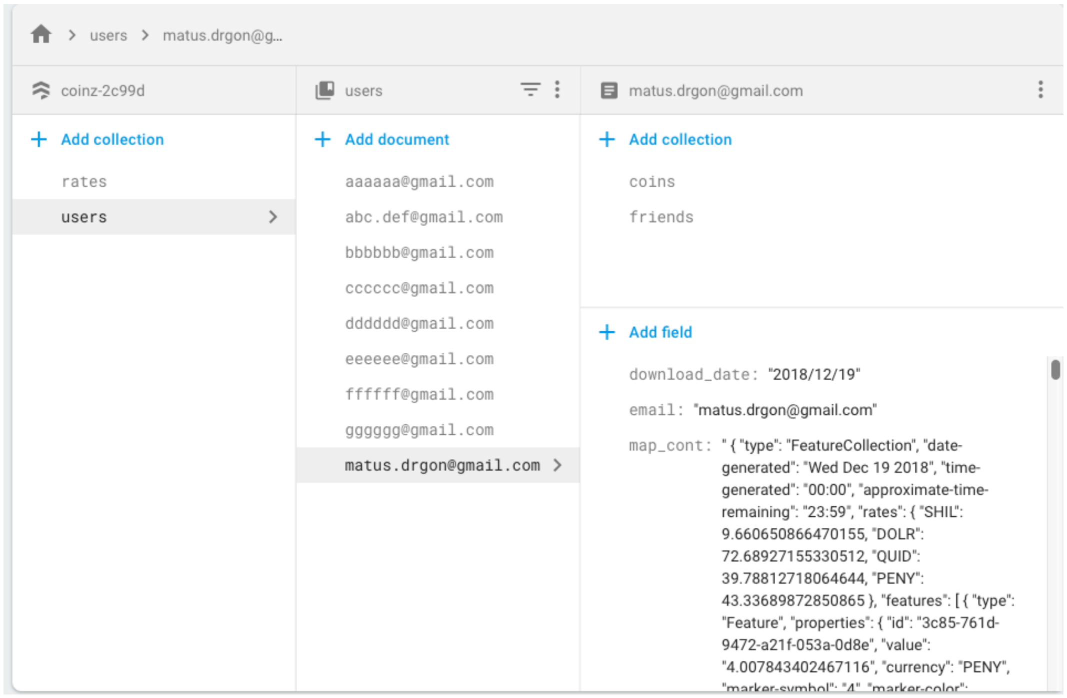
The database has 2 collections: users and rates. Collection rates contains only one document, exchange rates, which contains four fields for the corresponding currencies. It is updated into firebase the same way that the Geo-JSON Coinz map is updated - when user plays the game on a new calendar day for the first time.



DOLR: "72.68927155330512"
PENY: "43.33689872850865"
QUAD: "46.329061430662726"
QUID: "39.78812718064644"
SHIL: "9.660650866470155"

Figure 1: Collection of exchange rates.

Users collection has a unique document for every user and the name of the document is the user's email address. Within the document, there is a field for the date indicating latest download of the Coinz map from the URL, the string storing the map and a field indicating the total amount of gold the user has collected.
It further contains 2 collections - friends, which stores email addresses of friends the user has added, and coins, which stores is divided into banked and picked coins. Banked coins contains IDs of all the coins the user has banked, and picked coins stores all the coins the user has picked. Note that after the user banks a coin, it is removed from picked coins and added to the banked coins.



users > matus.drgon@g...

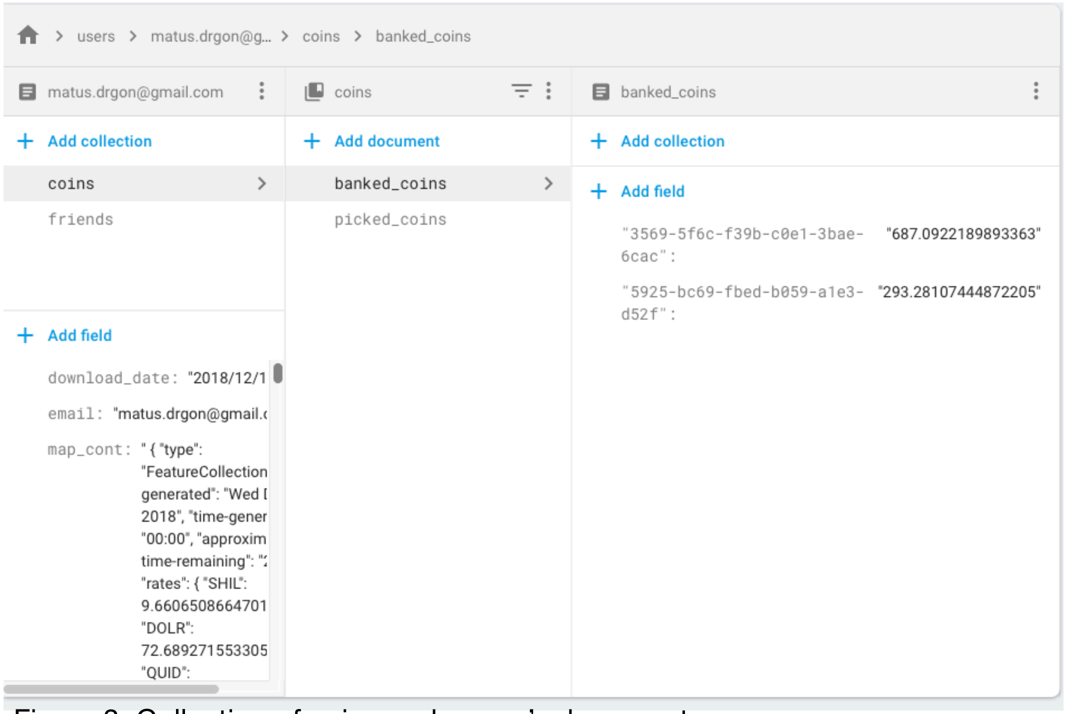
coinz-2c99d
+ Add collection
rates
users

users
+ Add document
aaaaaa@gmail.com
abc.def@gmail.com
bbbbbb@gmail.com
cccccc@gmail.com
dddddd@gmail.com
eeeeee@gmail.com
ffffff@gmail.com
gggggg@gmail.com
matus.drgon@gmail.com

matus.drgon@gmail.com
+ Add collection
coins
friends

+ Add field
download_date: "2018/12/19"
email: "matus.drgon@gmail.com"
map_cont: "{ "type": "FeatureCollection", "date-generated": "Wed Dec 19 2018", "time-generated": "00:00", "approximate-time-remaining": "23:59", "rates": { "SHIL": 9.660650866470155, "DOLR": 72.68927155330512, "QUID": 39.78812718064644, "PENY": 43.33689872850865 }, "features": [{ "type": "Feature", "properties": { "id": "3c85-761d-9472-a21f-053a-0d8e", "value": "4.007843402467116", "currency": "PENY", "marker-symbol": "4", "marker-color":

Figure 2: Collection of users and fields specific to the user



users > matus.drgon@g... > coins > banked_coins

matus.drgon@gmail.com
+ Add collection
coins
friends

+ Add field
download_date: "2018/12/1
email: "matus.drgon@gmail.c
map_cont: "{ "type": "FeatureCollection generated": "Wed [2018", "time-gener "00:00", "approxim time-remaining": "2 "rates": { "SHIL": 9.6606508664701 "DOLR": 72.689271553305 "QUID":

coins
+ Add document
banked_coins
picked_coins

banked_coins
+ Add collection
+ Add field
"3569-5f6c-f39b-c0e1-3bae-6cac": "687.0922189893363"
"5925-bc69-fbed-b059-a1e3-d52f": "293.28107444872205"

Figure 3: Collection of coins under user's document.

Persistent storage of user's information

1. Login

As user starts the game, menu in Figure 4 is displayed. They first have to sign up - this creates their account in the firebase. It creates a new document with their email address under "users" collection and initialises collections "friends" and "coins" within the document. These do not contain any fields at this point.

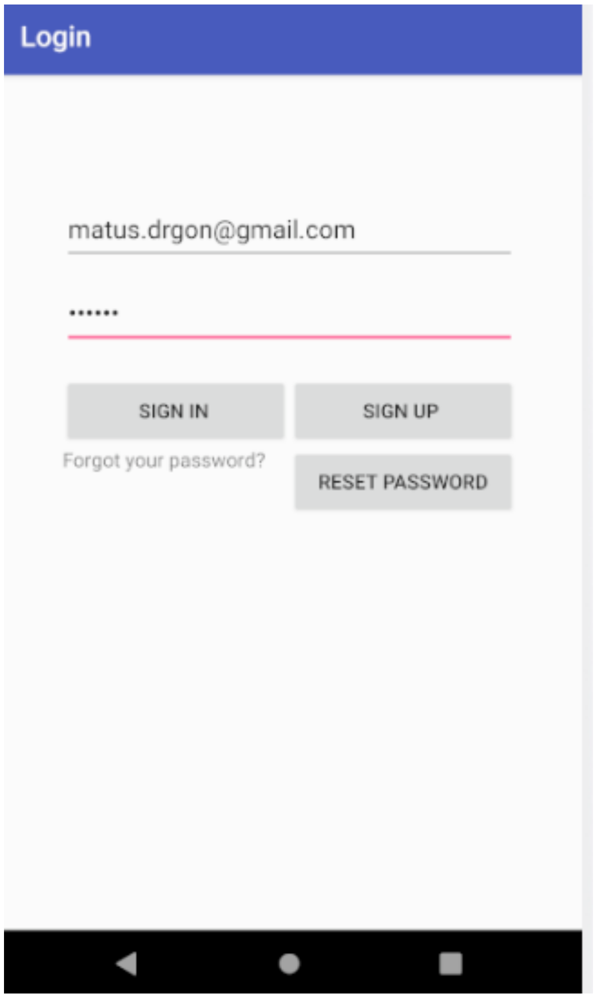


Figure 4: Login menu provided when user starts the game.

2. Main Menu

After the user has created their account, they can sign in. They only need to do this the first time they are launching the game - when they play the game later, they are automatically taken to the menu, skipping the signing process. The user can log out, in which case the signing process will be mandatory again.

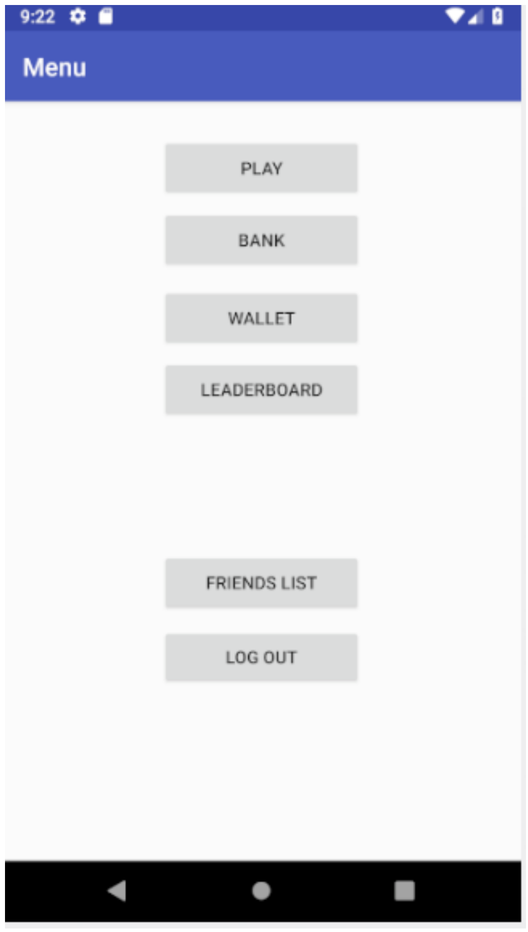
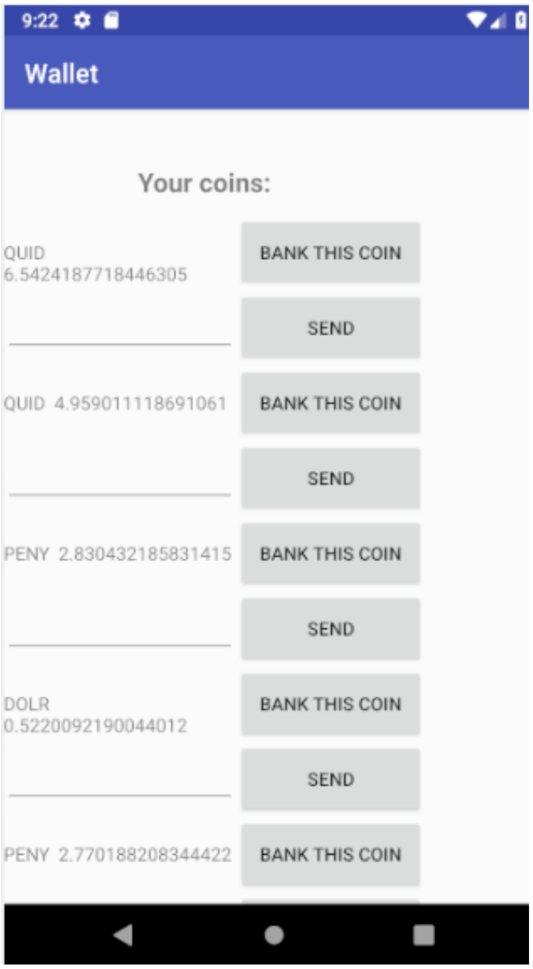


Figure 5: Main menu showing different parts of the game.

3. Banking and sending of the coins

When a user wishes to put some of their coins into bank, they do this in Wallet activity, which they can access from the main menu by clicking Wallet. This provides them with a list of all coins that are stored in their wallet. For every coin, there is a button that can be used to bank the coin, and a text field and another button that can be used to enter someone's email and send them the coin.
By sending a coin, it will be removed from the user's picked coins and added to the picked coins of the receiver with email that was entered. This way, the receiver can bank the same coin twice.



QUID 6.5424187718446305

QUID 4.959011118691061

PENY 2.830432185831415

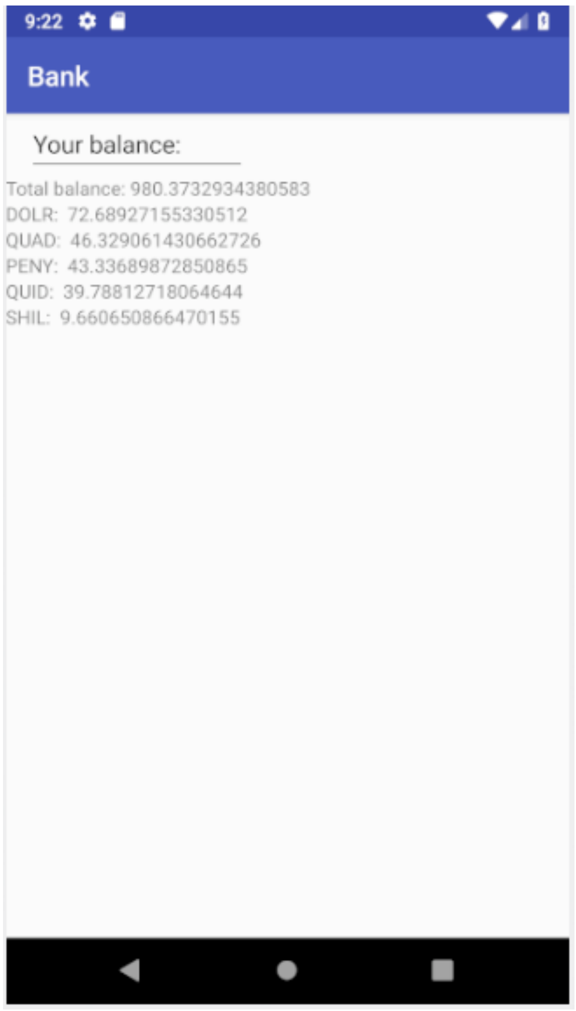
DOLR 0.5220092190044012

PENY 2.770188208344422

Figure 6: View of wallet. Coins can be banked and sent to other people.

4. Bank

This provides information of the current exchange rates and user’s total balance. Total balance is the amount of gold the user has banked since they created their user account. Thus, after banking a coin, it is only regarded in terms of its value in gold and in this way is stored between the banked coins.



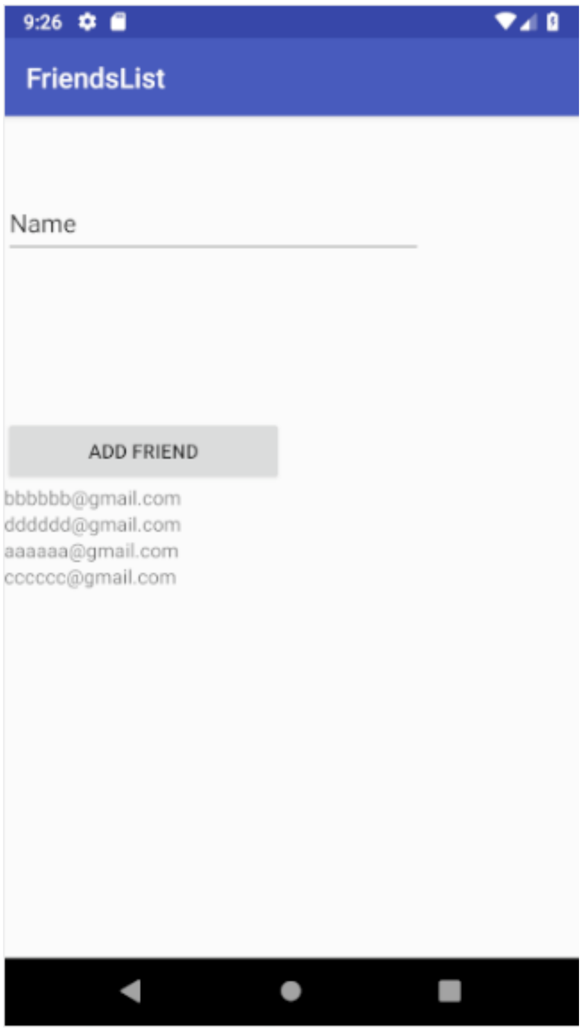
Your balance:

```
Total balance: 980.3732934380583
DOLR:  72.68927155330512
QUAD:  46.329061430662726
PENY:  43.33689872850865
QUID:  39.78812718064644
SHIL:  9.660650866470155
```

Figure 7: View of bank. Exchange rates and total balance are displayed.

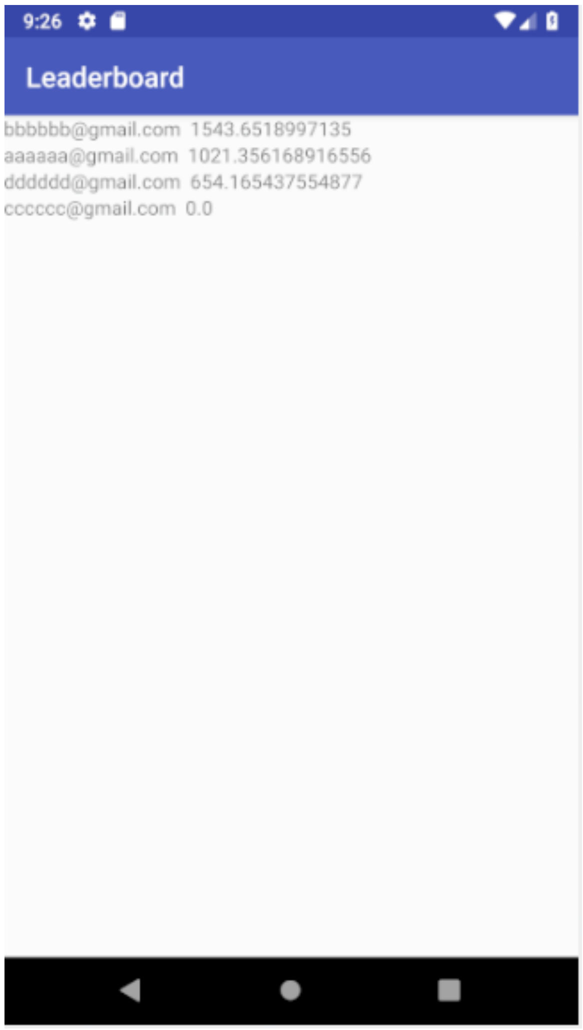
5. Friends list and Leaderboard

Friends list displays a list of friends the user has added. It also provides with a possibility to add a new friend by entering their email address and clicking corresponding button. It is associated with the Leaderboard, which shows list of all friends the user has added with their total balance. This way, the user can see how well they are doing in the game in comparison to their friends, which gives the game a competitive aspect.



bbbbbb@gmail.com
dddddd@gmail.com
aaaaaa@gmail.com
cccccc@gmail.com

Figure 8: Friends list



```
bbbbbb@gmail.com 1543.6518997135
aaaaaa@gmail.com 1021.356168916556
dddddd@gmail.com 654.165437554877
cccccc@gmail.com 0.0
```

Figure 9: Leaderboard shows friends and their total balance sorted in descending order of the balance.

Detection of the coins

There is a method for picking the coins called collectCoins(location) which takes one parameter and it is the location of the user. This is invoked every time the user's location changes. The coins are picked automatically when the user is within 25m radius of a coin's location - the user does not need to press on the coin to pick it. Picking the coins uses feature collection which remembers every coin rendered on the map. The user's location is compared to location of every coin stored in the feature collection. If a coin is within the necessary radius, the coin is added to the pickedCoins (a list of features storing the picked coins) and the firebase is uploaded with ID of this coin that has been picked.

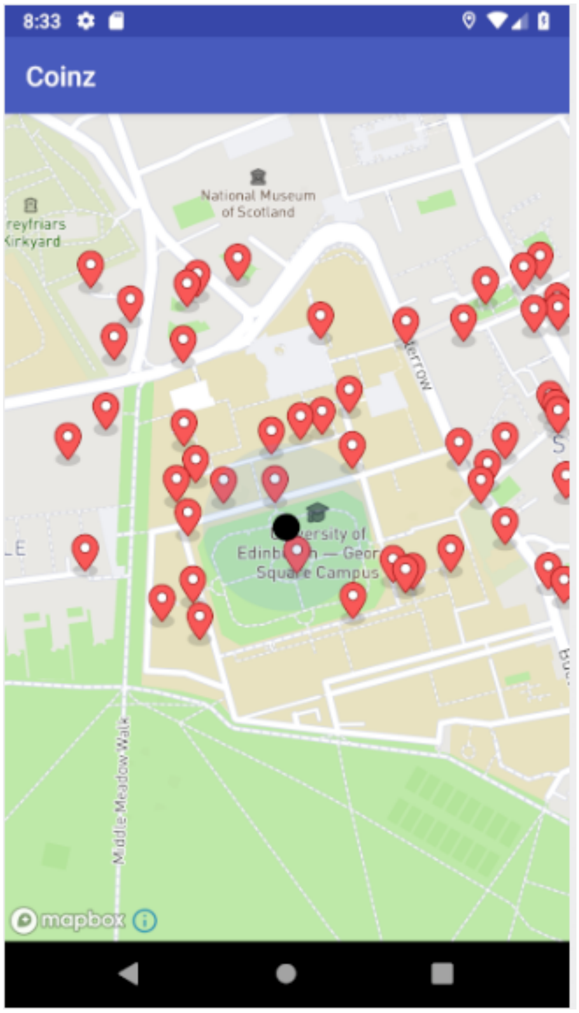

8:33
Coinz
National Museum of Scotland
reyfriars Kirkyard
University of Edinburgh — George Square Campus
Middle Meadow Walk
mapbox

Figure 10: Game has been launched, no coins have been picked yet.

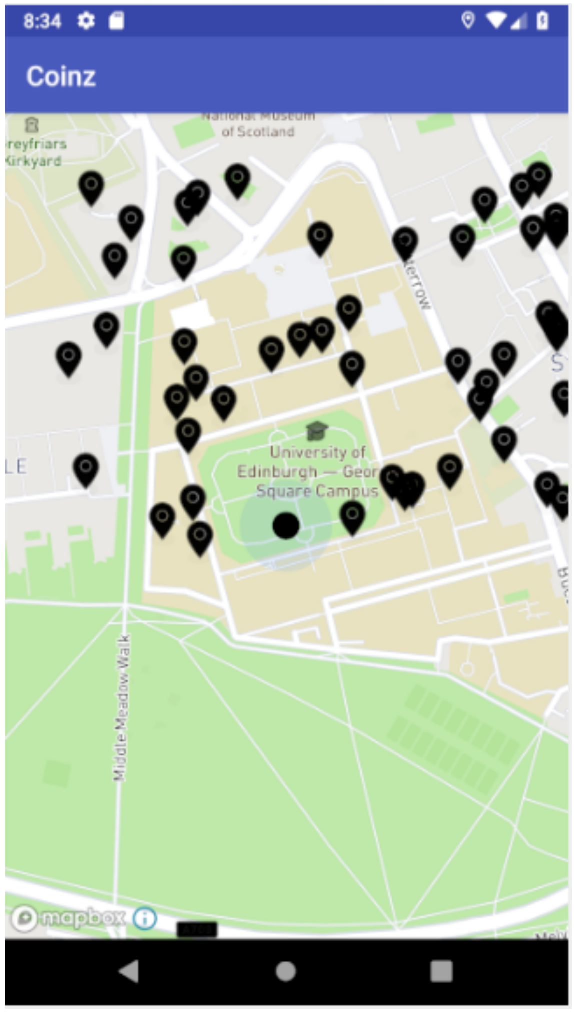

8:34
Coinz
National Museum of Scotland
reyfriars Kirkyard
University of Edinburgh — George Square Campus
Middle Meadow Walk
mapbox

Figure 11: User's location has changes, a few coins have been picked on the way.

Differences from design

As I started with the implementation, due to lack of time I was not able to fully follow my design. The first major difference is the bonus features I did not implement. I was planning to implement 3 bonus features, but ended up with implementing only the last one of them. I consider all bonus features an improvement to the game, mainly because they enhance its competitive nature. Therefore, I implemented the leaderboard and friends list, as I feel like having the possibility to compete with friends makes it more interesting than just solitarily picking the coins.
Another difference is that the wallet can store an unlimited number of coins and for an infinite amount of time, contrary to the plan of wallet's capacity of 100 coins and expiration date of 3 days for each coin.

Acknowledgements

Tutorial: http://tutorials.jenkov.com/java-json/gson-jsonparser.html
I struggled with downloading the exchange rates from the Geo-JSON from the URL. This short tutorial helped me resolve this issue.

Tutorial: https://www.baeldung.com/reading-file-in-java
This tutorial helped me with downloading of the Geo-JSON Coinz map from URL and loading it into a string.

Except these 2 specific acknowledgements, I used many forums like stackoverflow or quora, official tutorials and resources of Android, Firebase and Mapbox.