Mobile Device Programming Coursework 2

Name: LOH JIN XIAN

Student ID: 016763

Design of the fitness tracker

As requested by the mobile device programming coursework 2 handout, we are required to build a basic running tracker that have the following functions:

- Logging the movement of a user when they go running
- Saving the movement data in an appropriate manner
- Allowing the user to inspect their progress in some way
- Allowing the user to control how and when the logging occurs

Moreover, the mobile application must incorporate all major Android application components which are Activity, Service, Content Provider, and Broadcast Receiver.

After some trials and errors with the design of both the user interface and the database of the mobile application, I have finally settled with the following designs and they will be explained in detail per section. Before designing any mobile application, we must first determine the software requirement specification of the application. As for this fitness tracker app, the functional and non-functional requirements are listed below:

Functional requirements

1. **Start, pause, and stop the fitness tracker**

   The user can start, pause, resume and stop the fitness tracker at any time during the fitness tracking phase

2. **View the tracking data in a table form**

   The user can view all tracking data that the user has been recorded in the past

Nonfunctional requirements

1. **Recording and storing of data**

   Once the user has stopped the tracker, the data will be automatically stored into the database which can be retrieved later

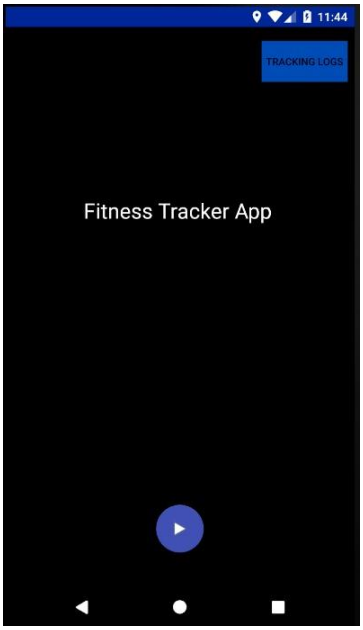
2. **Notification builder**

   When the user starts the tracking phase, a notification will automatically be built and shown to the user in the notification tray. The notification will be deleted when the user stopped the tracker

Detail explanation of the designs

1. User interface design

The design of the backend architecture of an application is essential to ensure a smooth performance when using an application. However, the front-end design of an application cannot be ignored as it is the core bridge between the user and the mobile application. Therefore, serious thoughts have been put into the design of the UI as it needs to be simple yet functional at the same time. The following are screenshots of the user interface:

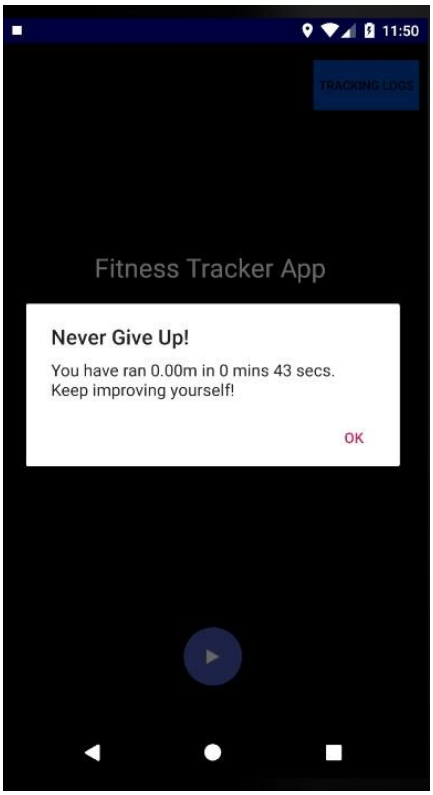


Home Page

There is only 1 homepage in the application equipped with 2 buttons on application startup. By clicking the rounded ‘play’ button at the bottom of the screen, the fitness tracker will start tracking the running activity of the user.

During the tracking phase, the user can either pause or stop the tracker, and notice that the ‘play’ button is automatically hidden too.

When user pauses the tracking, the ‘play’ button will appear again while the ‘pause’ button will be hidden. These design changes ensure the user will not be confused with what a button do but focusing on running instead.



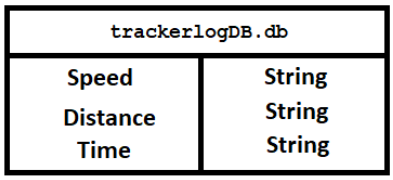
When the tracker is stopped, the user will be greeted with a simple alert dialogue to show how much he or she has run for how long. Besides that, the user running’s data is automatically stored in the database.

Tracking Log Page

By clicking the “tracking log” button at the top-right corner of the homepage, the user will be taken to a new activity screen that includes the user’s running data which includes the distance ran, the speed of running and the time taken to complete the run. By clicking the ‘cancel” button at the bottom of the screen, the user will be taken back to the homepage.

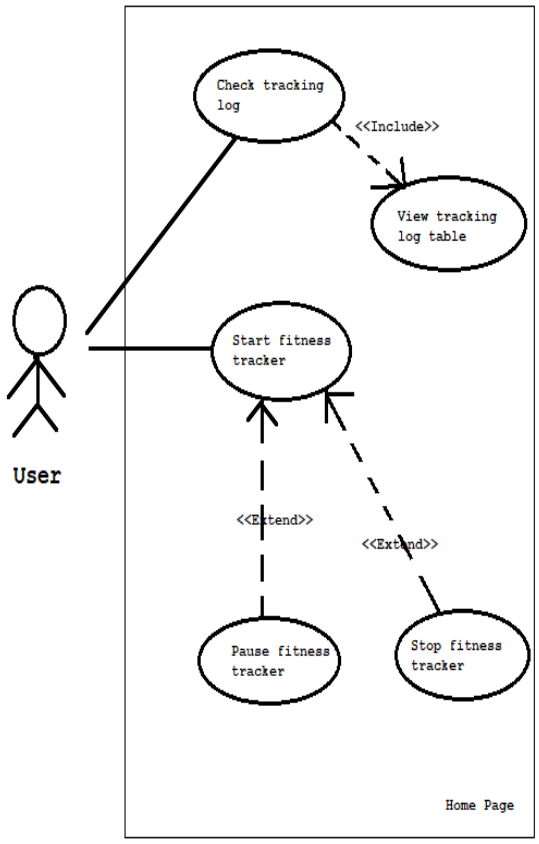
2. Database design

Since this fitness tracker needs to store the user's running data, an SQLite database has to be implemented into the application. The following diagram is the design of the database:



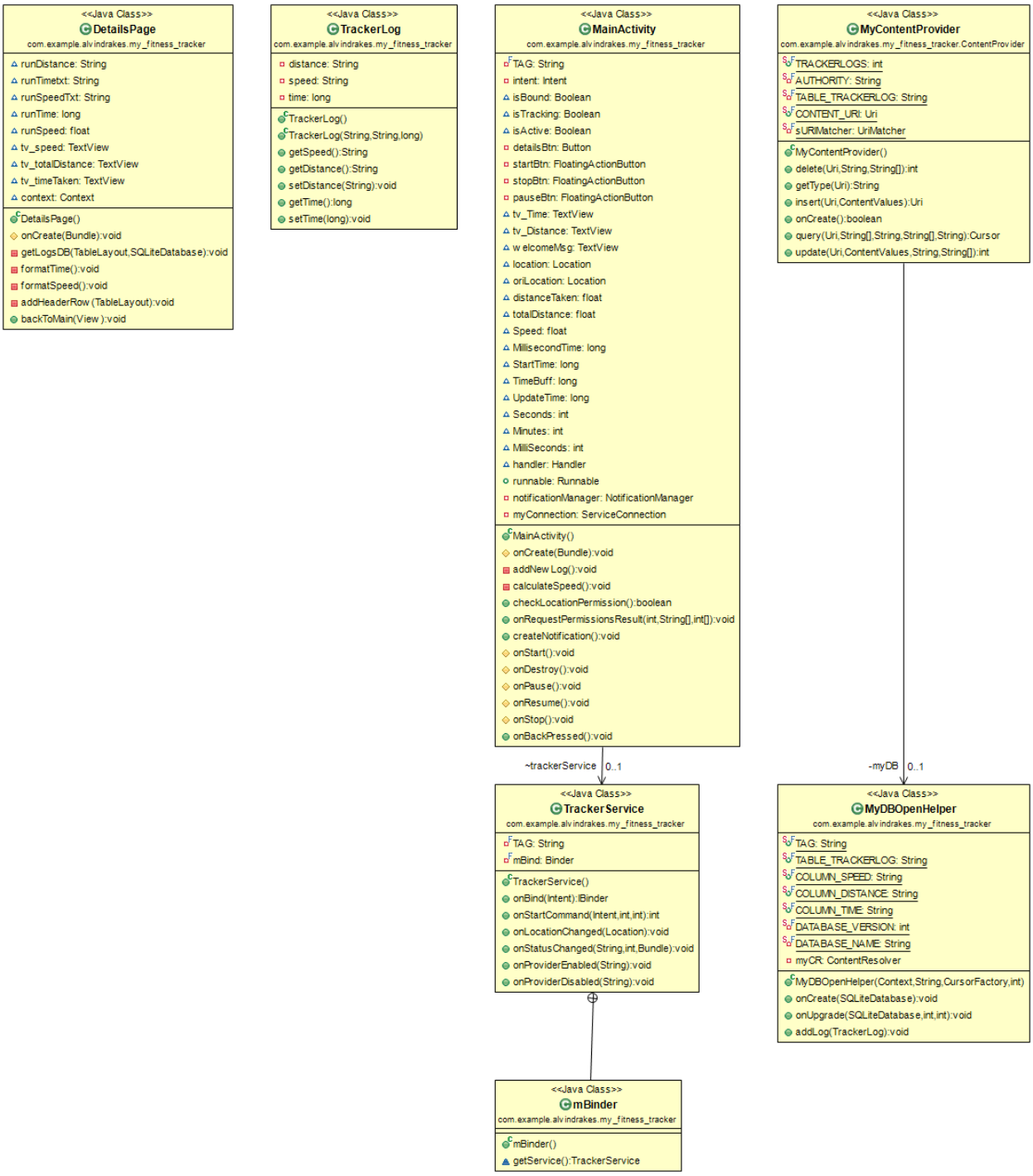
By having a simple database, the process of inserting and querying can be made more efficient and effective. When the user stops tracking the run, the total running time, total running distance and speed of running will be stored here. The purpose of this database is to be used to populate the table layout that is present on the tracking log page. Therefore, the data type of the variables are formatted into String format. At the same time, the user will have all the running data stored on the phone if he or she wishes to access the data at any given time.

3. Use case diagram



As seen in the use case diagram, the user will have the ability to start the tracker. When the tracker is started, the user can either pause or stop the tracker completely. Other than that, the user can access the tracking log page to view all previously stored data.

4. Class Diagram of the fitness tracker app



5. How are Android components used in the app?

Different Android components play different roles in the creation of this fitness tracker application. All 4 components' usage is explained below:

- **Activity**

  An activity is a single focus thing that the user can do. Thus, it is also the base component of an application. Activity component is implemented in both Homepage and Tracking Log Page, enable the user to interact with the UI that is contained in the activity.

- **Service**

  A Service is an application component that can perform long-running operations in the background, and it doesn't provide a user interface. In this fitness tracker application, the component is used to build the notification, time tracking and distance tracking while the tracking phase is running. This ensures the app can still run in the background when it is quit using the Android home button and can only be destroyed by clearing the app from the Android operating system.

  The service is bind when the user starts tracking and it will unbind when the tracking phase is stopped.

- **Broadcast receiver**

  Broadcast receiver receives the intent that is sent by the Broadcasts. Broadcast receiver is implemented here to receive location data that can be used to calculate how far did the user ran by calculating the distance between 1 marker to another marker.

  Broadcast receiver is started whenever there is a change in location data in terms of latitude and longitude during the tracking phase.

- **Content provider**

  Content providers help an application manage access to data stored by itself, stored by other apps, and provide a way to share data with other apps. In this application, content provider is used along with the SQLite database, so the user can add new tracking log and view the tracking log table that needs access to the database, at any given time.