Database of MRT Stations

CSC2007 Mobile Application Development Spring 2021

MRT Station Database

Fork the repo **csc2007-quiz02-2021** and inspect the code within the project.

The goal is to create an app using Android Architecture Components with a SQLite relational database that saves all the MRT Station data and displays them within a list in the application.

Hitting the Floating Action Button on the bottom right of the screen will open another separate Activity that will provide a screen to input the necessary fields for the MRT station to be saved to database. Returning to the main screen after adding the item will refresh and display the newly added MRT Station within the RecyclerView.

All screens should survive rotation and be refreshed with the latest data in the database.

Implement the main screen

Design the layout and logic of the `MainActivity` similar to the given screenshot. Implement the RecyclerView on the MainActivity to display the MRT Station data, and the floating action button. Each RecyclerView item should be able to display 3 rows of text that corresponds to the columns that are retrieved from the database.

Note: All MRT Station data should be saved and loaded from a SQLite Relational Database, using the Room architecture component. Refer to MRT_System_Map.pdf in the root folder for station names and the map.

IMPORTANT:

- **Ensure the RecyclerView id is named recyclerViewStations within the XML**
- **The 3 TextView fields ids in the RecyclerView row XML should be named textViewStationCode, textViewStationName and textViewLineName**
- **The floating action button id should be named fabAddStation in the XML**

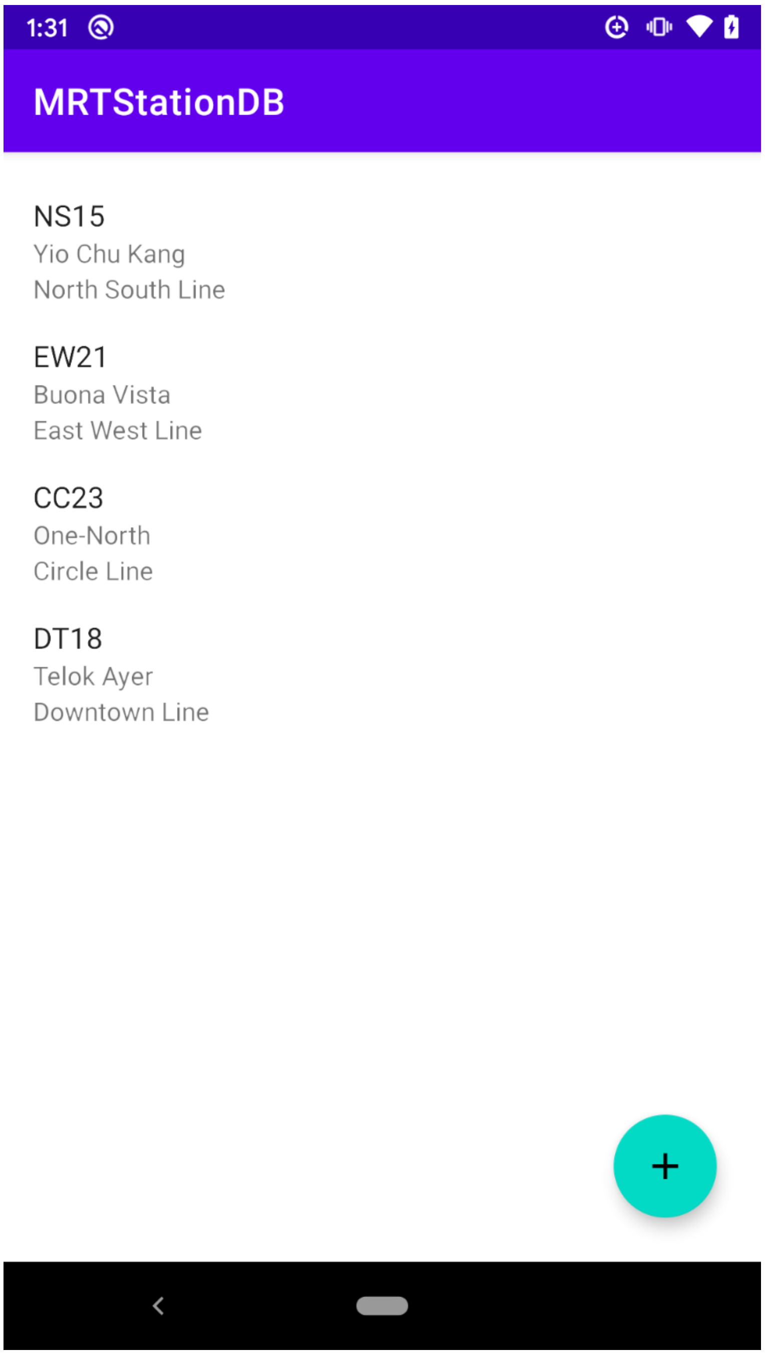
Hints:

A custom implemented RecyclerView item will be necessary for the display of 3 rows of text.

ViewModel might be necessary to survive rotation and keep the list updated.

Information on implementing the floating action button is here:
https://developer.android.com/guide/topics/ui/floating-action-button

Use the drawable ic_add_black_24dp to assist you in implementing the floating action button



NS15
Yio Chu Kang
North South Line

EW21
Buona Vista
East West Line

CC23
One-North
Circle Line

DT18
Telok Ayer
Downtown Line

MainActivity (contains a RecyclerView with custom item layout)

Implement the database

Implement the Room SQLite database to store the following information about an individual MRT Station: **the id, station code, station name** and **line name:**

- ID: <primary key autoincrement>
- Station Code: EW21
- Station Name: Buona Vista
- Line Name: East West Line

Implement the logic for the database access itself, and also to add and retrieve from this Room SQLite database with the necessary columns.

Enter a new MRT Station

Create a new activity called `AddStationActivity` that will provide the text fields and a spinner to add the necessary item into the database. Returning to the main screen will display the newly added MRT Station within the database.

The "Add" button adds the MRT Station to the database and closes the current activity and returns to the main screen. The "Clear" button clears all the fields. Hitting the "Back" arrow on the navigation bar simply returns to the main screen.

Do these simple checks on this Activity to prevent invalid entry of the fields:

- Check for **empty strings** being entered for the station code and station name
- Check that there are **no digits** in the station name

The checks should be done within a method with the following signature:

```
boolean isInputValid() {


}
```

Display a "Toast" message if there are invalid entries in the text fields. The toast message should EXACTLY say "Invalid data input"

IMPORTANT:

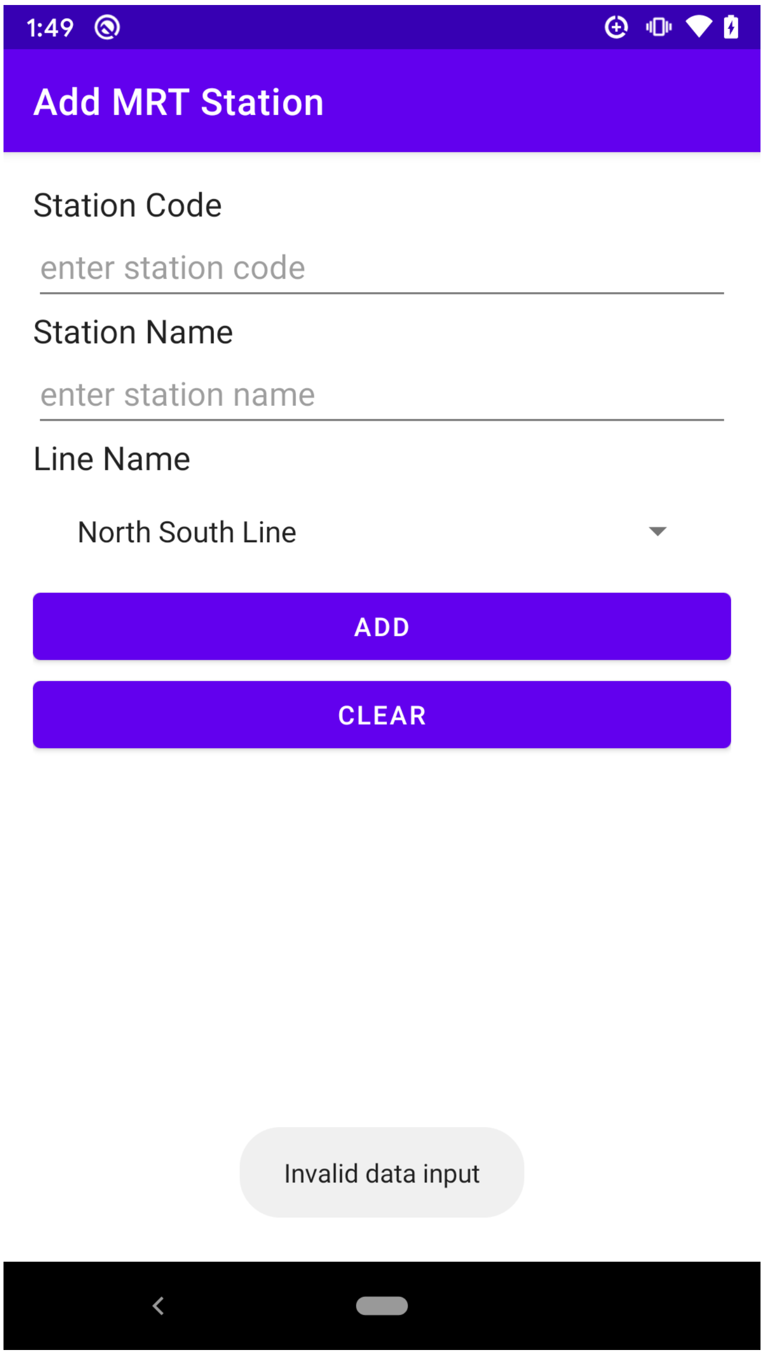
- **Ensure that the Activity class name is AddStationActivity**
- **Ensure there are no spelling errors on the buttons, text views and spinner**
- **The three field ids in the XML should be named: editTextStationCode, editTextStationName and spinnerLineName**
- **Toast message should say "Invalid data input"**

Hints: Information on how to implement a spinner is here: https://developer.android.com/guide/topics/ui/controls/spinner

Save last entered values

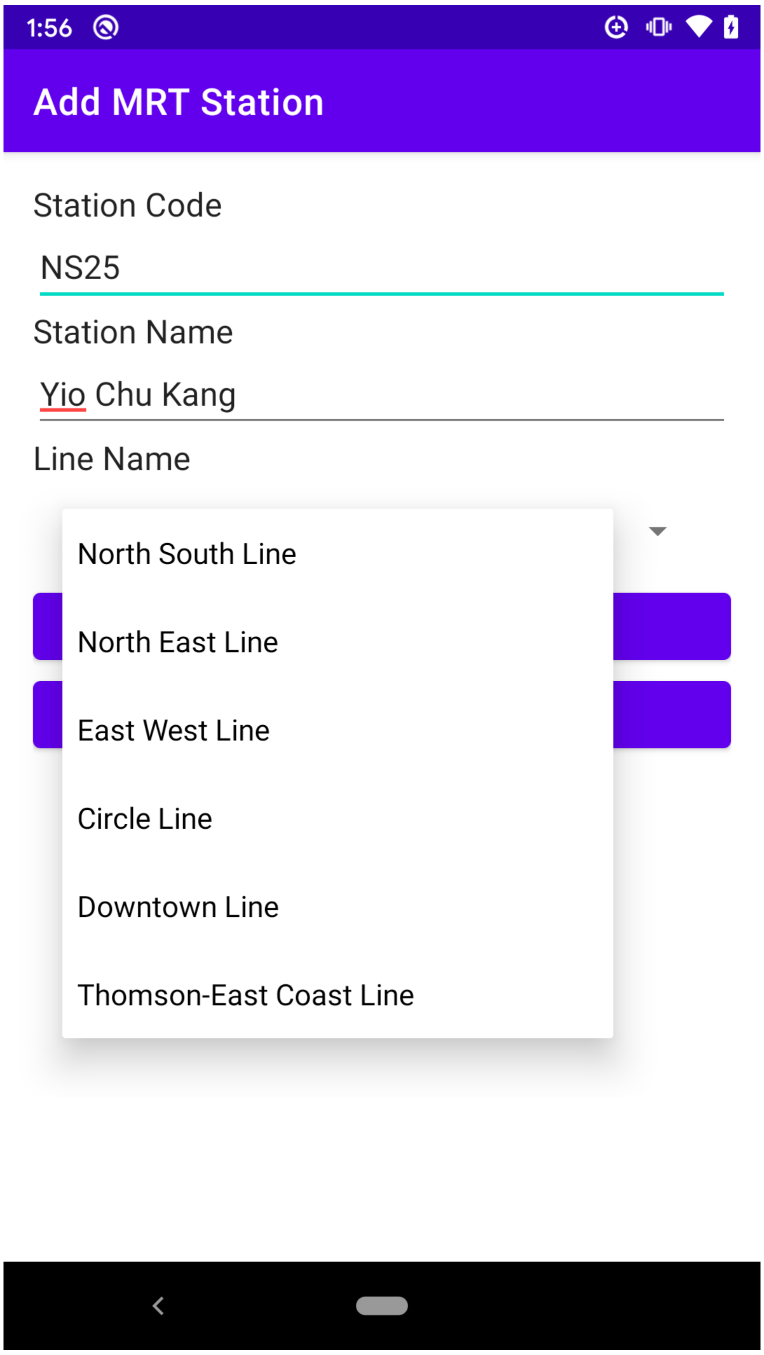
If the `AddStationActivity` is closed and reopened, or if the app process is killed and restarted, the last entered and added data of the station code, station name and line name should be reloaded and displayed in the `AddStationActivity`.

Implement the logic to save and reload the data that was last entered on the 2 text fields and the spinner on the screen. When the "Add" button is pressed, any valid values should be saved to Preferences DataStore. (And also inserted into the Room database as implemented earlier). Do not save to Preferences DataStore if the values are invalid.



AddStationActivity (empty fields, with toast)

IMPORTANT: Activity should be named AddStationActivity



Station Code

NS25

Station Name

Yio Chu Kang

Line Name

AddStationActivity (with fields filled in and spinner opened)

Lab Quiz 2

1. Fork the repo **csc2007-quiz02-2021.**
2. Design the layouts of each of the screens similar to the given screenshots.
3. Implement the logic for each of the Activities.
4. Implement the missing tags within AndroidManifest.xml to launch the correct Activities.
5. Implement the logic and classes to save the MRT Station data within a Room SQLite relational database.
6. Retrieve and display the MRT Station data from the Room database onto the MainActivity within a listview with list items that display 3 rows of text.
7. Implement the logic to add a new MRT Station to the Room database via the AddStationActivity.
8. Implement the following simple logic and validity checking before adding a new MRT Station, with the method called `isInputValid()`
   - Check for empty strings entered for the station code and station name
   - Check that there are **no digits** in the station name
9. Display a “Toast” message saying “Invalid data input” if there are invalid entries in the text fields.
10. Save the station code, station name and line name to shared preferences, and reload and display in AddStationActivity when re-opened
11. Retrieve and display the updated MRT Station data from the database onto the MainActivity in the RecyclerView **after** adding a MRT Station.
12. Remember to comment and indent the code, and implement it in a modular fashion, using different methods or classes as appropriate. Ensure it conforms to the Android Kotlin coding conventions.
13. Commit and **push** all changes to your forked repository **csc2007-quiz02-2021.**

IMPORTANT: Do not change the activity names, method names, method signatures or package name. Please ensure all spelling and ids are correct.

Grading Criteria

1. MainActivity layout and UI elements are similar to screenshot (1 mark)
2. AddStationActivity layout and UI elements are similar to screenshot (1 mark)
3. Open AddStationActivity when floating action button is pressed (1 mark)
4. Logic on the clear button to clear the fields (1 mark)
5. Check for zero length strings as invalid input (1 mark)
6. Check for no digits within the station name (1 mark)
7. Toast if the input is invalid (1 mark)
8. Successfully add new MRT station to database (1 mark)
9. Save the station code, station name and line name to shared preferences, and reload and display in the AddStationActivity when reopened (1 mark)
10. Add to database and close AddStationActivity, return to MainActivity (1 mark)
11. Refresh the RecyclerView with new record from database (1 mark)
12. MainActivity and AddStationActivity survive screen rotation and retain values (1 mark)
13. Modular code design and good naming conventions for methods and variables (1 mark)
14. Well-commented code (1 mark)

END OF DOCUMENT