Coursewrok 2 Report

For this runningTracker project, I have created 3 total activities for it which are MainActivity, AllRuns and UpdateActivity. All of the activities can be view either in portrait mode or landscape mode. When the app started, a permission to allow the app to access the user's current location will be prompted. The permission has to be allowed by the user in order for the app to record the running distance. If the permission is denied, the app does not start even the "START" button is clicked and an alert message will be displayed to the user.

MainActivity

In the MainActivty, the distance and the duration of the current run is shown in a text view. Besides, there are 4 more buttons which all serve different function such as starting, pausing and stopping the runs and also to check the history of all runs. When the "START" button Is clicked, the notification with the current distance progression is shown in the notification bar which can be view even when the MainActivity is killed. Tapping on the notification bar will move the user back to MainActivity.

AllRuns Activity

The AllRuns activity can be started by clicking the "ALL RUNS" button from the main activity. In this activity, a list of previous running records are shown and displayed in a recyclerview. Data such as DateTime, distance and duration which were retrieved from database are displayed in this activity. Other than those data, additional information that were added by user in the UpdateActivity are also displayed in this activity. In the recyclerView, onClicked method is set which bring the user to the last activity which is UpdateActivity. User can scroll on the recyclerView and click on the view to enter UpdateActivity to edit it.

UpdateActivity

In this Activity, users are able to add or modify the list such as adding notes for the records or inserting an external image from the phone storage to the app. The selected image is then stored in folder and the path of the folder was kept so that the app can trace to the image to display it to the user. An "update button" allowed user to store the changes once they finish editing and the changes were then stored in the database. All of the changes is then displayed in the AllRuns activity and users are still able to make changes on it if they wish to. Other than that, the "Delete" button allows user to delete the current records if they wish to. The "DELETE" button will only delete the specific record that is selected and the other records will remain unchanged. After the delete and update action, the app will redirect the user back to the AllRuns activity.

GPS Tracking

The main component of the project is GPS tracking where the source code is provided from the coursework sheet and added to my project. In the MyLocationListener class there is a method called onLocationChanged which is used to update the location changed every 0.5 second. For every location changed, the distance travelled is added from all the previous results and the updated distance is stored in a variable. A method called getDistance is created in MyLocationListener class to return the total distance travelled back to the service to be used. The distance is added up every time this method is called.

Service

A service is started when the main activity has created. When the “Start” button is clicked in the main activity, a method in the MyService class called onStartCommand. In this method, the location manager is started and a foreground notification is generated. The notification is present for the whole session to keep the service ongoing. While the service is ongoing, the notification in the notification bar is still updating even the activity has been killed. During the whole section, a callback I used using the binder to transfer all the data in the service back to the main activity to be displayed. A thread is used to keep the updating ongoing. The thread will keep updating the UI in the MainActivity using callback from the “ICALLBACK “class as well as the notificaition bar. The distance that are calculated is get from the MyLocationListener class’s get distance method. When the “pause” button is clicked, the counters for the distance travelled and timer are paused for a moment and the thread has been stopped. The thread will start again when the start button is clicked again and the pause time is being stored and calculated to prevent displaying the wrong duration time. The session is stop when “stop” button is clicked and the stopLocationManager method in MyService has been called. This method stops the foreground notification and insert the final result to the database.

Room Storage

A room storage has been used as the database of the project and it is designed based on the MVVM architecture. The database interacts with multiple activities of the project such as storing data into it and retrieving it. In TrackerDao, it consists of the queries to communicate with the Room on what action to do. When a run has stopped, a “Running” object is inserted to the database by using ViewModel from the main activity. Then the Repo pulls the live data from the ViewModel. From there, the data is insert to the Room using the queries stated in TrackDao and this also applies to deleting data in database. Retrieving data also followed the same steps but in a reverse way. The database for this project stores the data from the running app as well as data inserted the users.

Content Provider

A content provider is added to the app as it allows other apps to access to the application data. In MyContentProvider class, a method called query is implemented to allow the requester app to point and get the data of any column from the application database. The requester app is only available to read the data and not allowed to modify or delete the data since the method is not implemented.