**A Minor Project Report on**

**“HOVER Games”**

**Submitted to**



**Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal**

**Towards Partial Fulfillment for the Award of**

**Bachelor of Engineering**

**(Computer Science and Engineering)**

**Submitted By:**

**Udit Sen, Vaibhav Bhawalkar, Vinay Yadav**

**CS Third Year**

**Guided By**

**Ms. KavitaNamdev**

****

**Department of Computer Science and Engineering**

**Acropolis Institute of Technology and Research, Indore**

**January-June 2017**

**Acropolis Institute of Technology and Research, Indore**

**Department of Computer Science and Engineering**

**Certificate**

We, hereby declare that the project entitled “HOVER GAMES” submitted to **department of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore (M.P.)** in BE VI semester in session June 2018 is an authentic record of our own work carried out under the guidance of **Mr. Ritesh Khedekar** and that the project has been developed by us and not previously formed the basis for the award of any other degree.

This is to certify that the above statement made by the student is correct to the best of my knowledge.

**Internal Examiner External Examiner**

Date: Date:

Place: Indore

**Acropolis Institute of Technology and Research, Indore**

**Department of Computer Science and Engineering**

**PROJECT APPROVAL SHEET**

The project work entitled **“HOVER Games”** submitted by Udit Sen (0827CS161233), Vaibhav Bhawalkar (0827CS161236) and Vinay Yadav (0827CS161242)andis approved as partial fulfillment for the award of the **Bachelor of Engineering (Computer Science and Engineering)** degree by **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P).**

**Date: Mr. Ritesh Khedekar**

**Project Guide**

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

**Mr. Deepak Agrawal**

**Project Coordinator**

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

**Prof. Sanjay Bansal**

**Head of Department**

Computer Science & Engineering Department Acropolis Institute of Technology & Research, Indore

**Acropolis Institute of Technology and Research, Indore**

**Department of Computer Science and Engineering**

**STUDENT DECLARATION**

We the student of **Bachelors of Engineering** (Computer Science and Engineering) 2017, hereby declare that the work presented in this project synopsis entitled “**HOVER Games**” submitted towards completion of Minor Project in 6th semester of B.E. (Computer Science and Engineering) at Acropolis Institute of Technology & Research, Indore, is an authentic record of our own work. Due acknowledge have been made in the text to all other material used. The project was done in full compliance with the requirement and constraints of the prescribed curriculum.

Date: Udit Sen

(0827CS161233)

Vaibhav Bhawalkar

(0827CS161236)

Vinay Yadav

(0827CS161242)

**ACKNOWLEDGEMENT**

We would like to avail this opportunity to express our sincere thanks to all those who have helped us in making this project, even a most vivid collection of words yield to express our heart fully thank towards one and all who have successfully assisted us in our expenditure of carrying out this project.

We deeply indebted to our project guide **Mr. Ritesh Khedekar** who generously shared his wisdom and expertise with us, both as our project guide and teacher for providing us his valuable guidance and suggestions.

We express our hearty gratitude to **Prof. Sanjay Bansal, Professor & Head, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for his support, and help provided during the project tenure.

We also express our deep gratitude to **Dr. S.C. Sharma, Director, Acropolis Institute of Technology & Research, Indore** for all the help they provided for the completion of project.

We wish to express our deep sense of gratitude to **Mr. Deepak Agrawal** and the whole faculty members of the Computer Science Department for encouraging and giving moral support, not only regarding this project but also throughout our studies at the institute.

Last but not the least we place our deep sense of gratitude to our family and friends who have been constant source of inspiration and support during the preparation of this project work.

Udit Sen

(0827CS161233)

Vaibhav Bhawalkar

(0827CS161236)

Vinay Yadav

(0827CS161242)

**ABSTRACT**

There are buses made available for students, teachers and other staff members, but not many of those have complete information about these buses. Complete information about the bus arrival at their stop and real-time bus tracking through maps would help students, teachers and other staff members to track the current location of the bus and give the correct time for the bus to reach its bus stop. The proposed system deals with overcoming the problems stated above. The platform chosen for this kind of system is Android, reason being Android Operating System has come up on a very large scale and is owned by almost every second person. Also, Android is a user friendly platform, thereby enabling ease of access for all the users.

“HOVER GAMES”, runs on android smartphones and tablets. This proposed system allows organization to track their vehicles and to get exact location of vehicle. The system also allows those companies to monitor the travelled routes through a web client that uses the Google Maps API and shows colored markers on the map to indicate stops on route. This application saves the time of user who wants to travel through the bus. This application offers advantage to the users to retrieve information about the bus location and also to track the location of nearby buses. The location of bus would be displayed on the map using its coordinated (longitude and latitude).

**TABLE OF CONTENTS**

**1. INTRODUCTION 1**

**1.1 PROJECT OVERVIEW 1**

**1.2 PROJECT DELIVERABLES 1**

**2. PROJECT ORGANIZATION 2**

**2.1 SOFTWARE PROCESS MODEL 2**

**2.2 ROLES AND RESPONSIBILITIES 2**

**2.2.1 ADMIN ROLES 2**

**2.2.2 USER ROLES 3**

**2.3 TOOLS AND TECHNIQUES 3**

**3. PROJECT MANAGEMENT PLAN 3**

**3.1 TASKS 3**

**3.2 INFORMATION GATHERING 3**

**3.3 RESOURCES NEEDED 3**

**3.4 DEPENDENCIES AND CONSTRAINTS 4**

**3.5 RISKS AND CONTINGENCIES 4**

**4. GANTT CHART 5**

**5. INTRODUCTION 6**

**5.1 PRODUCT OVERVIEW 6**

**6 SPECIFIC REQUIREMENTS 6**

**6.1 EXTERNAL INTERFACE REQUIREMENTS 6**

**6.1.1 USER INTERFACES 6**

**6.1.2 SOFTWARE INTERFACES 6**

**6.1.3 COMMUNICATION PROTOCOLS 8**

**6.2 SOFTWARE FUNCTIONAL REQUIREMENTS 8**

**6.3 SOFTWARE NON-FUNCTIONAL REQUIREMENTS 9**

**6.3.1 RELIABILITY 9**

**6.3.2 AVAILABILITY 9**

**6.3.3 SECURITY 9**

**6.3.4 MAINTAINABILITY 9**

**6.3.5 PORTABILITY 9**

**6.3.6 PERFORMANCE 9**

**7.1 DESIGN OVERVIEW 11**

**7.1.1 DATA FLOW DIAGRAM 11**

**7.1.2 USE CASE DIAGRAM 13**

**7.1.3 CLASS DIAGRAM 14**

**7.1.4 ACTIVITY DIAGRAM 15**

**7.1.5 SEQUENCE DIAGRAM 19**

**7.2 DATABASE DSIGN 22**

**7.2.1 ENTITY RELATION DIAGRAM 22**

**8 SYSTEM ARCHITECTURE DESIGN 23**

**8.1 CHOSEN SYSTEM ARCHITECTURE 23**

**9 DETAILED DESCRIPTION OF COMPONENTS 23**

**10 USER INTERFACE DESIGN 23**

**10.1 DESCRIPTION OF USER INTERFACE 23**

**10.2 SCREEN IMAGES 26**

**11. INTRODUCTION 30**

**11.1 SYSTEM OVERVIEW 30**

**11.2 TEST APPROACH 30**

**11.3 TESTING OBJECTIVES 31**

**12. TEST PLAN 32**

**12.1FEATURES TO BE TESTED 32**

**12.2 FEATURES NOT TO BE TESTED 33**

**13. TEST CASES 33**

**13.1 UNIT TESTING 34**

**13.2 FUNCTIONAL TESTING 35**

**13.3 SYSTEM TESTING 35**

**13.4 INTEGRATION TESTING 36**

**13.5 VALIDATION TESTING 36**

**13.6 TOP DOWN INTEGRATION 37**

**13.7 BOTTOM UP INTEGRATION 37**

**14. CONCLUSION AND FUTURE SCOPE 38**

**15. REFERENCES 38**

**16. BIBLIOGRAPHY 38**

**LIST OF FIGURES**

**LIST OF TABLES**

**LIST OF ABBREVIATIONS**

1. **INTRODUCTION**
   1. **PROJECT OVERVIEW**

“HOVER GAMES” is an android application. This software is used to monitor the route followed by bus and to track the current location of bus. It can be configured to meet the requirements of any school transport system to manage their total institution transport management. This application uses GPS for locating the position of vehicle. We can also calculate distance travelled by bus in x seconds via its coordinates, which will help in identifying the speed of the vehicle in real time. Vehicles are tracked through android application using GPS to find out where a bus is. To start sending location to server, bus driver needs to login via his credentials.

Features of “HOVER Games”:

1. Bus Drivers application can run in background i.e. it doesn’t disturb his other activities like making or receiving calls, sending SMS, MMS and etc.
2. List of all bus stops nearby to user are displayed, so that if he misses his bus then he can get another bus from nearby stop.
3. This application displays the route of bus on Google Maps using its API V2 with markers on every stop on its route.
4. Changes of bus route are also shown to users.
   1. **PROJECT DELIVERABLES**

|  |  |
| --- | --- |
| DELIVERABLES | STATUS |
| Background Research | Done |
| Project Proposal | Done |
| Product Definition | Done |
| Alternative Analysis | Done |
| Prototype Design | Done |
| List of HW/SW components | Done |
| Cost/Implementation Analysis | Done |
| Benefits Analysis | Done |
| Report | Done |
| Presentation | Done |
| Application | Done |

Table 1.2 (a): - Project Deliverables

1. **PROJECT ORGANIZATION**
   1. **SOFTWARE PROCESS MODEL**

The software model used in our project is Incremental model.

The incremental model applies the waterfall model incrementally. The series of releases is referred to as “increments”, with each increment providing more functionality to the customers. After the first increment, a core product is delivered, which can already be used by the customer. Based on customer feedback, a plan is developed for the next increments, and modifications are made accordingly. This process continues, with increments being delivered until the complete product is delivered. The incremental philosophy is also used in the agile process model.

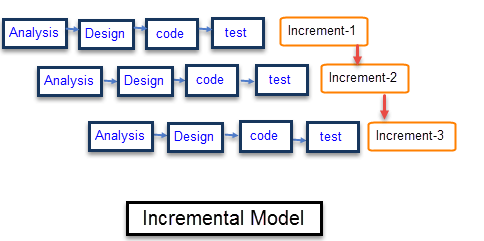


Figure 2.1 (a): - Incremental Model

* 1. **ROLES AND RESPONSIBILITIES**
     1. **ADMIN ROLES**

1. Add or delete bus routes.
2. Add or delete bus stops.
3. Add or delete bus numbers.
4. Provide login credentials to bus drivers.
5. Manage bus active sessions.
   * 1. **USER ROLES**
6. Registration with valid e-mail address.
7. Select his/her stop from the list of jobs.
8. Track the location of bus via bus number.
   1. **TOOLS AND TECHNIQUES**

The software tools used in our project are as follows:

1. Android Studio
2. NoSQL Database
3. Firebase
4. Google Cloud Function
5. **PROJECT MANAGEMENT PLAN**
   1. **TASKS**
      1. The major tasks are:
      2. Analysis of major deliverables
      3. Storing the data in database
      4. Documentation and modelling
      5. Approval to documentation.
      6. Creating a user interface
      7. Coding for interface
      8. Testing the project
   2. **INFORMATION GATHERING**

The major information that was gathered for HOVER Games:

1. Bus Management Department
2. Students Particulars
3. Analytics on different bus routes and schedules
4. Administrative Rights that are desired by Institute Authorities
   1. **RESOURCES NEEDED**

**HARDWARE**

1. Operating System: Android API 17 or higher
2. Hard disk: 200 MB or higher
3. RAM: 512 MB or higher
4. Processor: 1GHz or higher

**SOFTWARE**

1. Android Studio
2. NoSql Database
3. Firebase
4. Google Cloud Function
   1. **DEPENDENCIES AND CONSTRAINTS**

**DEPENDENCIES:**

* 1. The application must need internet while displaying bus routes. It must be done in online mode.
  2. All users must have Google Play Services 9.0+ installed.
  3. The platform used must be Android only.

**CONSTRAINTS:**

The users are the students, teachers and other staff. It is considered that the users have the basic knowledge of operating an android smartphone.

* 1. **RISKS AND CONTINGENCIES**

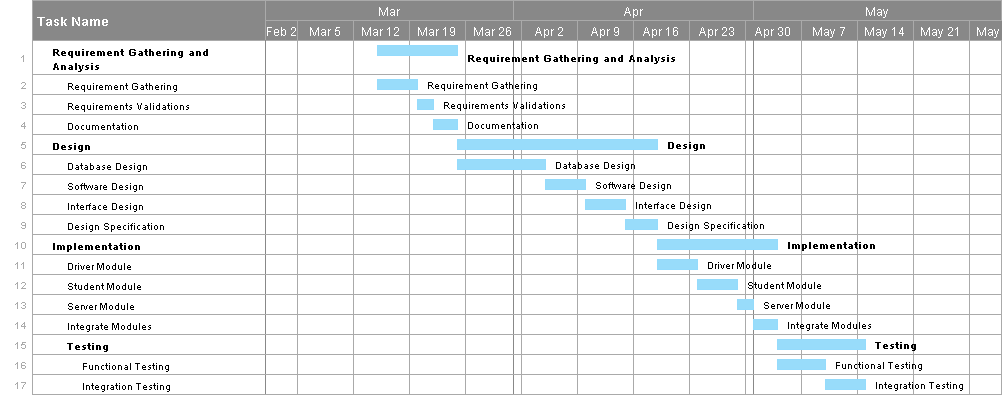
1. Slow Internet
2. Improper functioning of GPS
3. Non availability of Network
4. **GANTT CHART**

Figure 4 (a): - Gantt Chart

1. **INTRODUCTION**
   1. **PRODUCT OVERVIEW**

Android is the latest and a rapid growing technology available for all the users or customers in today’s market. An enormous increase in the end user acceptance has been experienced in the past few years. This project has been developed on the Bus Information System in AITR, Indore. This paper proposes an Android mobile phone application that gives information about buses, bus numbers as well as bus routes. Reason for Android platform - Android requires an open source development which is probably the most feasible and a present user friendly approach. This paper also deals with Location Based Services, which are used to track the current location of the bus as well as give an estimate remaining time for the tracked bus to reach its destination using the Client-Server technology. Also, it displays the required maps with the help of GPS. The main goal of the proposed work is to improve the Bus system by adding the necessary additional features into the application, like accurate bus timings, correct bus numbers and moreover adding a GPS tracker into it.

1. **SPECIFIC REQUIREMENTS**
   1. **EXTERNAL INTERFACE REQUIREMENTS**
      1. **USER INTERFACES**

The user interface is provided to user by the Android application where user can easily track the bus and can get real-time location of bus via its coordinates. Users have to Login to access the data from server. Route or location of bus will be displayed on Google Maps via its API V2.

* + 1. **SOFTWARE INTERFACES**

1. **Android Studio:**

Android Studio is the official integrated development environment (IDE) for the Android platform. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0.Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, macOS and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.



Figure 6.1.2 (a) Android Studio

1. **NoSQL Database:**

A NoSQL database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. NoSQL Database document pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents. Graph stores are used to store information about networks of data, such as social connections.

1. **Firebase:**

Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs.



Figure 6.1.2(b) Firebase

1. **Google Cloud Function:**

Cloud Functions provides a connective layer of logic that lets you write code to connect and extend cloud services. Listen and respond to events such as a file upload to Cloud Storage, an incoming message on a Cloud Pub/Sub topic, a log change in Stackdriver Logging, or a mobile-related event from Firebase. Cloud Functions augments existing cloud services and allows you to address an increasing number of use cases with event-driven code.

**6.1.3 COMMUNICATION PROTOCOLS**

The Communication between different parts of the system is important since they depend on each other. However, the way in which communication is achieved is not important for the functioning of system and is therefore handled by the underlined architecture for server and mobile application.

* 1. **SOFTWARE FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a system or its components. A function is described as a set of inputs, the behavior and outputs.

INPUTS:-

1. User would provide his stop details.
2. User would enter the bus number whose real-time location is to be found.

PROCESSING:-

1. Maintain Session Table:

A function will run on server using Google Cloud Function which will listen to any changes in database due to location send by driver application and will calculate speed of bus and estimated time to reach different stops on its route.

1. Send Location:

Return the speed of bus with its current location and estimated time to reach different stops on its route.

1. OUTPUT:-

Display the result obtained from server on Google Maps using current latitude and longitude of bus and displays all stops in the route of that bus using Markers with a custom marker at stop of the user which requested for location displaying expected arrival time of bus on his stop.

**6.3 SOFTWARE NON-FUNCTIONAL REQUIREMENTS**

Constraints on the service or functions offered by the system such as timing constraints, constraints on development process, standards etc.

* 2. 1. **RELIABILITY**

High Reliability is the measure of how a product behaves in varying circumstances and our project is reliable because there are less chances of errors and exceptions and works well in varying circumstances. The probability that application will perform required function without failure would depend on traffic on server as well as the good internet speed.

* + 1. **AVAILABILITY**

The application is available to all the students, teachers and other college staffs. The users are able to use the application anytime when the college buses run as per their respective schedule. The services are available to the users as long as there is a good internet connectivity.

* + 1. **SECURITY**

Security is the ability of the software to remain protected from unauthorized access. This includes both change access and view access. This application is well secured using multiple levels of security constraints. It would be taken care that only an authenticated person can access this location, by checking validated fields in the database.

* + 1. **MAINTAINABILITY**

Maintainability of system is done by admin. Admin adds or deletes records from database whenever new routes, stops or buses are added or remove from college database.

* + 1. **PORTABILITY**

This project works on an android application, mobile phones are highly portable and can be easily carried anywhere by users. Popularity of android is another important aspect as most of users are aware of android user interface which makes it easy to use and increases scope of this application

* + 1. **PERFORMANCE**

Performance requirement is concerned with the speed of operations of functions and their accuracy. For optimum performance of application some non-functional requirements are recommended:

1. Good Internet Connectivity
2. Google Play Services 9.4+
3. Android Version 4.4.4 or higher
4. **INTRODUCTION**

A software design description (a.k.a. software design document or SDD) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, the description is required to coordinate a large team under a single vision, needs to be a stable reference, and outline all parts of the software and how they will work.

* 1. **DESIGN OVERVIEW**
     1. **DATA FLOW DIAGRAM**

**Level 0: -**

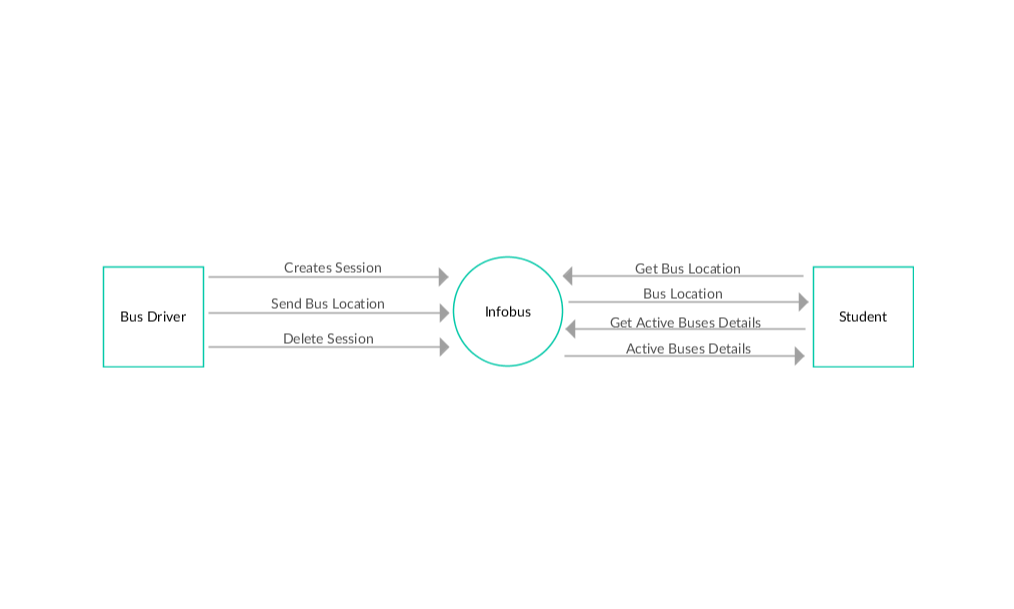
****

Figure 7.1.1 (a): - Data Flow Diagram (Level 0)

**Level 1: -**

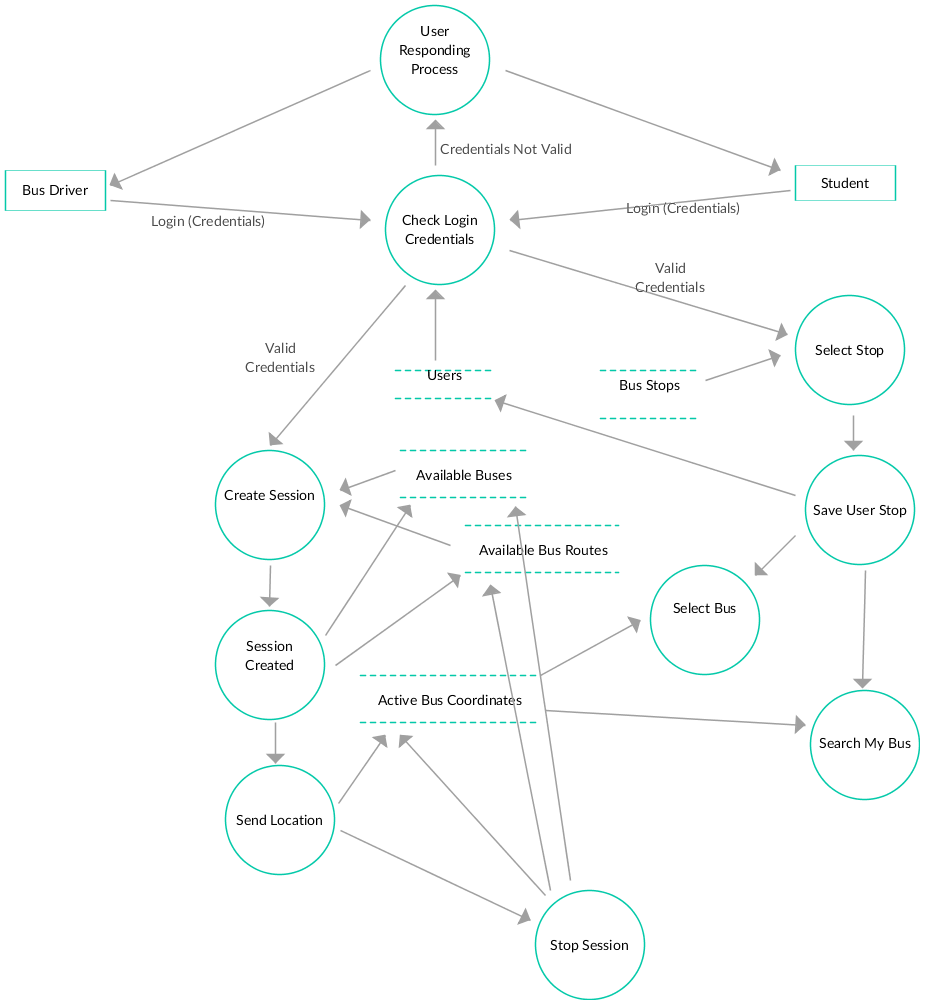
****

Figure 7.1.1 (b): - Data Flow Diagram (Level 1)

* + 1. **USE CASE DIAGRAM**

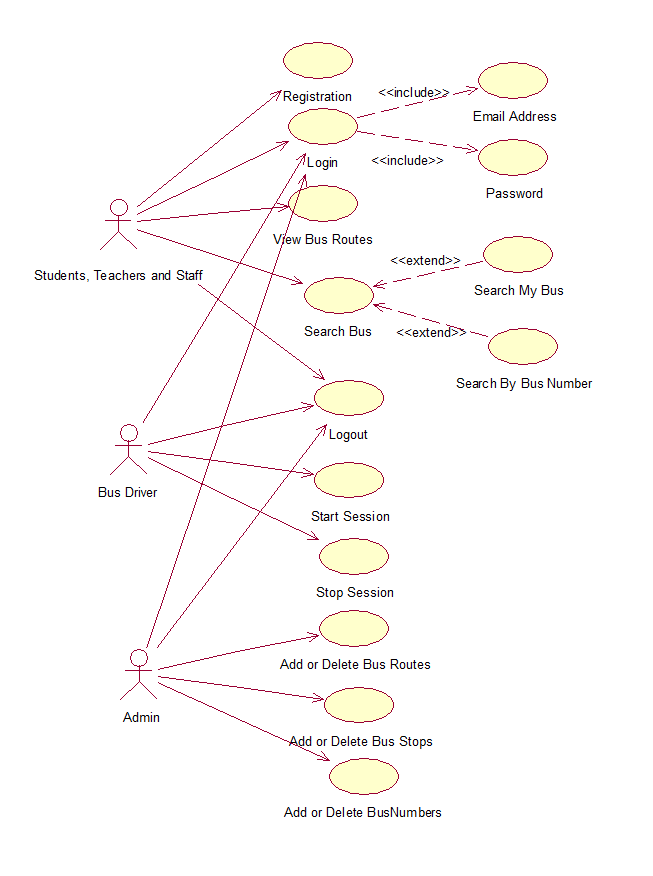


Figure 7.1.2 (a): - Use Case Diagram

* + 1. **CLASS DIAGRAM**

****Figure 7.1.3 (a): - Class Diagram

* + 1. **ACTIVITY DIAGRAM**

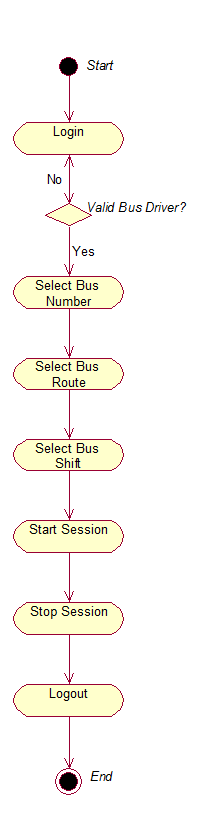


Figure 7.1.4 (a):- Activity Diagram (Bus Driver Module)

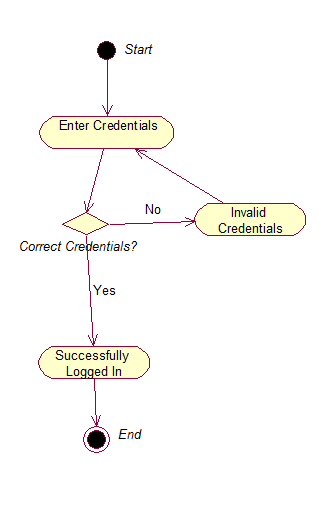


Figure 7.1.4 (b):- Activity Diagram (Login Module)

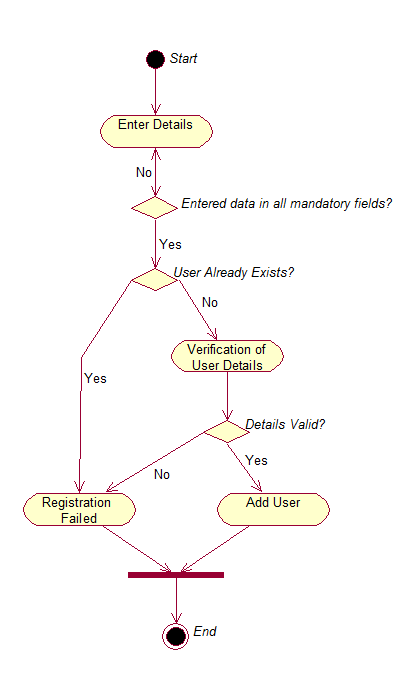


Figure 7.1.4 (c):- Activity Diagram (Registration Module)

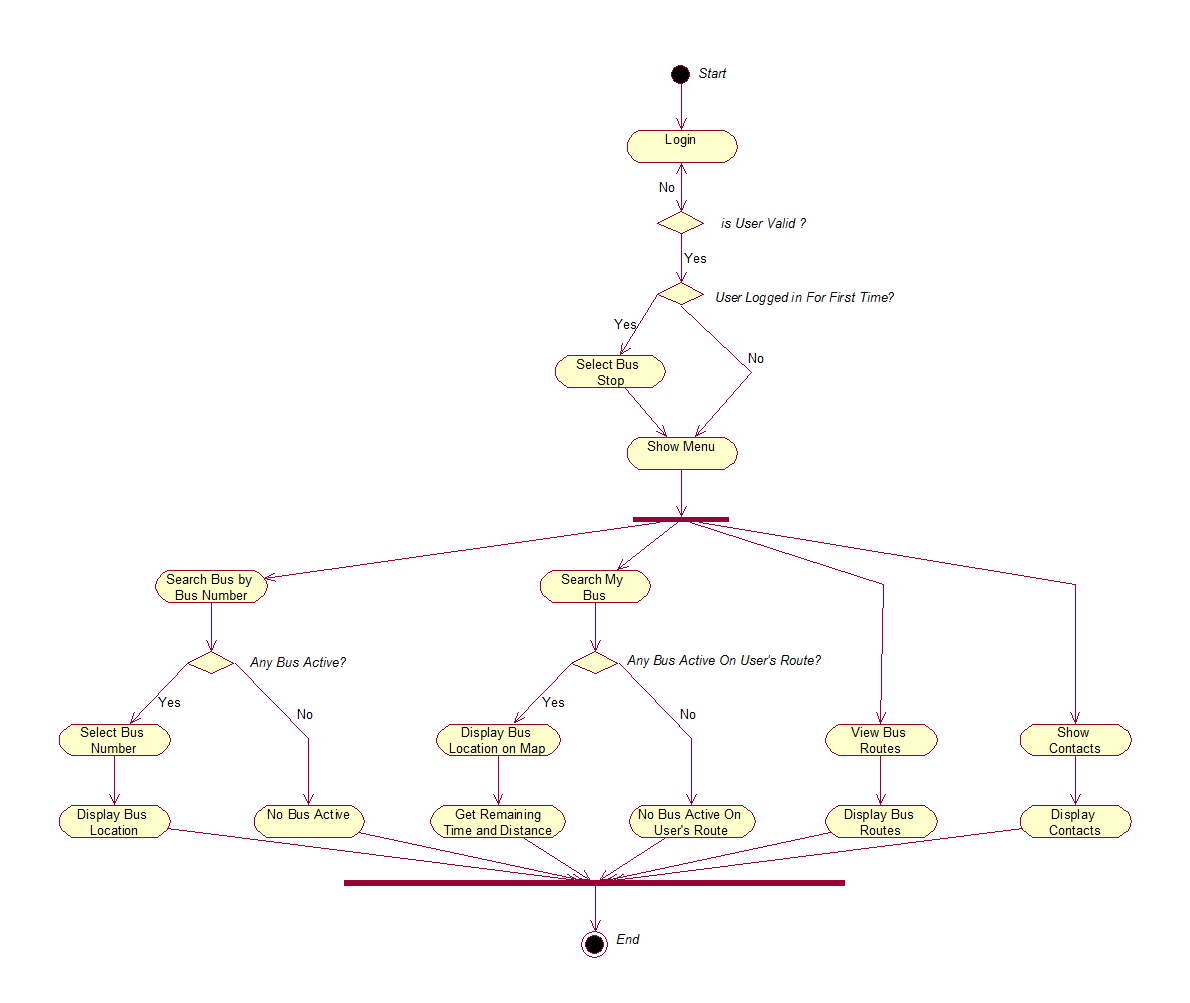


Figure 7.1.4 (d):- Activity Diagram (Student Application Module)

* + 1. **SEQUENCE DIAGRAM**

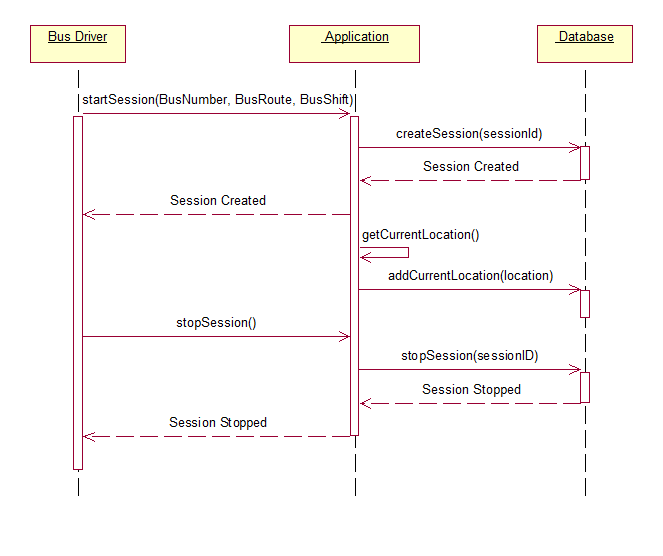


Figure 7.1.5 (a): - Sequence Diagram (Bus Driver: Create Session Module)

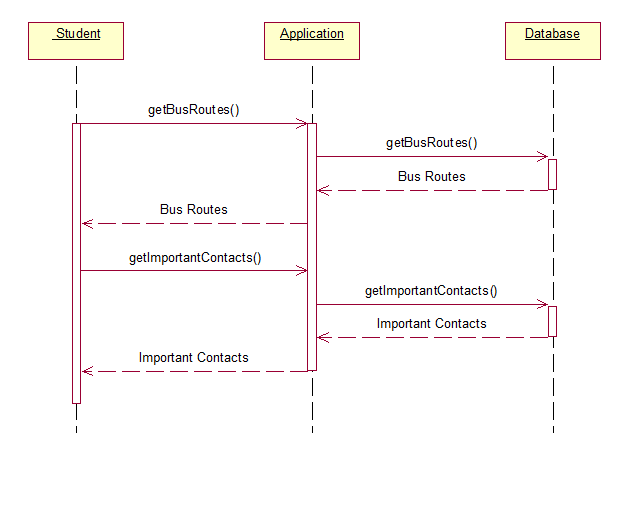


Figure 7.1.5 (b): - Sequence Diagram (Student Application)

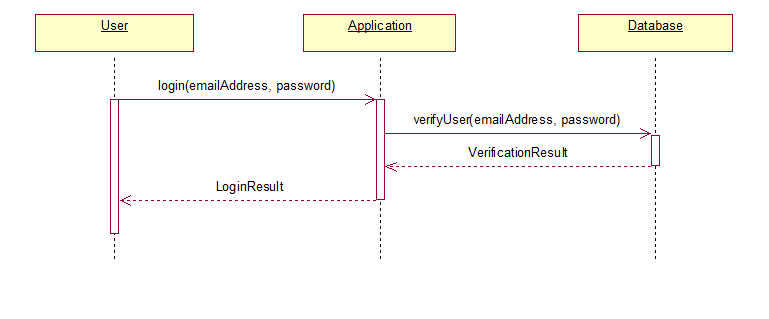
****

Figure 7.1.5 (c): - Sequence Diagram (Login Module)



Figure 7.1.5 (d): - Sequence Diagram (Student Module: - Registration)

* 1. **DATABASE DSIGN**
     1. **ENTITY RELATION DIAGRAM**

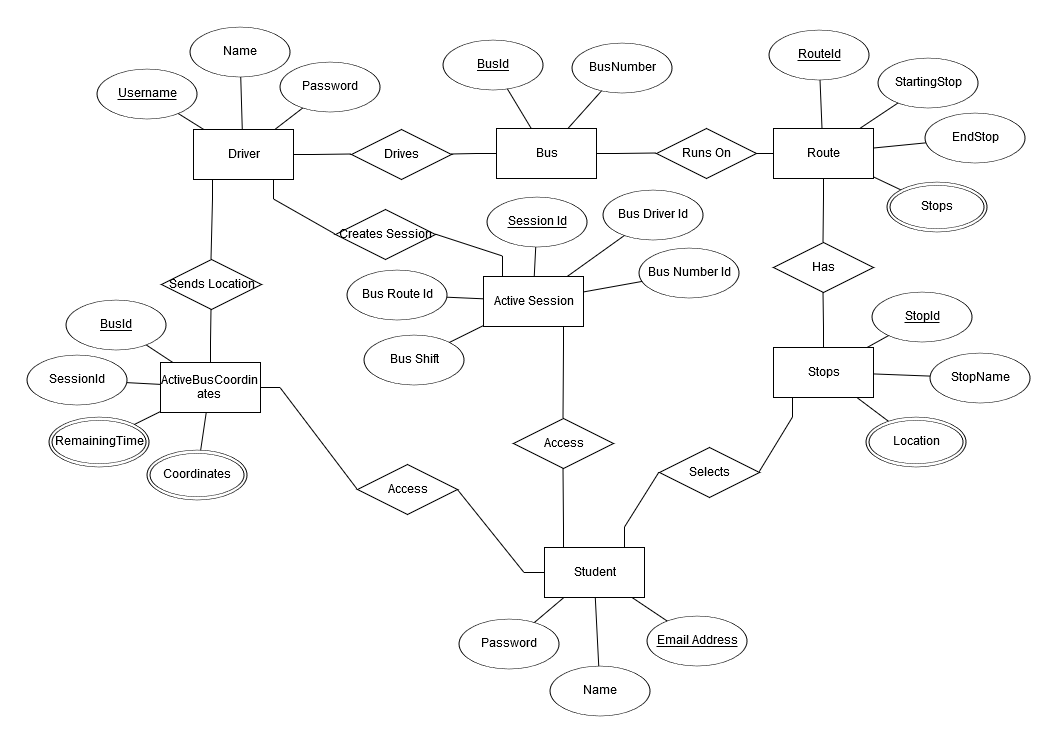
****

Figure 7.2.1 (a): - Entity Relation Diagram

1. **SYSTEM ARCHITECTURE DESIGN**

**8.1 CHOSEN SYSTEM ARCHITECTURE**

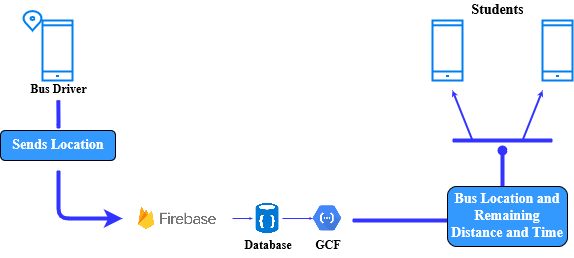


Figure 8.1(a): - System Architecture

1. **DETAILED DESCRIPTION OF COMPONENTS**

This system consists of 3 different module or components, each for: -

1. **Student:** - This module is used by students, teachers and other staff members of college. This module allows user to track any bus location and to know the precise arrival timing of bus on their respected stops.
2. **Bus Driver: -** This module will be used by bus drivers, this module will allow bus drivers to send their location to server. Through this module they can start and stop a session. A session has to be started when bus starts running on its route and session has to be stopped when bus reaches its final destination.
3. **Admin: -** Role of admin will be to manage bus numbers, bus routes, bus stops and to provide login credentials to bus drivers. Admin can do these things from firebase admin login.
4. **USER INTERFACE DESIGN**
   1. **DESCRIPTION OF USER INTERFACE**
5. **Login Activity: -**

A login, logging in or logging on is the entering of identifier information into this system by a user in order to access application. Login requires the user to enter two pieces of information, first a user name and then a password.

User can also login via his Google account.

1. **Registration Activity: -**

Registration is signing up for using this application.

User can register via his credentials.

Credentials include-

* Full Name
* Valid Email Address
* Password

1. **Forget Password: -**

User can change your password for security reasons or reset it if you forget it.

Password reset instructions will be sent to user email address.

1. **Select Stop: -**

Students can select their stop, which will be displayed on Google map from which user can select one. Stops are represents on map on based on user’s current location.

1. **Search Bus: -**

Students can track of any active bus, list of all active buses are shown to user from which he may select one whose exact real-time location would be shown on map with the route followed by it.

1. **Search My Bus: -**

This would show the current locations of bus that are running on route which consist of users stop. It also displays user the remaining distance and duration bus would take to reach students selected stop.

1. **Start Session: -**

Bus driver will start the session before starting the bus on its route, after starting session application will automatically send the location of bus to server in every x seconds.

1. **Stop Session: -**

Bus driver can stop the session once bus has reached its final destination.

* 1. **SCREEN IMAGES**

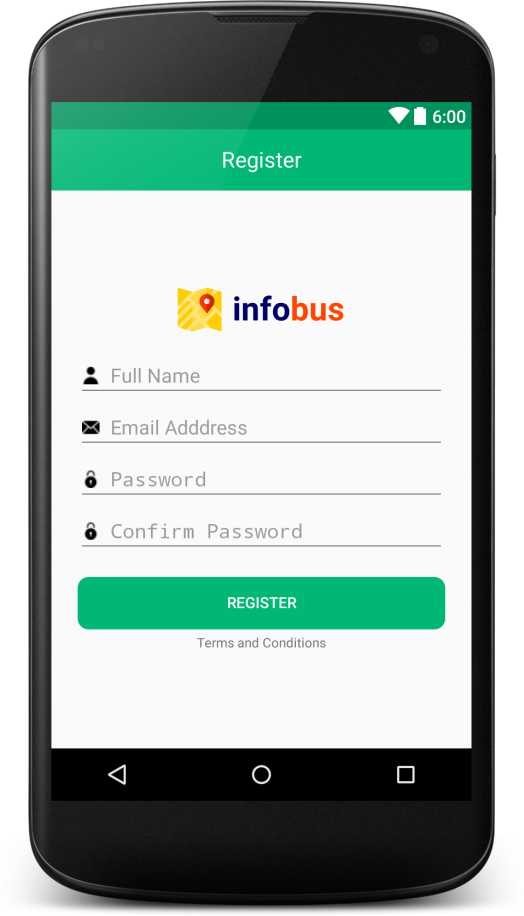
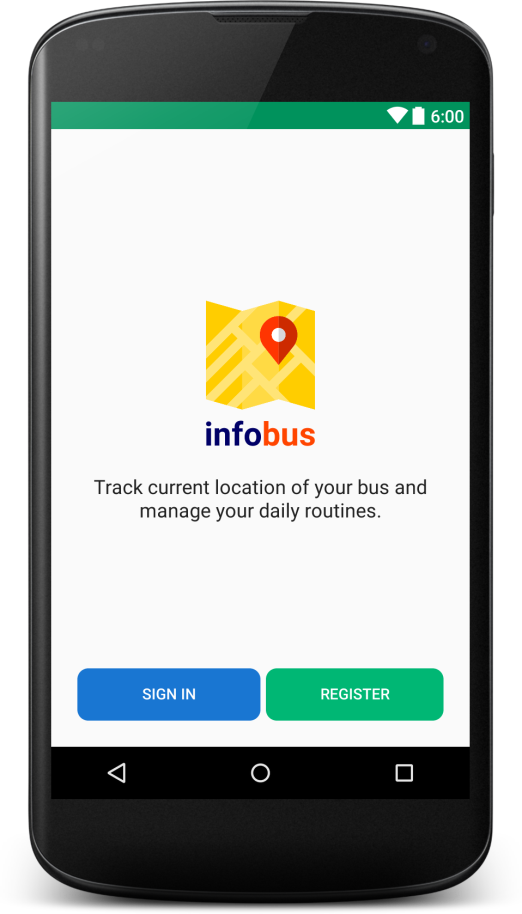


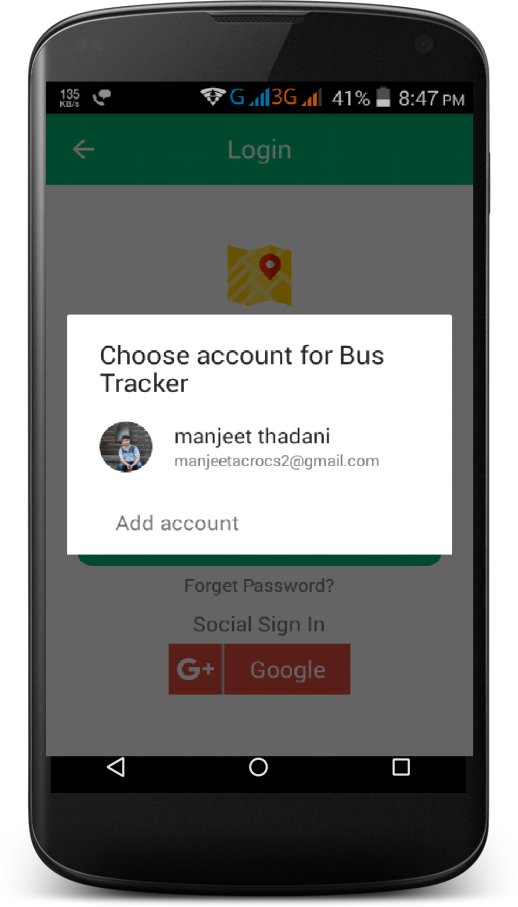
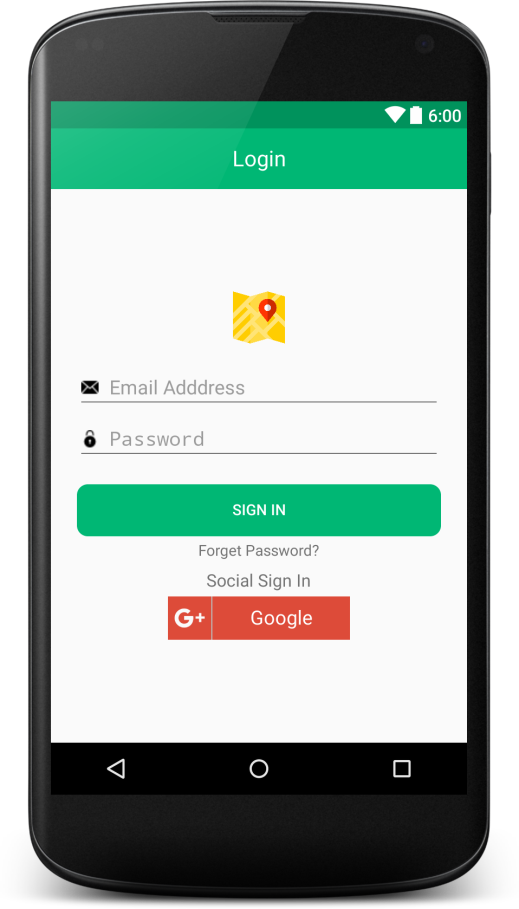
Figure 10.2 (a): - MainActivityFigure 10.2 (b): - Registration

Figure 10.2 (c): - Login

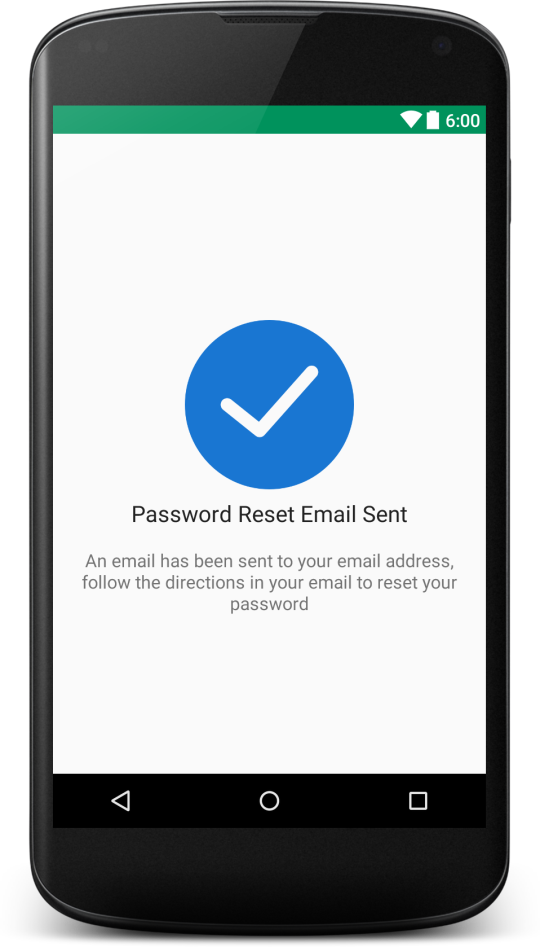
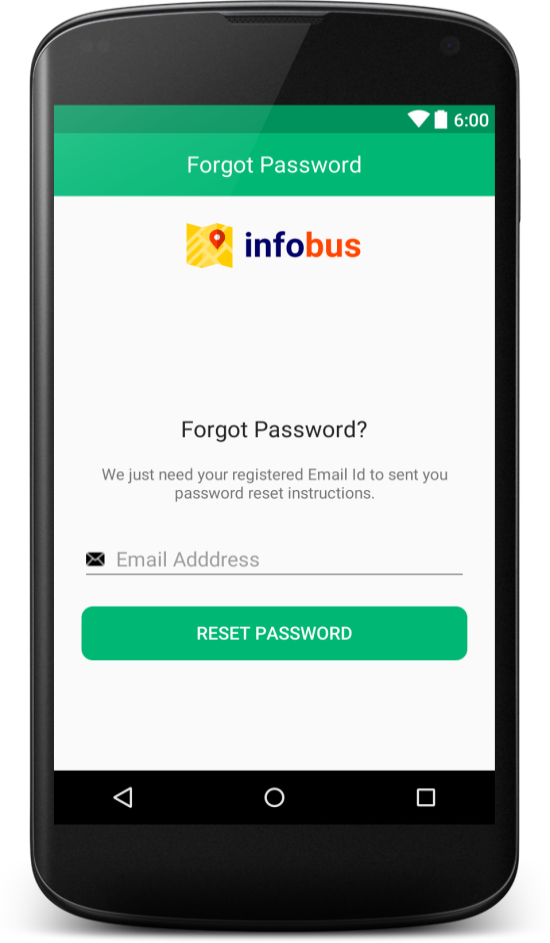


Figure 10.2 (d): - Forget Password

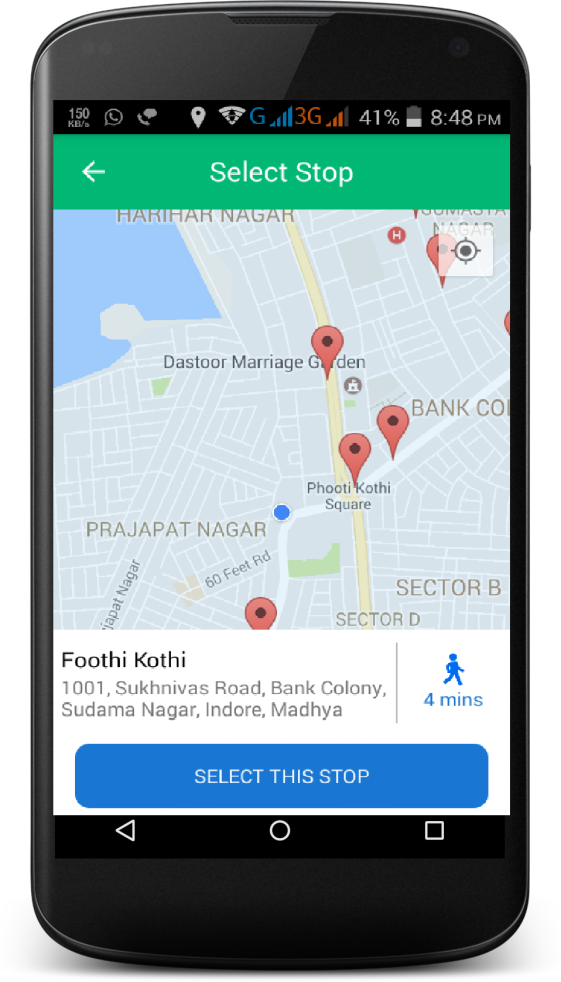
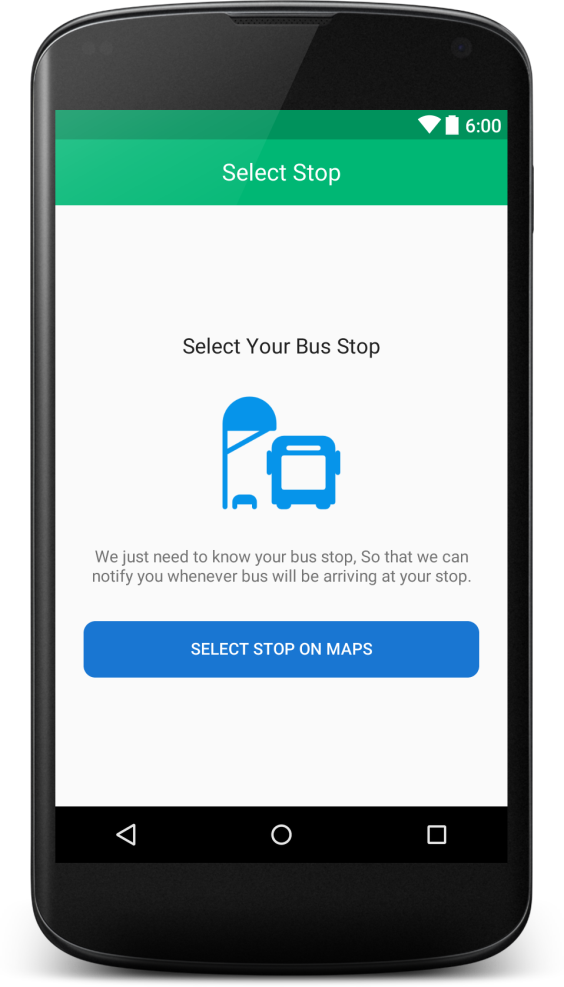


Figure 10.2 (e): - Select Stop

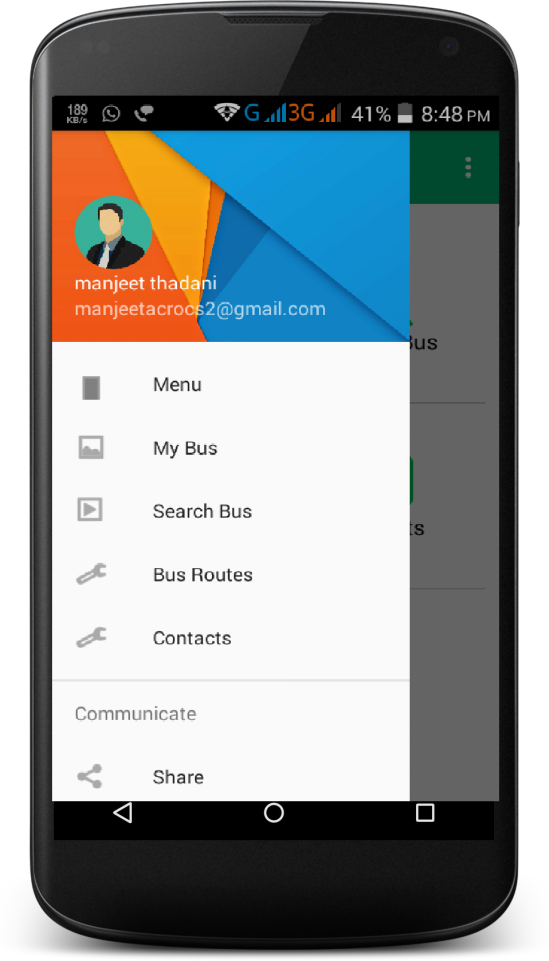
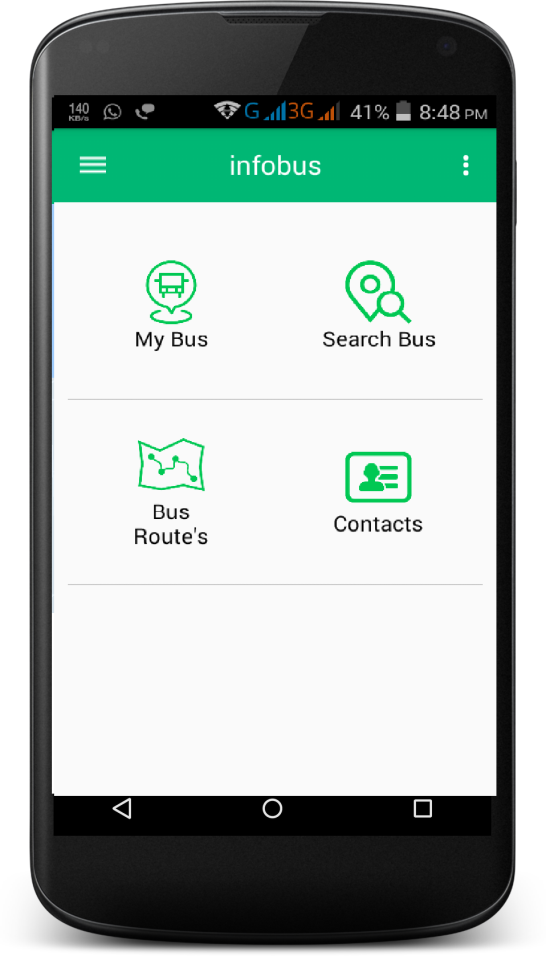


Figure 10.2 (f): - Menu

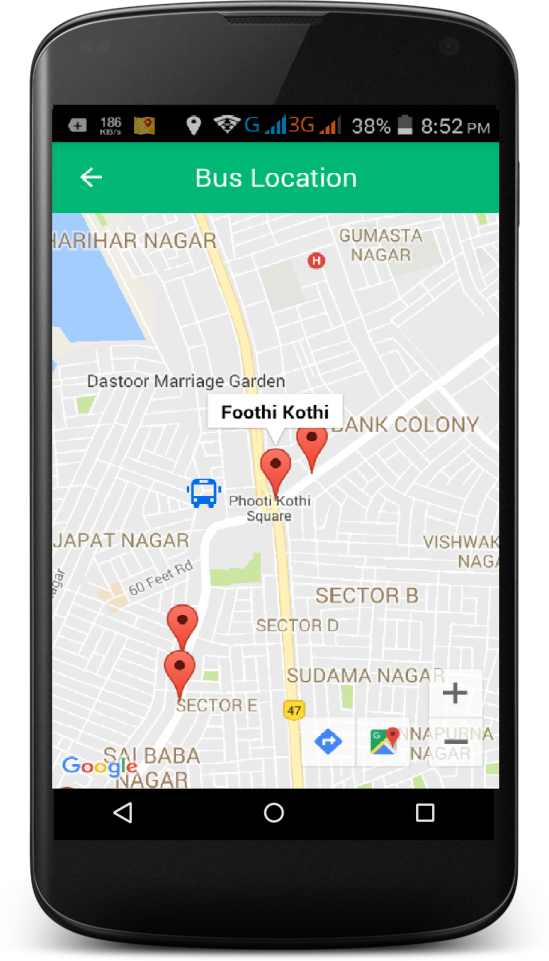
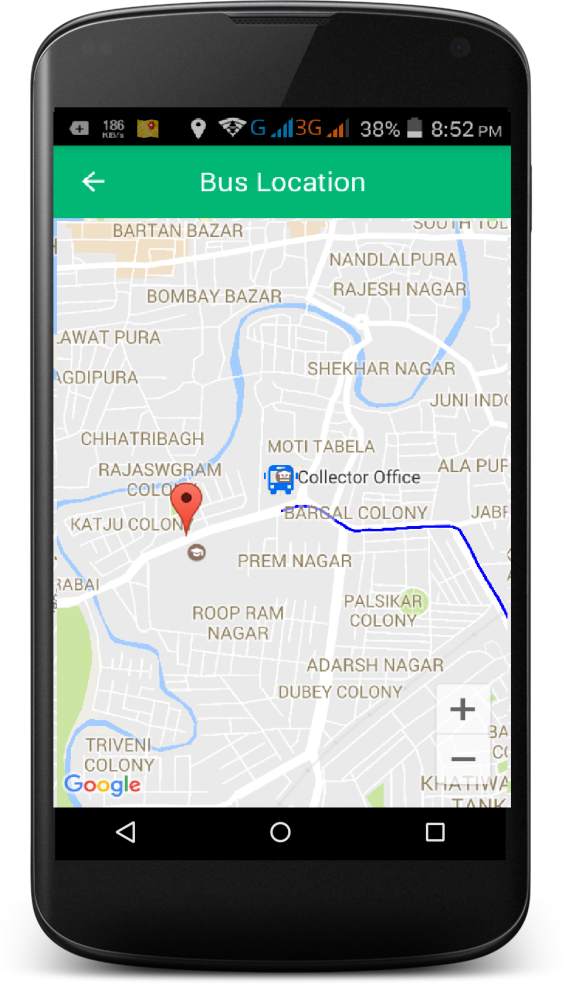


Figure 10.2 (g): - Search Bus Location

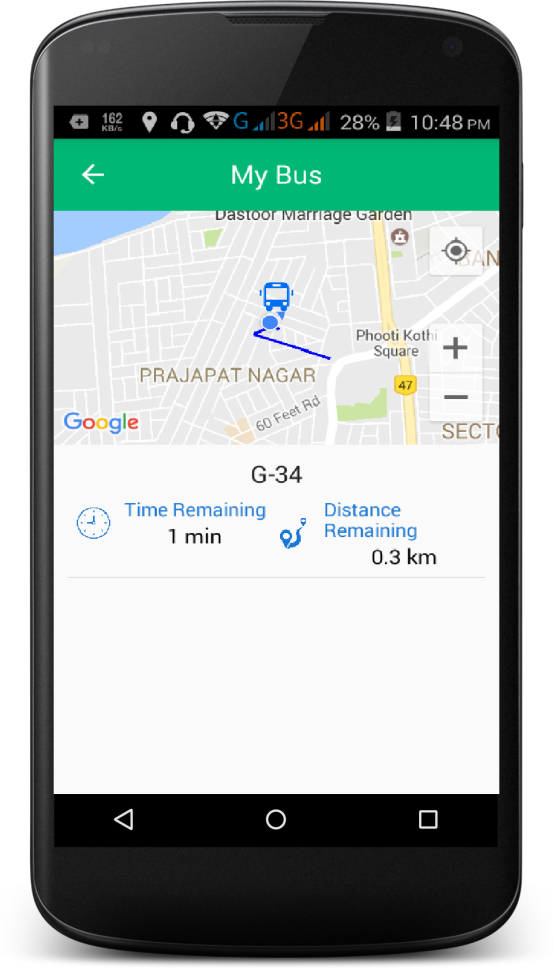


Figure 10.2 (h): - Search My Bus

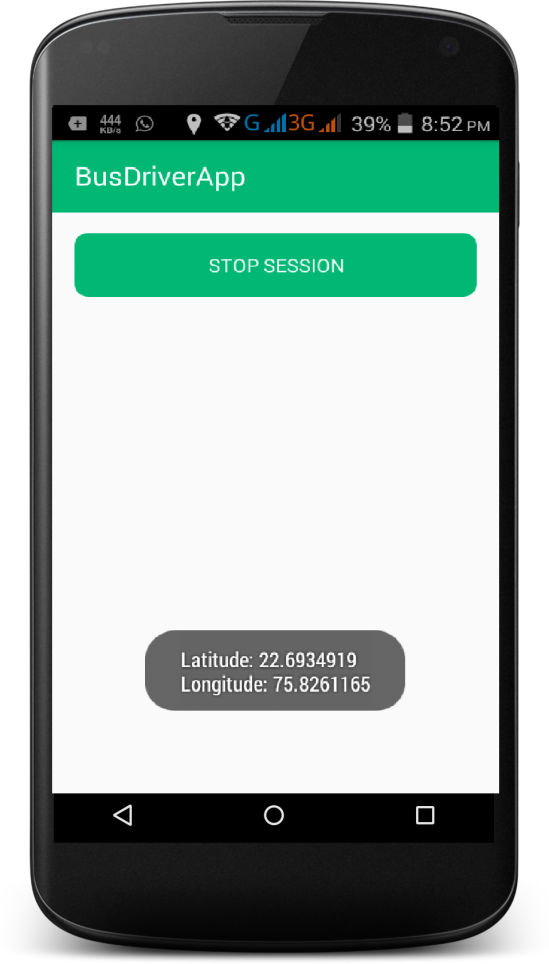
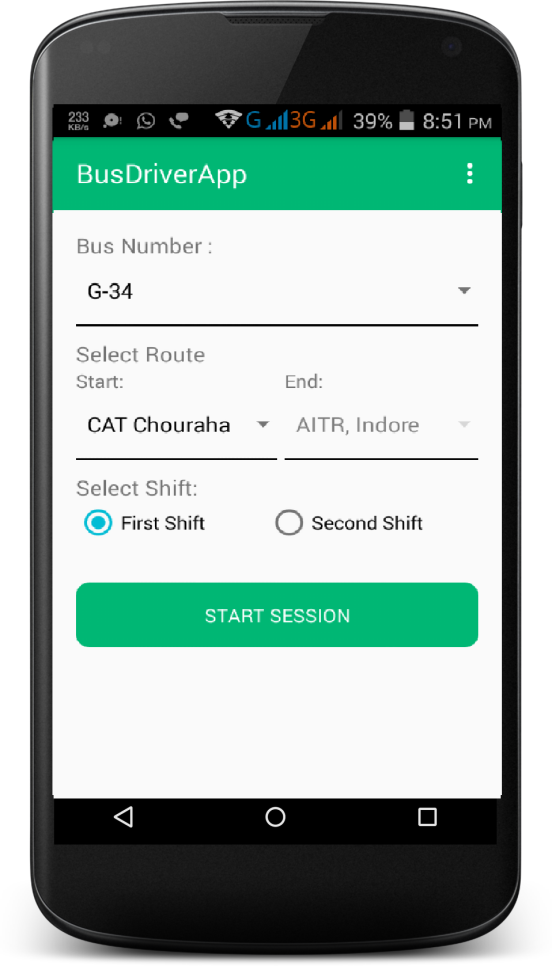


Figure 10.2 (i): - Start SessionFigure 10.2 (j): - Stop Session

1. **INTRODUCTION**

Testing documentation involves the documentation of artefacts that should be developed before or during the testing of Software. Documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing, etc. This section describes some of the commonly used documented artefacts related to software testing. Testing allows you to ensure your application works the way you think it does, especially as your codebase changes over time. If you have good tests, you can refactor and rewrite code with confidence. Tests are also the most concrete form of documentation of expected behaviour, since other developers can figure out how to use your code by reading the tests.

* 1. **SYSTEM OVERVIEW**

Android is the latest and a rapid growing technology available for all the users or customers in today’s market. An enormous increase in the end user acceptance has been experienced in the past few years. This paper proposes an Android mobile phone application that gives information about the bus arrival at users stop and real-time bus tracking through maps would help students, teachers and other staff members to track the current location of the bus and give the correct time for the bus to reach its bus stop. Reason for Android platform - Android requires an open source development which is probably the most feasible and a present user friendly approach. This paper also deals with bus routes and stops details, which are conveyed to the students directly using the Client-Server technology. Also, it displays the current location of bus on maps with the help of GPS. The main goal of the proposed work is to improve the Bus Tracking and Management System of India by adding the necessary additional features into the application.

* 1. **TEST APPROACH**

Various software-testing strategies have been proposed so far. All provide a template for testing. Things that are common and important in these strategies are: Testing begins at the module level and works “outward”: tests which are carried out are done at the module level where major functionality is tested and then it works towards the integration of entire system.

Different testing techniques are appropriate at different point of time: Under different circumstances, different testing methodologies are to be used which will be the decisive factor for software robustness and scalability. The developer of the software conducts testing and if the project is big then there is a testing team: All programmers should test and verify that their results are according to the specification given to them while coding. In cases where programs are big enough or collective effort is involved for coding, responsibilities for testing lies with the team as a whole.

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

**Proactive -**An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

**Reactive -**An approach in which the testing is not started until after design and coding are completed.

There are many strategies that a project can adopt depending on the context and some of them are:

* + 1. Dynamic and heuristic approaches
    2. Consultative approaches
    3. Model-based approach that uses statistical information about failure rates.
    4. Approaches based on risk-based testing where the entire development takes place based on the risk
    5. Methodical approach, which is based on failures.
    6. Standard-compliant approach specified by industry-specific standards.
  1. **TESTING OBJECTIVES**

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

Objectives of testing are: -

1. Finding defects which may get created by the programmer while developing the software.
2. Gaining confidence in and providing information about the level of quality.
3. To prevent defects.
4. To make sure that the end result meets the business and user requirements.
5. To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
6. To gain the confidence of the customers by providing them a quality product.
7. **TEST PLAN**

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice and any risks requiring contingency planning.

## 12.1FEATURES TO BE TESTED

Features are changes that add new functionality or significantly modify existing functionality. They are designed to be useful, effective, and attracting. Though, users are inventive and use the functionality unexpected ways. Also the application is complex and all pieces have to cooperate with each other. Any change might have surprising side effects in the related parts. Finally, developers are just humans and they might simply forget to implement a piece.

Feature testing should catch the above mentioned problems and make sure that users will be happy with the new stuff.

Features to be tested:-

1. Login Module: That always a valid and registered user is able to login into the system.
2. Student Registration Module: - Student registers through a valid email address.
3. Session created by Bus Driver: - To check that bus driver always creates a valid session.
4. Bus Location: - Location is transmitted even in background using services and only the location with higher accuracy is send.
5. Remaining Time Generation: - Remaining time to reach each stop on the route (on which bus is travelling) is calculated and updated frequently.
6. Delete Session: - Session is cancelled successfully by bus driver.

## 12.2 FEATURES NOT TO BE TESTED

This is a listing of what is 'not' to be tested from both the user's viewpoint of what the system does and a configuration management/version control view. This is not a technical description of the software, but a user's view of the functions.

Some features are not to be tested and they include:

1. Maintaining bus records
2. Maintaining bus routes record
3. Maintaining bus stop records
4. Maintaining bus drivers record
5. **TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Number** | **Test Case** | **Description** | **Test Result** |
| T-001 | Login Module | That always a valid and registered user is able to login into the system.  Students can also login via Google Account. | Successful |
| T-002 | Registration Module | Student registers through a valid email address | Successful |
| T-003 | Select Bus Stop Module | User selects his stop from the all valid bus stops. | Successful |
| T-004 | Search Bus | User searches location of that bus which is active at that instant. | Successful |
| T-005 | Get My Bus Module | To display correct distance and time remaining to user with validations. | Successful |
| T-006 | Create Session Module | To check that bus driver always creates a valid session | Successful |
| T-007 | Get Device Location Module | Location is transmitted even in background using services and only the location with higher accuracy is send | Successful |
| T-008 | Modularity Test | To check if module work well on different systems or not | Successful |

Table 13 (a): - Test Cases

**13.1 UNIT TESTING**

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

The following are the tests that are performed during the unit testing:

**Data Fetch from server using Async Methods: -** To check bus session and coordinates information fetched through server.

1. **Network Failures: -** To check network failures if any while data transmission between client and server.
2. **GPS Locations accuracy: -** To check accuracy of location obtained.
3. **Boundary Conditions: -** It is observed that much software fails at boundary conditions. That’s why boundary conditions are tested to ensure that the program is properly working at its boundary conditions.
4. **Independent Paths: -** All independent paths are tested to see that they are properly executing their task and terminating at the end of program.

**13.2 FUNCTIONAL TESTING**

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

Functional Testing is a testing technique that is used to test the features of the system or Software, should cover all the scenarios including failure paths and boundary cases. React forces us to build everything as “components.”

**13.3 SYSTEM TESTING**

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

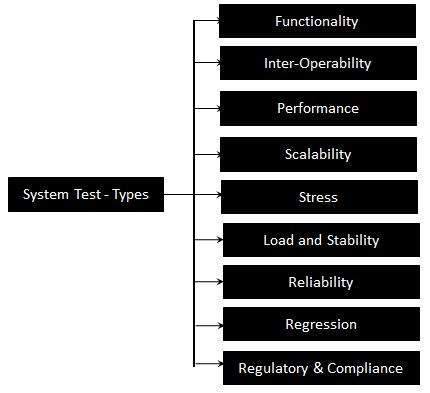


Figure 13.3 (a):- System Testing Types

**13.4 INTEGRATION TESTING**

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

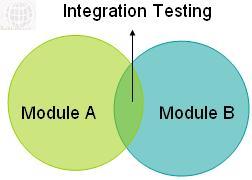


Figure 13.4 (a): - Integration Testing

**13.5 VALIDATION TESTING**

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment. It answers to the question, Are we building the right product?

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.

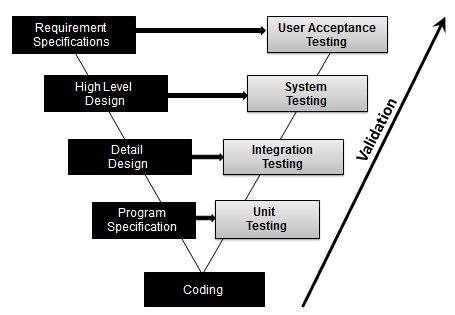


Figure 13.5 (a): - Validation Testing

**13.6 TOP DOWN INTEGRATION**

Top-down integration testing is an integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated. Stubs are the modules that act as temporary replacement for a called module and give the same output as that of the actual product.

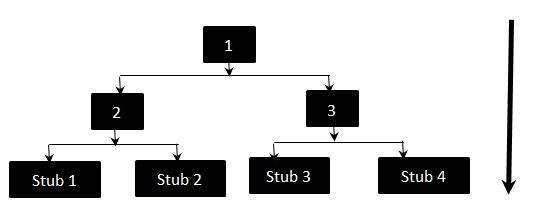


Figure 13.6 (a): - Top Down Integration

The above diagrams clearly states that Modules 1, 2 and 3 are available for integration, whereas, below modules are still under development that cannot be integrated at this point of time. Hence, Stubs are used to test the modules. The order of Integration will be:

**13.7 BOTTOM UP INTEGRATION**

Each component at lower hierarchy is tested individually and then the components that rely upon these components are tested.

## Bottom up Integration - Flow Diagram

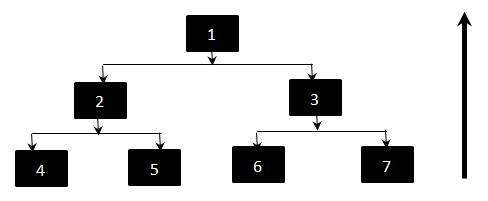


Figure 13.7 (a): - Bottom Up Testing

Though Top level components are the most important, yet tested last using this strategy. In Bottom-up approach, the Components 2 and 3 are replaced by drivers while testing components 4,5,6,7. They are generally more complex than stubs.

1. **CONCLUSION AND FUTURE SCOPE**

**Conclusion**

The conclusion of this study suggests that knowledge of specific domain improves the results. This Project has been implemented on Android platform. Also, different attributes are added to project which are advantageous to the system. The requirements and specifications have been listed above. This project uses firebase as backend infrastructure. Using GPS system, the application will send its location to server, which can be accessed via a valid user and location can be tracked and displayed on maps using Google Maps V2 .

A basic measurement between locations provides remaining duration and distance to users.

Due to simple NoSQL structure, data can be easily updated in future whenever required.

**Future Scope**

1. This system can be implemented for AICTSL buses.
2. It can be beneficial for every frequent bus travellers or tourists.
3. Similar applications can be developed for windows or IOS users.
4. **REFERENCES**
5. Available at <http://www.vogella.com/tutorials/AndroidServices/article.html>
6. Available at <https://developer.android.com/guide/components/services.html>
7. Available at <https://developers.google.com/maps/documentation/android-api/intro>
8. Available at <https://developers.google.com/maps/documentation/android-api/polygon-tutorial>
9. Available at <https://developer.android.com/studio/write/firebase.html>
10. Available at <https://www.udacity.com/course/firebase-essentials-for-android--ud009>
11. Available at <https://firebase.google.com/docs/functions/>
12. **BIBLIOGRAPHY**
13. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)
14. Android Recipes: A Problem-Solution Approach
15. Google Android Firebase: Learning the Basics, Book by Bill Stonehem

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| **Figure Number** | **Figure Name** | **Page Number** |
| Figure 2.1 (a) | Incremental Model | 3 |
| Figure 4 (a) | Gantt Chart | 6 |
| Figure 6.1.2 (a) | Android Studio | 8 |
| Figure 6.1.2(b) | Firebase | 8 |
| Figure 7.1.1 (a) | Data Flow Diagram (Level 0) | 12 |
| Figure 7.1.1 (b) | Data Flow Diagram (Level 1) | 13 |
| Figure 7.1.2 (a) | Use Case Diagram | 14 |
| Figure 7.1.4 (a) | Activity Diagram (Bus Driver Module) | 16 |
| Figure 7.1.4 (b) | Activity Diagram (Login Module) | 17 |
| Figure 7.1.4 (c) | Activity Diagram (Registration Module) | 18 |
| Figure 7.1.4 (d) | Activity Diagram (Student Application Module) | 19 |
| Figure 7.1.5(a) | Sequence Diagram (Bus Driver: Create Session Module) | 20 |
| Figure 7.1.5 (b) | Sequence Diagram (Student Application) | 21 |
| Figure 7.1.5 (c) | Sequence Diagram (Login Module) | 22 |
| Figure 7.1.5 (d) | Sequence Diagram (Student Module: - Registration) | 22 |
| Figure 7.2.1 (a) | Entity Relation Diagram | 23 |
| Figure 8.1(a) | System Architecture | 24 |
| Figure 10.2 (a) | MainActivity | 27 |
| Figure 10.2 (c) | Login | 27 |
| Figure 10.2 (d) | Forget Password | 28 |
| Figure 10.2 (e) | Select Stop | 28 |
| Figure 10.2 (f) | Menu | 29 |
| Figure 10.2 (g) | Search Bus Location | 29 |
| Figure 10.2 (h) | Search My Bus | 30 |
| Figure 10.2 (i) | Start Session | 30 |
| Figure 13.3 (a) | System Testing Types | 36 |
| Figure 13.4 (a) | Integration Testing | 37 |
| Figure 13.5 (a) | Validation Testing | 37 |
| Figure 13.6 (a) | Top Down Integration Testing | 38 |
| Figure 13.7 (a) | Bottom Up Integration Testing | 38 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **LIST OF TABLES** | | |
| **Table Number** | **Table Name** | **Page Number** |
| Table 1.2 (a) | Project Deliverables | 8 |
| Table 13 (a) | Test Cases | 40 |

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **LIST OF ABBREVATIONS** | |
| **Abbreviation** | **Meaning** |
| GPS | Global Positioning System |
| GSM | Global System for communication |
| GCM | Google Cloud Messaging |
| UI | User Interface |
| Firebase | Firebase gives you the tools and infrastructure you need to build better apps and grow successful businesses. |
| NoSQL | It provides a mechanism for storage and retrieval of data which is modelled in means other than the tabular relations used in relational databases |
| API | Application Programming Interface |