Candidate number: 462

Android programming exam report

Introduction

In the following report, I will try to explain, as short as possible, all decisions and problems I have faced during the development of the application.

Layouts

Constraint layout is one I used most, besides I have used **linear layout** and **frame layout** (as the holder for fragments).

UI Components

To display data from the database I have used **recycler view** as instructed in assignment text. Each element inside of recycler view is a **card view.** Other components used were: **Seek bar** (adjusting how often to fetch data), **Checkbox**, **Progress bar** (to show that data is being saved in the database), **Web view** (to show comments from API as they are in HTML format), **EditText**, **TextView**, **Button.**

Patterns

I have used the **DAO pattern** with my database to satisfy requirements for Room persistence library, that needs DAO (among other things) to be used. I also have used the **singleton pattern** to create a connection to the database as it is an expensive operation and we need only one connection to the database.

Libraries

I have used the following libraries in developing this application:

- **Retrofit 2** – used for downloading data from https://www.noforeignland.com/ endpoints
    - As part of retrofit, we have also Gson converter, for converting data received from API to Kotlin objects
- **Room persistence library** for database CRUD operations
- **Picasso** to display an image in single location activity
- **Google maps** for displaying the location of a port on the map

Of course, I have imported needed libraries, such as Room compiler for Kotlin, etc. but do not feel that they are worth discussing.

Implementation – decisions, and challenges

Splash screen activity

This activity has as main goal downloading data from API (no foreign land) and saving it to the database. Data is downloaded from /home/api/v1/places endpoint.

To create classes for **data parsing** I have used **JSON to Kotlin class plugin** that has been recommended in class. To download data I used **Retrofit 2**. And to parse received data I used **Gson converter** from **Retrofit 2**. In combination with the retrofit, I have used **listener class with callback methods** for success, error, and progress.

To prevent downloading data on each run of application I am saving information about the last update of data in the database to **shared preference**. This way data is fetched **automatically** every 30 days. It is possible to change the frequency of updates or to trigger the update manually from settings activity, also it is possible to disable downloading of data completely. The percentage of data written to the database is shown to the user.

If there is no need for downloading data, we make a short delay to show the splash screen and go to the main activity.

I used **finish()** method to prevent a user from going back to the splash screen from the main activity.

Main activity

The main challenges in this activity were displaying data from the database in the recycler view and implementing search for given data. To implement search I used **filter function,** on data received from the database, and compared it to input from the user. The number of locations displayed in recycler view is shown under **Edit Text**, on typing, this value is updated accordingly. To add on **text change listener** I had to implement **TextWatcher** which was a minor challenge as I did not meet with it before

Single location activity

This activity could be seen as a two-part activity as it is a holder for one of two fragments.

Single location description fragment Web view is used to display comments from API as they are in HTML, the picture has on touch listener to zoom in it, and on clicking pin map fragment with location is shown.

Google map fragment can be invoked by clicking pin from either main and single location activities. On clicking pin user is prompted for permissions to access the location of the device.

Extras

In this section, I am going to briefly discuss all the things that were not originally part of the assignment and that I did not discuss previously.

- **Theming** – I decided to choose blue and white as the main colors for my application, thinking that simple is best.
- **Toolbar** – For easier navigation, I styled toolbar such that it is holding the title for the current screen, back button, and menu (3 dots) where we can close the application or go to settings of the application.
- **Settings activity** – This activity is used for choosing if and how often data should be downloaded and to re-download data manually by clicking on button re-download data. Challenge was the implementation of a slider component.
- **Shared preference controller** – I felt that since I am manipulating shared preference data on multiple places in code it would make sense to have a class that is going to hold common methods and variables for such manipulations
- **Functional programming** – As mainly Java programmer I did my best to focus on functional programming (using the filter and for each instead of for loops…)