DistanceTracker

Documentation

I. Design

This part will explain how and why the app is designed the way it by going through the different layouts.

MainActivity:

When a user starts the app, this screen appears.

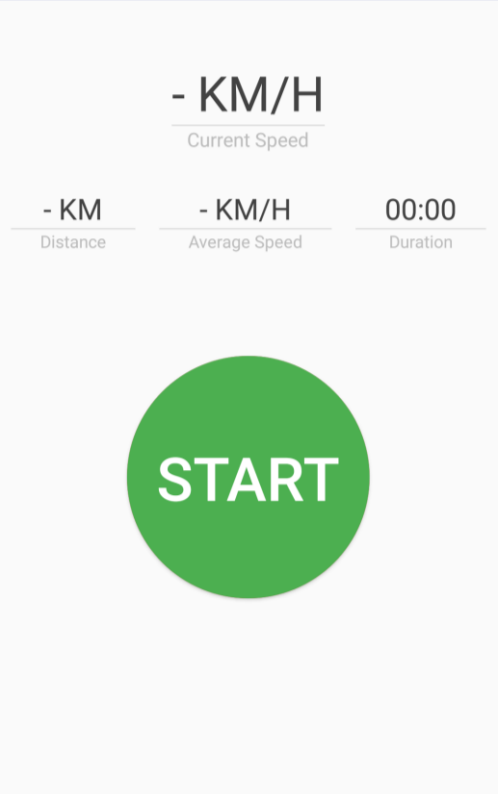
I placed the different elements of the UI strategically, the current speed is at the top and is the biggest in size since it's the main info the user will need.

Then, there are the secondary info just below the speed: distance, average speed and duration.

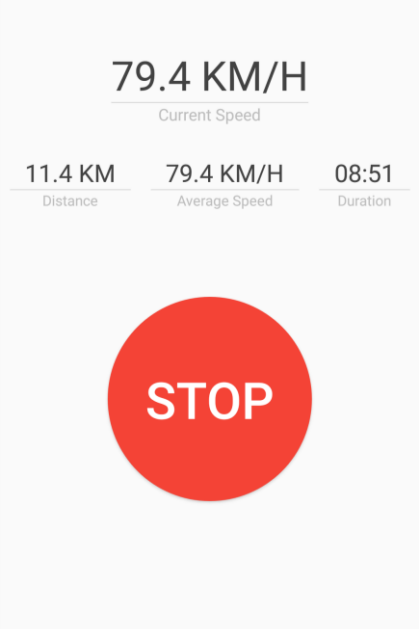
At the bottom of the activity, there is the big button that will be used to interact with the app. It has two states: START and STOP.

When you click START, the button change its colour to red and the text as well to "STOP".

When you click STOP, the app will go to the Summary Activity.



Initial State



Running State

SummaryActivity:

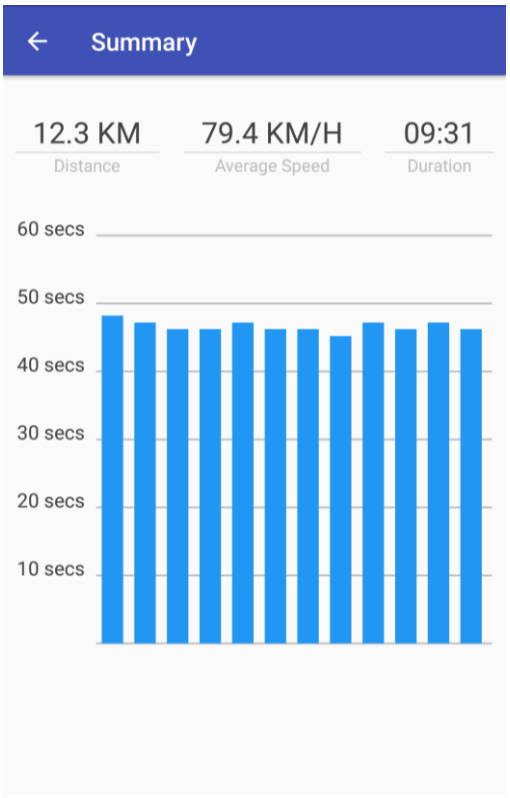
When the user finished his running, he clicks “STOP” and this activity starts.

I placed the same secondary info the same way it was in the MainActivity to don’t confuse the user.

Then, there’s the graph that sums up the running where each bar represents a kilometre.

The graph will scale to the kilometre that was the slowest.

The user can go back to the MainActivity by pressing the back arrow in the ToolBar.



Summary Activity

II. Code

This part will explain the code.

MainActivity:

Override of onRequestPermissionsResult:

This method will be called when the user respond to the request permission for the GPS and will call the method to initialize the location listener.

initComponents:

This method will initialize all the components contained in the "activity_main.xml".

initListeners:

This method will set the correct listeners for components contained in the "activity_main.xml".

initLocationListener:

This method will initialize the location listener by overriding "gotLocation" from the inner abstract class "MyLocation.LocalisationResult" then pass it to the constructor of MyLocation. When the location changes, it will call the correct methods to update the UI.

updateDistance:

This method that take a Location object, will handle the distance covered since the tracking was started thanks to the received location. Each time a kilometre is passed, the time spent for this kilometre is stored for future usage. It also updates the UI.

updateSpeed:

This method that take a Location object, will handle the current speed thanks to the received location. It also updates the UI.

updateAvgSpeed:

This method that take a Location object, will handle the average speed thanks to the received location and the previous speeds. It also updates the UI.

getAverage:

This method computes the average speed from the array that stores every current speed at each location received.

reset:

This method resets the UI and every variable needed in the activity.

startSummary:

This method launches the SummaryActivity and give it every information it needs like the distance covered, the average speed, the duration of the run and the array that represents every time for each kilometre.

MyLocation:

This class is used in the MainActivity, its usage is to get the current location and call the correct methods in the MainActivity.

constructor:

The constructor takes a “LocationResult” as parameter which is an inner abstract class. Thanks to this LocationResult, MyLocation will know what to do when the location changes thanks to “LocationResult.gotLocation”.

startLocation:

This method starts the listening of the GPS and Network modules through two LocationListeners (locationListenerGps and locationListenerNetwork). The refresh interval time is set to 1 second.

stopLocation:

This method stops the listening of the two modules used to get the location.

SummaryActivity:

init:

This method initializes the activity by getting the components through “findViewById” and setting the correct texts for distance, average speed and duration. It also gives the array of time for each kilometre to the custom view “TrackerGraph” which is part of this activity.

TrackerGraph:

This class extends from View and is the graph that represents the time spent for each kilometre.

init:

This method initializes the several Paint objects used for the lines of the graph, the texts and the bars.

Override of onDraw:

onDraw draws the lines of the graphs, the texts that appears next to each line which represents the time and the rectangles(bars) that represents the time spent for each kilometre.

Override of onMeasure:

onMeasure will force the view to be a square.

giveTimes:

This public method will set the array of times used to draw the bars.

Override of onMeasure:

This method will be called each time onMeasure is called. It computes the correct scale for the y-axis (the time) of the graph with the max time in the array of times. It also computes the size used in onDraw to know how to divide the space for each bar.