

Bus Tracking System

University of North Texas

CSCE 5430 - Software Engineering

Fall 2016

Project Report

Bus Tracking System

**Android Developers**

Anurag Chitins

Gill Wasserman

Satyanarayana Chivukula

Nitesh Sharma

Contents

[2 User Manual 5](#_Toc468206803)

[2.1 Overview 5](#_Toc468206804)

[2.1.1 What is BusTrackingSystem? 5](#_Toc468206805)

[2.1.2 Required Equipment 5](#_Toc468206806)

[2.2 Getting Started 5](#_Toc468206807)

[2.2.1 Starting Bus Tracking System App 5](#_Toc468206808)

[2.2.2 Basic Settings 6](#_Toc468206809)

[2.2.3 Required Connections 6](#_Toc468206810)

[2.3 Using BusTrackingSystem App 6](#_Toc468206811)

[2.3.1 Selecting Route 6](#_Toc468206812)

[2.3.2 Looking at the stops 7](#_Toc468206813)

[2.3.3 Looking at schedule 7](#_Toc468206814)

[2.3.4 Tracking Bus Route 10](#_Toc468206815)

[2.3.5 Driver Login 10](#_Toc468206816)

[2.3.6 Notification 12](#_Toc468206817)

[2.4 Troubleshooting 13](#_Toc468206818)

[2.4.1 General Issues 13](#_Toc468206819)

[3 Instructions to compile and run the project 13](#_Toc468206820)

[3.1 Building and running the project 13](#_Toc468206821)

[3.2 Running the test suites 14](#_Toc468206822)

[4 Testing 16](#_Toc468206823)

[4.1 Android Studio Testing 16](#_Toc468206824)

[4.2 Firebase Test-Lab 17](#_Toc468206825)

[4.3 Group Experiences 19](#_Toc468206826)

[5 Features 19](#_Toc468206827)

[5.1 Features implemented successfully 19](#_Toc468206828)

[5.2 Features that we couldn’t complete 20](#_Toc468206829)

[5.3 Future Scope 20](#_Toc468206830)

[6 Meeting minutes 20](#_Toc468206831)

[6.1 'Android Developers' meeting minutes 20](#_Toc468206832)

[6.1.1 Agenda items 21](#_Toc468206833)

[6.1.2 Discussion: 21](#_Toc468206834)

[6.2 'Android Developers' meeting minutes 22](#_Toc468206835)

[6.2.1 Agenda items 22](#_Toc468206836)

[6.2.2 Discussion: 22](#_Toc468206837)

[6.3 'Android Developers' meeting minutes 23](#_Toc468206838)

[6.3.1 Agenda items 24](#_Toc468206839)

[6.3.2 Discussion: 24](#_Toc468206840)

[7 Member contribution table 25](#_Toc468206841)

# User Manual

## Overview

### What is BusTrackingSystem?

Your UNT bus services has gone a step forward!!

UNT Bus Tracking System is an android app developed to track the UNT bus services running on different routes and E-ride services on android compatible mobiles devices.

Bus Tracking System tracks the bus location for the route you selected and displays it on your mobile app. Bus Tracking System allows you to choose five different routes provided by the UNT bus services and e-ride services. This app allows you to choose different bus stops on a given route and lets you see the route’s active buses’ current locations. Bus Tracking System allows you to view the bus schedule of departure stops of the respective bus operating on the route.

BusTrackingSystem lets you:

* Select the route from a list of routes provided by Bus Tracking System
* Look at the stops of the selected route
* Look at the schedule of the departure bus stops of the selected route
* Track the current location of the bus of the selected route
* Driver login (used only by driver to share their location)

### Required Equipment

Bus Tracking System app operates on mobile devices using the Android operating system. It is compatible with Android API level 19 (KitKat) and higher.

## Getting Started

### Starting Bus Tracking System App

Once your Bus Tracking System app is installed successfully in your mobile device, the icon below will appear in your applications.



Start the app by tapping the above icon and your app should start successfully as no user ID or password is required.

### Basic Settings

We will provide an APK file to install for usability testing initially. In general, the user will need to install it using Google Play or another Android app store (such as the Amazon App Store). There is no user id or password required for the application to start.

### Required Connections

The Bus Tracking System app requires internet connection in order to save and retrieve data on the required database. The app also requires a working GPS receiver to receive the geographic co-ordinates (latitude and longitude coordinates) automatically, which is in turn used to display the driver’s location. Data saved can be seen on Firebase database from any internet browser. The GPS is also usedfor the normal users (non-drivers) to display their current location in relation to the bus and stops. Bus Tracking System app can be used without any further configuration.

## Using BusTrackingSystem App

### Selecting Route

As soon as the user taps on the Bus Tracking System app/icon, the route list is available to the user as shown in the Figure 1.1:

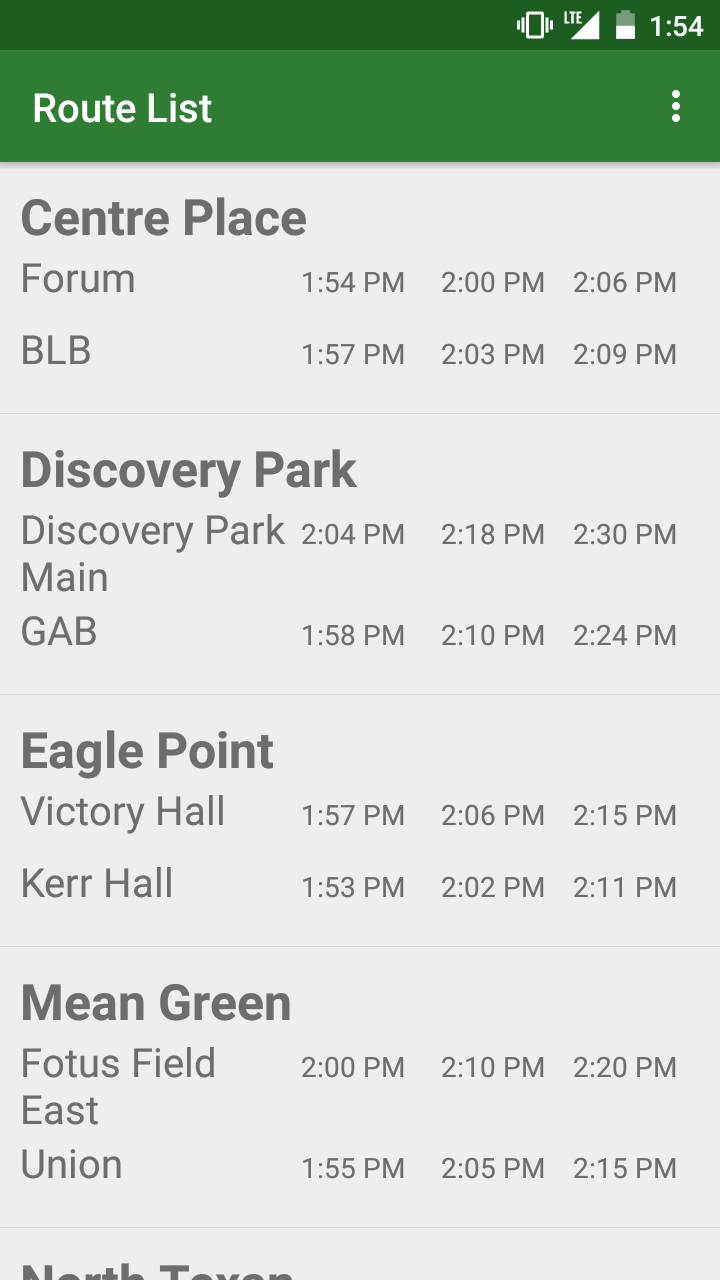
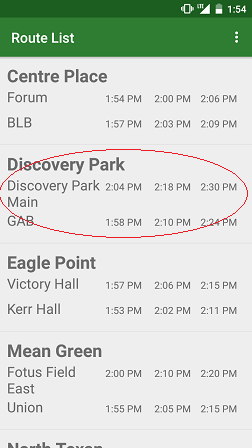
 

Figure 1.2

Figure 1.1

To select a specific route, the user only should tap on the desired route and they will be redirected to the map view.

### Looking at the stops

To view the different bus stops available on the specific route, user has to check the check box “Show Stops” displayed after the user selects a specific route as shown in Figure 1.3:

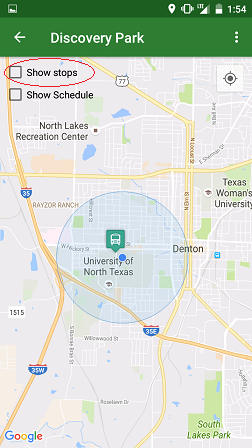
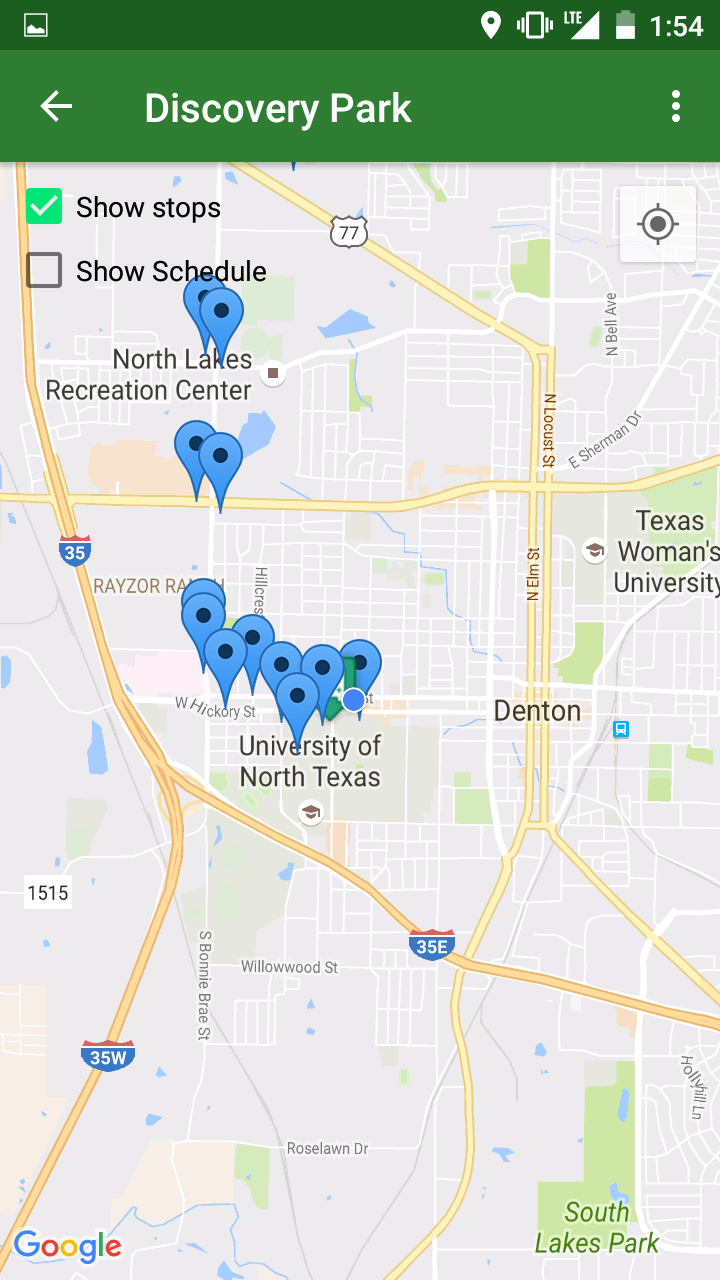


Figure 1.4

Figure 1.3

Once a user select this check box, all stops for specific route are displayed using a blue marker icon (Figure 1.4).

### Looking at schedule

To view the departure timings of the bus from the departure points of the specific route, the user should check the check box “Show Schedule” as seen in Figure 1.5:

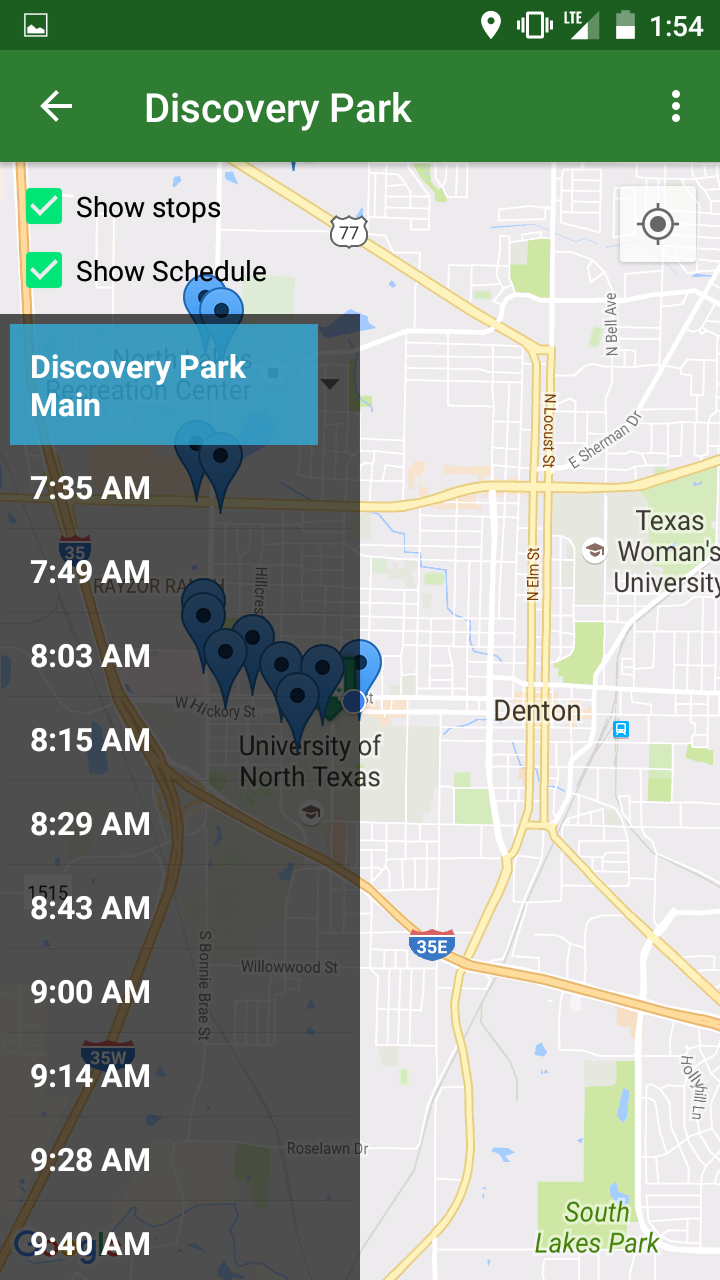
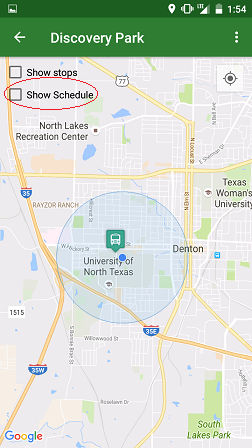


Figure 1.6

Figure 1.5

Once a user checks this box, the schedule list for the departure bus stop will be displayed. To view the schedule of a different departure stops, the user has to tap on the departure bus stop. This is a spinner that will display the other departure bus stops. Clicking on any different bus stop will refresh the schedule list with the relevant information (see Figure 1.7 below).

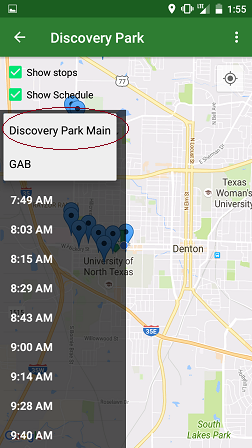


Figure 1.7

### Tracking Bus Route

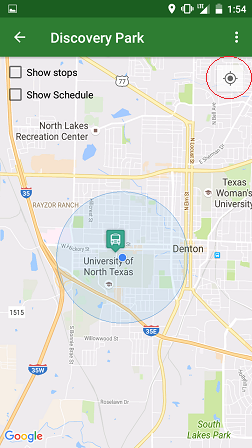
To track the bus location of the specific route user has to click on the navigation icon in the top right corner of the screen (circled in red in Figure 1.8). When the user taps the button, the application displays the current geographical location of the user (the blue dot in Figure 1.8).

Figure 1.8

### Driver Login

This module of the application is for the driver login. Once a driver logs in, the application tracks the driver location using the GPS location of the driver’s device and updates it in the Firebase database for real-time tracking.

To access this module, tap the application icon to open the app. Once the route list is displayed, tap at the right corner as shown in the Figure 1.9. The login is also accessible from the map view in the same location (top right, see Figure 1.10).

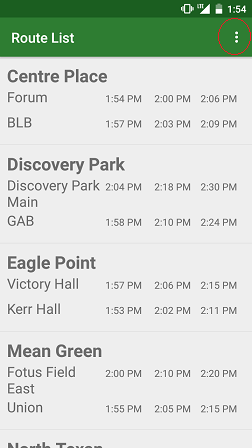
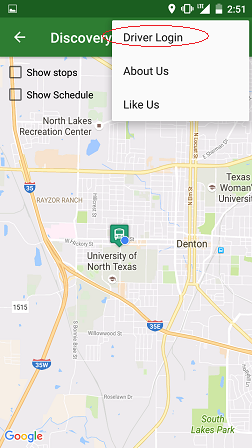
Once user taps this, it prompts for user ID (the user’s registered email) and password where only registered users can log in. Figure 1.11 shows the form that the driver will see. They will enter their user information and then select the vehicle ID of their current vehicle as well as the route that they are assigned to. These 4 fields are required to sign in and begin location tracking. Once signed in, the driver will see a notification stating that his location is being shared to the app as seen in Figure 1.12

Figure 1.10

Figure 1.9

### Notification

Figure 1.12

Figure 1.11

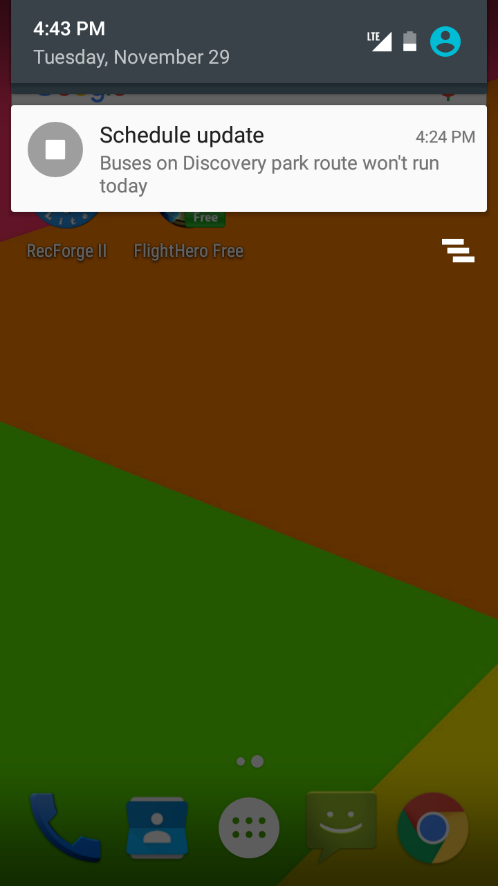
In case of emergency or other news update, the app will also push notifications to the user using the notification services within Android and Firebase. Figure 1.13 displays an example of this.

Figure 1.13

## Troubleshooting

### General Issues

* In case the app does not display any of the data, ensure that the device is connected to a working internet connection (WiFi or mobile networks).
* In case the user does not see any current location of the bus, the user should check the bus schedule and verify that the bus route is active and has stops remaining for the day.
* If you find any issues even after following the above-mentioned steps, please check your android phone version and any running apps on the device that may be interfering with network communication.

# Instructions to compile and run the project

## Building and running the project

Our project is the android application. We used Android studio for application development and white box testing. Instruction to compile and run the project using android studio are as follows:

1. Open android studio
2. Go to Files > New > Import Project. This will open the file browsing window.
3. Open the base level project directory ‘android-dev’. The project will be imported in android studio.
4. Android studio will automatically start building the project. If auto build option is off, go to Build > Make Module ‘app’. It may take several minutes to build the project.
5. Ones the project build is finished, you will see a message that Gradle build is finished at the bottom message bar.
6. Go to Run > Run ‘app’ to execute the application. You should either have a virtual device setup done to open it on emulator or an android device connected to system in USB debugging mode.
7. This operation will push the application APK file to the device and launch it.
8. ‘Play’ button in the android studio can be used to directly compile and execute the application as shown in Figure 1.

## Running the test suites

1. Right Click on the ‘androidTest’ package in the android studio and click on ‘Run <package-name>’ to execute the test cases.
2. Test cases can be executed individually by performing the above step on individual file.
3. Test cases start executing and the result gets displayed on the ‘test-cases’ window as shown in the Figure 2.
4. Instructions to run the test cases on command line can be found at <https://developer.android.com/studio/test/command-line.html>

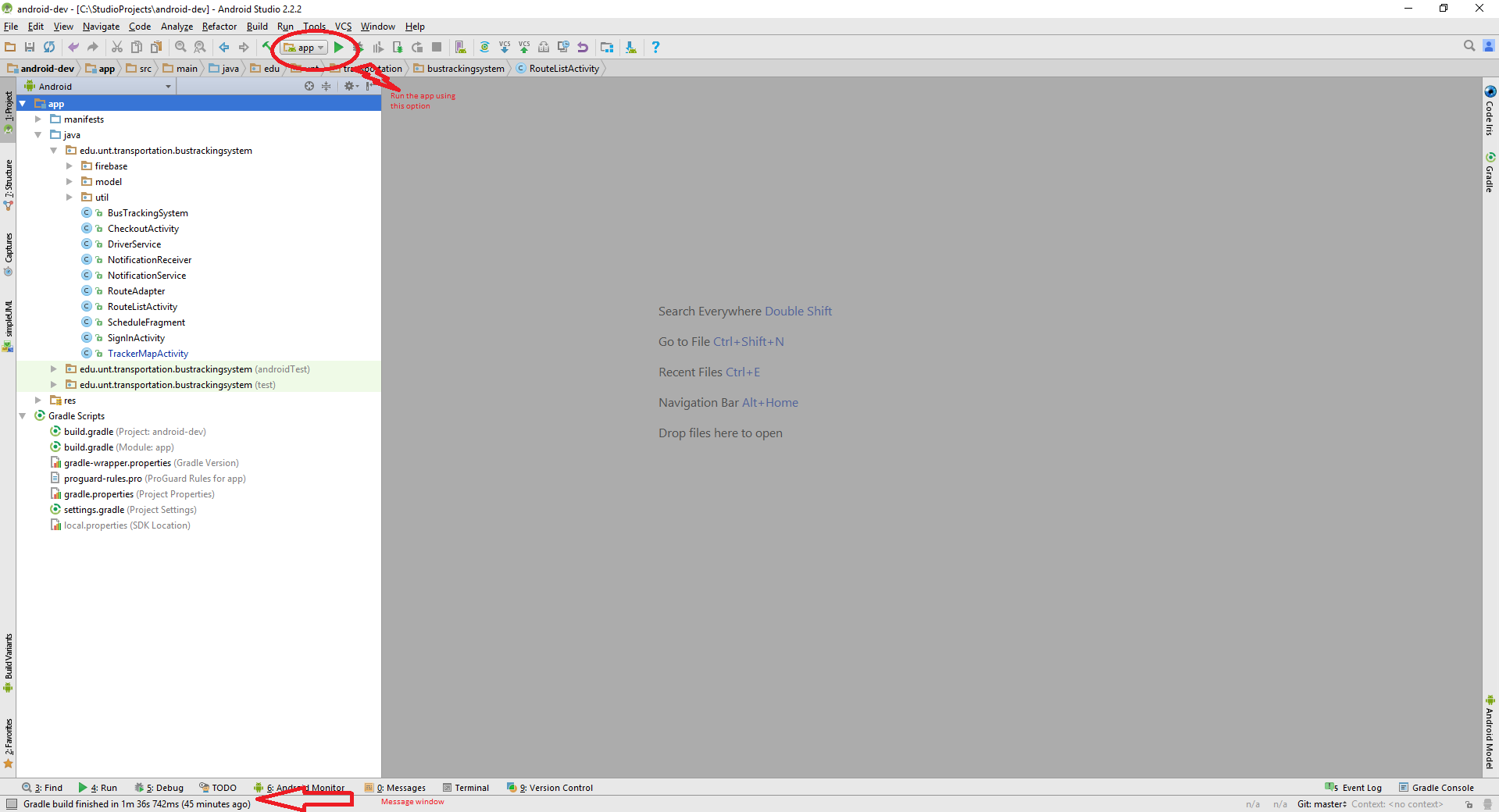


Figure 1 : Compiling and executing the Application

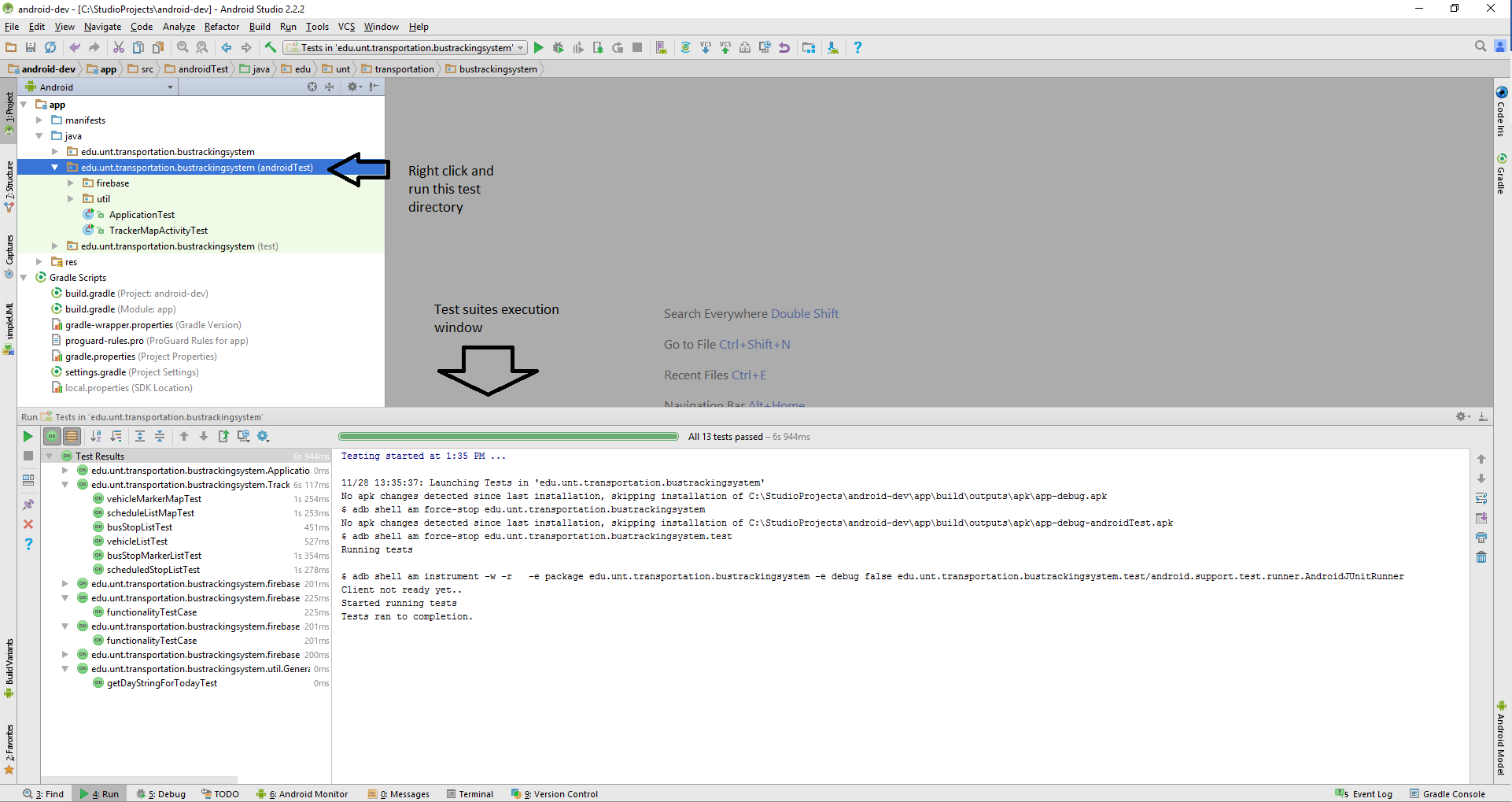


Figure 2 : Test suite execution instructions

# Testing

## Android Studio Testing

We used the Junit test case framework to test our product. In android, we have following two types of test cases available:

1. Local unit tests: These are the simple Junit test cases which are dependent just on the JVM. They are located at module-name/src/test/java/.
2. Instrumented tests: These are the test cases which need android context and other android resources to execute. They are located at module-name/src/androidTest/java/.
3. In our project, we have used instrument test cases as all our test scenarios require android context or android resources.

## Firebase Test-Lab

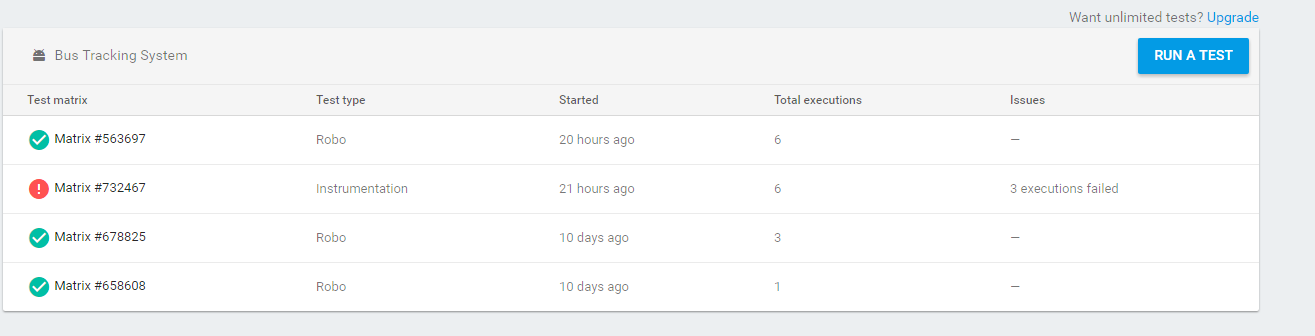
* We are using Firebase test lab to test our product as well.
* Firebase has provided the option of Robo-test, where we can run the automated test scenarios on various devices of different android versions. This provides us with the feedback from huge number of devices, which is generally difficult to test with in real.
* We have test suite running option in firebase test-lab as well, where we can run our test cases on the array of devices. We just need to upload the APK of our test files and our application APK.
* When we specify the devices and the platform version we want to run the test cases on, it starts on its own.
* The following are figures demonstrate our usage of the Firebase Test-Lab
  + 

Figure 3.1: Test Lab Start Screen

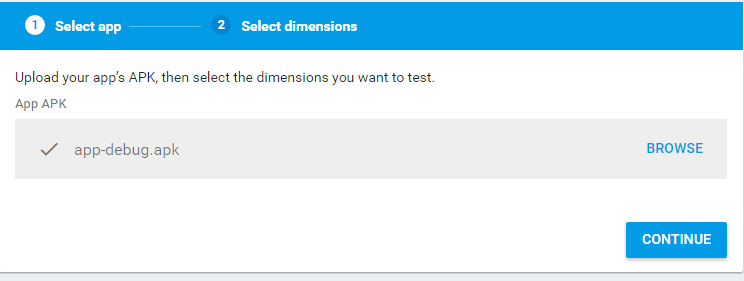
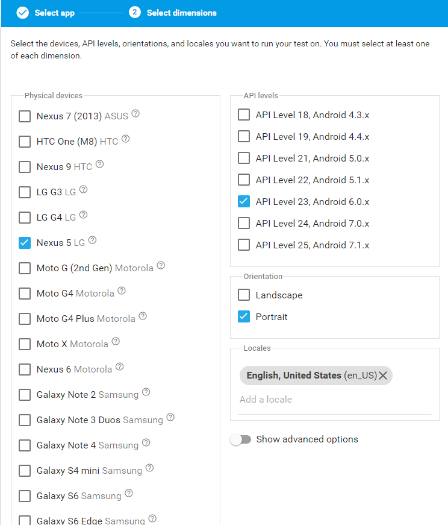
* + 

Figure 3.2 and Figure 3.3: Selecting app APK and test environment

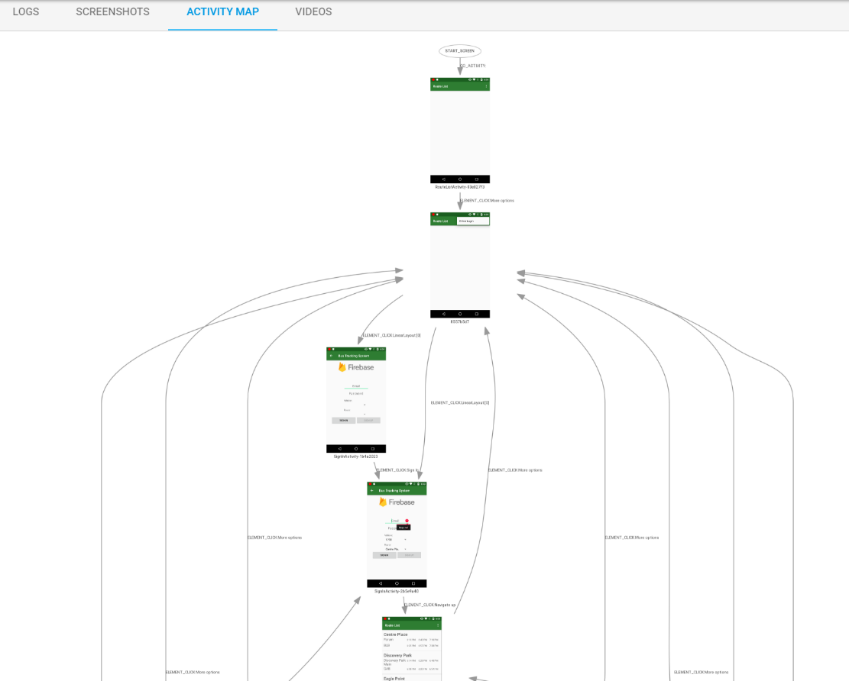
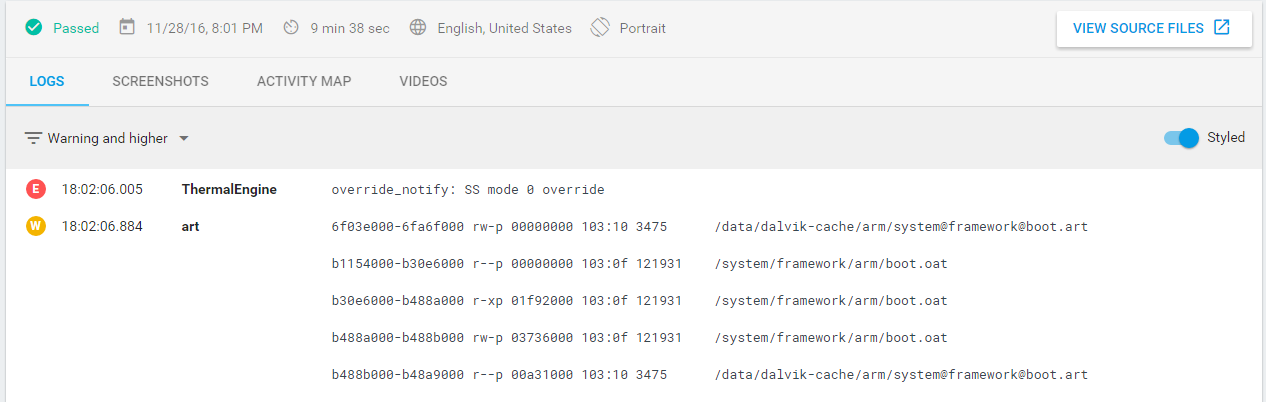
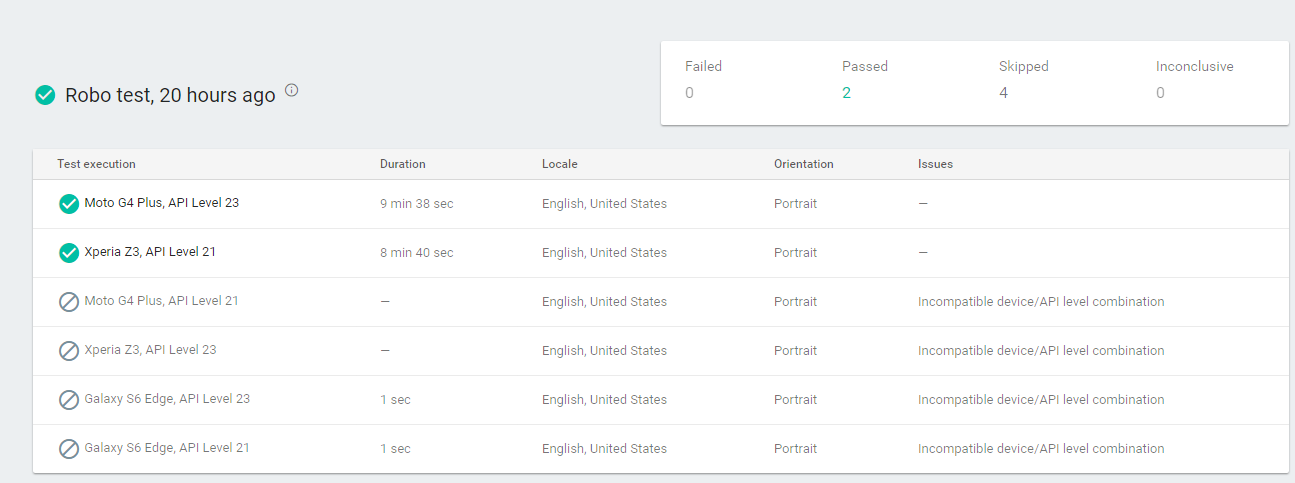


Figure 3.4: Results of RoboTest

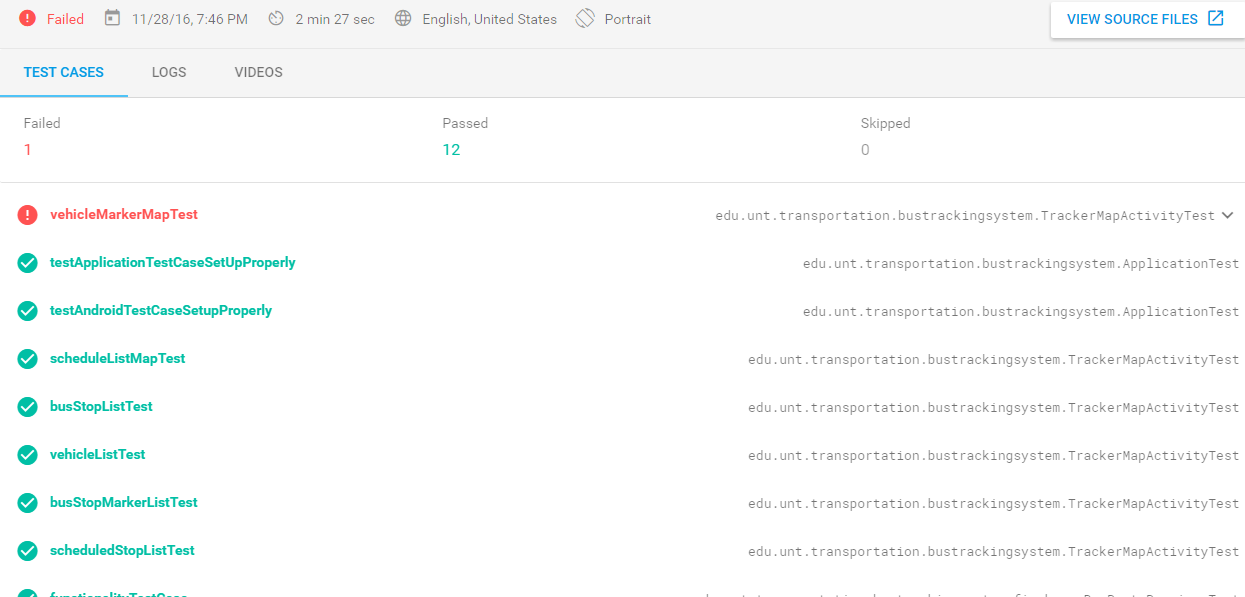
Figure 3.1 shows the starting page for the Test Lab. It displays all test matrices that have been set up, displaying whether they are pending, succesfully completed, or completed with errors. Figure 3.2 and 3.3 show the simple interface used to set up the environment and apk to test with. The result of these tests can be seen in Figure 3.4 and Figure 3.5. The Test Lab returns all system logging information, including the error level, as well as displays screenshots and videos of the automated testing process. The Test Lab also shows an activity map that shows the navigation tree that the test case followed and the method it used to navigate to each location. The instrumentation test matrix shows the resulting jUnit tests that we created, running them on the previously selected test devices.

Figure 3.5: Instrumentation Test Cases

## Group Experiences

* We found bugs regarding vehicle id and route id discrepancies when testing multiple routes and bus together.
* While updating the latest location, service is getting destroyed, so when testing with the group, we changed IntentService to Service in background thread.

# Features

## Features implemented successfully

* Displaying of all the available routes in the form of List View.
* Display of the scheduled next three stops on each route (This feature helps user to see the next buses on the same screen without having him to press anymore buttons)
* Display all the buses running on the route (clicked by user) on Google Map.
* Display user’s location when user clicks on ‘my location’ icon on the screen.
* User can click on the location of bus stop and click on directions option provided by google map to navigate to that bus stop.
* Displays the bus stops on the selected route when user clicks on ‘show stops’
* Displays schedule of the buses on the day of use of application.
* Offline Caching: We cache the data ones it is loaded so that the user will be able to see the static data like schedule and bus stop location even when internet connection is not available.
* Driver Login: Driver can login into the application using the provided credentials. He must provide the ‘bus number’ which he is driving and the route on which he is driving.
* Shows notification to the driver on notification bar that the location is being shared with everyone with an option to logout which stops sharing of location.
* Transportation services can send notification to the users about any changes in the schedule on some special days, route change notification, etc.
* Transportation services can add new routes, stops, buses etc. form the firebase database. So, that user will see those changes without having to update the application.

## Features that we couldn’t complete

* Highlighting the selected route on the google map, because of data points (locations), collecting them goes beyond the scope of the project.
* Search feature: A search feature for user to reach to a destination taking number of buses.
* Drawing a line along the route from users’ location to nearest bus stop.

## Future Scope

* Addition of ‘on screen widget’ to display the schedule of favorite routes on the welcome screen.
* Implementation of search feature, where user will be able to enter the destination and navigate to it using the available bus routes.
* Display the detailed schedule of all the days, routes and stops in a different activity.
* Show the list of received notifications.
* Request E-ride: User should be able to request e-ride from the application.
* Showing up multiple routes on the map simultaneously.

# Meeting minutes

## 'Android Developers' meeting minutes

|  |  |
| --- | --- |
| Location: | Discovery Park NTDC B157 |
| Date: | 11/15/2016 |
| Time: | 10:40 a.m. |
| Author: | Anurag Chitnis |
| Attendees: | Dr. Hyunsook Do, Anurag Chitnis, Nitesh Kumar Sharma, Satyanarayana Chivukula, Gil Wasserman |

### Agenda items

1. Discussion on the progress of project
2. Assigning new task
3. Bug Fixing

### Discussion:

* Discussed about the Driver Activity, in our current implementation we have Driver Activity which driver gets to see after he logs in. We decided that we don’t need that activity anymore and we will use the service instead which will run in background.
* Decided to use Notification when the driver logs in and provide option for driver in notification bar, so that he can stop sharing his location.
* Discussed the UI for Route List Activity. Finalized that, it will display the Route title along with the names of scheduled bus stops and next 3 stop timings for each stop.
* Discussed about the time schedule fragment. Gill will start working on schedule fragment this week.
* Discovered following bugs in the current implementation. Decided to fix them.
  + Title on each Activity is not appropriate
  + Schedule list does not show schedule of some routes.
  + Crash observed when schedule list is empty.
* Addition of unit test cases for each module. Decided to use expresso to simulate UI interaction in android.

| **Sr. No.** | **Action items** | **Owner(s)** | **Deadline** | **Status** |
| --- | --- | --- | --- | --- |
| 1 | UI enhancement of Route List Activity | Gil | 11/21/2016 | Completed |
| 2 | Addition of new module to show the list of schedule of all days and all stops. | Gil | 11/21/2016 | Completed |
| 3 | Implementation of Service which will replace the Driver Activity. | Satya | 11/21/2016 | Completed |
| 4 | Implementation of Notification bar to show the status of location sharing to driver. | Satya | 11/21/2016 | Completed |
| 5 | Addition of Unit Test cases. | Nitesh, Gill, Satya | 11/21/2016 | Completed |
| 6 | Addition of comments and code refactoring | Nitesh, Anurag | 11/21/2016 | Completed |
| 7 | Code Review | Nitesh, Anurag | 11/21/2016 | Completed |
| 8 | Addition of feature to show the schedule of the day on google map. The schedule should be of the day when app is being used. | Anurag | 11/21/2016 | Completed |

## 'Android Developers' meeting minutes

|  |  |
| --- | --- |
| Location: | Discovery Park NTDC B157 |
| Date: | 11/17/2016 |
| Time: | 10:40 a.m. |
| Author: | Nitesh Kumar Sharma |
| Attendees: | Dr. Hyunsook Do, Anurag Chitnis, Nitesh Kumar Sharma, Satyanarayana Chivukula, Gil Wasserman |

### Agenda items

1. Discussion on the progress of project
2. Reviewing assigned task
3. Bug Fixing

### Discussion:

* Discussed and replaced the driver activity so that it runs in background as against the proposed implementation to achieve the desired results.
* Implemented the notification module so that driver can logout whenever he wants to logout and the latest location of the driver is saved.
* Implemented and formatted the UI for Route List Activity. Implementation of display the Route title along with the names of scheduled bus stops and next 3 stop timings for each stop.
* Discussed about the time schedule fragment. Gill will start working on schedule fragment this week and is Completed.
* Discovered following bugs in the current implementation. Decided to fix them and Completed and decided to finish the coming weekend by team.
  + Title on each Activity is not appropriate
  + Schedule list does not show schedule of some routes.
  + Crash observed when schedule list is empty.
* Addition of unit test cases for each module. As of now few test cases are implemented using espresso.

| Sr. No. | Action items | Owner(s) | Deadline | Status |
| --- | --- | --- | --- | --- |
| 1 | UI enhancement of Route List Activity | Gil | 11/21/2016 | Completed |
| 2 | Addition of new module to show the list of schedule of all days and all stops. | Gil | 11/21/2016 | Completed |
| 3 | Implementation of Service which will replace the Driver Activity. | Satya | 11/21/2016 | Completed |
| 4 | Implementation of Notification bar to show the status of location sharing to driver. | Satya | 11/21/2016 | Completed |
| 5 | Addition of Unit Test cases. | Nitesh, Gill, Satya | 11/21/2016 | Completed |
| 6 | Addition of comments and code refactoring | Nitesh, Anurag | 11/21/2016 | Completed |
| 7 | Code Review | Nitesh, Anurag | 11/21/2016 | Completed |
| 8 | Addition of feature to show the schedule of the day on google map. The schedule should be of the day when app is being used. | Anurag | 11/21/2016 | Completed |

## 'Android Developers' meeting minutes

|  |  |
| --- | --- |
| Location: | Discovery Park NTDC B157 |
| Date: | 11/15/2016 |
| Time: | 10:40 a.m. |
| Author: | Gil Wasserman |
| Attendees: | Dr. Hyunsook Do, Anurag Chitnis, Nitesh Kumar Sharma, Satyanarayana Chivukula, Gil Wasserman |

### Agenda items

* Discussion on the progress of project
* Assigning new task
* Bug Fixing

### Discussion:

* Discussed about the Driver Activity, in our current implementation we have Driver Activity which driver gets to see after he logs in. We decided that we don’t need that activity anymore and we will use the service instead which will run in background.
* Decided to use Notification when the driver logs in and provide option for driver in notification bar, so that he can stop sharing his location.
* Discussed the UI for Route List Activity. Finalized that, it will display the Route title along with the names of scheduled bus stops and next 3 stop timings for each stop.
* Discussed about the time schedule fragment. Gill will start working on schedule fragment this week.
* Discovered following bugs in the current implementation. Decided to fix them.
  + Title on each Activity is not appropriate
  + Schedule list does not show schedule of some routes.
  + Crash observed when schedule list is empty.
* Addition of unit test cases for each module. Decided to use expresso to simulate UI interaction in android.

| **Sr. No.** | **Action items** | **Owner(s)** | **Deadline** | **Status** |
| --- | --- | --- | --- | --- |
| 1 | UI enhancement of Route List Activity | Gil | 11/21/2016 | Completed |
| 2 | Addition of new module to show the list of schedule of all days and all stops. | Gil | 11/21/2016 | Completed |
| 3 | Implementation of Service which will replace the Driver Activity. | Satya | 11/21/2016 | Completed |
| 4 | Implementation of Notification bar to show the status of location sharing to driver. | Satya | 11/21/2016 | Completed |
| 5 | Addition of Unit Test cases. | Nitesh, Gill, Satya | 11/21/2016 | Completed |
| 6 | Addition of comments and code refactoring | Nitesh, Anurag | 11/21/2016 | Completed |
| 7 | Code Review | Nitesh, Anurag | 11/21/2016 | Completed |
| 8 | Addition of feature to show the schedule of the day on google map. The schedule should be of the day when app is being used. | Anurag | 11/21/2016 | Completed |

# Member contribution table

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note  (if applicable) |
| Anurag Chitnis | Project: TrackerMapActivity class, SignIn Activity, Code Integration, Firebase controller classes, Test cases  Document: Features, Testing, Compile and Running instructions | 26.66% |  |
| Gil Wasserman | Project: Route List Activity, Firebase model classes, test cases  Document: Testing, User Manual | 26.66% |  |
| Satyanarayana Chivukula | Project: Location Sharing service, SignIn Activity, Notification Receiver, Persistent Notification  Document: Landing page document, Table of contents | 26.66% |  |
| Nitesh Kumar Sharma | Project: About Us Page, Facebook page, Dataset creation, Test cases  Document: User Manual | 20% |  |