7th of May, 2021

Assignment 2

Harry Singh – hars@itu.dk

Project made with Ellen-Sofie Frydenlund

Harry Singh
ITU COPENHAGEN, BSC IN SOFTWARE DEVELOPMENT 4TH. SEMESTER – MOBILE APP DEVELOPMENT COURSE

Content

Most important design choices, including functionality and user interface

The layout of the user interface and how is it implemented in activities / fragment

Below (figure 1) there is a screenshot of the user interface layout of all pages in the application. See 'appendix' folder to get the full resolution.

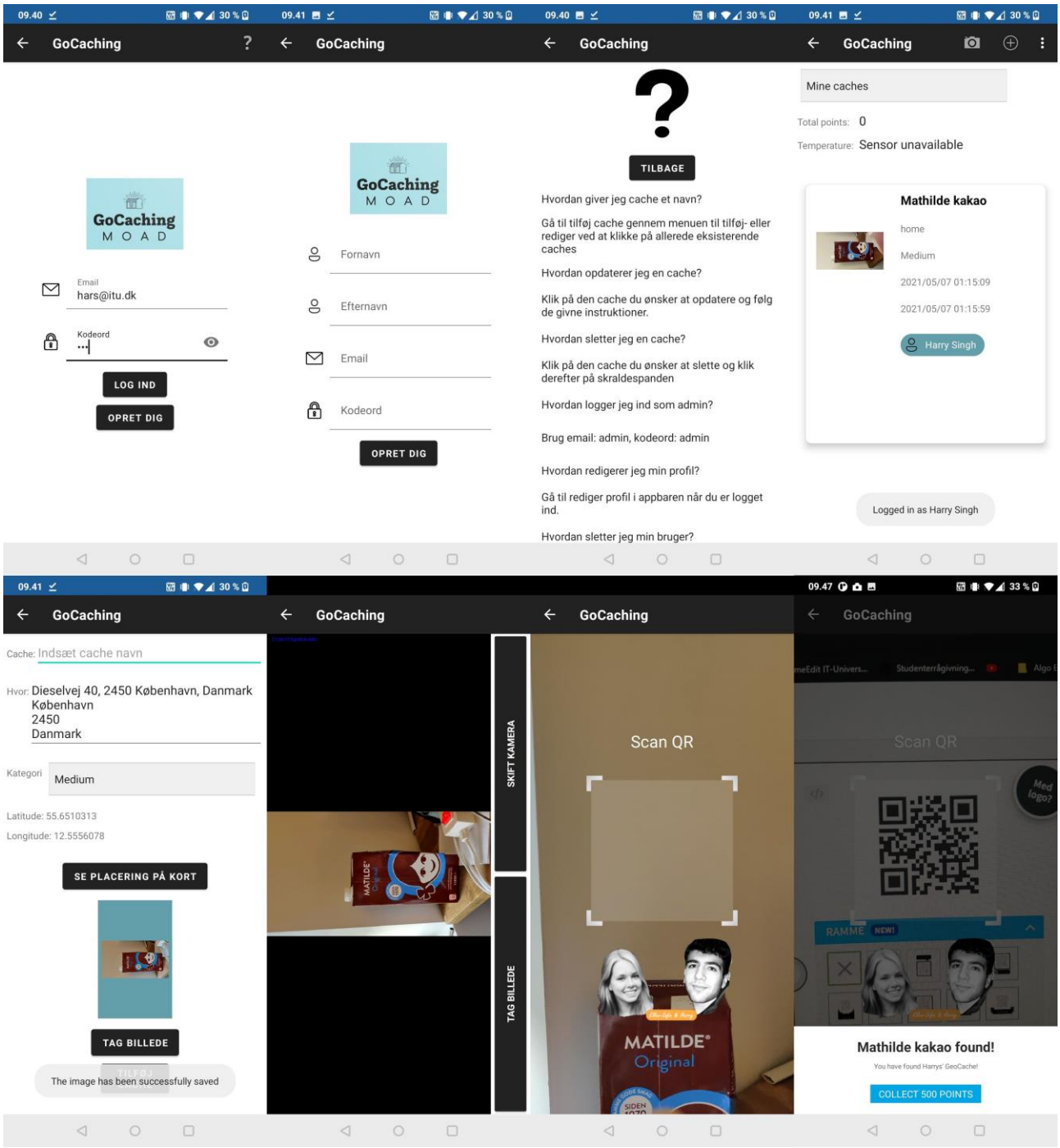

Figure 1: Screenshots from the app

There is a lot of different Kotlin files and XML files, and it can be quite difficult to navigate through and make sense of what is connected to what. I have created an image (figure 2) that show the link between Activities, Fragments and their XML files. It is sorted in a hierarchy of who is the parent. LoginActivity and SplashActivity is at the top, then we have GoCachingActivity which is a child of LoginActivity. CameraActivity, MapActivity and ScanQrActivity is all child of GoCachingActivity.

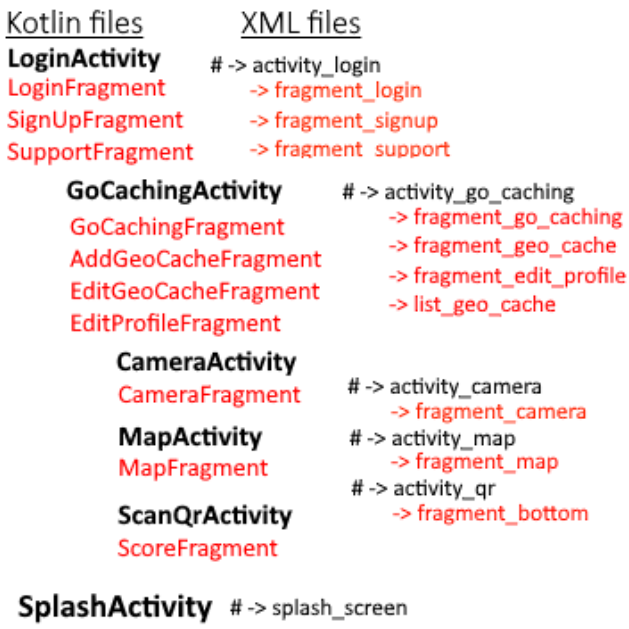

Figure 2: Links between files

I found a way to make AddGeoCacheFragment and EditGeoCacheFragment to use the same XML file as Layout, fragment_geo_cache. This is quite useful to create less redundant code and reuse layout files that are almost similar.

Structure of my database

The database is created with MongoDB Realm and consist of two tables defined in the code: GeoCache and User (figure 3). The relationship between the two tables can be seen where we have a one-to-one relationship starting from GeoCache: user that connects a GeoCache to the User who create it. I also tried implementing one-to-many relationship from User: geoCaches where a User is able to add found GeoCaches in their list. This is used to calculate the users' total points in the Dashboard.

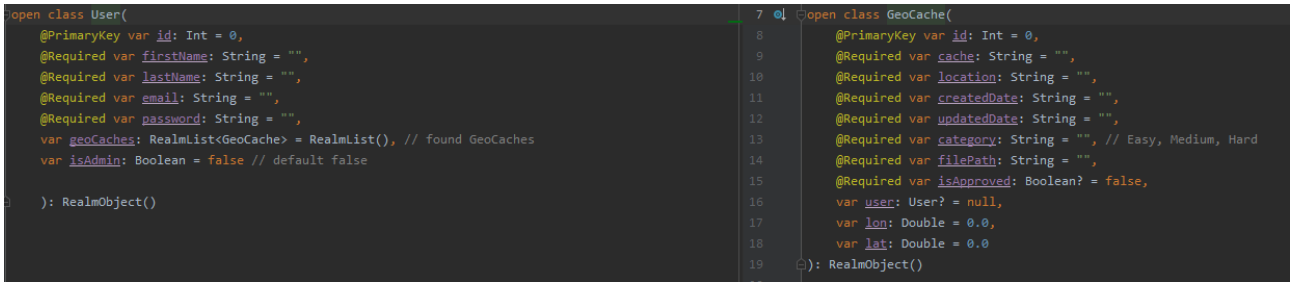

Figure 3: Realm table User and GeoCache

Diagram of the application structure

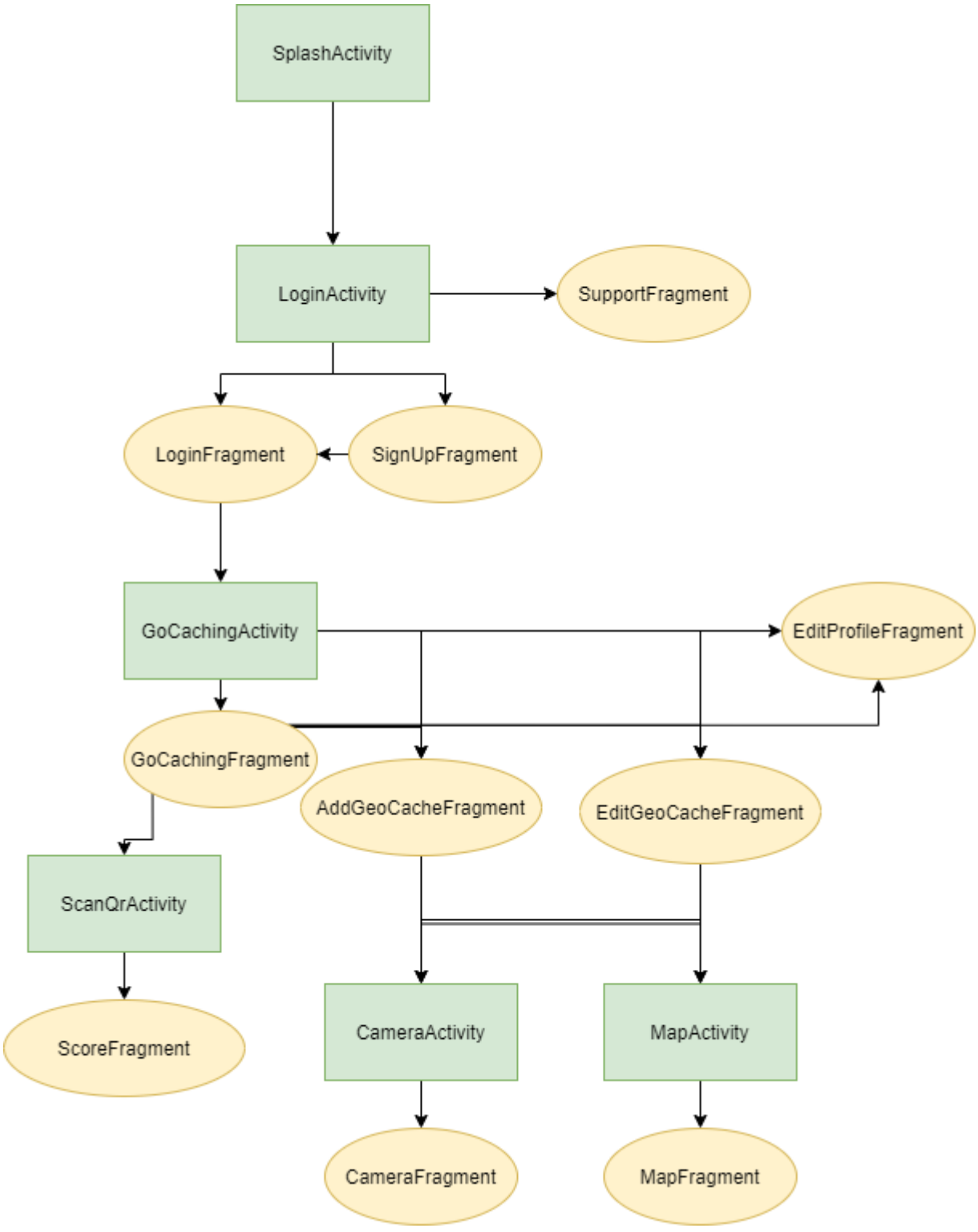

Figure 4: Diagram of application

Primary functions

Register a GeoCache

To register a GeoCache a User will have to use the AddGeoCacheFragment. The User fill out the information relevant to the GeoCache and the clicks on the button “ADD CACHE”. The backend code now executeTransactionAsync on the variable mRealm which is an instance of a Realm class.

1. First the latest id gets collected from the database: If no id (GeoCaches) has been created before, set id to 0 otherwise plus the latest id with Integer 1, for the next GeoCache id in the Realm database.
2. Creates a temporarily GeoCache: newGeoCache and fills out all the @Required and relevant fields from the Realm GeoCache table.
3. At last, the realm inserts GeoCache object into the database.
4. By default, the GeoCache isApproved is false because an Administrator must approve the GeoCache before it can get displayed for all regular users.

Load GeoCaches in dashboard (GoCachingFragment)

The onViewCreated in GoCachingFragment calls a function: listAssociatedCachesToUser(user: User).

1. If myUser (the user that is currently logged in) isAdmin = true, then display the GeoCache associated to user from the parameter.
   An admin should be allowed to see all GeoCaches associated to a specific user also those who is not approved.
2. If myUser is not admin, then just display the GeoCaches associated to the selected user from the spinner – but only display those who isApproved is equal to true.
   A user should only be allowed to see approved GeoCaches.
3. When it comes to displaying the results we use the GeoCacheAdapter and loops through each results and display the list of geocache in ascending order from id.
   First created geocache gets displayed at top and last at bottom.
4. GeoCacheAdapter is also where we implement the functionality to edit GeoCache by clicking on the GeoCache card called itemView.setOnClickListener.
5. If myUser is not admin do not allow to edit GeoCache not created by the logged in user.

Scan QR code

This functionality is created by Belal Khan on September 30, 2020: https://www.simplifiedcoding.net/qr-code-reader-android-kotlin/. The code is his, but I have modified it to make it work on my implementation. I use the ScanQrActivity, ScoreFragment and MyImageAnalyzer to scan the QR code and add the GeoCache. In the other fragments we ask permission on runtime with a popup display but Belal Khan just redirect the user to Android Settings where the User must allow access to the camera. This is an implicit intent way to do it and I liked the way, so I choose to keep it instead of editing it. The MyImageAnalyzer “readBarcodeData” collects the Barcode data and I use the TYPE.TEXT to read the Barcode.

When scanning a QR code:

1. Retrieve the GeoCache Id with function: getGeoCacheIdFromQrCode(..) and basically just returns the digits in the text from GeoCache.
   Example: “geocacheid 14” returns “14”.
2. The Scorefragment() receives the geocache id from MyImageAnalyzer the function ScoreFragment().displayGeoCache() gets called.
3. A temporarily value foundGeoCache see if the GeoCache id exists and gets returned the GeoCache object from the ScanQrActivity.
4. If the GeoCache is found and isApproved is true, then display information and allow user to collect their points. The user cannot collect points if the foundGeoCache is already in the list of geoCaches in the RealmList<GeoCache> from Realm User (Cannot cheat and collect repeatedly same points from same GeoCache).
5. Add to RealmList if the user have not found the GeoCache before. (**This function does not work and the application crashes, this issue is addressed in the ‘Problems in the app’ section**)

Short user guide

YouTube demonstration of my app can be seen here: https://www.youtube.com/watch?v=rQCIozxnLBo

1. Create a user by clicking 'Sign Up' button on the start page and fill out all the details and make sure to use an email that is not already in use.
2. Login with your account email and password
3. On the dashboard you can see your own caches, load other Users' caches, see your total points and if your device is equipped with a temperature sensor, you are able to see the temperature in Celsius. You can also see five menu items on the top right corner: Scan QR, Add GeoCache, Edit Profile, Log out.
4. Add a new GeoCache in the system by clicking on the (+)-add sign on the top right corner.
5. Give the app permission to collect your GPS data and after approximately 5 seconds the nearest registered address is added in the "Where" text field. Your current Latitude and longitude are also printed out.
6. Fill out your cache information and you must submit a picture by clicking the 'TAKE PICTURE' button. When done, click the button 'ADD CACHE' To add your Geocache.
7. Your cache is now awaiting approval from the game manager (administrator). The administrator can choose to edit your GeoCache and still approve it or they can just approve it as you submitted it. The administrator has access to all Geocaches and can also remove users' Geocaches. **You** as a **user** is **not allowed** to **edit** others **Geocaches**.

For fun, try creating lots of different users and add different Geocaches at different coordinates. Category 'Easy' gives a user 200 points, medium gives 500 point and hard gives 1000 points.

8. View approved Geocache by accessing the Geocache map from the Add Geocache page or Edit geocache page by clicking the button 'SEE LOCATION ON MAP'.

Remember, when updating your GeoCache it must get approved again by administrator.

9. Scan others QR code and collect the points from the found Geocache. You cannot collect the same Geocache more than once.
10. Generate a QR code for your Geocache and place it on where your Geocache is located – so others users is able to find your Geocache and scan it. Create a QR-code text that includes the ID of your Geocache. (e.g. For GeoCache id 42 write "geocache 42"
11. Edit your profile by clicking on the top right corner with three dots placed vertically. Here you can also choose to Log out.

Administrators' approval users' cache:

By default, the system provides an administrator user:

Email: admin

Password: admin

1. Log in as administrator from the start page.
2. Click on the spinner to show other users in the system and click on their name to see their Geocaches.
3. Click on their Geocache to approve or disapprove (remove) a Geocache – when clicking on Geocache, you can edit the Geocache. **Approve** it by clicking (+)-approve button on the top right corner or **disapprove** it by clicking the trash can.
4. As administrator, you can edit all Geocaches in the system and you can also add your own (just remember to approve your own Geocaches!)

Test of app

Test case #1 (Sign up, log in, edit profile and delete profile)

1. Sign up a user without a last name should return “Last name must be provided”.
2. Sign up the user by filling out all fields.
3. Try sign up a different user with the same email should return “Email already exists!”
4. Log in with your created user, edit your profile and save changes.
5. Edit your email to “admin” and should return “Email already in use!”
6. Delete your user by clicking on Edit profile and clicking the trash can on top right corner.
7. Try log in again with same user you just deleted should return “Email do not exist”.

RESULT: Works smoothly, no problem.

Test case #2 (Add GeoCache, Edit GeoCache, Approve GeoCache, Scan QR, Collect Points)

This test requires you have created two different users and you are using the admin to approve GeoCache.

1. Create a GeoCache without taking a picture should return dialog “Cache name, Where, Category and Photo is required!”
2. Where should be automatically filled with the nearest registered address.
3. Create a GeoCache by filling out what is required.
4. Log out and log in with another regular user and create a different GeoCache
5. Create one QR-codes with text-format: “geocache 1” from QR code generator (e.g. https://www.qr-code-generator.com/)
6. Try scan the QR code and it should return “GeoCache does not exist”. This is correct because the GeoCache has not been approved yet.
7. Log out and log in as Admin email: ‘admin’, password: ‘admin’. Select the users from the spinners click on their GeoCaches and on the top right corner click the (+) icon which approve the GeoCaches but try edit one the cache names to something else.
8. Log in and you can see the users own GeoCache and the other users but one of them has a different cache name.
9. Log out and log in with a user (doesn’t matter which one) and scan the QR code again and the GeoCache information should be returned and how many points you can collect, depending on you choose ‘Easy’, ‘Medium’ or ‘Hard’ which equals to ‘200 points’, ‘500 points’ or ‘1000 points’.
10. Click collect points and your total point should be displayed at the top of GoCachingFragment (dashboard)

RESULT: SUCCESS UNTIL STEP 10. Backend fails on adding points (GeoCache) to the User.

Test case #3 (See GeoCache markers on mapFragment)

Before testing create four GeoCaches. Two of which have been approved by admin and two of which is still waiting approval.

1. Go in the Add GeoCache fragment by clicking on the (+) icon from Dashboard and wait about 5 seconds to let the GPS register your location. Click on 'SEE LOCATION ON MAP'.
2. You should be able to see your position marked as a blue circle and you should only see the two approved GeoCaches and by clicking on them you can see their Cache name.

RESULT: Success, works.

Test case #4 (Cannot edit others Geocaches)

This test case requires you to have created and approved different Geocaches with at least two users.

1. Log in with one of your users.
2. Select another user from the spinner in the dashboard.
3. Try editing their Geocache should return "You are not allowed to edit others Cache!"

RESULT: Success work. You cannot edit others cache as a regular user.

Test case #5 (Editing your profile information to empty)

This test requires you creating a new user.

1. Edit your profile.
2. Delete your first name by making the field empty.
3. Click save changes and you should not be allowed to save empty first name.

RESULT: Fails. This error handling is not covered only in Sign up but not when editing profile.

Problems with my app

The application has a lot of bugs and small errors. I will start by listing some of them:

- Cannot collect points from found Geocache when scanning QR-code.
- You can delete the administrator users on runtime (however, it will always be added again on new run as default).
- If you move camera too fast when taking a picture, the OpenCV camera implementation gets overloaded and crashes.
- You can edit your profile information to empty – no control, but email cannot be changed to one that already exists (which is the most important error handling).
- The CameraFragment is sideways meaning that the camera is not vertical when the phone is vertical – gives a weird layout. However, rotation is able in the camera.
- Landscape mode is not possible in the app – when changing to landscape and rotate the phone back it will reset and go back to the LoginFragment.
- The back button on top left corner confuses the user and makes them think that it goes back to last fragment, but goes to parent fragment (which often lead my test person, roommate, to log out – when he tested the app.)
- The XML layout could have been improved – a lot of unnecessary code as LinearLayout, etc.
- When adding a Geocache with let us say an address field with a lot of next lines makes the Geocaches push down the other information in the dashboard (GoCachingFragment).
- A lot of messages in the program is hardcoded.

The biggest two problems are:

1. Cannot collect points from found Geocache when scanning QR-code.
2. Cannot use landscape.

Collecting points

The theory code behind collection points and counting points including all error handling to it should work. However, I was stuck with a problem which I could not solve in time (due to deadline). The first error I got is that: “I cannot execute transaction on a realm that is from another thread” which I fixed by making sure that there was only happening one realm executing at the time. Then I get the error as seen in Figure 5, which is: “I cannot copy an object from Realm instance” – still the same type of error. The problem is I am working with two different realms in one realm transaction execution.

I could not manage to solve this in time but theoretically I would have to make sure that I do not mix other realm objects in the executing happening in the function addGeoCacheToUser from ‘ScoreFragment’. myUserId is from another realm I think that is causing the error.

```
E/AndroidRuntime: FATAL EXCEPTION: main
    Process: dk.itu.moapd.gocaching, PID: 12467
    io.realm.exceptions.RealmException: Async transaction failed
        at io.realm.Realm$1$2.run(Realm.java:1705)
        at android.os.Handler.handleCallback(Handler.java:938)
        at android.os.Handler.dispatchMessage(Handler.java:99)
        at android.os.Looper.loop(Looper.java:233)
        at android.app.ActivityThread.main(ActivityThread.java:8010) <1 internal call>
        at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:631)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:978)
     Caused by: java.lang.IllegalArgumentException: Cannot copy an object from another Realm instance.
        at io.realm.RealmModelListOperator.checkCanObjectBeCopied(ManagedListOperator.java:309)
        at io.realm.RealmModelListOperator.appendValue(ManagedListOperator.java:212)
        at io.realm.ManagedListOperator.append(ManagedListOperator.java:92)
        at io.realm.RealmList.add(RealmList.java:252)
        at dk.itu.moapd.gocaching.view.ScanQrActivity$addGeoCacheToUser$1.execute(ScanQrActivity.kt:149)
        at io.realm.Realm$1.run(Realm.java:1649)
        at io.realm.internal.async.BgPriorityRunnable.run(BgPriorityRunnable.java:34)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:462)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1167)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:641)
        at java.lang.Thread.run(Thread.java:923)
```

Figure 5: Scan QR - Collect point error

Cannot use the landscape

Again, due to time pressure I was not able to implement the landscape view but theoretically I would have to create a shared preference and when rotating it I can make sure the activities and fragments are updated from the shared preferences. Some of the implementation can be seen in the AddGeoCacheFragment but is not really in use yet.

Detailed information about the additional functionalities I have implemented in my GoCaching app

Only administrator can see the (+)-approve button in the menu item (figure 6). I have created isAdminAndIsNotApproved() which return true if the user is an Admin and the GeoCache has not yet been approved. So that if a GeoCache is approved the button will still not show for the administrator. I like those small details that makes the user experience better.

```
private fun isAdminAndIsNotApproved(): Boolean {
    // return true if isAdmin = true and isApproved = false
    if (myUser.isAdmin) {
        return myGeoCache.isApproved != true
    }
    return false
}

override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
    super.onCreateOptionsMenu(menu, inflater)
    inflater.inflate(R.menu.fragment_geo_cache, menu)

    val approve = menu.findItem(R.id.approve_cache)

    approve.isVisible = isAdminAndIsNotApproved()


}
```

Figure 6: Display approve button