|  |
| --- |
|  |
| Testing Strategy Document |
| Remind My Stop |

|  |
| --- |
| Zaeem Siddiq |

Testing Strategy Document

# Introduction

Remind my stop is an application which, after passing through vigorous stages of testing, has reached to its completion. As RMS was purely GPS data based app so testing was not less than a challenge. Testing was performed after every key functionality was performed whether it was simple displaying of user’s location on the map to more complex tasks such as calculating distances between two points. Although android studio 2.0 has now enabled the GPS location simulation procedure where we can plug in array of points or simply send in singleton locations to jump to latitude and longitude. However, this strategy turned out to be very slow performing.

By doing a bit of research on internet I came to know android devices can mock their locations using a set of instructions. In order to automate this mocking functionality, android developers out there have already created user friendly applications which are quiet handy.

The application which I used is called “Fake GPS location” [1]. My strategy included the following steps:

(*NOTE: All the testing was performed on HTC ONE M7*)

1. Starting my application (Remind My Stop RMS).
2. Minimizing RMS.
3. Launching Fake GPS location.
4. Setting my desired location and pressing “Set Location”
5. After pressing Set Location button, Fake GPS automatically goes to background. Because RMS was running in background so I automatically switched to RMS (app resumed)
6. When the location changed, RMS sensed the new location and performed the required computations.
7. Later on, I found out another attractive and feature of ease, which was by bookmarking the GPS coordinates into a list. E.g. Drag marker to Chadstone SC and press Star button, after giving it a desired name it automatically persisted into the file. Which is used later onwards at any point of time by simply selecting that location from the list.

# Sample Test Case

I have conducted a test which is based on the bus route number 900. This bus would depart from Chadstone SC and will travel all the way to Caulfield Station.

## Stops List

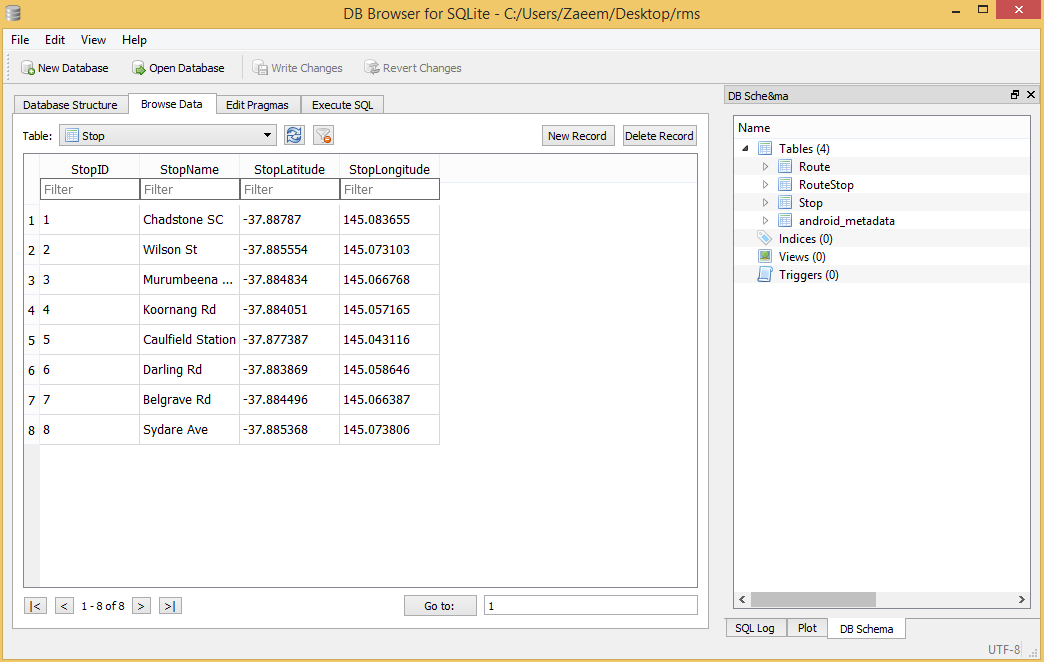
1. Chadstone SC (Start)
2. Wilson St.
3. Murumbeena Rd.
4. Koornang Rd.
5. Caulfield Station. (Finish)

## Behavior

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stop | Previous Stop | Next Stop | Reminder | Expected Output | Actual Output |
| Chadstone SC | Chadstone SC | Wilson St. | NO | Should not generate notification. Update the journey info screen, set Stops to dest to 4 and set next stop to Wilson and previous to Chadstone SC and Update the progress bar. | As expected |
| Wilson St. | Chadstone SC | Murumbeena Rd | NO | Should not generate notification. Update the journey info screen, set Stops to dest to 3 and set next stop to Murumbeena and previous to Chadstone SC and Update the progress bar. | As expected |
| Murumbeena Rd. | Wilson St. | Koornang Rd | NO | Should not generate notification. Update the journey info screen, set Stops to dest to 2 and set next stop to Koornang and previous to Murumbeena SC and Update the progress bar. | As expected |
| Koornang Rd. | Murumbeena Rd | Caulfield Station | YES | Should remind user (if the app is minimized, create notification else create dialog alert) that their destination stop is next. Start preparing, set the remaining stops to 1 and set previous and next stop to Murumbeena and Caulfield respectively, Update the progress bar. | As expected |
| Caulfield Station | Koornang Rd | Caulfield St | YES | Stop the plan, Generate notification and update the views. | As expected |

# SQlite Testing

As I was syncing the data from firebase and storing it to my local SQLite database. I needed to check if the data saved successfully and in the rite format. For that I pulled out my .dbo file from data folder (accessing from ADM device monitor) and opening it into SQLite Browser. SQLite Browser allowed me to try and execute queries before implementing them to my code.



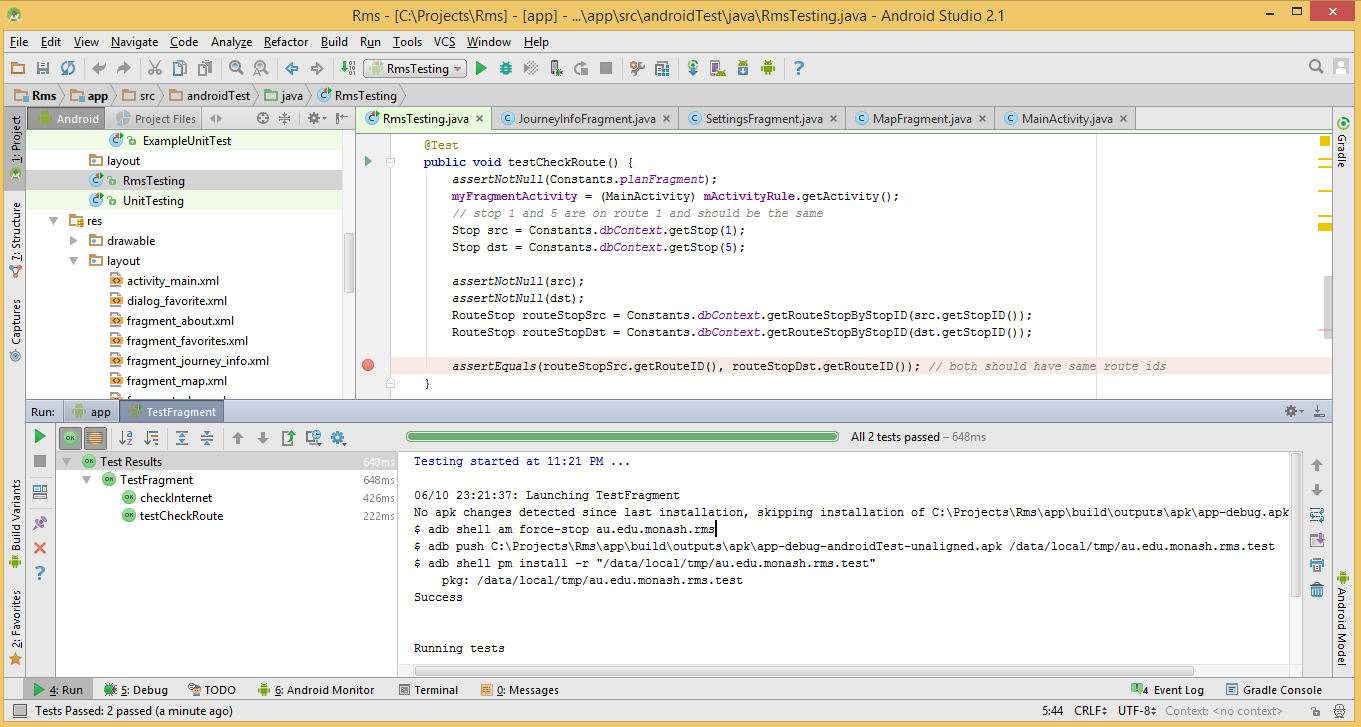
# Unit Testing

## Check whether two stops are on the same route using ID’s

This test checks whether the 2 stops i.e. src and dst lie on the same bus route or not. I have hardcoded the stop ids as 1 and 5 which lie under bus route 1 and should pass the test.

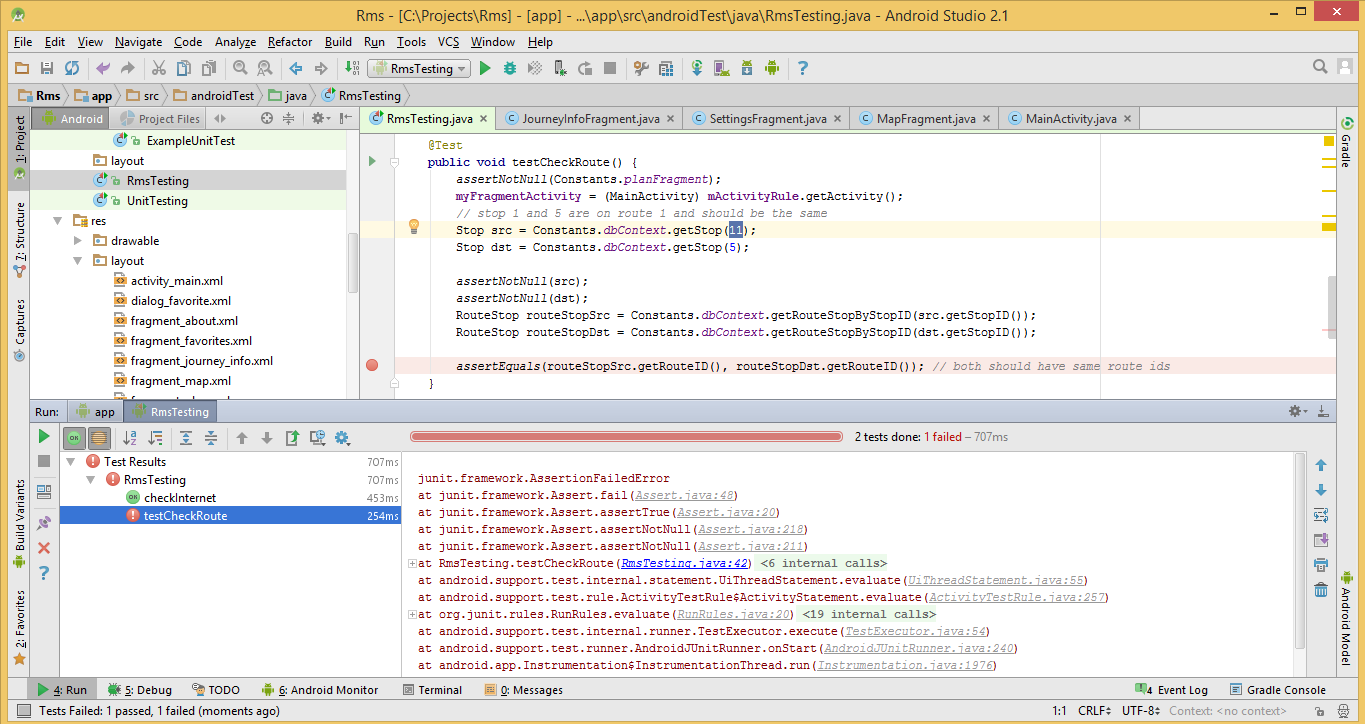
As per the Screenshot it did pass the test

Results:



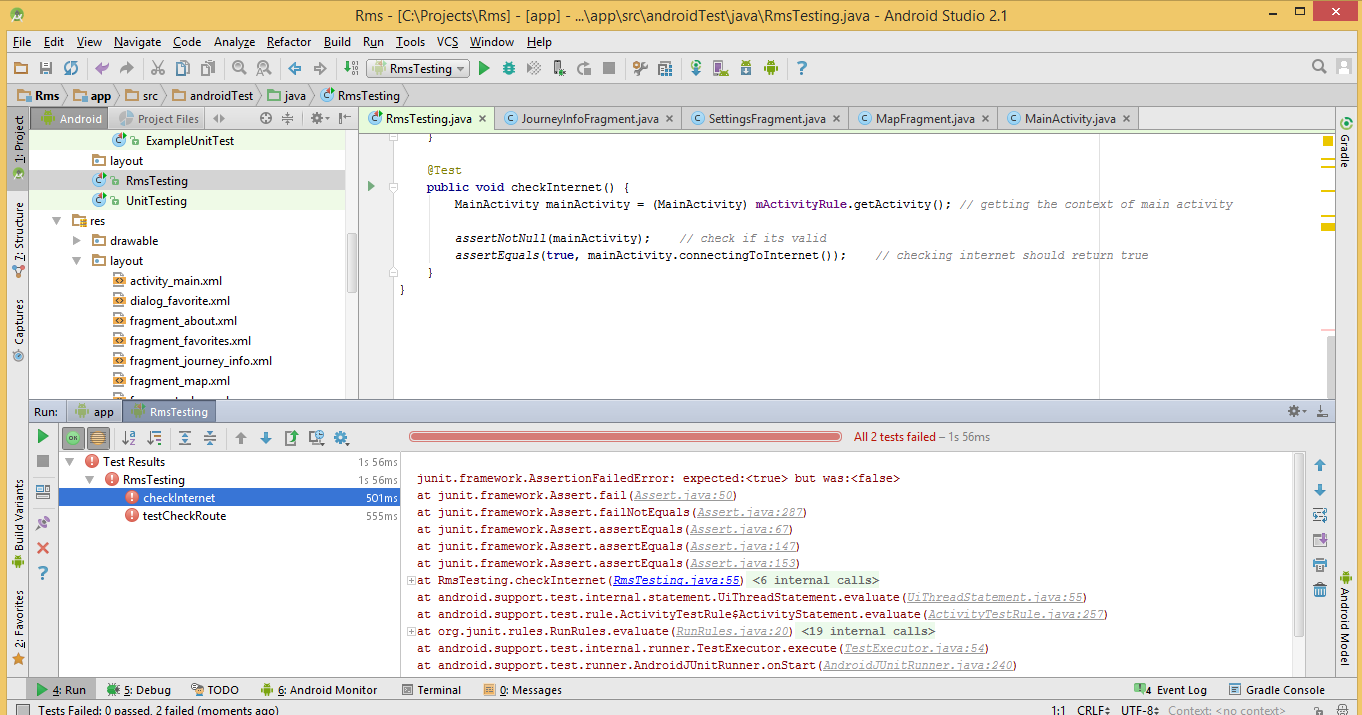
Upon changing the values of src from 1 to 11, it should fail the test.

Results:

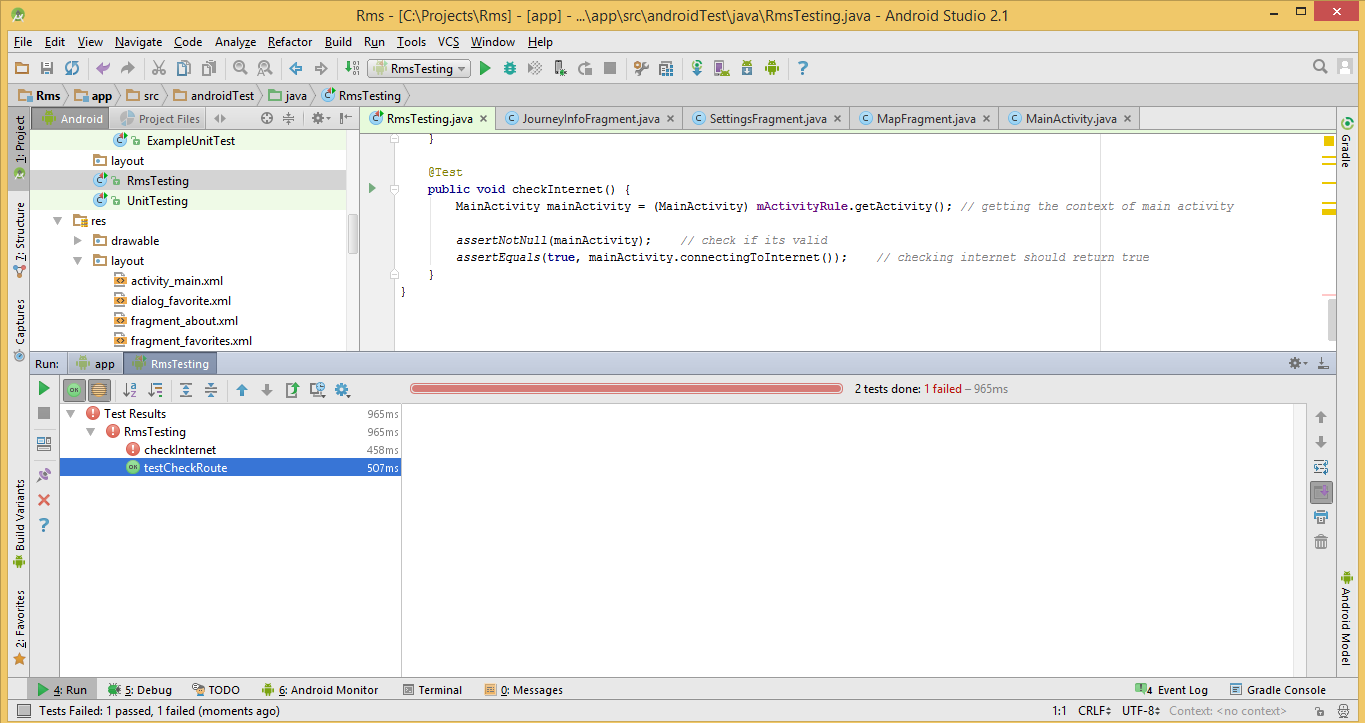


## Internet Connection

I switched off my mobile devices wifi and checked if it returned false.



Upon turning the internet on again. The test passed successfully



# References

[1] https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=en