Running Tracker: Strada

## Introduction

I created this app primarily as a running tracking device, and in being a runner myself I knew which features I would want from the app. Having said that the app can be used for walking as well, but it would be somewhat less optimal and is designed to track when the app is in use and does not use a remote service. The app tracks the distance, duration and the pace of the run – with the pace being a key feature that runners use to track how fast they run, measured as time per kilometre in the app. The user is also able to add a name to their run and add comments so that they can remember how they felt or the conditions on that day. The date of the run is also recorded. In the Record Activity a MapView was implemented to show the user’s current location on the run, with a black line being drawn where the user has currently run. This feature is particularly useful if the user has a route in mind where they would like to run.

## Main Activity

The Main Activity shows the users most recent activity, with the newest runs appearing at the top of the ListView. These runs can be clicked which takes the user to the Edit Run Activity where the name and comments can be updated, or the run can be deleted. After this the user will return to the Main Activity and the recent runs will be updated to show any changes. At the bottom of the activity the user can choose to record a new activity or see the statistics of their runs. When the record button is clicked, the Main Activity starts a local broadcast, which the Location Broadcast Receiver receives.

## Location Broadcast Receiver

This component is used to start the Location Service, receiving the broadcast from the Main Activity and starting the service via an intent from itself. It also displays a toast that tells the user the service has been started. If the app were to be extended this would allow other activities to start the Location Service.

## Edit Run Activity

This activity shows the details of a particular run. It comprises of a series of TextViews for the distance, date, duration and pace. The name and comments are EditTexts so that the user can edit those within the activity, whereas the others should not be altered by the user. If the activity has been accessed from the Main Activity then the run’s name and comments can be updated via the content provider. The run can also be deleted if the user wishes to discard the run. If the activity has been accessed from the Record Activity, the save button inserts a new record into the database via the content provider. If the delete button is pressed the run is not saved and the information is discarded. When the Edit Run Activity is started it is passed a function value from the bundle, which allows the activity to decide whether to update or insert, or to delete or discard data.

## Strada Provider

Strada Provider in the Content Provider for the app. This in addition to the Strada Provider Contract class abstracts the underlying SQLite database (StradaDB) where the information about each run is stored. The provider allows the Edit Run Activity to update, insert and delete entries from the database. In addition, the provider allows the Main Activity and the Statistics Activity to query the database to show the selected records.

## Record Activity

The Record Activity allows the user to see their current location through the MapView and the Google Maps API. The map follows the user as they move, using the Polyline feature to draw a line from the previous location to the current location. This allows the user to see where they have been on the current run, and where they might want to head to reach a certain distance or find their way home. This is the activity that binds to the Location Service and implements an ICallback, that gets updated every second by a thread in the service. This updates the Views in the activity that display the values from the service. There are play, pause and finish buttons in the activity, with the activity initially started in the paused state. Below there is text to show the user which state, recording or paused, the service is in. Once the play button is clicked, the timer will start and the distance will increase as the use moves. The pace will also show how fast the user is moving. Once the user is finished they can pause the service and click the finish button, which will then take them to the Edit Run Activity.

## Location Service

The Location Service handles all the distance and duration of the app. When created the service creates a Location Listener that updates the current location and distance stored in the service class. It then also creates the notification, which when clicked allows the user to return to the record activity. The service can be bound to via a binder class within the service, which calls internal methods of the service, so that they can be accessed from the Record Activity. The main process of the service is to run the Tracking thread, which uses a RemoteCallbackList to call a function in the Record Activity every second, to update the values from the service to the TextViews in the activity. This only happens if the service is in the RECORDING state as compared to the PAUSED state, which is checked in every call of the function.

## Statistics Activity

This activity allows the user to see all their previous runs, displayed with the most recent runs first. There is a Spinner in the activity that allows the user to select to see all their runs, all their runs from the current week or the current month. At the bottom of the activity they can see the total distance covered from their Spinner selection in kilometres.

See below a diagram showing the components of the app and their relationships:



