DECLARATION

We declare that this project "NSTU Bus Tracking System" submitted to Institute of Information Technology, Noakhali Science and Technology University to fulfil the requirements for the Bachelor of Science in Software Engineering (BSSE) in 5th Semester. This project work and project related software has done under the guidance of Md.Auhidur Rahman Sumon, Assistant Professor & Director (Acting), Institute of Information Technology, Noakhali Science and Technology University. Though this project is mainly developed for the student and teacher of NSTU, this is certified that we did this project under SE-3112, Software Project Lab -II Course and it has not been submitted elsewhere for the requirement of any degree or for any other purposes.

Members of the Project

1. Md. Al Adnan
2. Md. Faisal Ahammed
3. Moon Moon Das

Supervisor of the Project

Md.Auhidur Rahman Sumon

Assistant Professor & Director (Acting)

IIT, NSTU

ACKNOWLEDGEMENT

This project is completed with the support of our project supervisor and collective effort. We would like to express our deepest thanks who contribute with us by their knowledge and effort that helps us to complete this successful project.

We would like to thank to our Project Supervisor Md.Auhidur Rahman Sumon very much because of his valuable suggestion and direction contribution that helps us in many ways to complete this project.

Finally we would like to thanks the great almighty “Allah” who gives us the patient to do such a work like “NSTU Bus Tracking System”.

Table of Contents

Chapter ONE

NSTU BUS TRACKING SYSTEM PREFACE

Chapter 1

1.1 Purpose:

The purpose of this document is to present a detailed description of the NSTU Bus Tracking System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system. It also serves as a contract between the customer and the supplier to approve the requirements contained here, as specified in the acceptance criteria. The audience of this document (Software Requirements Specification) primarily includes the Project Team, the Client and the Project Supervisor.

1.2 Overview

Travelling by bus to go for university is the most common thing and also a hassle for students and teachers waiting for the bus. Many times, students and teachers wait for the bus which is not available at that time and it might take a lot of time to reach University. Due to this problem many times they waste their precious time since they do not have the information about the alternative buses and many a times they get late for their exams. Therefore, this project is focusing to provide a solution to this problem by giving the proper and accurate information about the bus to the students and teachers. We also ensure the necessary security of information and other processes.

1.3 Scope of the Project

We want to build a system which will track the current position and direction of the bus and show it to the user who wants to travel by the bus. The driver will share his location that’s will show to the user the current position of bus. The system will reduce the waiting time for travelling.

1.4 Evolution of this Document

Our Team will endeavor to elicit all of the requirements for the proposed system in the early stages of the project life-cycle. However, it may be impracticable for some requirements to be specified during the initial stages. As such, this document has been produced as thoroughly as is possible at this time. Any changes deemed necessary by the client in conjunction with our Team will be formally documented by revising this document in a traceable manner.

In order to prevent alterations to the requirements at a later phase in the production process, our Team will provide the client with a prototype of the system, showing all interfaces, and allowing the client to suggest any changes.

1.5 Glossary

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

- **API** – Application Programming Interface
- **SRS** – Software Requirement Specifications
- **UI** – User Interface
- **SDLC** – Software Development Life Cycle
- **DBMS** – Database Management System

1.6 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Chapter TWO

PROJECT PLAN

Chapter 2

2.1 User Classes and characteristics

There are three types of stakeholders in our “NSTU bus tracking system”. Such as:

Driver: Driver share their location which used to provide the user the current location of the bus.

Student: Student can get the current direction and location of the bus when they once logged in to the system.

Teacher: Teacher can get the current direction and location of the bus when they once logged in to the system.

2.2 Proposed Timeline:

Task	**Deadline**
Proposal	Within 3rd week of February, 2020
Requirement Analysis, Specification	Within 21th July, 2020
Designing, Study	Within 25th October, 2020
Coding	Within 24th September, 2021
Final Testing	Within 11th October, 2021

Chapter Three

SOFTWARE REQUIREMENTS SPECIFICATION FOR NSTU BUS TRACKING SYSTEM

Chapter 3

3.1 Design and implementation constraints

Design and implementation constraints are those that we have used to implement this project make successful. It also describes tool that enables developers and testers to view and interact with the user interface (UI) elements of this application.

User Interface Technology

3.1.1 Programming Language

We will use java as programming language to develop the system.

3.1.2 Database Server

We will use firebase server to store all of the information of this system. The reason behind to choose the database server are given below:

- Security
- Reporting and Data Mining
- Replication
- Fault tolerance

3.2 Functional Requirement Specifications

Every system must have some functional requirements. Functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional software requirements help to capture the intended behavior of the system. Now, we are going to mention functional requirements associating with our project.

Table 01: Show bus location in real time

FR-1	**Show information to user in real time**
Description	When the user request to system the location of the bus then it will show the current location of the bus.
Stakeholders	Student, Teacher, Driver
Priority	High

Table 02: Save real time location records of bus when driver start ride

FR-2	**Save real time location records of bus when driver start ride**
Description	When driver start ride the real time location records of bus will be stored on database so that later it will be process to ready for students and teachers so that they can see the bus location.
Stakeholders	Driver
Priority	High

Table 03: Allow user to retrieve information from mobile device

FR-3	**Allow user to retrieve information from mobile device**
Description	User can retrieve information from mobile device when they once logged in to the NSTU Bus Tracking System.
Stakeholders	Student, Teacher
Priority	High

Table 04: Verify the user with institutional email

FR-4	**Verify the user with institutional email**
Description	When user register to the system, a verification email will send to user institutional email.
Stakeholders	Student, Teacher, Driver
Priority	High

Table 05: Show the currently available bus location in a map

FR-5	**Show the currently available bus location in a map**
Description	When the user logged in to the system then system will show currently available buses location in the google map.
Stakeholders	Student, Teacher
Priority	High

Table 06: Save real time location of currently logged in users

FR-6	**Save real time location of currently logged in users**
Description	When user logged in to the system then the system will retrieve the user location using their device location.
Stakeholders	Student, Teacher
Priority	Medium

Table 07: Administrator manage all the user

FR-6	**Administrator manage all the user**
Description	Administrator will manage all the registered user and bus schedule as required.
Stakeholders	Administrator
Priority	High

Table 08: Emergency contact with administrator

FR-6	**Emergency contact with administrator**
Description	User can contact with administrator in any emergency situation.
Stakeholders	Driver
Priority	Medium

3.3 Use case diagram and description for NSTU Bus Tracking System:

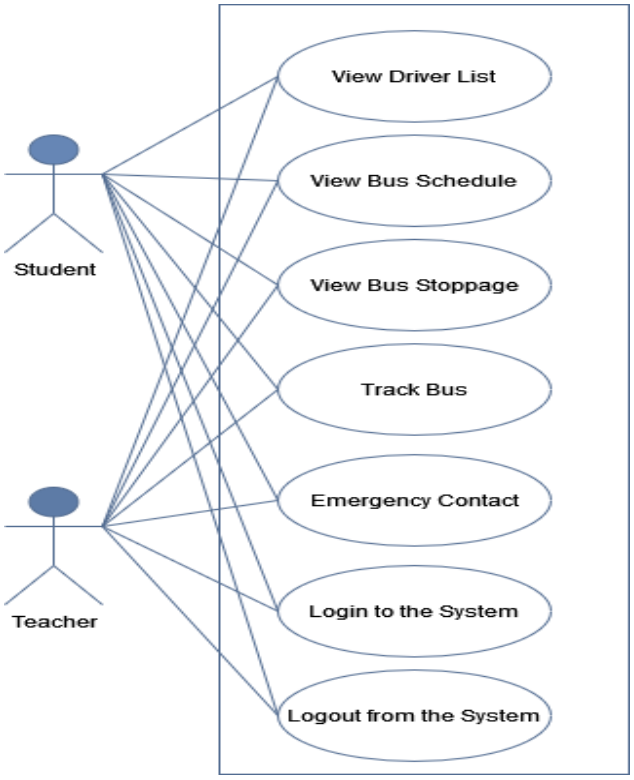


Fig-1: Teacher and Student activities

3.3.1 Use case description for user activities:

Table 09: Use Case Description of Log in to System

Use case Id	**01**	
Use case Name	Login to the system	
Goal	As a valid user must be able to log in to the system.	
Preconditions	Users must be registered to the system.	
Success End Condition	User will successfully logged in to 'NSTU Bus Tracking System'.	
Failed End Condition	User can't logged in to the system.	
Primary Actors: **Secondary Actors:**	Student, Teacher	
Trigger	Click on 'Locate Your Bus'.	
Main Success Flows	**Step**	**Action**
	1	User gives required information.
	2	System check the information of user.
	3	System find that user is unique, and information are right.
	4	User successfully logged in to the system.
Alternative Flows	**Step**	**Branching Action**
	1	The system find that user is not unique, and information are wrong.
	2	User can't log in to the system.
Quality Requirements	Not applicable	

Table 10: Use Case Description of Log out from System

Use case Id	**02**	
Use case Name	Log out from System	
Goal	User will log out from the system	
Preconditions	Users must be registered to the system	
Success End Condition	User will successfully logged out from 'NSTU Bus Tracking System'	
Failed End Condition	User can't logged from the system	
Primary Actors: **Secondary Actors:**	Student, Teacher	
Trigger	User is in the system.	
Main Success Flows	**Step**	**Action**
	1	User log in to the system
	2	User log out from the system
Alternative Flows	**Step**	**Branching Action**
	1	User does not want to log out
Quality Requirements	Not applicable	

Table 11: Use Case Description of View Driver List

Use case Id	**03**	
Use case Name	View Driver List	
Goal	User will see all the available driver list.	
Preconditions	User must have to be logged in into NSTU Bus Tracking System.	
Success End Condition	User will successfully able to see all the available driver list.	
Failed End Condition	User can't able to see driver list.	
Primary Actors: **Secondary Actors:**	Student, Teacher Administrator	
Trigger	Click on View Pilot	
Main Success Flows	**Step**	**Action**
	1	User log in to the system.
	2	User click 'Menu' button.
	3	User click 'View Driver List'.
	4	User see the list of driver.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 12: Use Case Description of Track Bus

Use case Id	**04**	
Use case Name	Track Bus	
Goal	As a valid user must be able to track bus location.	
Preconditions	Users must be logged in to the system.	
Success End Condition	User will successfully able to see the location of buses.	
Failed End Condition	User can't track the location of bus.	
Primary Actors: **Secondary Actors:**	Student, Teacher Driver	
Trigger	Click on 'Locate Your Bus'.	
Main Success Flows	**Step**	**Acton**
	1	User click on 'Locate Your Bus'.
	2	User will see the location of buses.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 13: Use Case Description of View Bus Schedule

Use case Id	**05**	
Use case Name	View Bus Schedule	
Goal	As a valid user must be able to view bus schedule.	
Preconditions	Users must be logged in to the system.	
Success End Condition	User will successfully able to see the schedule of buses.	
Failed End Condition	User can't see the schedule of buses.	
Primary Actors: **Secondary Actors:**	Student, Teacher Administrator	
Trigger	Click on 'Bus Schedule'.	
Main Success Flows	**Step**	**Action**
	1	User logged in to the system.
	2	User click on 'Menu'.
	3	User click 'Bus Schedule'.
	4	User see the schedule of buses.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 14: Use Case Description of Emergency Contact

Use case Id	**06**	
Use case Name	Emergency Contact	
Goal	User will be able to emergency communication with administrator.	
Preconditions	Users must be logged in to the system.	
Success End Condition	User will successfully able to see the schedule of buses.	
Failed End Condition	User can't see the schedule of buses.	
Primary Actors: **Secondary Actors:**	Student, Teacher Administrator	
Trigger	Click on 'Emergency Contact'.	
Main Success Flows	**Step**	**Action**
	1	User logged in to the system.
	2	User click on 'Menu'.
	3	User click 'Emergency Contact'.
	4	User able to communicate with administrator.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

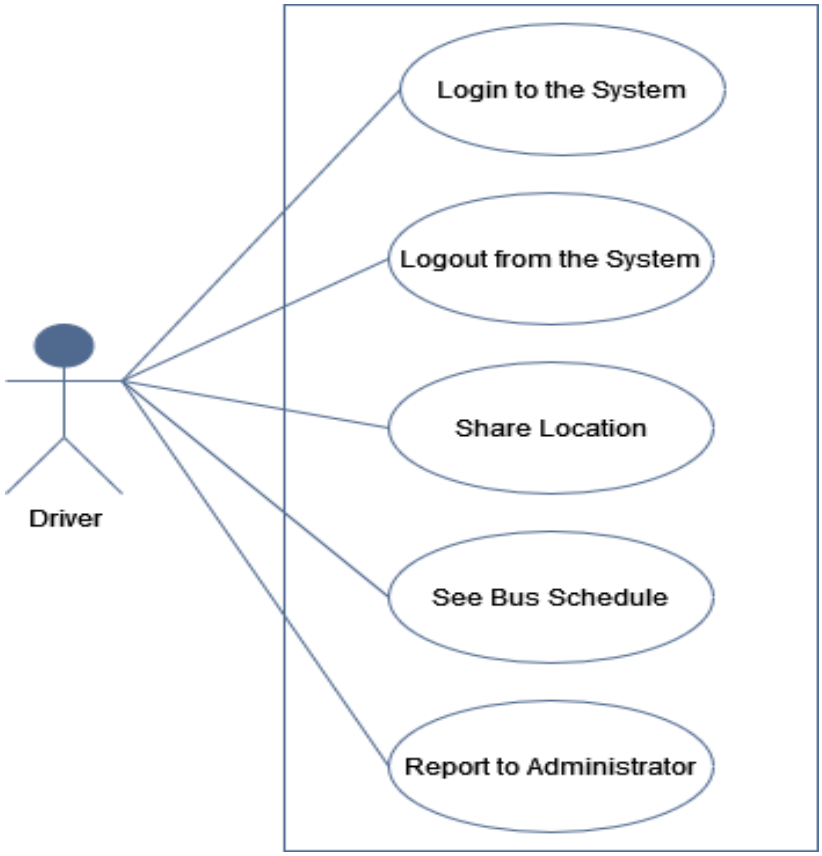


Fig-2: Driver activities

3.3.2 Use case description for driver activities:

Table 15: Use Case Description of Share Location

Use case Id	**07**	
Use case Name	Share Location	
Goal	Driver will share bus location.	
Preconditions	Driver must be registered & logged in to the system.	
Success End Condition	Driver share bus location.	
Failed End Condition	Driver don't share bus location.	
Primary Actors: **Secondary Actors:**	Driver Administrator	
Trigger	Click "share location" button.	
Main Success Flows	**Step**	**Action**
	1	Driver logged into the system.
	2	Driver click the share location button.
	3	System process the position data received from bus Positioning module.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 16: Use Case Description of See Bus Schedule

Use case Id	**08**	
Use case Name	See Bus Schedule	
Goal	Driver wants to see his bus schedule which set before by administrator	
Preconditions	Driver must be registered & logged in to the system.	
Success End Condition	Driver see his bus schedule.	
Failed End Condition	Driver don't see his bus schedule.	
Primary Actors: **Secondary Actors:**	Driver Administrator	
Trigger	Click "See Bus Schedule" button.	
Main Success Flows	**Step**	**Action**
	1	Driver logged into the system.
	2	Driver click the see bus schedule button.
	3	System shows the bus schedule.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 17: Use Case Description of Report to Administrator

Use case Id	**09**	
Use case Name	Report to Administrator	
Goal	Driver wants to send emergency situation report to administrator	
Preconditions	Driver must be registered & logged in to the system.	
Success End Condition	Driver send a emergency situation report to administrator.	
Failed End Condition	Driver can't send a emergency situation report to administrator.	
Primary Actors: **Secondary Actors:**	Driver Administrator	
Trigger	Click "Report to Administrator" button.	
Main Success Flows	**Step**	**Action**
	1	Driver logged into the system.
	2	Driver faced a problem.
	3	Driver send a report to administrator.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

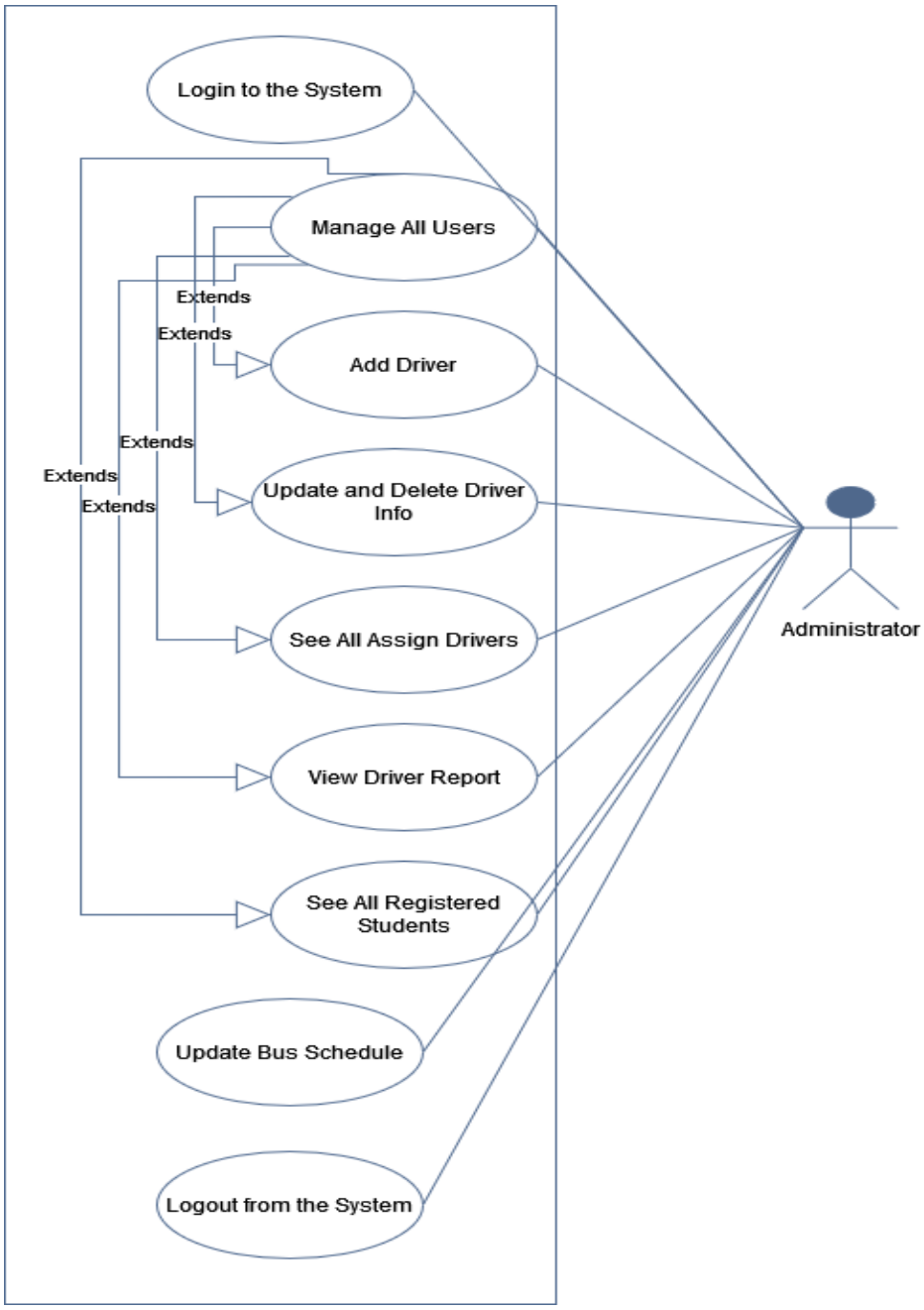


Fig-3: Administrator activities

3.3.3 Use case description for administrator activities:

Table 18: Use Case Description of Manage all Users

Use case Id	**10**	
Use case Name	Manage All Users	
Goal	Administrator wants to manage all users of their system.	
Preconditions	Administrator must add users for managing them.	
Success End Condition	Administrator successfully manage users those are include in their system.	
Failed End Condition	Administrator can't manage users of their system.	
Primary Actors: **Secondary Actors:**	Administrator Teacher, Student, Driver	
Trigger	Click on Manage all users.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator create profile for users.
	2.1	Fill up necessary information for users.
	4	Administrator manage and monitor users that are include in their system.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 19: Use Case Description of Update Driver Info

Use case Id	**11**	
Use case Name	Update or delete driver Info	
Goal	Administrator wants to update or delete drivers info which add before.	
Preconditions	Administrator must be logged in to the system.	
Success End Condition	Administrator successfully update or delete drivers info in their system.	
Failed End Condition	Administrator can't update or delete drivers info in their system.	
Primary Actors: **Secondary Actors:**	Administrator Driver	
Trigger	Click on Update or Delete Info.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator successfully update or delete driver's info which add before.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 20: Use Case Description of Add Driver

Use case Id	**12**	
Use case Name	Add Driver	
Goal	Administrator wants to add drivers of their system.	
Preconditions	Administrator must be logged in to the system.	
Success End Condition	Administrator successfully add drivers in their system.	
Failed End Condition	Administrator can't add drivers in their system.	
Primary Actors: **Secondary Actors:**	Administrator Driver	
Trigger	Click on Add drivers.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator create profile for drivers.
	2.1	Fill up necessary information for drivers.
	3	Administrator add users.
	4	Finally click add button to add drivers to the system.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 21: Use Case Description of Update Bus Schedule

Use case Id	**13**	
Use case Name	Update Bus Schedule	
Goal	Administrator wants to update bus schedule.	
Preconditions	Administrator must be logged in to the system.	
Success End Condition	Administrator successfully update the bus schedule.	
Failed End Condition	Administrator can't update the bus schedule.	
Primary Actors: **Secondary Actors:**	Administrator Driver	
Trigger	Click on Update Bus Schedule.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator add day and time in bus schedule.
	3	Fill up necessary information for drivers.
	4	Administrator successfully update bus schedule.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

Table 22: Use Case Description of View Driver Report

Use case Id	**14**	
Use case Name	View Driver Report	
Goal	Administrator wants to view driver report which sent from driver module.	
Preconditions	Administrator must be logged in to the system.	
Success End Condition	Administrator successfully view the driver report.	
Failed End Condition	Administrator can't view the driver report.	
Primary Actors: **Secondary Actors:**	Administrator Driver	
Trigger	Click on View Driver Report.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator successfully view driver report which sent from driver module.
Alternative Flows	**Step**	**Branching Action**
	1	Don't have any report and administrator can't view the report.
Quality Requirements	Not applicable	

Table 23: Use Case Description of See All Assign Drivers

Use case Id	**15**	
Use case Name	See All Assign Drivers	
Goal	Administrator wants to see all assign drivers which add before.	
Preconditions	Administrator must be logged in to the system.	
Success End Condition	Administrator successfully see all assign drivers.	
Failed End Condition	Administrator can't see all assign drivers.	
Primary Actors: **Secondary Actors:**	Administrator Driver	
Trigger	Click on See All Assign Drivers.	
Main Success Flows	**Step**	**Action**
	1	Administrator logged into the system.
	2	Administrator add drivers in the system.
	3	Administrator see all assign drivers in the system.
Alternative Flows	Not applicable	
Quality Requirements	Not applicable	

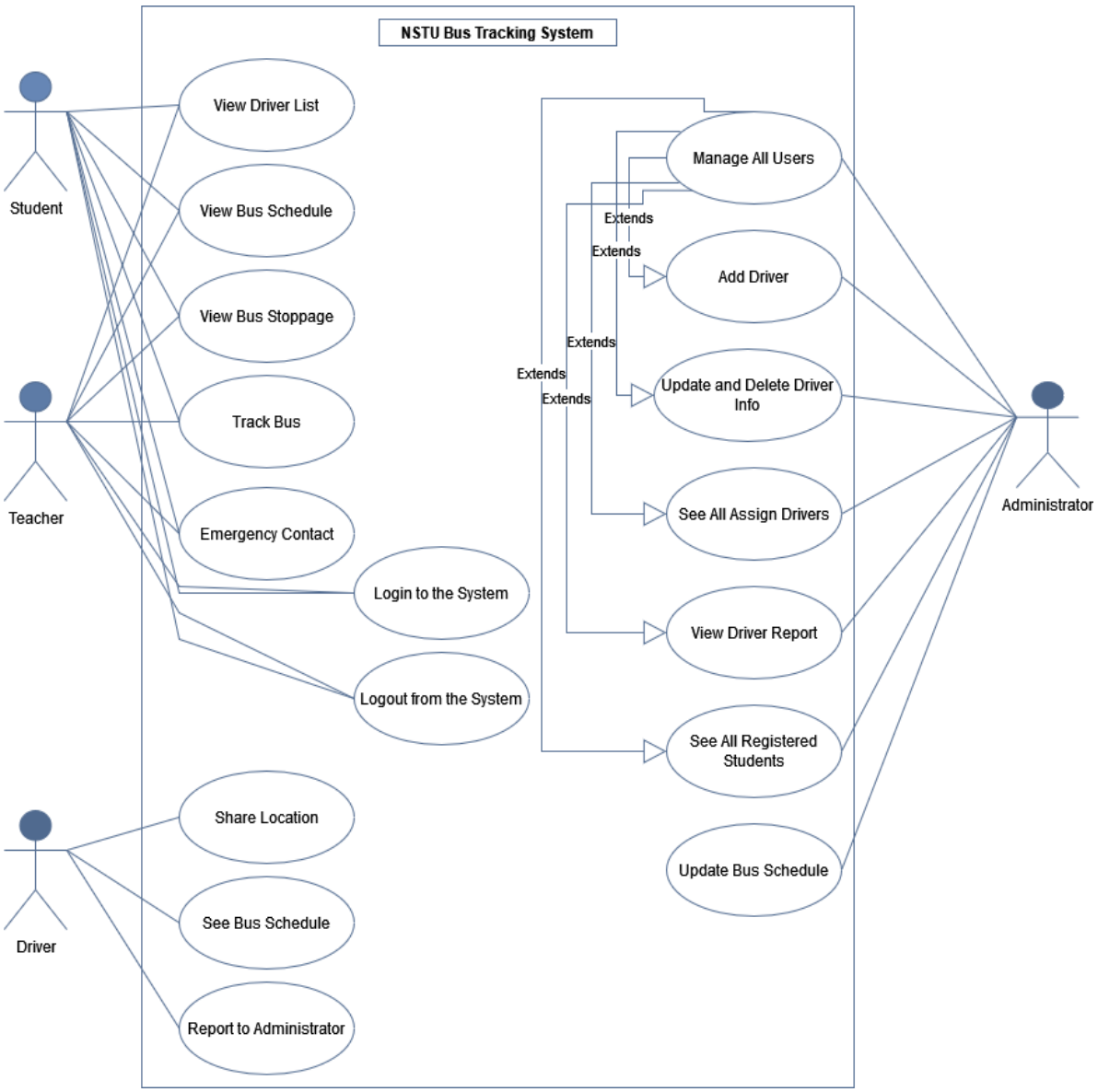


Fig 4: Use Case Diagram for NSTU Bus Tracking System

3.4 Non Functional Requirements

3.4.1 Data Requirements

For defining data requirements, we need to build the model. For our application maximum data would be loaded from remote user. And for that purpose, we need to focus on some major points. Such as:

- Types of entity of the system
- Route data locations
- Capacity and resources of the data requirements
- Data source sequence
- Data availability schedules
- Quantity of data
- Availability of data

3.4.2 Performance Requirements

It is very important to maintain performance of any software system. To ensure performance, we need to maintain some steps. Now, I will explain some perspective by which we are going to enhance the performance of our project.

3.4.3 Speed & Latency Requirements

Speed and latency requirements must be ensured while retrieving data from the bus positioning module.

Table 24: Showing the bus position in a map view must be faster

SLR-1	**Showing the bus position in a map view must be faster**
Description	When the user request to system the location of the bus then the system will provide the current location within 1 second.
Stakeholders	Student, Teacher
Priority	High

3.4.4 Precision & Accuracy Requirements

Result that is to be shown to the end user is need to be accurate. Because, wrong information might be ruined the whole business process.

Table 25: Showing the bus position in a map view must be accurate

PAR-1	**Showing the bus position in a map view must be accurate**
Description	When the system will provide the current location of the bus it must have to be accurate.
Stakeholders	Student, Teacher
Priority	High

3.4.5 Capacity Requirements

The developed system by us must be capable to handle user data, provide accurate information, handling database, manage http request etc.

Table 26: The system will handle thousands of data

CR-1	**The system will handle thousands of data**
Description	The system needs to handle data thousands of data every moment.
Stakeholders	Student, Teacher
Priority	High

3.5 Dependability Requirements

The term dependability is measured based on four dimensions. Such as:

- Availability
- Reliability
- Safety
- Security

If we want to say that our application system is dependable then it must fulfil the four dimensions. But there are other tasks. Like there is no way to make mistakes or our system should have the ability to detect and then remove errors. Besides that, it is also very important to limit the damage which might be caused by system failure.

3.5.1 Reliability & Availability Requirements

Now, we will mention requirements which are related to reliability and availability.

Table 27: The system must be available on time

RAR-1	**The system must be available on time**
Description	Our system must be available every day on time • The system must be updated regularly • System must be malware free
Stakeholders	Student, Teacher
Priority	High

3.5.2 Robustness or Fault-Tolerance Requirements

To ensure robustness and fault-tolerance facilities to the end users, it is urgent to ensure 0% crush. Moreover, it must show accurate results.

Table 27: The system handles all user access without system errors

RFT-1	**The system handles all user access without system errors**
Description	Thousands of users might hit our application system at a time. All their requests must be handled without any fault.
Stakeholders	N/A
Priority	High

3.5.3 Safety-Critical Requirements

There are no safety-critical requirements in our project.

3.6 Maintainability & Supportability Requirements

It is very important to provide after service or support to the end users.

3.6.1 Supportability Requirements

Supportability requirements may have related to some extends. Like:

- Testability
- Extensibility
- Adaptability
- Maintainability
- Compatibility
- Configurability
- Serviceability
- Install ability

Our application meets all of the above requirements related to supportability.

3.6.2 Adaptability Requirements

There are no adaptability requirements in our application system.

3.7 Security Requirements

Making software security as a requirement is very important. Software security requirements should be its functional requirement. Software security enforces security of an application system.

Functionality related to software security can either be directly tested or observed. Some security related requirements are given below:

- Signing in with an educational email in a student/teacher module
- Provide an otp code for confirmation of a valid user.
- Get access according to logged in user
- Signing out as a student/teacher

While accessing to the system, each and every module must provide a central authentication mechanism. There is also a process to prevent entering into the system by ensuring hashed password for the unauthenticated users.

3.7.1 Access Requirements

For accessing to our application system, there remains some authentication and authorization techniques. And every module of our system will provide it. Now we will provide an explanation below.

Table 29: Application provides security mechanism

AR-1	**Application provides security mechanism**
Description	Every module is designed in such a way that it only gives access to the authorized and authenticated users.
Stakeholders	Student, Teacher, Bus Driver
Priority	High

3.7.2 Integrity Requirements

Integrity requirements refers to a security system which ensures an expectation of data quality. It also ensures that all data of the system would never be exposed to the malicious modification or accidental destruction. For that reason, we will store our user passwords as encrypted format which is impossible to decrypt. It is also called hashed password.

3.7.3 Privacy Requirements

It is very important to ensure privacy of the system users. Privacy requirements enhances to protect stakeholder's privacy. In this way, all data or a partial part of data are going to be disclosed according to system's privacy policy. To ensure privacy, the central database should be protected by the anonymous. Users are permitted to get access to those data which are being associated by them which can be ensured by the user log in system.

3.8 Usability and Human-Interaction Requirements

The main target of developing any system is to make the system user friendly and easy to usable for the end users.

3.8.1 Ease of Use Requirements

Our application is easy to use and also easily understandable.

Table 30: Application must be usable for the end users

EUR-1	**Application must be usable for the end users**
Description	This app is enough usable to the teacher, student and bus driver by which they can operate this system easily.
Stakeholders	Student, Teacher, Bus Driver
Priority	High

3.8.2 Personalization and Internationalization Requirements

There are not any personalization and internationalization requirements to our system. This maiden version of our application is only be operated by Noakhali Science and Technology University.

3.8.3 Understand ability and Politeness Requirements

It is already said that the application which we are going to develop, is understandable enough. The system provides hints to users whether any error occurred or wrong. By reading those errors users can be able to operate the system easily.

3.8.4 Accessibility Requirements

There are no specific accessibility requirements associated to our system yet.

3.8.5 User Documentation Requirements

Documentation are mainly two types. One is internal documentation which is generally written by the application engineers. It is prepared to make development life cycle easier for the system engineers or system analysts.

Table 31: The system engineer documentation

UDR-1	**The system engineer documentation**
Description	To develop our application named "NSTU Bus Tracking System", firstly we have made a system analysis team as well as documentation team.
Stakeholders	System analysts or software developers
Priority	Medium

3.8.6 Training Requirements

Training requirements involved in after service of any application. It is very necessary to properly train up end users to the system so that they would be capable to operate easily. After launching the full package to the market, firstly we provide training to the different end users like bus drivers, students and teachers.

3.9 Look and Feel Requirements

Look and feel requirements mainly refers how the system will look like and how the user interface or graphical user interface of our system will display to the user.

3.9.1 Appearance Requirements

Students and all other user must know which input fields are required and which are not. For that reason, we will use labels for all input fields. Input fields might be text type, radio, checkbox, spinner etc.

Table 32: Labels of mandatory fields must be bold

AR-1	**Labels of mandatory fields must be bold**
Description	The mandatory field's label must be bold and all input fields must have placeholder to make it easier for the users.
Stakeholders	Teacher, Student, Bus Driver
Priority	Medium

3.10 Operational and Environmental Requirements

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc.

3.10.1 Expected Physical Requirements

There are no expected physical requirements in our system.

3.10.2 Requirements for Interfacing with Adjacent Systems

There are no requirements for interfacing with adjacent system for our project.

3.10.3 Release Requirements

There are no specific release requirements in our system.

3.11 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policy of any organizations. The terms and condition of our application is that, no third-party software or person are allowed to engage to use our data for their business purpose.

3.11.1 Compliance Requirements

There are no specific compliance requirements for our system.

3.11.2 Standards Requirements

There are no specific standards requirements for our system.

3.12 Requirement Engineering Process

Requirement engineering refers to the process of defining, documenting and maintaining requirements in the engineering design process. It is a common role in systems engineering and software engineering.

3.13 Requirement Elicitation Techniques

Requirement elicitation is the process of collecting and refining stakeholder's requirements. It is perhaps the most difficult, most error-prone and most communication intensive software development. It can be successful only through an effective customer-developer partnership. It is needed to know what the users really need.

3.13.1 Perform Document Analysis

Existing documentation can help reveal how systems currently work or what they are supposed to do. Documentation includes any written information about current systems, business processes, requirements specifications, competitor research. Reviewing and analysing the documents can help identify functionality that needs to remain, functionality that isn't used.

3.13.2 Survey/Questionnaire

For Survey/Questionnaire, a set of questions is given to stakeholders to quantify their thoughts. After collecting the responses from stakeholders, data is analyzed to identify the area of interest of stakeholders. Questions should be based on high priority risks. Questions should be direct and unambiguous. Once the survey is ready, notify the participants and remind them to participate.

Chapter Four

DESIGN AND ARCHITECTURE OF NSTU BUS TRACKING SYSTEM

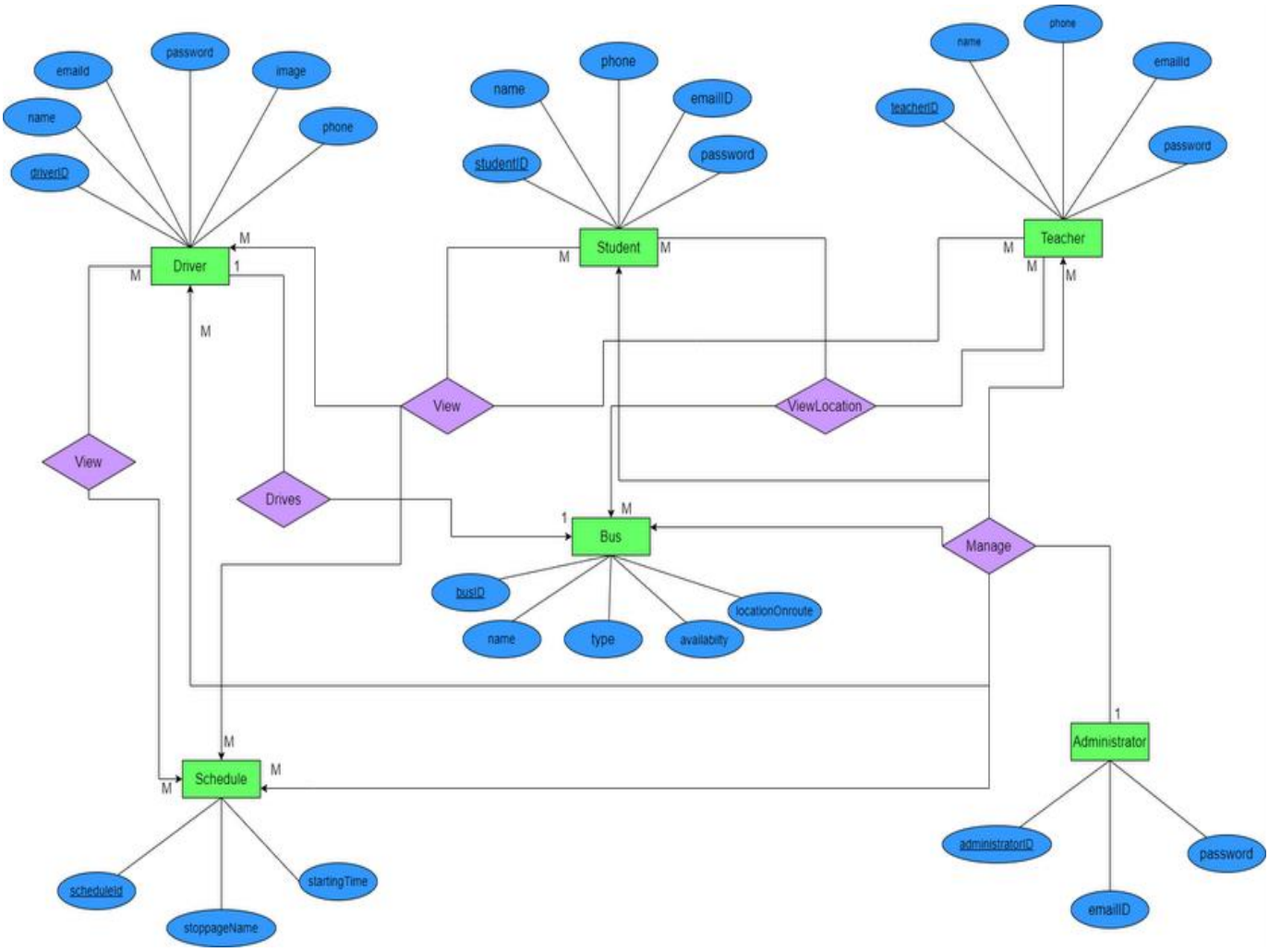


Fig 5: Entity Relationship Diagrams of NSTU Bus Tracking System

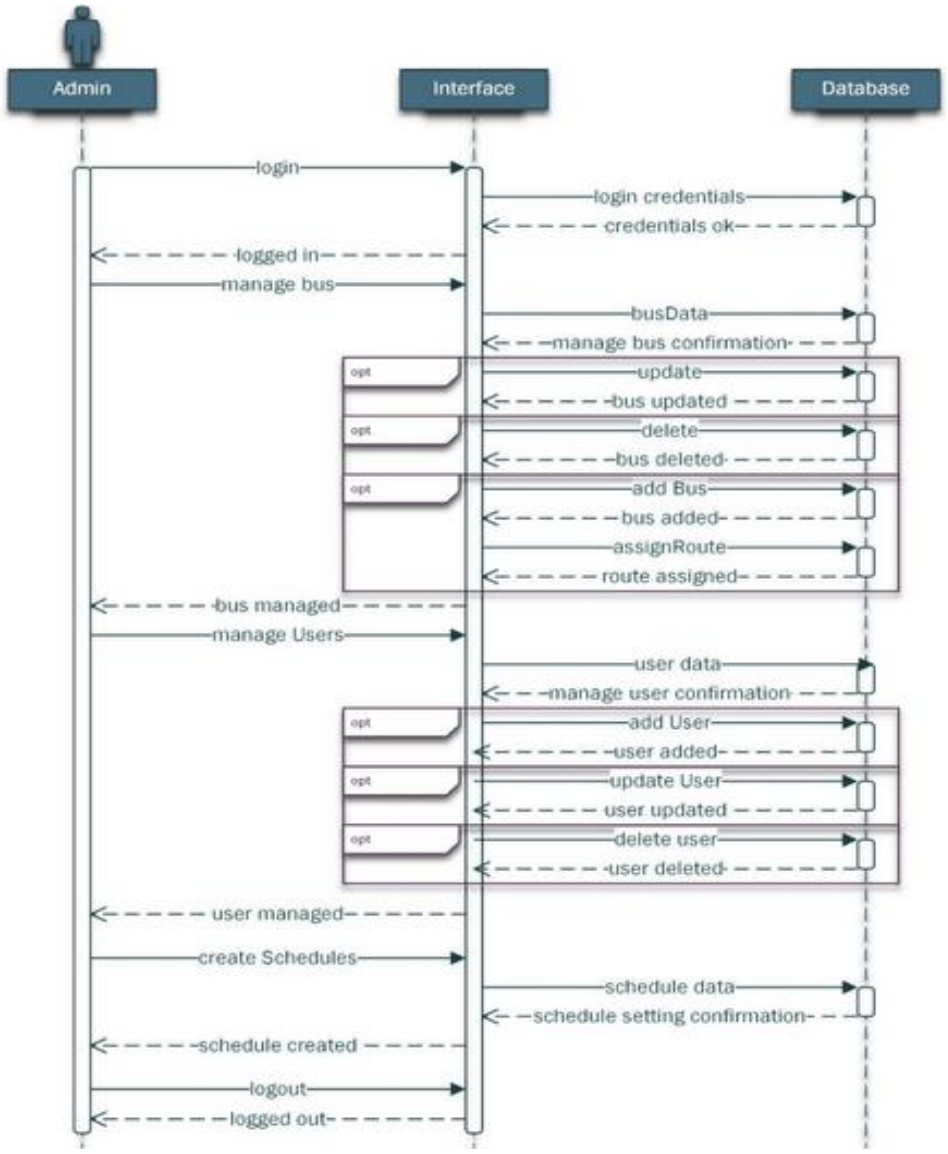


Fig-6: Admin Sequence Diagram

Chapter Five

TESTING

Testing is the process to find any deviation from the expected working of the system. If there is no deviation from the expected behaviour of the system then the project is successful otherwise failure. Testing can't be done in a full-fledged manner because of the time and budget constraints.

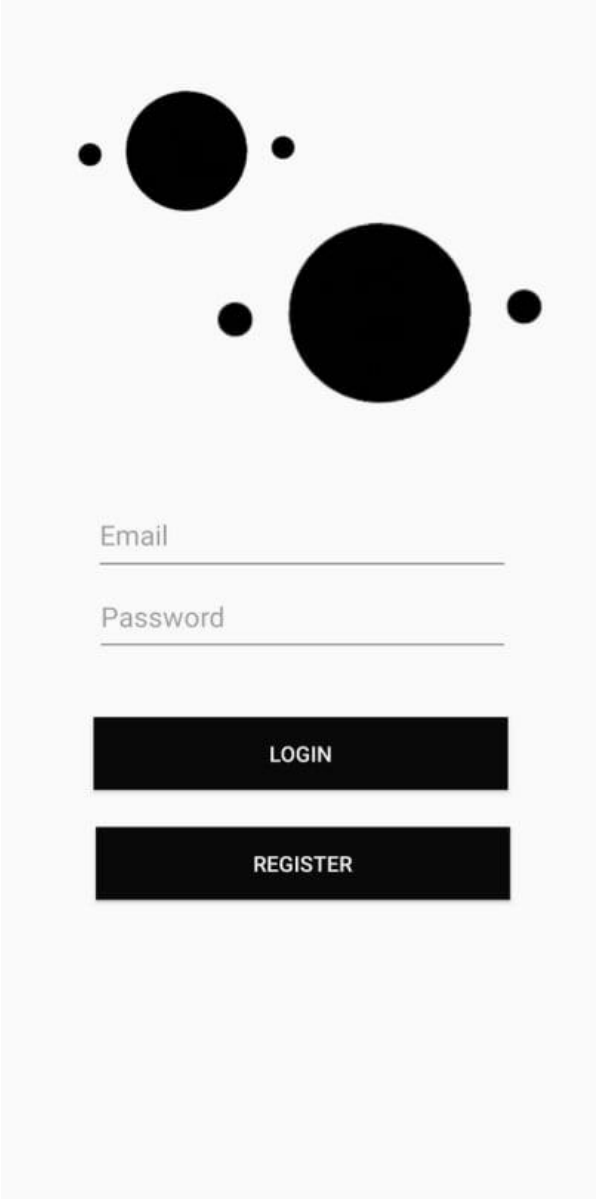
Chapter Six

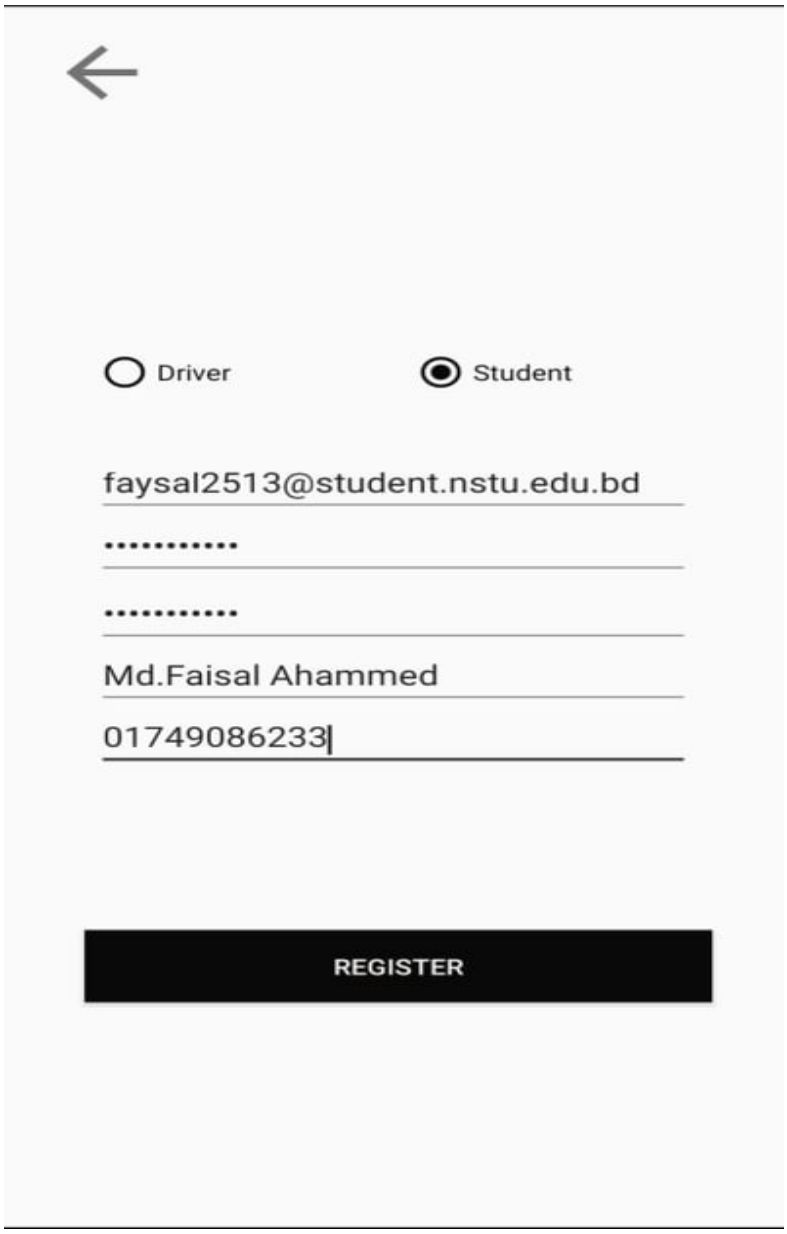
USER GUIDEBOOK OF NSTU BUS TRACKING SYSTEM

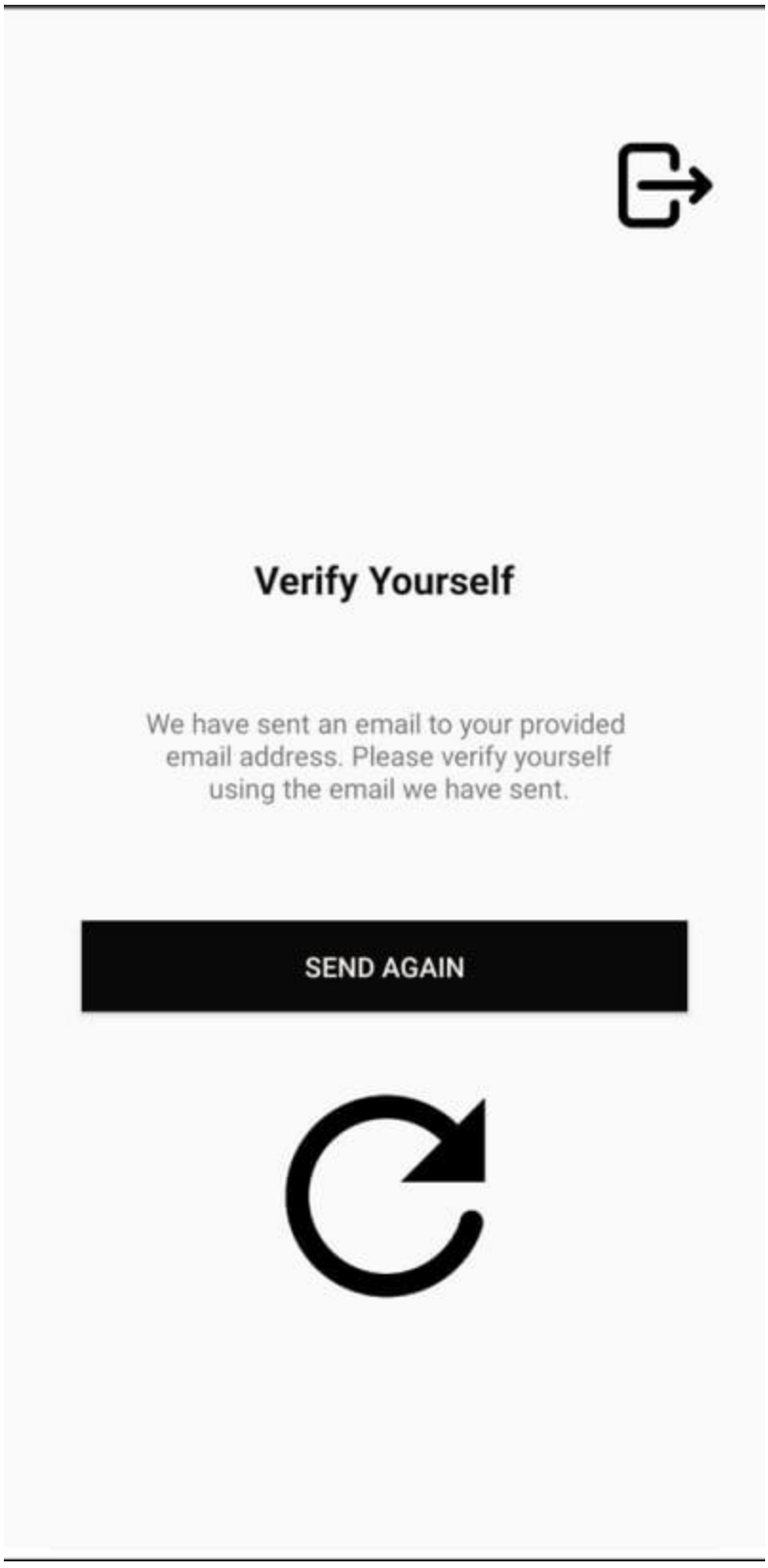
Chapter Four

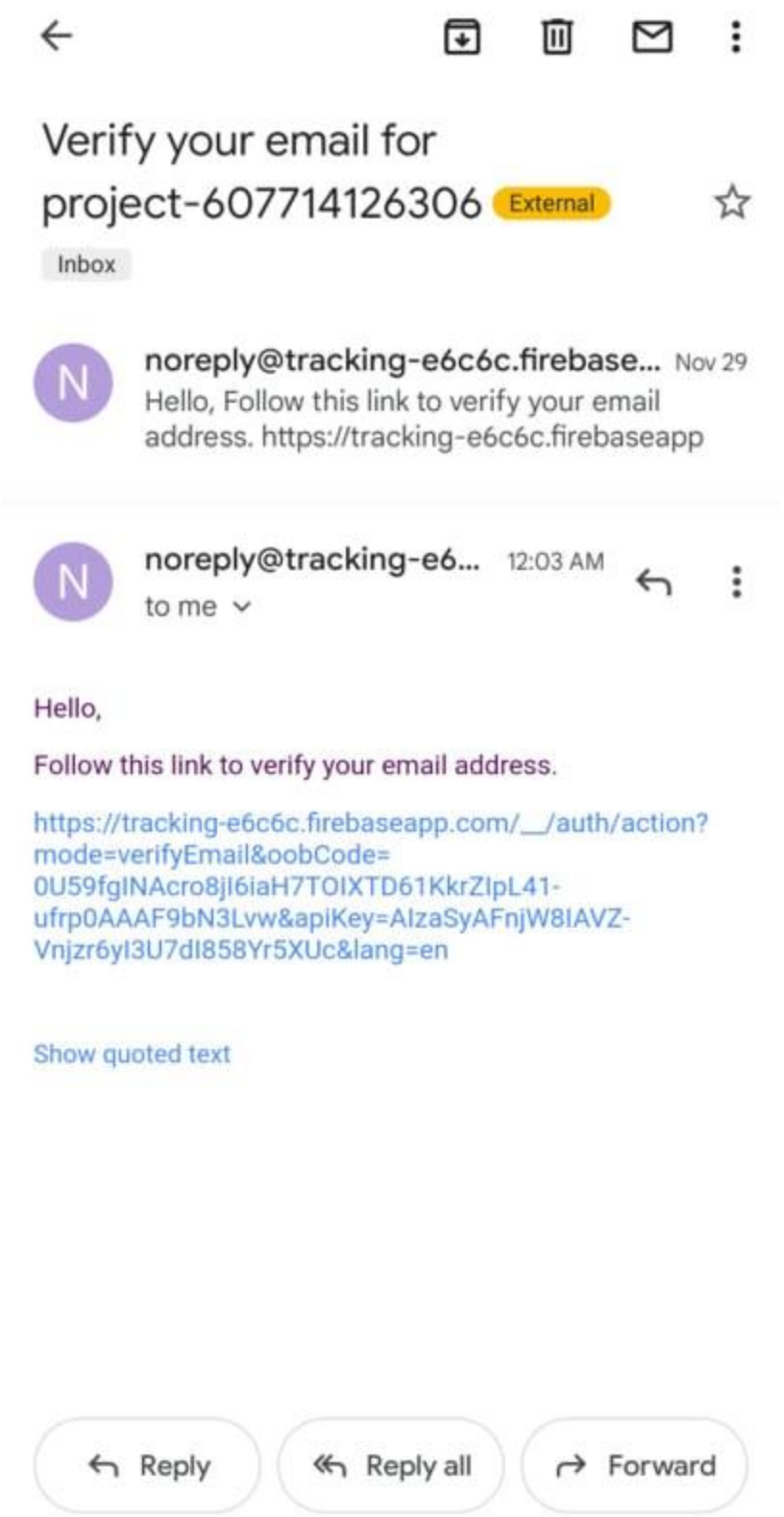
7.1 Registration

To register as a user of NSTU Bus Tracking System user at first need to click “REGISTER”. Then user need to fill all the required field and click the ‘Register’ button.



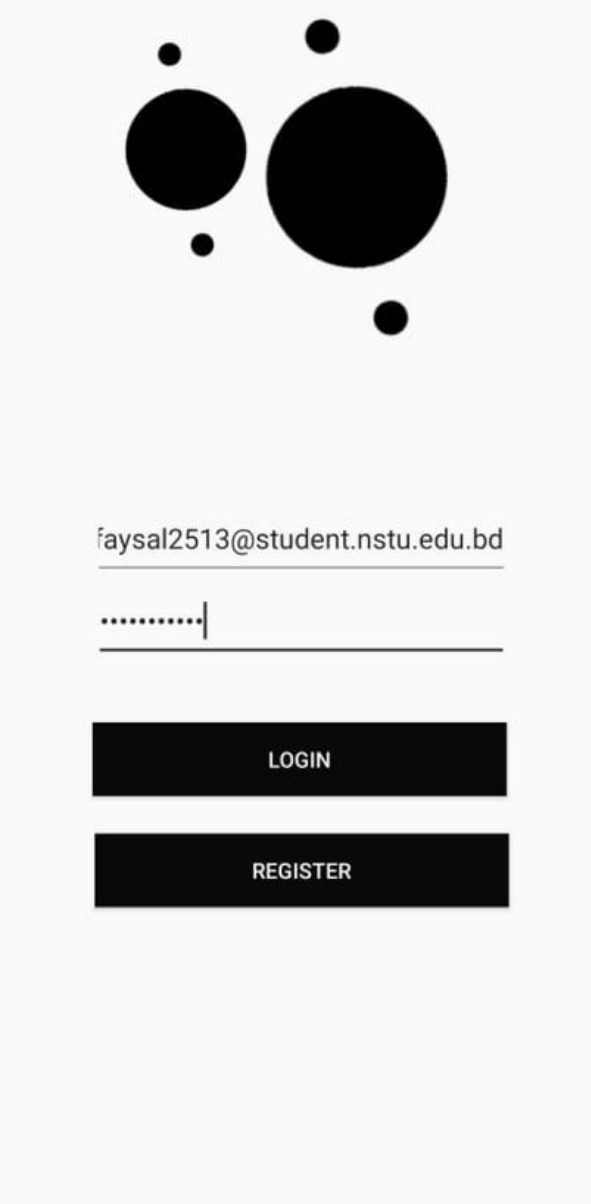






User have to provide institutional email and a valid password and then have to verify using the provided link to their institutional email to successfully register.

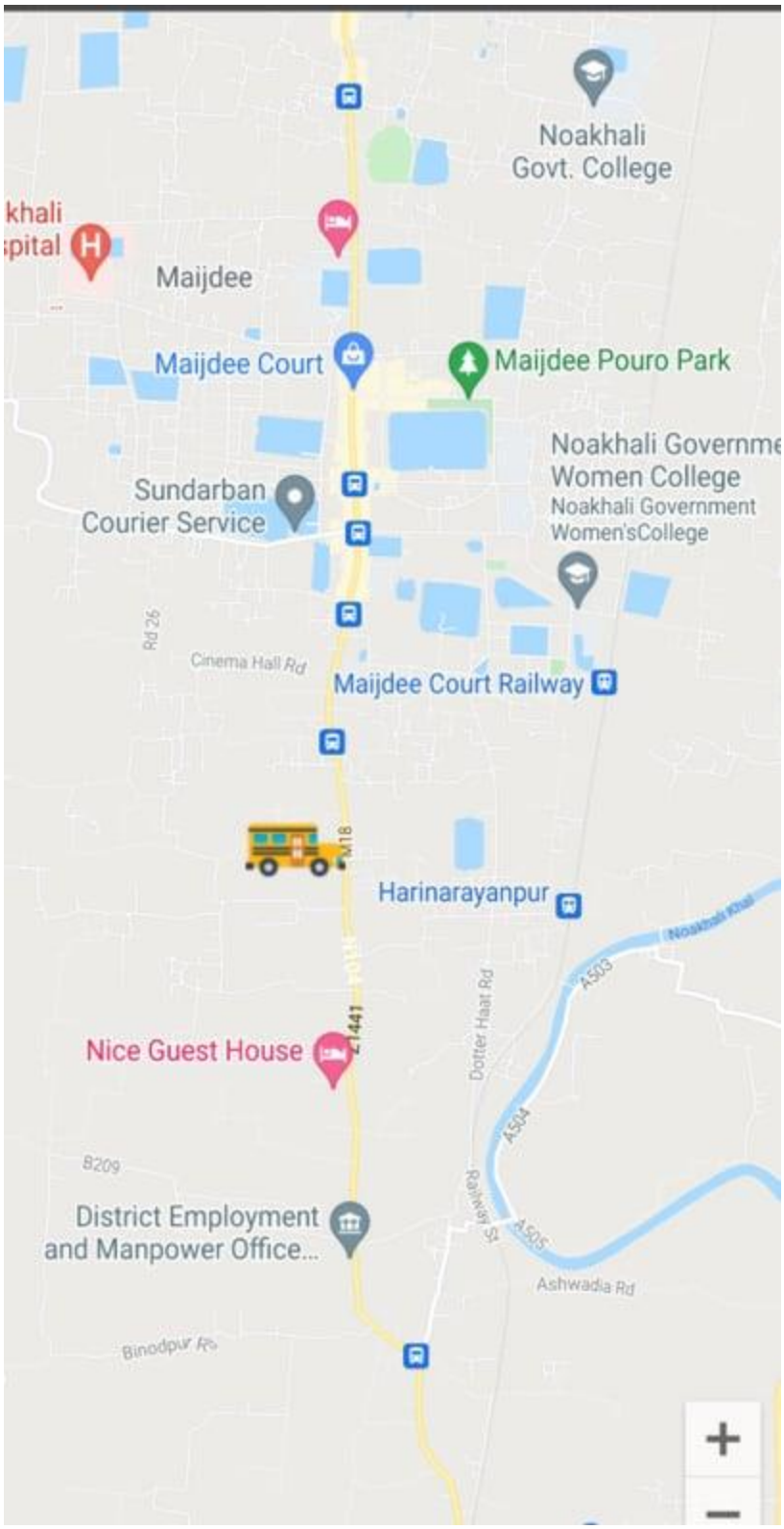
7.2 Log in



After providing a valid email and password by clicking LOGIN button a valid user can log in to the sytem

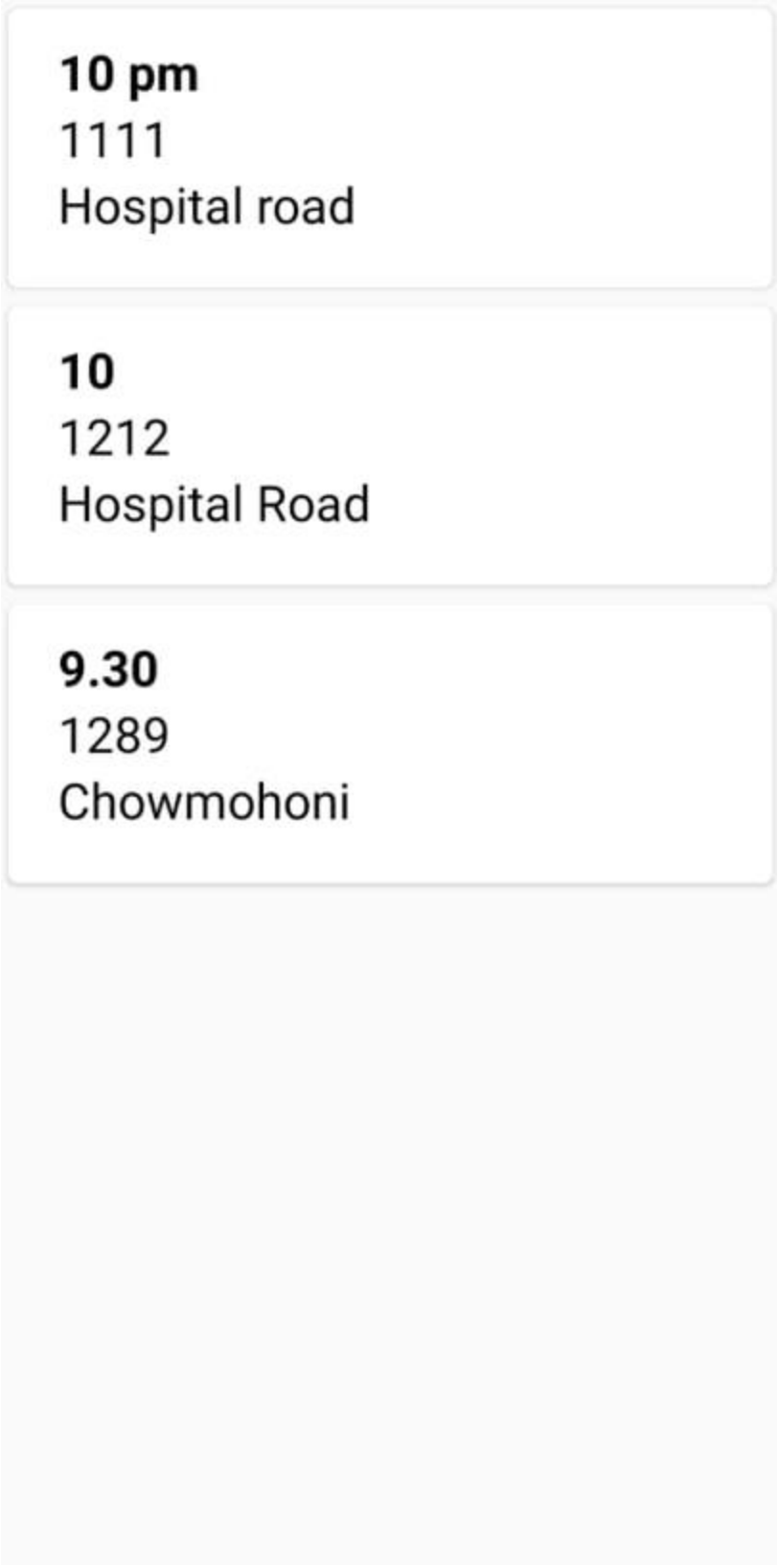
7.2 Locate Bus

A valid user can locate bus by providing legitimate email and regarding password and then click the ‘Map’ button.



7.4 Schedule

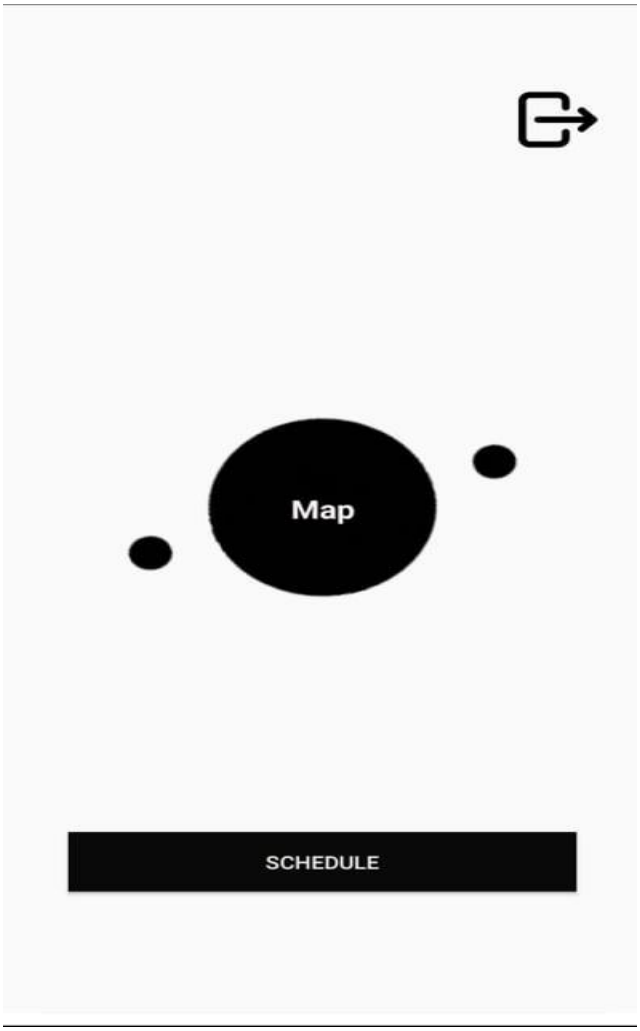
A valid user can view the bus schedule by clicking SCHEDULE button.



Administrator can create new schedule by clicking ADD button after filling the these field(Time, BusID , From).

Administrator can update a existing schedule by clicking on UPDATE button

7.5 Log out



User can Log out from the system by clicking on logout icon placing upper right side of the system.

Reference

Reference:

[1] Jisha R.C, Aiswarya Jyothindranath, Sajitha Kumary L “IoT Based School Bus Tracking and Arrival Time Prediction” 978-1-5090-6367- 3/17 ©2017 IEEE.

[2] Sumit S. Dukare, Dattatray A. Patil, Kantilal P. Rane,” Vehicle Tracking, Monitoring and Alerting System: A Review”, International Journal of Computer Applications (0975 – 8887) Volume 119 – No.10, June 2015.

[3] Ahlam, M. A.,(2016) "Taibah Track Bus Mobile Application", Taibah University, Almadinah Almunawarra, Saudia.

[4] Priya, B.,(2015) "A Mobile Application for Tracking College Bus Using Google Map", International Journal Computer Science and Engeneering Communications, vol. 3, no. 3, pp. 1057-1061.

[5] L. Singla, “GPS Based Bus Tracking System,” (2016) IEEE International Conference on Computer Communication and Control, IC4 2015, art. no. 7375712.