COMP3018 COURSEWORK 2
– Running Tracker

Student Name: Chaeryeong Kim

Student ID: 20206447

Word Count: 1286

Declaration: *I confirm that this coursework submission is all my own work, except where explicitly indicated within the text.*

1. Project Title and Brief Description of the functionalities

The project was to implement a running tracking application named run tracker. As people today are getting more and more interested in their health, they try to record their work-out data using papers, mobiles, and other instruments. Among them, the mobile is quite convenient in the sense that we carry it when we work out especially when we walk or run. This makes easy measurement of the data (e.g. distance, average speed, time etc.) comparably easy with the GPS. Also, inspecting the useful data from the several points of views is clear, simple, and well-organised to see with the mobiles.

The mobile application "run tracker" I implemented supports 4 main functionalities as listed below.

- Log the movement of a user when they run or walk using GPS
- Save the useful data computed from the logs in an appropriate manner
- Allow the user to inspect their data in a useful manner
- Allow the user to annotate their data in a useful manner

All this functionalities are abstracted in the application implementation so that user can record and inspect their data in a user-friendly fashion which also means it's very easy for user to use the application by hiding all the GPS raw data and serving the processed meaningful data.

2. Description of the design and technical architecture

2.1. 4 Components and Their Communication

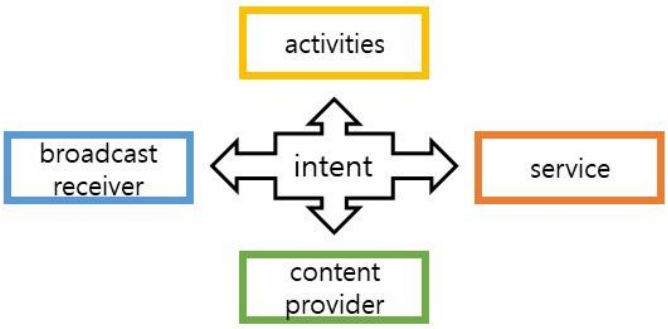
There are 4 components which makes up an android application – activity, service, content provider, and broadcast receiver. All these 4 components are properly supporting the functionalities of the application. For the operating system to notice each of the

components, all of them are specified in the manifest files as below.

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".AddActivity" />
<activity android:name=".StartStopActivity" />
<activity android:name=".HistoryActivity" />
<activity android:name=".TodayActivity" />
<activity android:name=".SingleRecordActivity" />
<activity android:name=".AnnotateActivity" />
<activity android:name=".MonthYearActivity" />
<activity android:name=".TotalActivity" />
```

```
<service
    android:name=".MyService"
    android:enabled="true" />

<provider
    android:name=".myProvider"
    android:authorities="com.example.runtracker"
    android:exported="true" />

<receiver
    android:name=".MyReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="com.example.runtracker.DATE_TODAY" />
    </intent-filter>
</receiver>
```

How these components are related each other and how they communicate can be summarized in the following diagram.



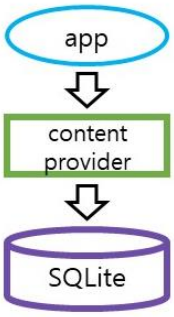
Each component communicates with one another all through "intent" including some data when necessary. There are 9 activities in the application and they navigate each other and pass the data through intent. The following diagram is the collection of the activities of the application.

Depending on the nature of the functionality the activity performs in the application, I divided the activities into 4 groups. The main activity is like the main menu so it's an entry point to other activities via the buttons. To be specific, each button is an entry point of each task that user might perform.

The second group of the activity supports retrieving inputs from users. They are intermediate activities which mean they exist to pass the user inputs to the following activities.

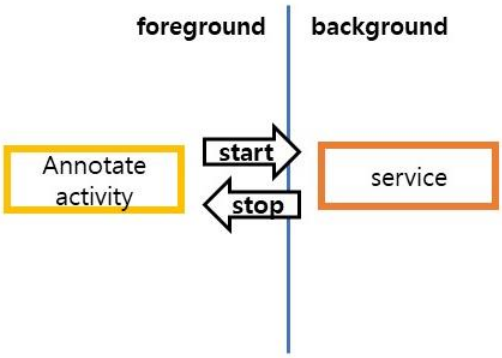
The activities in the third group are in charge of displaying data to users through appropriate views and layouts.

The last two activities are quite important in the point that they communicate between not only other activities but between other components. Annotate activity gets user input and update the data to the lowest data layer through content provider with a contract of data abstraction which is the third component of the application. Within the application, it's easy to hire the content provider to access the data layer. Even more with the consideration of extension of the application, other applications can access to the data run tracker application through content resolver and content provider further. Here is a diagram of relationship between the application and content provider.



The application accesses to the lowest data layer through content provider in a data-safe way and SQLite supports the database fundamentally.

The StartStop activity also access to the data through content provider to log and save the movement data. Additionally, the activity interacts with the service component through "start service" fashion like below.



The annotate activity is in the foreground which interacts with users and has user interfaces. It has a control over the background service which is non-visible to users and explicitly starts or stops the service through the user input via the buttons. Therefore, when the user presses the start button, the service keeps on working at the background over the life cycle of the activity until explicitly stopped by the user when stopping walking or running.

The last component of the application is the broadcast receiver which responds to system-wide broadcasts from the operating system or other applications. In the run tracker, I implemented the broadcast receiver to make toast of the date of today and notice it to the user so that user can type correct date of the day through the broadcast sent by the calendar application for example in the future.

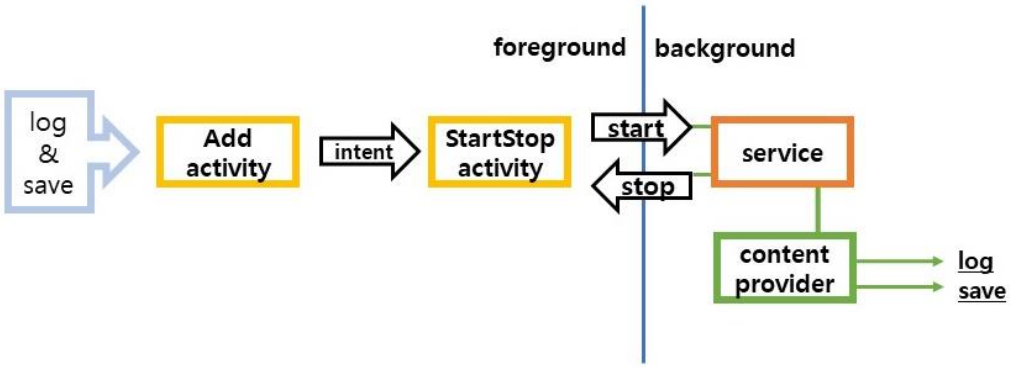
2.2. Tasks and Its Decomposition to Activities and Service

There are mainly 3 big tasks the user might take actions on. Each task spans within or over components – activities, service, content provider. Here are the big 3 tasks I designed.

- Log and save the movement data retrieved with GPS while running or walking
- Annotate the records with feeling, weather, and comment tags to function as diary
- Inspect the data in a useful and easy manner

Following small sections gives explanation of how each task can be decomposed into and achieved through the combination of multiple components.

2.2.1. Log and save the movement data when running or walking



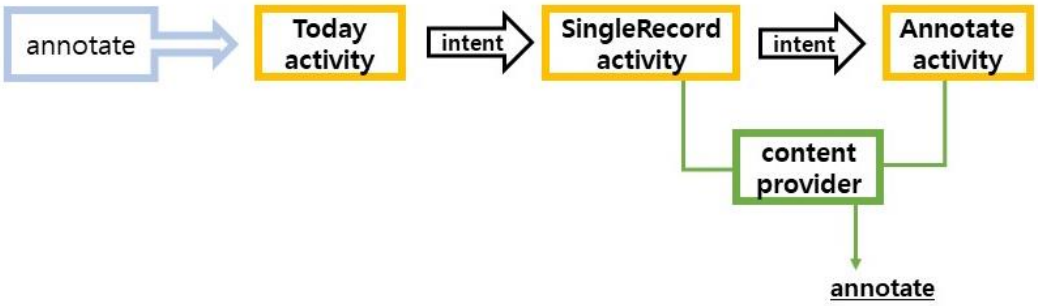
To log and save the movement data using GPS information, the date information of year, month, and day is retrieved from user in the Add activity. Then it passes the date through the intent to StartStop activity which interacts with the user, service, and content provider. When the user presses the start button of the StartStop activity, the service started.

Then with the minimum time interval and distance interval restrictions to update the latitude, longitude, and time of the location is logged through content provider. To efficiently notice the user that it's being logged, the background music is infinitely played during the running or walking until the user presses stop button so that the user doesn't need to watch the screen on moving.

When the user decides to stop logging his/her movement by pressing the stop button of the StartStop activity, the service is finished explicitly noticing that the logging is stopped. Right after the logging stops, the data processing is done to save useful and meaningful data (i.e. distance, average speed, duration of the time) using the logged data. After computing and saving the processed data called "record", the logs are deleted to save space and to easily separate the old from the next new logs.

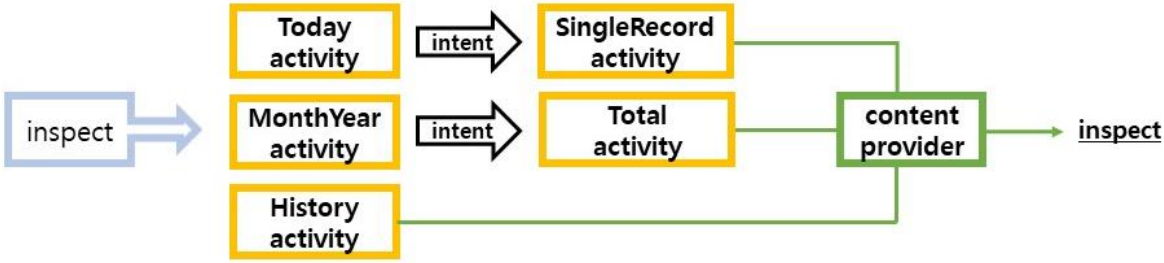
For exception handling of when user tries to run or walk and save the data multiple times in a day, using the date information of the record the content provider updates the record of the day by merging the previous record and newly added logs.

2.2.2. Annotate the records like a diary



To annotate the record, user give date inputs to the Today activity and it is passed through intent to SingleRecord activity displaying the record of the given date by querying through content provider. When the user wants to annotate tags or update comments to the record, the intent with date is passed to the Annotate activity where EditText views retrieve the new annotations and update it to the record through content provider. Here, the pair of year, month, and date is acting like a primary key for the record for querying and updating the rows.

2.2.3. Inspect the data

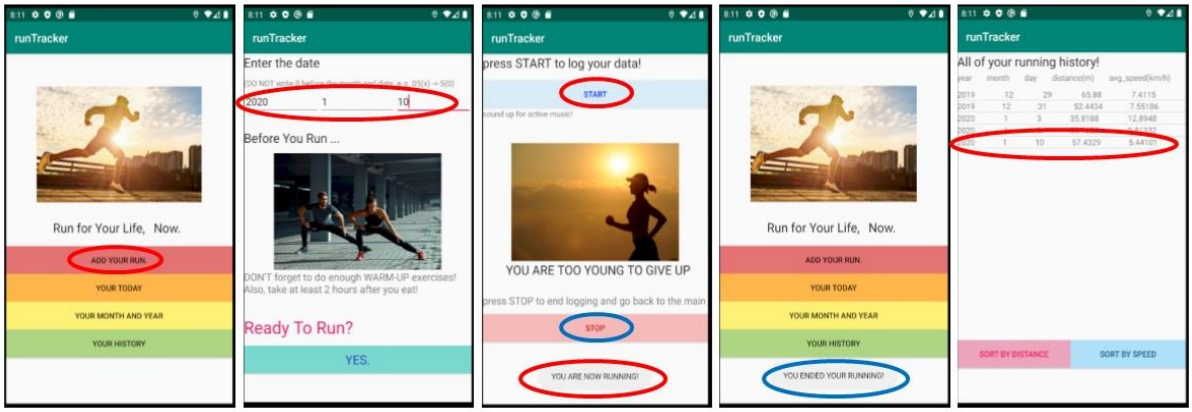


This application gives the user 3 ways to inspect their records. Through Today and MonthYear activity, user gives a date input and it's passed each to SingleRecord activity and Total activity. SIngleRecord activity through content provider displays all the relevant, useful data of the given day including date, distance, average speed, feeling, weather, and comment. Total activity shows total distance the user recorded through the application in the month and in the year. History activity shows all records in a list view and also provides sorted list views either in distance descendant or average speed descendant manner.

2.3. Behaviour of the Application from Users' View

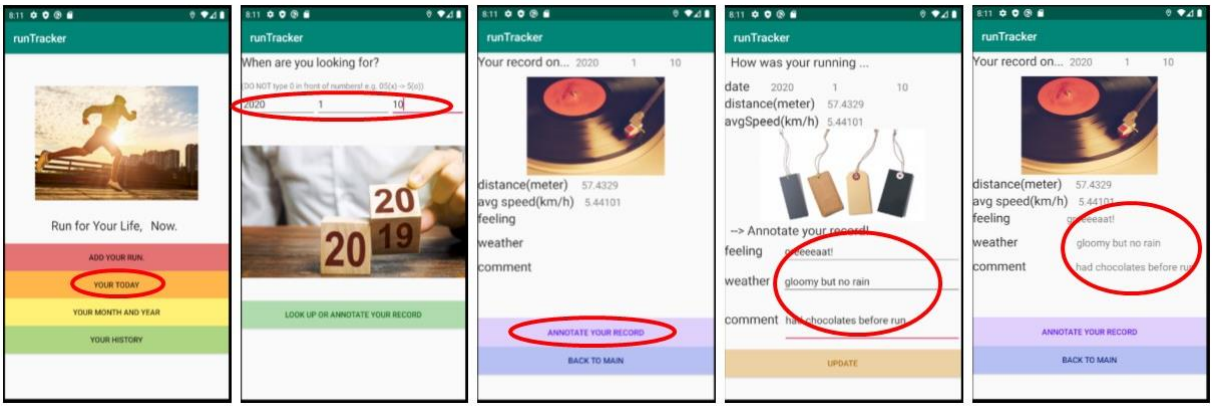
When user tries to achieve the tasks in the run tracker, the application behaves like below for each task. Following diagrams shows the behavior of the application with screenshots of the emulator.

2.3.1. Log and save the movement data



After entering the date of today, if the user presses start button, the music is being played and the toast comes out. When the user stops walking or running and presses stop button, the music stops and another toast displayed. Through inspecting the history, the user can check that their new data has been recorded and saved.

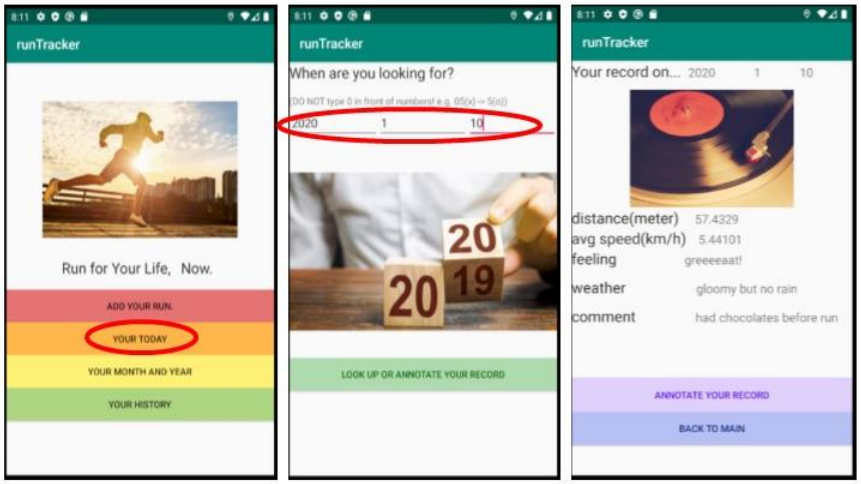
2.3.2. Annotate the records



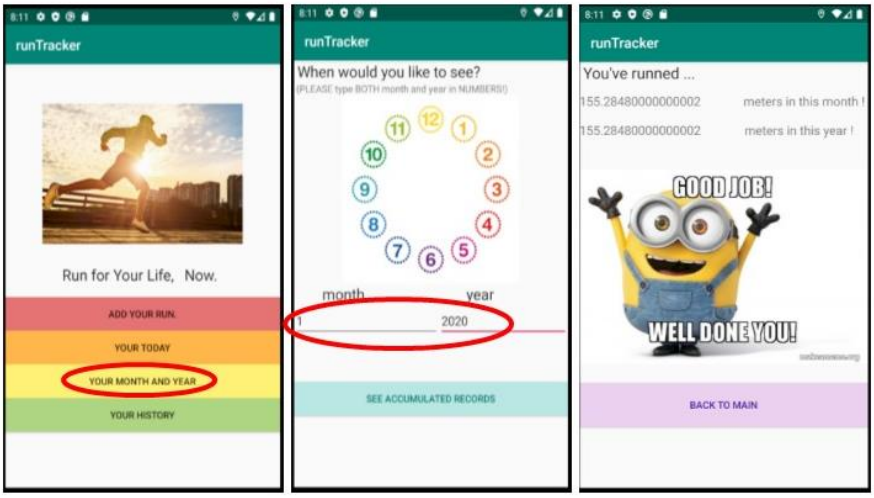
When the user gives one of the date inputs form the dates contained in records, the SingleRecord activity shows all records of the day and the user can add annotation as the screenshot shows. After update, the newly added records are shown to the user.

2.3.3. Inspect the data

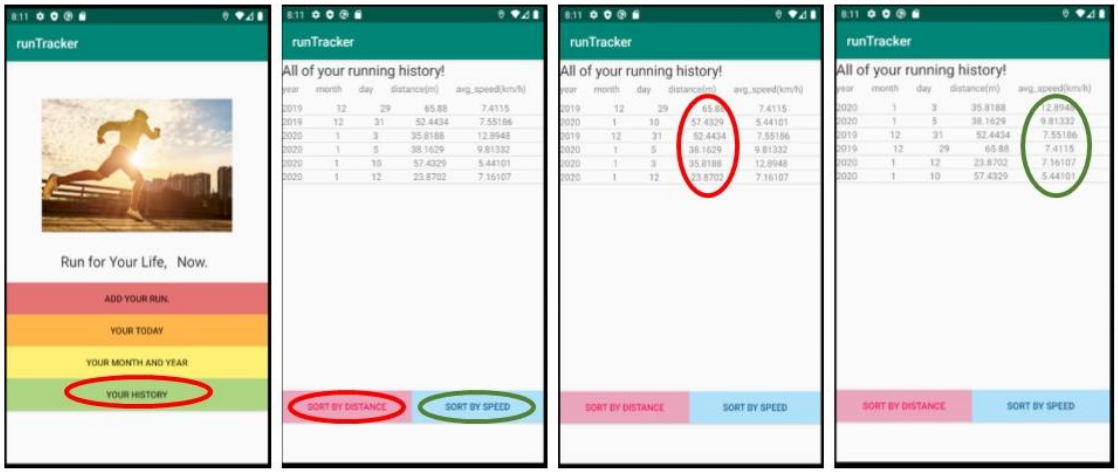
There’re 3 ways to inspect your data.



The user can see all the records of one given day.



If the user wants to see how long he/she've run through the month and the year, it's also possible to view total distances of the month and year as the screenshots show.



Through History activity, the user can see all the list of records in one window and also sorting the list by distance or average speed is possible.

Comments

A demo video including behavior of the application from the installation to how to use the application is included in the zip file.