



SCHOOL OF INFORMATICS, UNIVERSITY OF EDINBURGH

Informatics Large Practical Coursework 2

CHENG LI

s1603732

December 10, 2018

1 Introduction

This report describes the implementation of my Coinz game.

2 Implementation Structure

2.1 Project Structure

2.1.1 Versions

Android Studio: 3.1.4

Gradle: 3.1.4

Kotlin: 1.2.50

Android SDK Tools: 26.1.1

Android Platform Version: API 28 revision 6

Firebase Core: 16.0.4

2.1.2 Emulator

Emulator: Nexus 5X

API: 26

Target: Android 8.0 (Google Play)

Screen size: 5.2

Resolution: 1080 × 1920 (420 dpi)

2.1.3 App Structure

- See figure 1 on page 2 for a general structure. Blues rectangles represent essential activities, while the pink ones represent bonus activities. The yellow rectangle is the navigation drawer, which is not an activity, but plays a significant role in connecting activities.
- More details are given at section 3 on page 7.

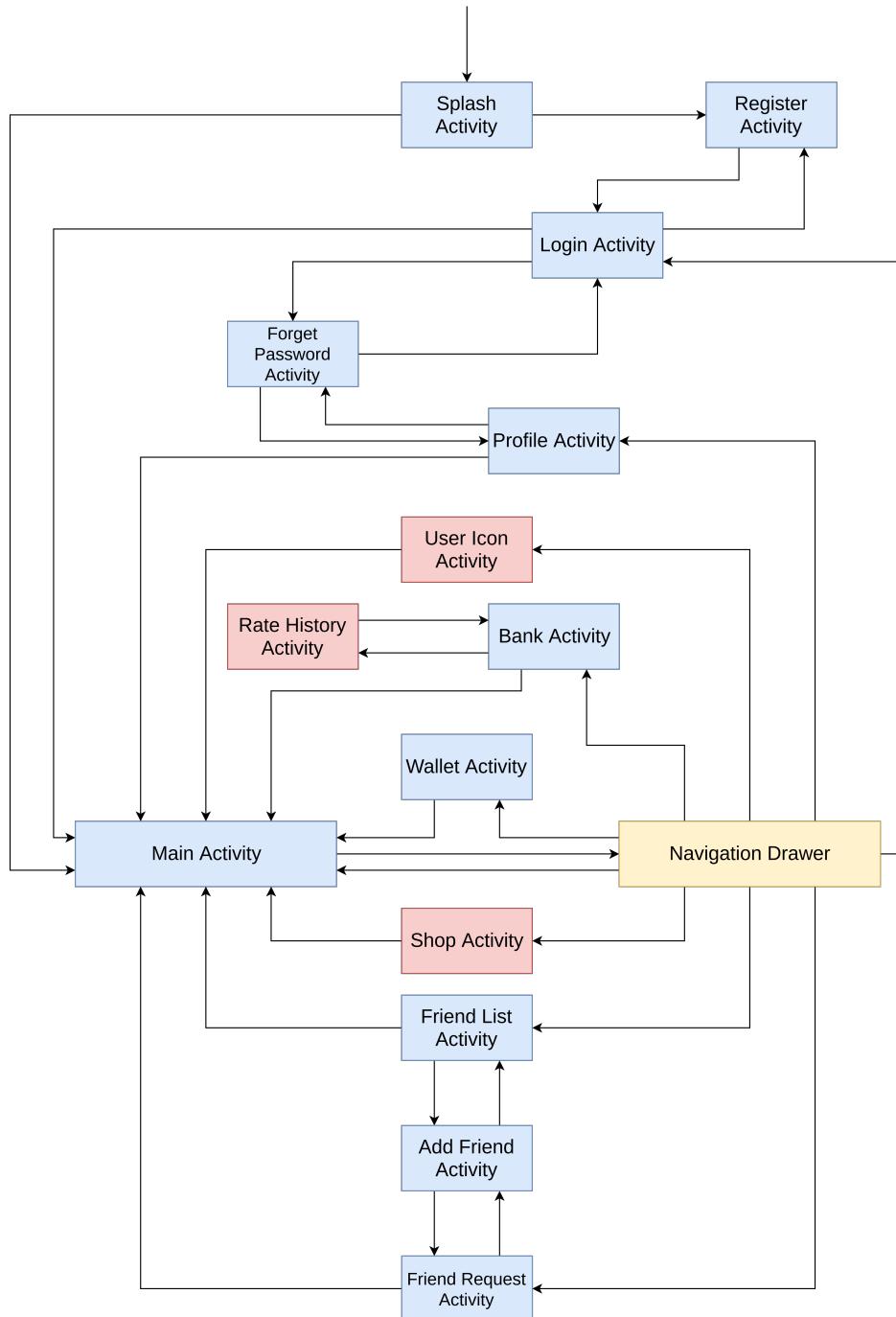


Figure 1: This is the structure of the app.

2.2 Database

2.2.1 Versions

Firebase Firestore: 17.1.2

Firebase Storage: 16.0.4

2.2.2 Structure

See figure 2 on page 4 for a general structure.

- **Firebase Firestore:** See figure 3 on page 5 for an example. Stores all data except user icon images. Yellow folders represent the collections; blue-white rectangles represents documents, where the blue bars show how each document is named and the white parts show what data together with the types are stored in each document.
- **Firebase Storage:** See figure 4 on page 5 for an example. Stores only user icon images. Users would have icon image files named as their email addresses stored in the yellow folder, if they upload their icon images.

2.2.3 Firebase Firestore

bank accounts: Since bank accounts do not involve other players, in order to improve level of security, it is set as the only collection where documents are named by user IDs generated by Firebase Authentication. Each document records user's balances for the four currencies and gold.

exchange rate: Since exchange rate is public, in order to access and retrieve the data more conveniently, it is set as the only collection where documents are named by dates. Each document records the date, and the corresponding rates from the four currencies to gold.

friends: From now on, all documents are named by the user email. Each document records whether there is a new friend request for the user. Also, each document records a friend list and a waiting confirmation list.

shopping carts: Each document records the date, and whether the user has completed the mission today. Also, each document records the current amounts of meat, bread, water and pill obtained.

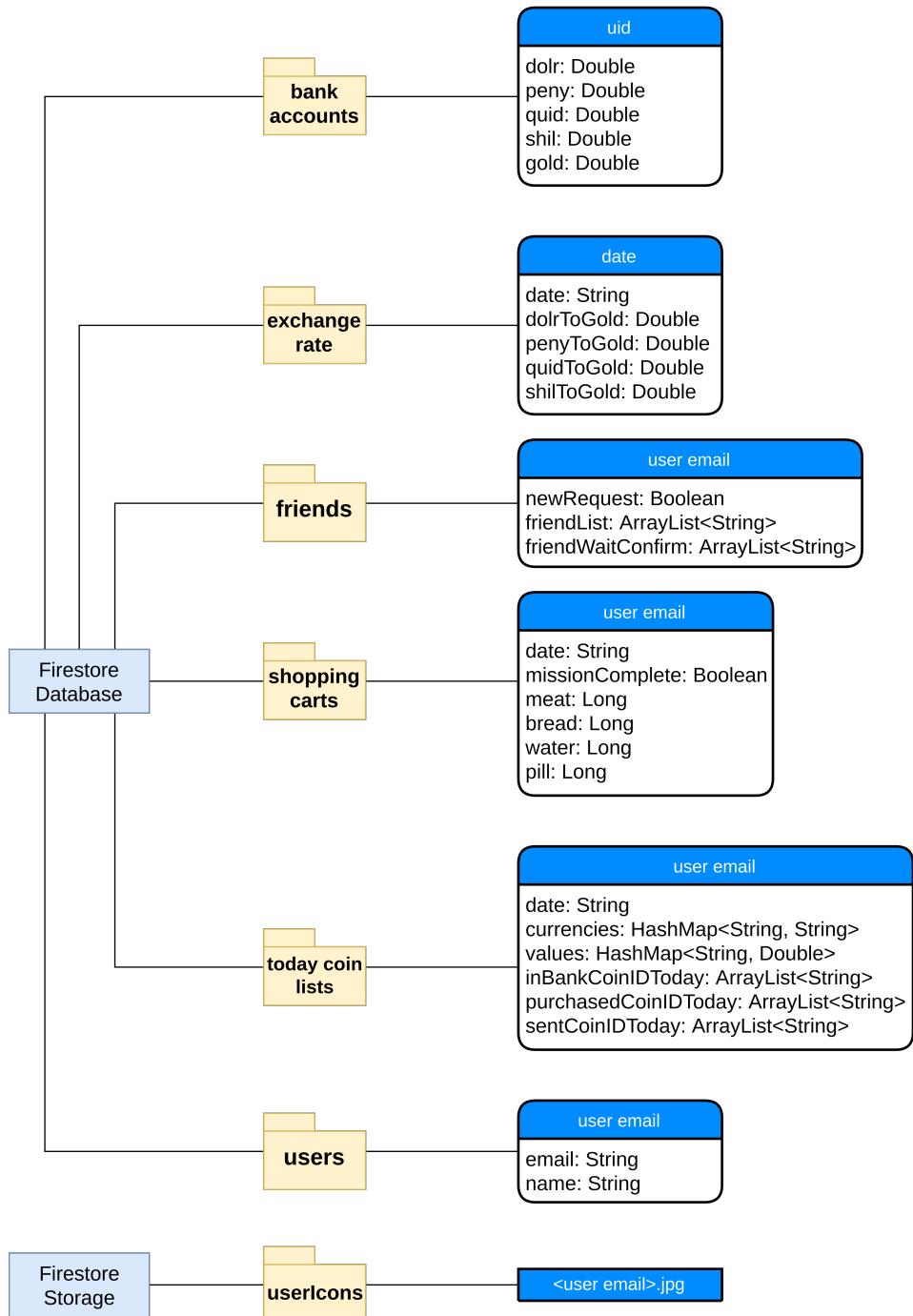


Figure 2: This is the structure of the database.

The screenshot shows the Firebase Database interface for a project named 'Coinz'. On the left, a sidebar lists various services: Authentication, Database (selected), Storage, Hosting, Functions, and ML Kit. Below these are sections for Quality, Analysis, and Development. Under Development, there's a 'Spark' section with a free trial offer. The main area is titled 'Database' and shows a hierarchical view of a 'bank accounts' collection. A specific document, 'YoWs01VsK4WCzB03hqAu5SZl3yh2', is selected, displaying its fields: R3vea2Hyu5TQtBT7VqLsfH09xP2, exchange_rate, friends, shopping_carts, today_coins_list, users, dolr, gold, peny, quid, and shil.

Figure 3: This is an example of a document in the Firebase Firestore.

The screenshot shows the Firebase Storage interface for the same 'Coinz' project. The sidebar includes Authentication, Storage (selected), Hosting, Functions, and ML Kit. The main area shows the 'Storage' tab with a list of files in a folder named 'gs://coinz-zhanhenlc.appspot.com/userIcons'. The files listed are: 2856483627@qq.com.jpg (216.05 KB, image/jpeg, 2018年12月7日), 815753783@qq.com.jpg (92.46 KB, image/jpeg, 2018年12月7日), and borisyuan024@163.com.jpg (480.65 KB, image/jpeg, 2018年12月7日).

Figure 4: This is the userIcons folder in the Firebase Storage.

today coin lists: Each document records the date, a currencies map and a values map, which are used for storing information of the coins user has collected today. Also, each document records lists which record the using of coins: banked in, purchased for food or sent to friends.

users: Each document records the email addresses and the names of users.

Main Functions used: *addSnapshotListener()*, *get()*, *update()*, *set()*

2.2.4 Firebase Storage

userIcons: This folder stores icon images of all users as .jpg file and all images are named by the user email.

Main Functions used: *getFile()*, *putBytes()*

2.3 User Authentication

2.3.1 Version

Firebase Authentication: 16.0.5

2.3.2 Functions

See figure 5 on page 7 for the authentication console. An account is required to use the app. Email/Password is the only sign-in method for this app, and emails need to be verified before signing in. Also, users could receive a reset password email when they forget their passwords.

2.4 Map

2.4.1 Versions

Google Management System Play Location Service: 16.0.0

Mapbox Android SDK: 6.6.1

Mapbox Android Plugin Location Layer: 0.5.2

The screenshot shows the Firebase console for the 'Coinz' project. On the left, there's a sidebar with navigation links for Project Overview, Authentication (selected), Database, Storage, Functions, ML Kit, Crashlytics, Performance, Test Lab, Dashboard, Events, Conversions, Predictions, A/B Testing, Cloud Me..., Spark (free: \$0/月), and Upgrade. The main area is titled 'Authentication' and shows a table of users. The table has columns for ID, 提供商 (Provider), 创建时间 (Created Time), 上次登录日期 (Last Login Date), and 用户的 UID (User's UID). There are three entries:

ID	提供商	创建时间	上次登录日期	用户的 UID
borisyuan024@163.com	✉️	2018年12月7日	2018年12月7日	R3vea2Hyu5TQtBTTVqLsftH09xqP2
2856483627@qq.com	✉️	2018年12月5日	2018年12月7日	YoWs01VsK4WC2B03hqAuSSZ3y...
815753783@qq.com	✉️	2018年12月7日	2018年12月7日	b6UJ5QWfeuczF2k2fnh179YxAhs1

At the bottom of the table, it says '每页行数: 50 < 第1-3位, 共3位 >'.

Figure 5: This is the structure of the database.

2.4.2 Map View

Collecting coins on the map is the fundamental function of this game. A different set of 50 coins is scattered around George Square every day for players to collect. The map and location service are provided by Mapbox and Google, and coins are provided by <http://homepages.inf.ed.ac.uk/stg/coinz>.

3 Activities

3.1 Splash Activity

3.1.1 General Functions

See the first image of figure 6 on page 8 for the Splash Activity. This activity displays a welcome screen for 4 seconds, and download the GeoJson file for the coins in background at the mean time. If there is a valid user logged in, the main activity is started; otherwise, the register activity is started.

3.1.2 GeoJson File

First, the app would check whether the last download date is the same as today's date and whether the file is broken or not (by checking the file

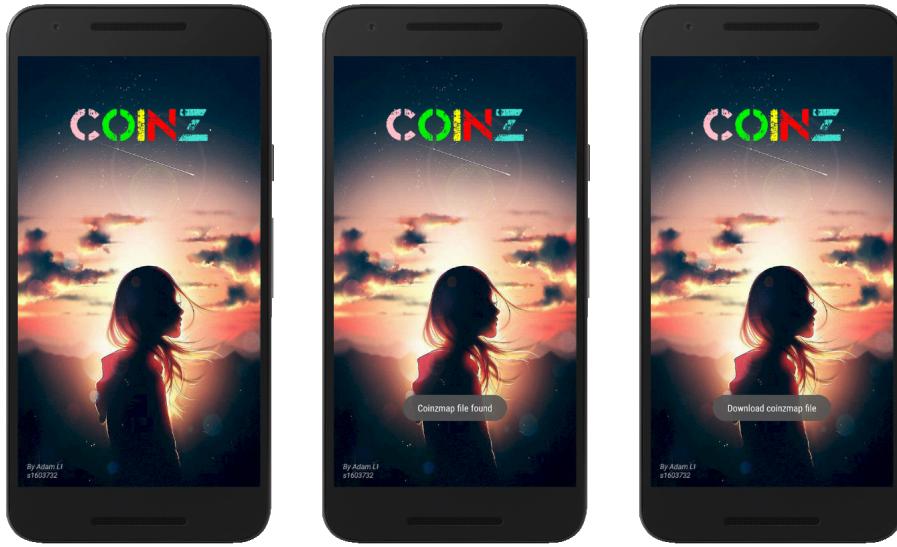


Figure 6: Splash Activity

size). If the last download date is today and the file is not broken, which means there is a valid file, the second image of figure 6 on page 8 would be seen; otherwise, the app would download the newest file and update the last download date, and the third image of figure 6 on page 8 would be seen.

The access and edit of the last download date was done by using `getSharedPreferences()`.

Downloading the coins file was done by implementing a `AsyncTask` interface and a customised interface called `DownloadCompleteListener`. Since we do not need to use the `Progress` type and both of the `Params` and the `Result` types are `String`, i.e. `AsyncTask<String, void, String>`, we only need to override `doInBackground()` and `onPostExecute()` functions. A URL string is passed in and data is parsed from the website as string, and when data is parsed, `downloadComplete()` function overridden before hand is called `DownloadCompleteListener` to store the data in a local file.

The check and update last download and download file methods are identical to the methods taught in the lectures.

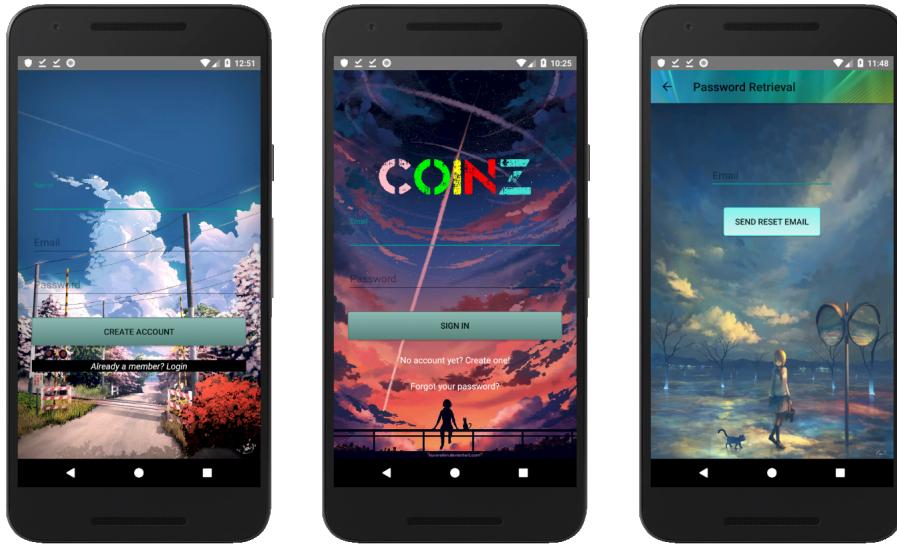


Figure 7: Register Activity, Login Activity and Forget Password Activity

3.2 Register Activity

See the first image of figure 7 on page 9 for Register Activity. You would be able to register an account if you input a valid email address (not registered by others and could receive verification email), a valid name (not empty) and a valid password (with 6 or more characters). After pressing the register button, users would receive a verification email and the app would start the login activity. Also, by clicking the text below, the app would directly start the login activity.

When registering the user using Firebase Authentication, user data would also be created at the same time, which include one document in each of the three folders: users, bank accounts and friends. Documents in other folders would be created later.

3.3 Login Activity

See the second image of figure 7 on page 9 for Login Activity. Users would not sign in unless the email they registered with is verified. Also, register activity and forget password activity (see the third image of figure 7 on page 9) could be reached at this point.

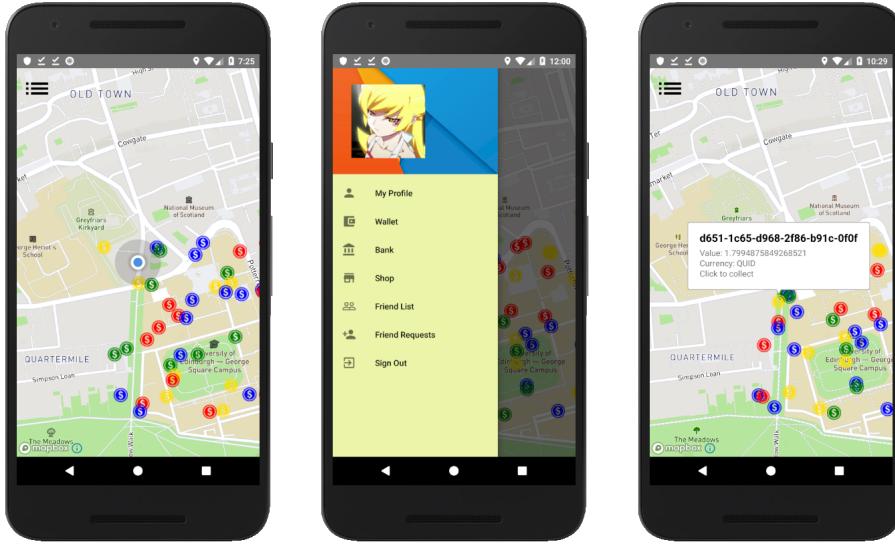


Figure 8: Main Activity

3.4 Main Activity

3.4.1 General Functions

The main activity is just a map with coins as markers on it (see the first image of figure 8 on page 10), together with a navigation drawer (see the second image of figure 8 on page 10)

3.4.2 Map

When the main activity is started, map view is initialised and the app would check whether the permission of location is granted or not and ask for permission if not. Then, the location service is initialised by `initialiseLocationEngine()` and `initialiseLocationLayer()` functions.

For the `onMapReady()` function, the app would parse user's today coin lists data and check the date. If the date is not today or there is no available data in Firestore, a new `CoinToday` class with date set as today would be uploaded to the cloud; otherwise, user's collected coins information would be parsed. Coins in today's GeoJson file would be shown on map in marker format if they have not been collected by the user. Each time a coin marker

is clicked, an info window would pop up to show information of that coin (see the third image of figure 8 on page 10). User needs to click the info window in order to make a collection. If distance between user and coin is less than 25m, the coin would be successfully collected, the database would be updated (*currencies* and *values* HashMaps would both add one instance) with the collection and the collected coin marker would be removed by using the function *remove()*.

3.4.3 Distance by Longitude and Latitude

Haversine Formula:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

where ϕ is latitude, λ is longitude, R is earth's radius (mean radius = 6713 km); all angles need to be in radians

3.4.4 Navigation Drawer

Navigation Drawer could be opened either by clicking the top left image button or by dragging the screen from left edge to right. It is a hub of switches with access to all the other activities. Everything, including the user icon image is clickable.

3.4.5 Daily Renewal

The coins information updates daily, so users who are still playing the game at midnight would be notified to restart the game in order to obtain the latest version.

3.5 User Icon Activity

It is a bonus feature not mentioned in the design, which allows users to customise their own user icons.

See figure 9 on page 12. Images which are not squares could be cropped as square and uploaded to Firebase Storage by the app.

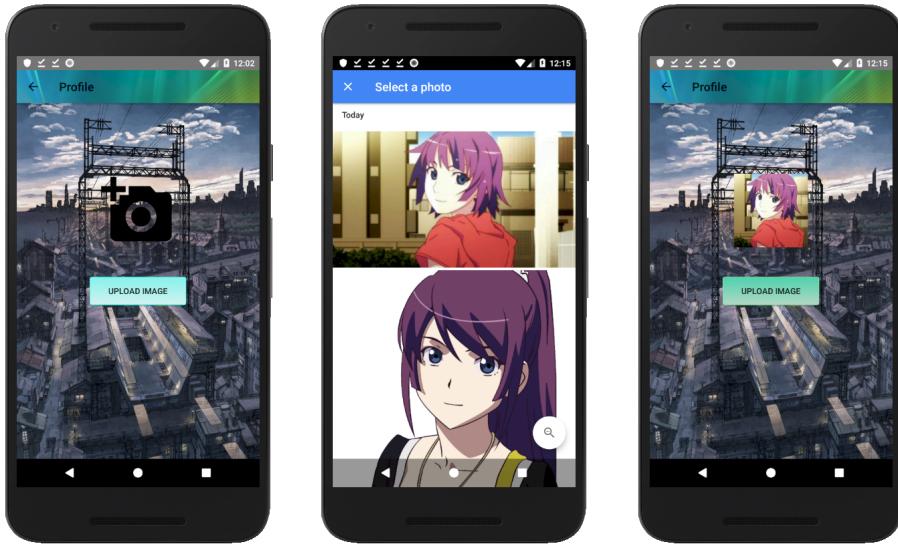


Figure 9: User Icon Activity

3.6 Profile Activity

See the first image of figure 10 on page 13. It is a basic activity that allows users to view their emails and names and to make name and password changes.

3.7 Wallet Activity

Wallet activity could not be started unless today's mission in shop activity is completed. It shows balance of users' bank accounts and coins (see the second image of figure 10 on page 13) that could be banked in (collected and did not used in buying things or sending to friends). Also, if a user banks in a coin, the coin would disappear from the coin list and the corresponding balance would increase (see the third image of figure 10 on page 13). Data are also updated to Firestore instantly.

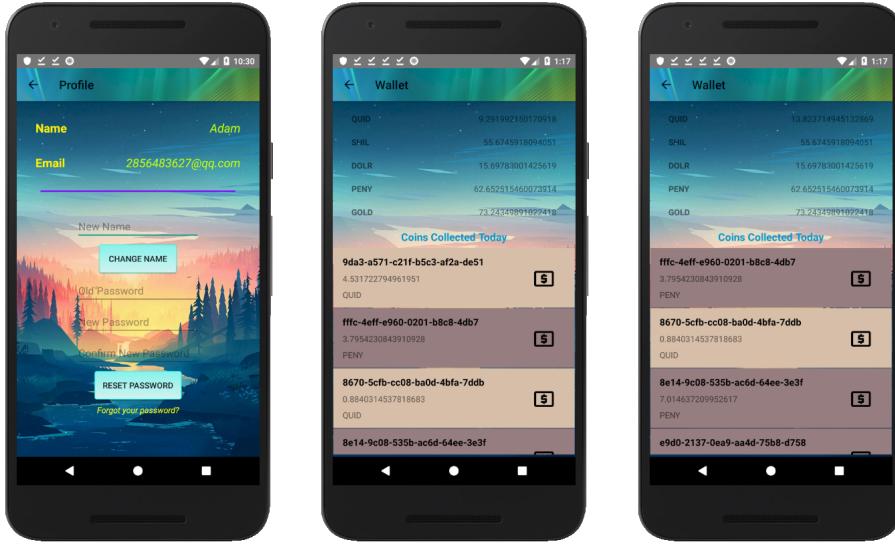


Figure 10: Profile Activity and Wallet Activity

3.8 Bank Activity

3.8.1 General Functions

Bank activity allows users to make currency exchanges, and check today and previous exchange rates.

3.8.2 Exchange Rates

Exchange rate is randomly (with reasonable fluctuations) generated automatically by the first user who starts the bank activity everyday (generate a set of rates and then upload to Firestore).

3.8.3 Exchange

All of the four currencies could be exchanged to gold and vice versa, and no direct exchange could be made among the four currencies. Also, bank would charge a fee of 5% for each transaction. Transactions modes could be changed by clicking the buttons on the right of the exchange rates. See figure 11 on page 14 for a transaction example, and please be aware of the colour changes.

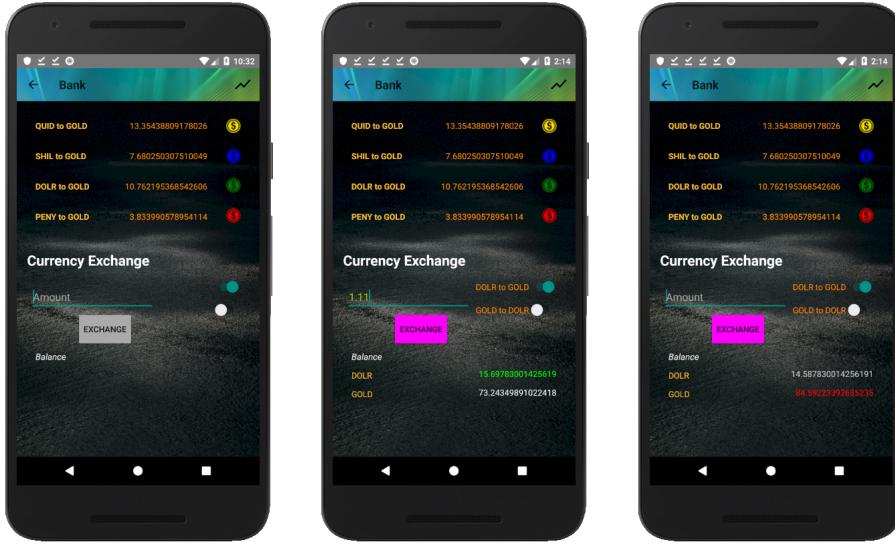


Figure 11: Bank Activity

3.9 Rate History Activity

It is a bonus feature not mentioned in the design, which allows user to check previous exchange rates. See figure 12 on page 15.

3.10 Shop Activity

It is a bonus feature mentioned in the design, which allows user to buy things with coins collected and complete daily mission. See figure 13 on page 15 for a mission complete example.

A popup menu is used to store the coins which are available to pay with.

3.11 Friend List Activity

Friend List Activity shows all the friends user has added, and allow user to send coins to friends. See figure 14 on page 16.

A popup menu is used to store the coins which are available to send away.

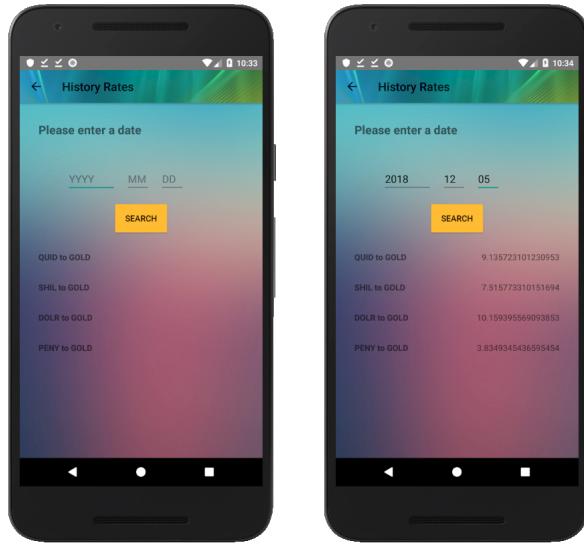


Figure 12: Rate History Activity

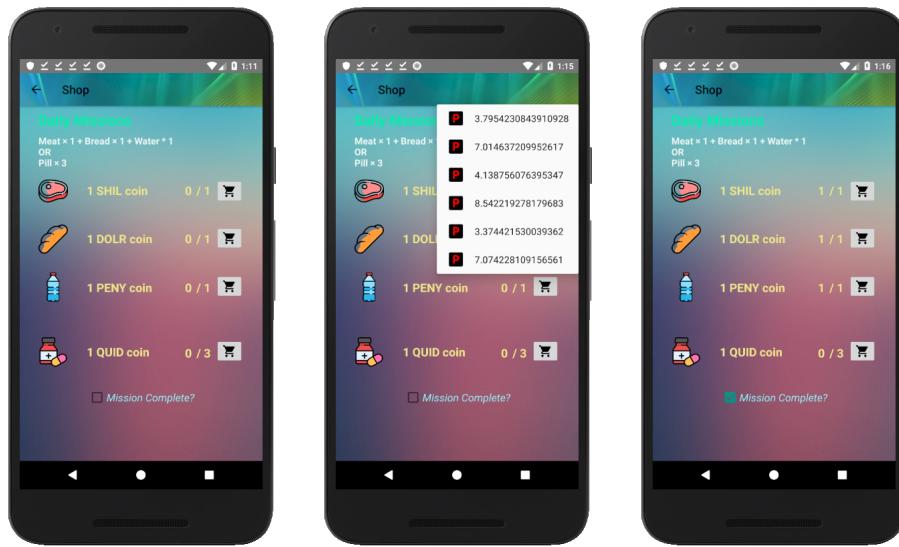


Figure 13: Shop Activity

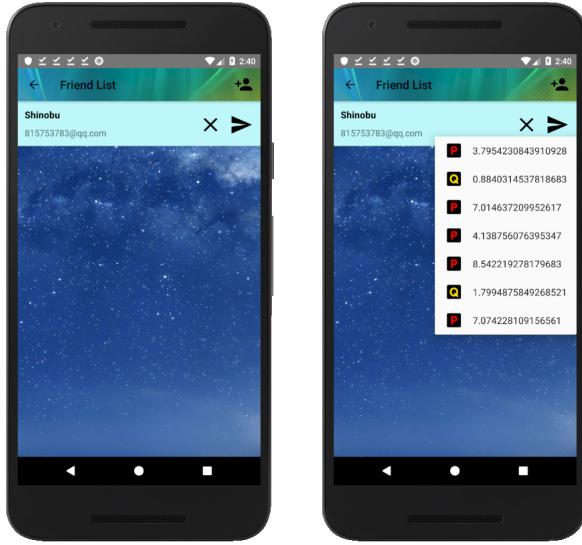


Figure 14: Friend List Activity

3.12 Add Friend Activity

Add friend activity allows users to add others as friend by sending a request. See the first image of figure 15 on page 17.

3.13 Friend Request List Activity

Friend request list activity shows all the persons that are willing to add the user as a friend and provides the user options on whether to accept or refuse. See the second image of figure 15 on page 17.

3.14 Unimplemented Bonus Features

Due to limitation of time, those bonus features mentioned in the design was not realised: distance meter and pay-in helper.

4 Acknowledgement

Background Images: www.pixiv.net

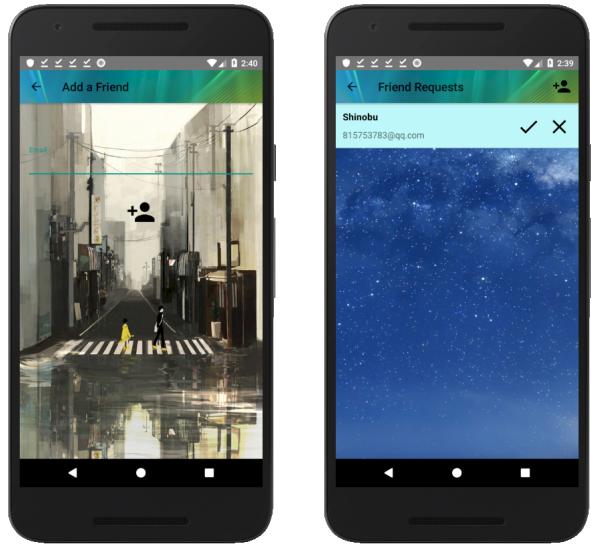


Figure 15: Add Friend Activity and Friend Request Activity

Background Images: <https://www.pinterest.co.uk/>

Icon Images: <https://www.flaticon.com/packs/simpleicon-ecommerce>

Haversine Formula: <https://www.movable-type.co.uk/scripts/latlong.html>

Popup Menu add images: <https://www.youtube.com/watch?v=ncHjCsoj0Ws>

Mapbox: <https://www.youtube.com/watch?v=p9fOTyRqdV0&t=652s> (whole series)

List View: https://www.youtube.com/watch?v=EwwdQt3_fFU&t=823s