



Daffodil
International
University

Daffodil Bus Tracker

Submitted By:

Md. Rashedul Islam
181-35-293

Department of Software Engineering
Daffodil International University

Supervised By:

Mr. S A M Matiur Rahman
Associate Professor

Department of Software engineering
Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.

APPROVAL

This **project** entitled on “**Daffodil Bus Tracker**”, submitted by **Md. Rashedul Islam (ID: 181-35-293)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University



External Examiner

Md. Tanvir Quader
Senior Software Engineer
a2i Programme



Internal Examiner 1

S A M Matiur Rahman
Associate Professor
Department of Software Engineering
Daffodil International University



Internal Examiner 2

Syeda Sumbul Hossain
Lecturer (Senior Scale)
Department of Software Engineering
Daffodil International University

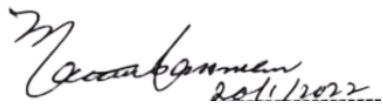
Declaration

I hereby declare that, this project has done by me under the supervisor of Mr. S A M Matiur Rahman, Associate Professor and Associate Head, Department of Software Engineering, Daffodil International University. I also declare that neither this project nor any part of this project has submitted elsewhere for award of any degree of diploma.



.....
Md. Rashedul Islam
ID: 181-35-293
Batch: 25
Department of Software Engineering
Faculty of Science and Information Technology

Certified By:



.....
Mr. S A M Matiur Rahman
Associate Professor and Associate Head,
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Acknowledgement

First, I would like to thank almighty Allah for giving me the proper time to successfully complete the project. Then I want to thank my respectable supervisor “Mr. S A M Matiur Rahman” Faculty of software Engineering, Daffodil International University, Dhaka for his excellent guidance and for having trust in my capability to complete this project. I want to express our heartiest gratitude to Head, Department of SWE, for encouraging me and providing me with such an opportunity. I am grateful to my parents as well as family members for their understanding, encouragement and endless support throughout the duration of my studies. I would like to extend my best wishes to all the teachers, friends and staff member of our department.

Project Summary

This is a web-based system to provide user about bus information in real time. User can easily search their expected bus by providing query. User can see details of a bus such as departure time, routes etc. User can select a specific bus and see its real-time location. As a result, User will not worry about Bus current location and they can utilize their valuable time.

Table of Contents

Contents

APPROVAL	II
DECLARATION.....	III
ACKNOWLEDGEMENT	IV
PROJECT SUMMARY	V
TABLE OF CONTENTS	VI
CHAPTER 01: INTRODUCTION.....	1
1.1 INTRODUCTION	2
1.2 PURPOSE	2
1.3 PROJECT OVERVIEW	2
1.4 PROBLEM STATEMENT	2
1.5 GOALS	2
1.6 PROJECT SCOPE.....	3
1.7 STAKEHOLDERS	3
1.8 PROJECT SCHEDULE.....	4
1.9 RELEASE PLAN	4
CHAPTER 2: SYSTEM ANALYSIS.....	5
2.1 REQUIREMENT ANALYSIS	6
2.2 REQUIREMENT GATHERING.....	6
2.3 FEATURES:	7
2.4 REQUIREMENTS SPECIFICATION	7
2.4.1 Functional Requirement	7
2.5.2 Non-Functional Requirements:	11
2.6 USE CASE OF PROPOSED SYSTEM	12
2.6.2 Use case table.....	13
CHAPTER 3: SYSTEM DESIGN	18
3.1.1 ACTIVITY DIAGRAM	19
3.1.2 ACTIVITY DIAGRAM	20
3.1.3 ACTIVITY DIAGRAM	21

3.2.1 DATA FLOW DIAGRAM (LEVEL-0)	22
3.2.2 DATA FLOW DIAGRAM.....	23
3.3.1 ER DIAGRAM	24
3.4.1 CLASS DIAGRAM	25
CHAPTER 4: TESTING	26
4.1 INTRODUCTION	27
4.2 FEATURES TO BE TESTED	27
4.3 TEST CASE TABLE:.....	28
CHAPTER 5: DEVELOPMENT.....	34
5.1 TOOLS AND TECHNOLOGIES:	35
5.2 REASON BEHIND CHOOSING	35
CHAPTER 6: USER MANUAL	36
6.1 HOW TO SEARCH BUS:	37
6.2 HOW TO SEE LOCATION OF A BUS:	39
6.3 HOW TO ADD TO FAVORITE LIST:	40
6.4 HOW TO LOGIN	40
6.5 HOW TO LOGOUT	40
CHAPTER 7: CONCLUSION.....	43
7.1 GITHUB LINK:.....	44
7.2 PROJECT SUMMARY	44
7.3 LIMITATIONS	44
7.4 FUTURE SCOPE.....	44

List of Figures:

FIGURE 1: USE CASE DIAGRAM	12
FIGURE 2: ACTIVITY DIAGRAM FOR TRAVELER.....	19
FIGURE 3: ACTIVITY DIAGRAM FOR ADMIN	20
FIGURE 4: ACTIVITY DIAGRAM FOR DRIVER	21
FIGURE 5: DATA FLOW DIAGRAM	22
FIGURE 6: DATA FLOW DIAGRAM (LEVEL-1)	23
FIGURE 7: ENTITY RELATIONSHIP DIAGRAM.....	24
FIGURE 8: CLASS DIAGRAM	25
FIGURE 9: BUS SEARCHING (USER MANUAL).....	37
FIGURE 10: PLACE SELECTION (USER MANUAL)	37
FIGURE 11: SEARCH	37
FIGURE 12: BUS LIST (USER MANUAL)	38
FIGURE 13: DAY SELECTION (USER MANUAL).....	38
FIGURE 14: TAP LOCATION BUTTON (USER MANUAL)	39
FIGURE 15: BUS LOCATION ON MAP (USER MANUAL).....	39
FIGURE 16: START SERVICE BUTTON.....	41
FIGURE 17: LOCATION PERMISSION REQUEST (USER MANUAL)	41
FIGURE 18: STOP BUTTON ACTIVE (USER MANUAL)	42

List of Tables:

TABLE 1: PROJECT SCHEDULE.....	4
TABLE 2: RELEASE PLAN	4
TABLE 3: LOGIN (FUNCTIONAL REQUIREMENT)	7
TABLE 4: SEARCH BUS (FUNCTIONAL REQUIREMENT)	7
TABLE 5: INPUT SEARCH QUERY (FUNCTIONAL REQUIREMENT)	8
TABLE 6: SEE REAL TIME LOCATION (FUNCTIONAL REQUIREMENT)	8
TABLE 7: SEE BUS DETAILS (FUNCTIONAL REQUIREMENT).....	8
TABLE 8: SET FAVORITE BUS (FUNCTIONAL REQUIREMENT).....	9
TABLE 9: ADD USER (FUNCTIONAL REQUIREMENT).....	9
TABLE 10: CHANGE USER CREDENTIALS (FUNCTIONAL REQUIREMENT)	9
TABLE 11: ADD NEW USER (FUNCTIONAL REQUIREMENT).....	9
TABLE 12: REMOVE SPECIFIC BUS (FUNCTIONAL REQUIREMENT).....	10
TABLE 13: MODIFY INFORMATION (FUNCTIONAL REQUIREMENT).....	10
TABLE 14: LOGOUT (FUNCTIONAL REQUIREMENT).....	10
TABLE 15: ACCURACY (NON-FUNCTIONAL REQUIREMENT)	11
TABLE 16: AVAILABILITY (NON-FUNCTIONAL REQUIREMENT)	11
TABLE 17: SPEED (NON-FUNCTIONAL REQUIREMENT)	11
TABLE 18: LOGIN (USE CASE)	13
TABLE 19: SEARCH BUS (USE CASE)	13
TABLE 20: SHOW LOCATION (USE CASE)	14
TABLE 21: SEE BUS DETAILS (USE CASE)	14
TABLE 22: CHANGE USER PASSWORD & ID (USE CASE)	15
TABLE 23: ADD NEW USER (USE CASE)	15
TABLE 24: REMOVE USER (USE CASE)	16
TABLE 25: MODIFY EXISTING BUS (USE CASE).....	16
TABLE 26: ADD NEW BUS (USE CASE).....	17
TABLE 27: LOGIN (TEST CASE)	28
TABLE 28: LOGOUT (TEST CASE)	29
TABLE 29: BUS SEARCHING (TEST CASE).....	30
TABLE 30: ADD BUS.....	31
TABLE 31: REMOVE BUS.....	32
TABLE 32: MODIFY BUS INFORMATION (TEST CASE).....	33

Chapter 01: Introduction

1.1 Introduction

This documentation will give a good understanding of the project. In this documentation, I have discussed detailed information about the project. The reader can have a good understanding of the project after reading this.

1.2 Purpose

The prime purpose of this documentation is a written description of the project. Normal users and developers can benefit from this. After reading, a usual user can better utilize the system. Developers can understand the function for further modification of the system without effort.

1.3 Project Overview

The project "Daffodil Bus Tracking" is a web-based project. Users can see the real-time location of a specific bus. They will be updated all the time they want to trace. Users can also search buses and get a list of buses according to the user's search query. Users can set their favorite bus list to get a location without searching bus. They also have a list of the recent searched bus.

1.4 Problem Statement

Waiting for a bus is a time-consuming and monotonous job. Sometimes, people think about the bus location. Sometimes, they get tense. Sometimes they want to about availability of bus. People need to know about the route of the bus. These all issues will solve by using this system.

1.5 Goals

The main goal of this project is to give a user-friendly system to get the real-time location of a bus. The system will be fast, responsive, well organized. Any type of person can use it.

1.6 Project Scope

By using this system, a person can do several actions. Such as bus searching, seeing the routes of the bus, and getting the real-time location of a specific bus. They can also get the schedule of the bus. Users can see instant information about the bus's current state.

1.7 Stakeholders

There are three types of stakeholders in this system. Such as,

1. Traveler
2. Admin
3. Driver

Following are descriptions of stakeholders:

1. Traveler: They are the non-registered user. They need not register to use this system. They can search the bus, see the router of a specific bus and see the location of a particular bus. They can also see the bus schedule.
2. Admin: Admin is a superuser. Admin needs to log in to enter this system. Admin can add/remove and modify any information.
3. Driver: The driver is another user. A driver operates a locator device. It always stays active in the bus while the bus is in service.

1.8 Project Schedule

Table 1: Project Schedule

Activities	Duration(in week)	Total week
Brainstorming	Week-1, Week-2, Week-3	3
Problem Identification	Week-4, Week-5	2
Requirements Analysis	Week-6,week-7,week-8	3
Sketching	Week-9	1
Design Specification	Week-10,week-11	2
Database design	Week-12	1
Implementation	Week-13,week-14, week-15	3
Testing	Week-16, week-17	2
Delivery	Week-18	1

1.9 Release Plan

Table 2: Release Plan

Version	Features	Date
V-0.0.1	Map integration, Bus schedule, Login from locator device, Favorite bus list, search functionality	05-11-2021
V-0.0.2	Recently searched bus list, faster system	10-12-2021

Chapter 2: System Analysis

2.1 Requirement Analysis

Things that are needed to present in the project are requirements. Requirements analysis is a process to select what type of requirements we need to have in the system to successfully run the system. Therefore, this part is very much important to complete the project. Good analysis help to make a good quality project.

2.2 Requirement Gathering

Requirements gathering is a process to collect requirements. There are some ways to gather requirements. I have used brainstorming, questioning, interviewing, observing another system, customer observation.

Brainstorming: I have gotten many ideas from brainstorming. I have thought from the user's perspective and driver's perspective and got many ideas.

Questioning and Interviewing: I have questioned some travelers and frequently asked them several questions and got more ideas.

Customer Observation and other system observation: I have observe some other similar system like Uber and google map to get more idea.

2.3 Features:

- Bus Real Time Location.
- Bus Searching.
- Frequently searched bus.
- Favorite bus.
- Bus Filtering.
- Bus Active Status.
- Show Bus Information.
- Bus Data Management

2.4 Requirements Specification

2.4.1 Functional Requirement

Functions that a system must offer are functional requirements. It describe what function can be performed using the system.

Table 3: Login (Functional Requirement)

ID: FR-01
Requirement Name: Login
Description: user can login by providing valid credentials.
Stakeholders: Admin, Bus Driver

Table 4: Search Bus (Functional Requirement)

ID: FR-02
Requirement Name: Search Bus
Description: user can search bus by providing valid search query such as destination place, day, starting place.
Stakeholder: Traveler

Table 5: Input Search Query (Functional Requirement)

ID: FR-03
Requirement Name: Input Search Query
Description: user can input search query from drop down menu.
Stakeholder: Traveler

Table 6: See Real Time Location (Functional Requirement)

ID: FR-04
Requirement Name: See Real Time Location
Description: user can see real-time location of a specific bus.
Stakeholder: Traveler

Table 7: See Bus Details (Functional Requirement)

ID: FR-05
Requirement Name: See Bus Details.
Description: User can search bus and get a list according to search query and can see bus details such as, route information, starting time.
Stakeholder: Traveler

Table 8: Set Favorite Bus (Functional Requirement)

ID: FR-06
Requirement Name: Set Favorite Bus.
Description: User can search bus set this as a favorite. A list will appear besides search box containing favorite list.
Stakeholder: Traveler

Table 9: Add User (Functional Requirement)

ID: FR-07
Requirement Name: add user
Description: Only admin can register a new user. User can be a new bus or another admin.
Stakeholder: Admin

Table 10: Change User Credentials (Functional Requirement)

ID: FR-08
Requirement Name: Change user credentials
Description: Admin can change user credential. Suppose imagine a situation where a bus locator device operator forget password and then they need to inform admin and then admin can change the password and give them new password.
Stakeholder: Admin

Table 11: Add New User (Functional Requirement)

ID: FR-09
Requirement Name: Add new Bus
Description: Admin can add new bus information.
Stakeholder: Admin

Table 12: Remove Specific Bus (Functional Requirement)

ID: FR-10
Requirement Name: Remove specific bus
Description: Admin can remove existing bus information.
Stakeholder: Admin

Table 13: Modify Information (Functional Requirement)

ID: FR-11
Requirement Name: Modify information
Description: Admin can modify information about bus and user also.
Stakeholder: Admin

Table 14: Logout (Functional Requirement)

ID: FR-12
Requirement Name: Logout
Description: registered user can logout from the system.
Stakeholder: Admin, Bus Driver

2.5.2 Non-Functional Requirements:

Non-functional requirements of this project are given below:

Table 15: Accuracy (Non-Functional Requirement)

ID: NFR-01
Requirement Name: Accuracy
Description: System should be accurate enough to broadcast real time location among users and quick response after performing search.
Stakeholder: Traveler

Table 16: Availability (Non-Functional Requirement)

ID: NFR-02
Requirement Name: Availability
Description: System must be available in service time.
Stakeholder: all user

Table 17: Speed (Non-Functional Requirement)

ID: NFR-01
Requirement Name: Fast
Description: System must be fast enough to response.
Stakeholder: all user.

2.6 Use case of proposed system

2.6.1 Use case Diagram:

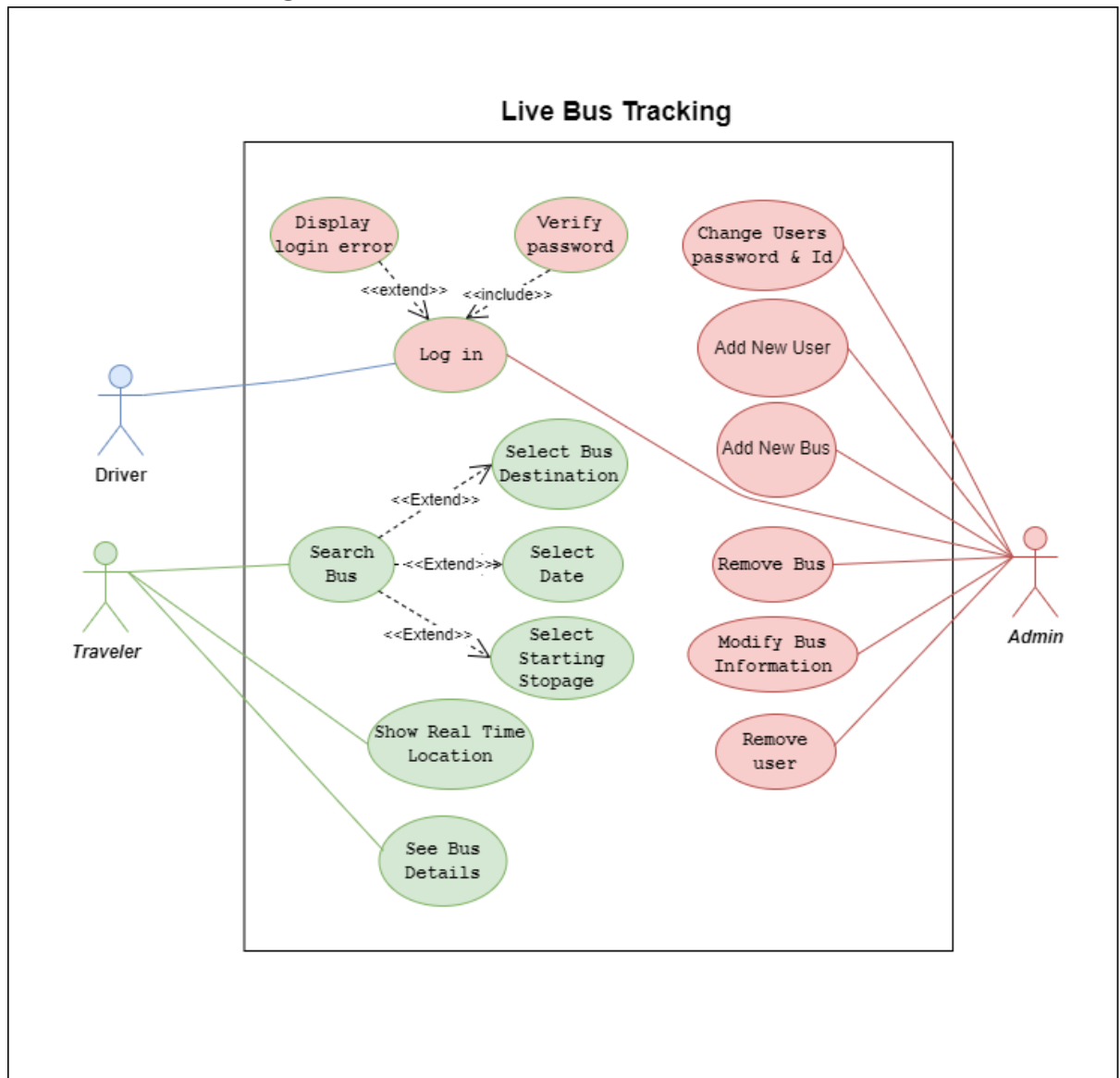


Figure 1: Use Case Diagram

2.6.2 Use case table

Table 18: Login (Use Case)

Use Case ID	UC-01
Use Case Title	Login
Precondition	1. User must be registered 2. User must fill up the form
Actors	Admin, Driver
Success End State	Show home page
Failure End State	Show error message in same page.
Trigger	Redirect to home page
Description	By login user can prove they are valid user. This is for security purpose.

Table 19: Search Bus (Use Case)

Use Case ID	UC-02
Use Case Title	Search Bus
Precondition	User must fill up at least one search input field.
Actors	Traveler
Success End State	Show a list bus.
Failure End State	Show not found message
Trigger	Show a list of bus in same page
Description	To perform search user need to fill up at least one input field. User can select value from dropdown menu and then perform search.

Table 20: Show Location (Use Case)

Use Case ID	UC-03
Use Case Title	Show location
Precondition	User must select a bus by performing search / from a favorite list/ recent searched list.
Actors	Traveler
Success End Condition	User can see the bus location in google map.
Failure End Condition	Show an error message.
Trigger	User can see the bus real time location in same page.
Description	User must select a bus and click location button.

Table 21: See Bus Details (Use Case)

Use Case ID	UC-04
Use Case Title	See Bus Details
Precondition	1. User must select a bus.
Actors	Traveler
Success End Condition	User can see details of a bus
Failure End Condition	See error an error message
Trigger	After clicking details, button user can see details about the bus.
Description	User can see bus details after performing this action. Details like departure time, routes etc.

Table 22: Change User Password & Id (Use Case)

Use Case ID	UC-05
Use Case Title	Change user password & ID
Precondition	User must be logged in.
Actors	Admin
Success End Condition	Can see changes
Failure End Condition	Show error message
Trigger	Can see changes in same page
Description	User ID & password can be change by performing this.

Table 23: Add New User (Use Case)

Use Case ID	UC-06
Use Case Title	Add new user
Precondition	1. User must be logged in. 2. Input field must be filled.
Actors	Admin
Success End Condition	Added successfully.
Failure End Condition	Show error message
Trigger	Added user will show.
Description	Input new user information and click add user button to add new user.

Table 24: Remove User (Use Case)

Use Case ID	UC-07
Use Case Title	Remove user
Precondition	1. User must be logged in. 2. User must select a user.
Actors	Admin
Success End Condition	User will be removed.
Failure End Condition	Show error message in same page.
Trigger	Show success message and remove user from list.
Description	To remove a user. Admin must select one or more user and perform remove.

Table 25: Modify Existing Bus (Use Case)

Use Case ID	UC-08
Use Case Title	Modify existing bus
Precondition	1. Admin must logged in. 2. Admin must select bus to modify.
Actors	Admin
Success End Condition	Show success message and modified information.
Failure End Condition	Show error message.
Trigger	Show success message and modified information.
Description	User must select bus and input updated value and perform action.

Table 26: Add New Bus (Use Case)

Use Case ID	UC-09
Use Case Title	Add new bus
Precondition	1. User must logged in. 2. Fill up input field with new value.
Actors	Admin
Success End Condition	Show added value and success message.
Failure End Condition	Show error message with instruction.
Trigger	Added new information in same page.
Description	To add new bus information user must input new information and perform addition.

Chapter 3: System Design

3.1.1 Activity Diagram

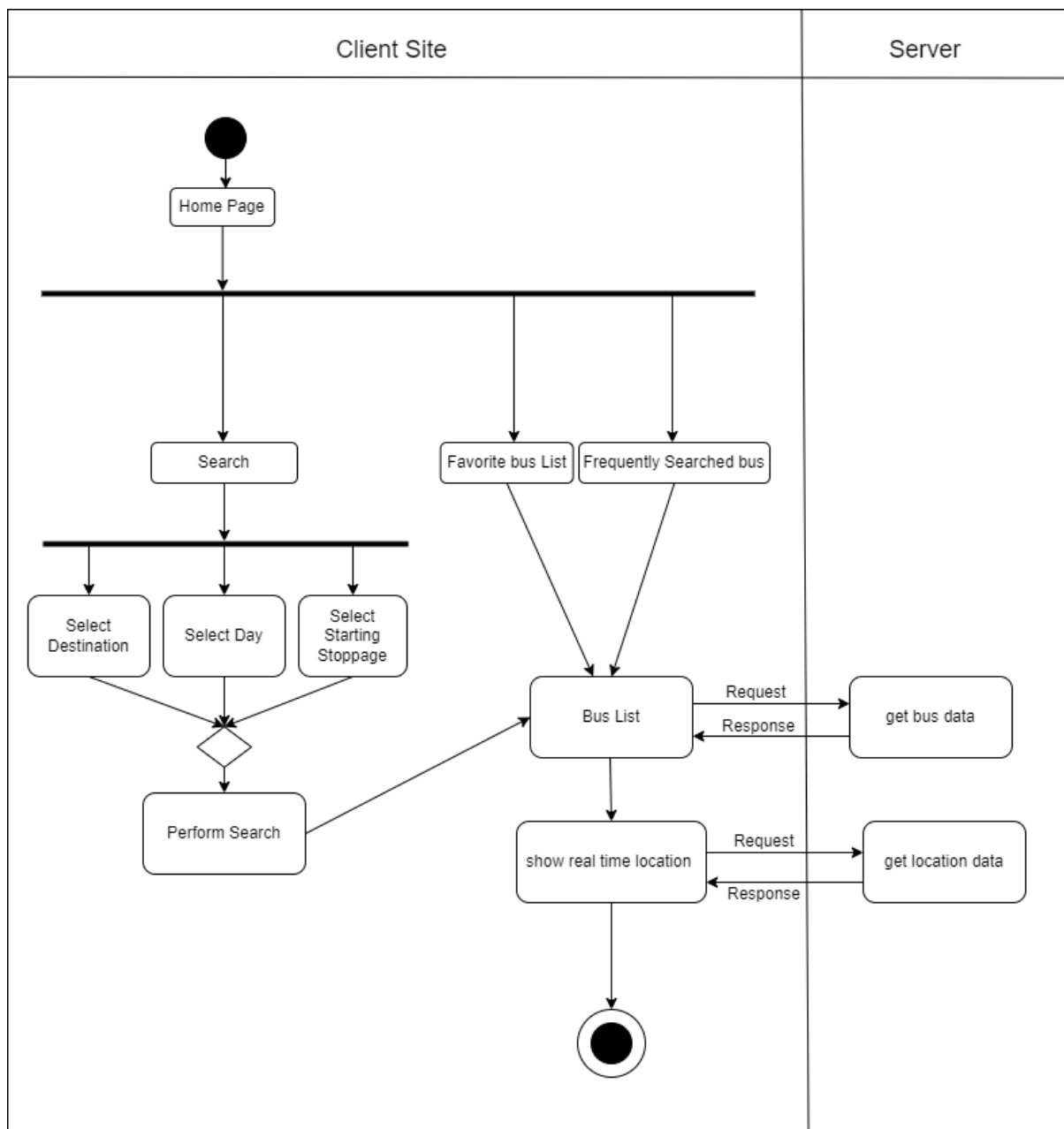


Figure 2: Activity diagram for Traveler

3.1.2 Activity Diagram

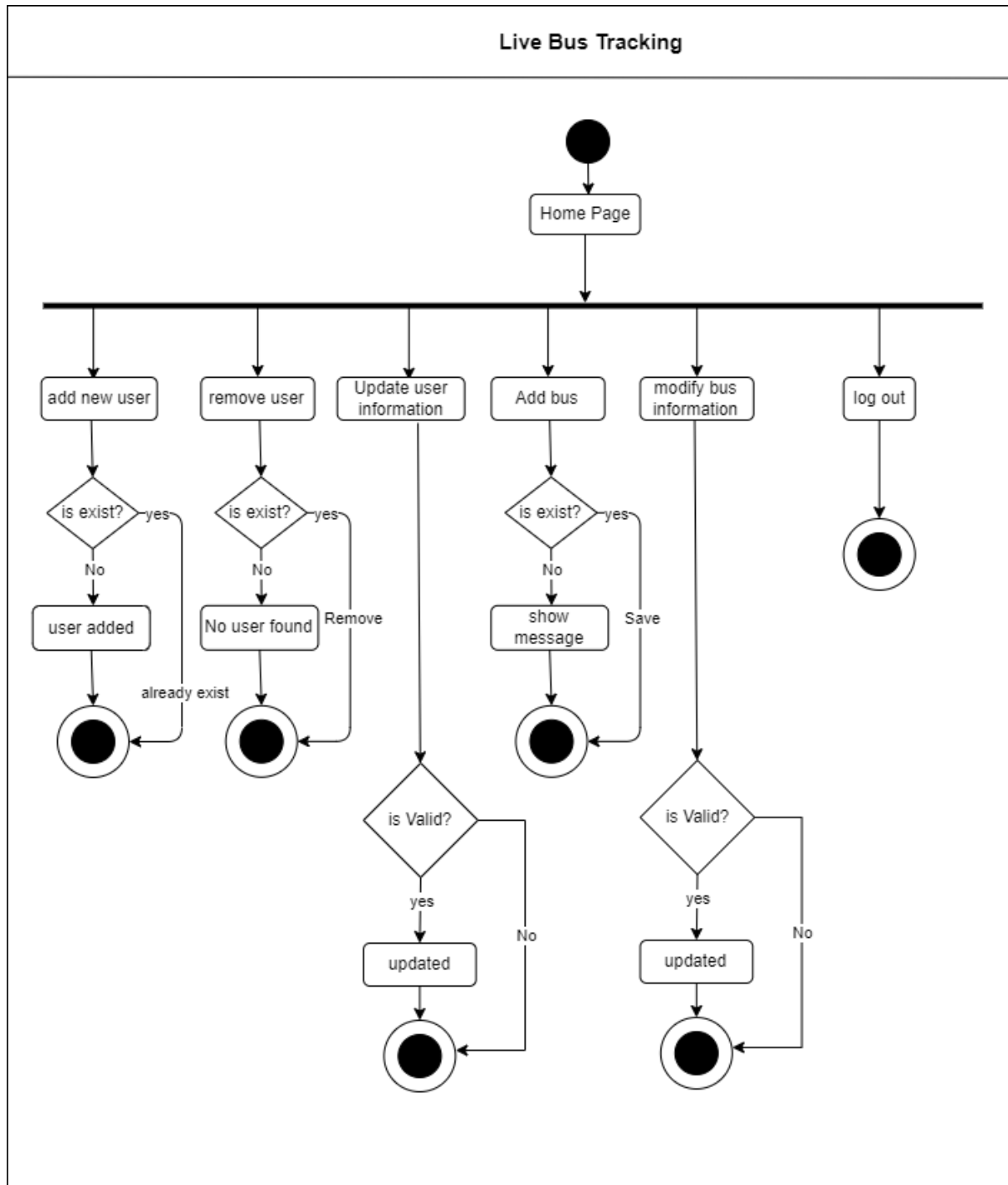


Figure 3: Activity diagram for Admin

3.1.3 Activity Diagram

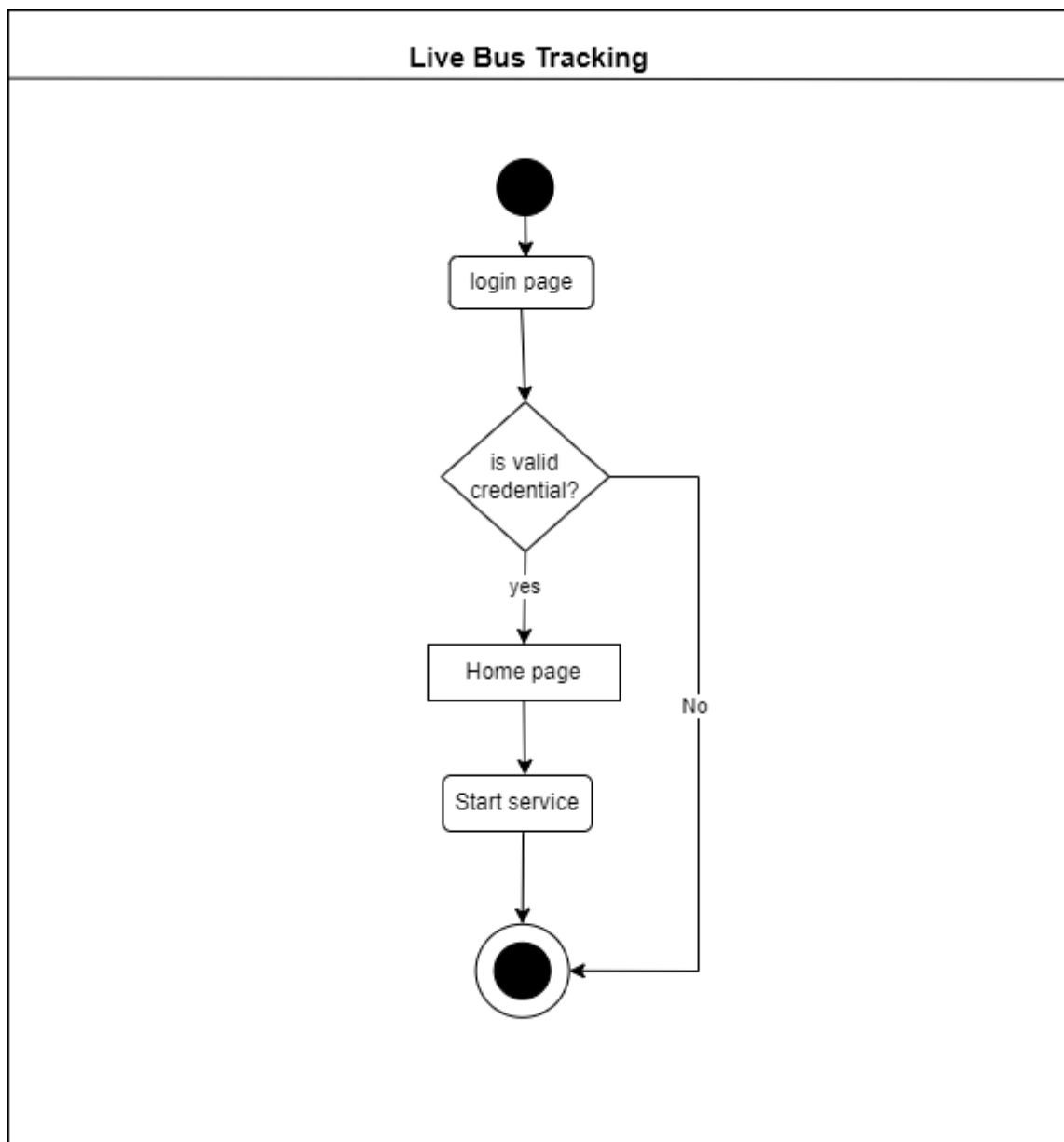


Figure 4: Activity Diagram for Driver

3.2.1 Data flow diagram (level-0)

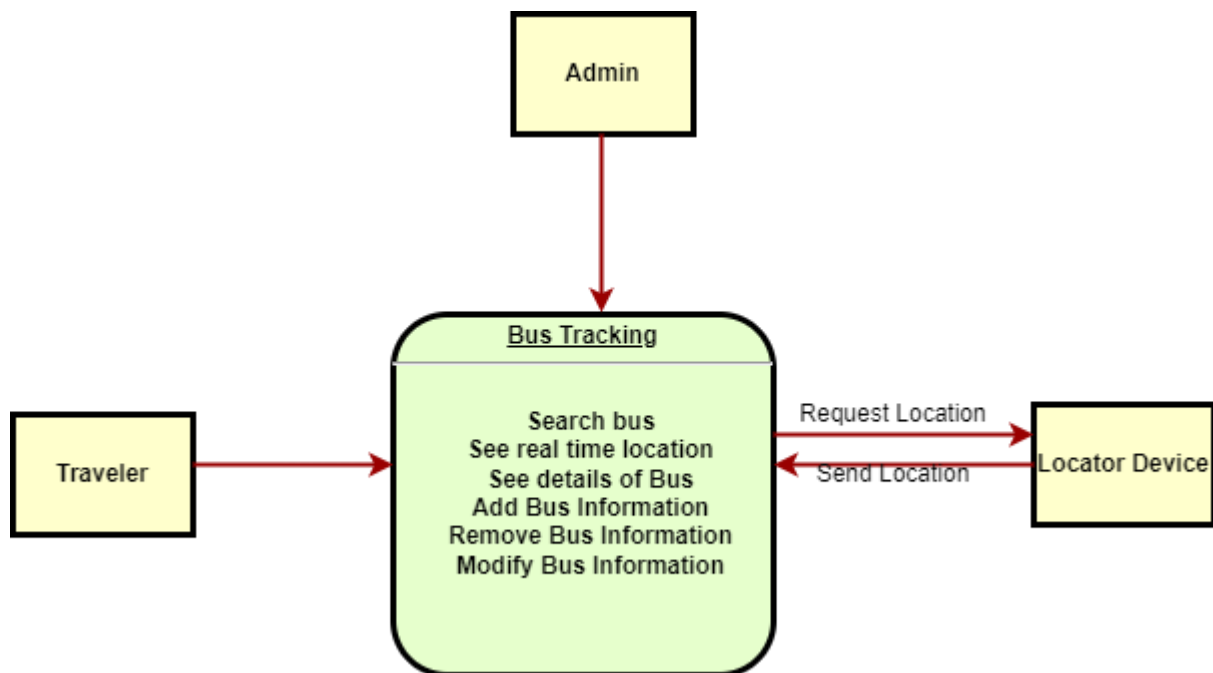


Figure 5: Data Flow Diagram

3.2.2 Data flow diagram

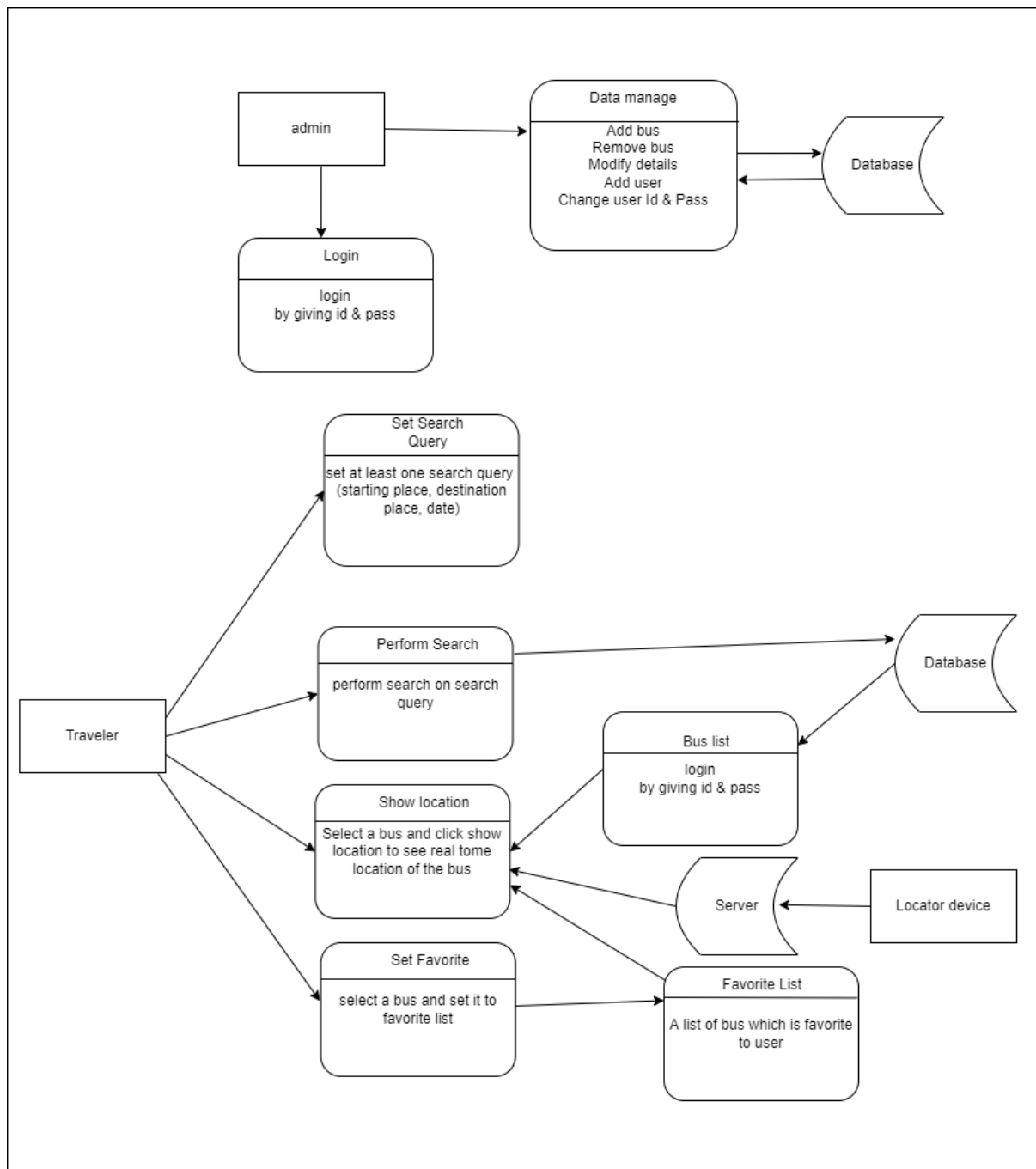


Figure 6: Data Flow Diagram (Level-1)

3.3.1 ER Diagram

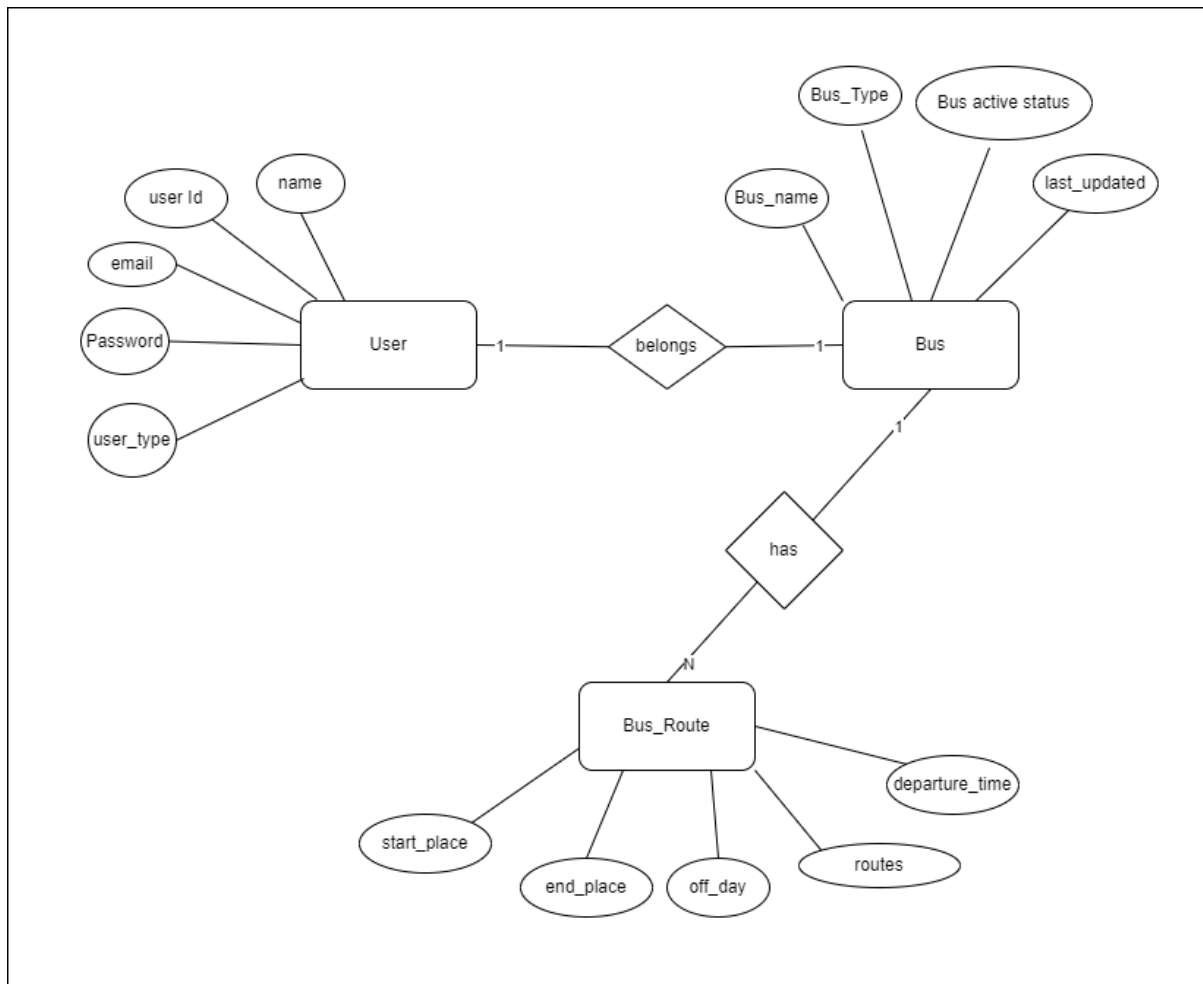


Figure 7: Entity Relationship Diagram

3.4.1 Class Diagram

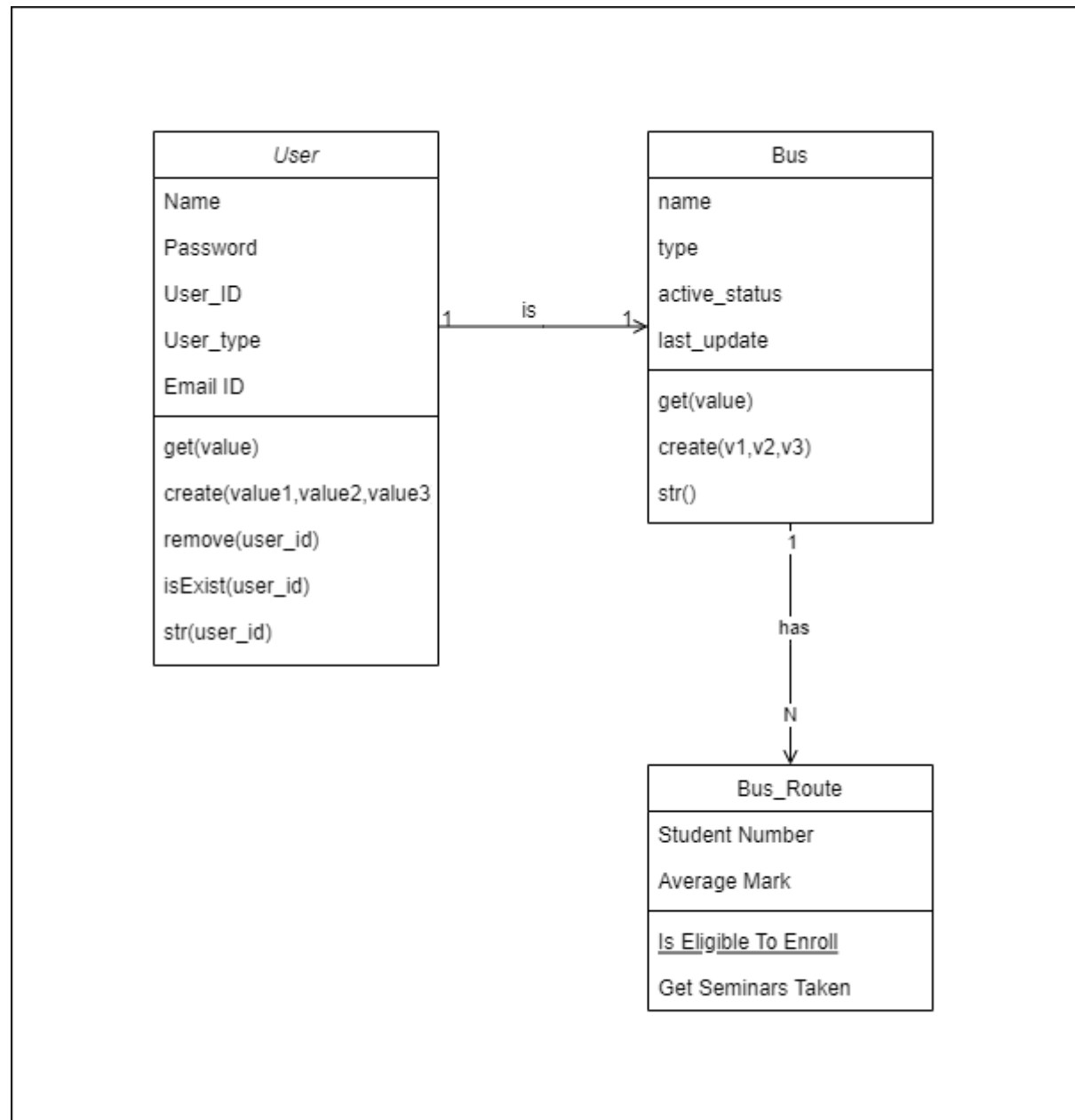


Figure 8: Class Diagram

Chapter 4: Testing

4.1 Introduction

“Live bus tracking” is a web based service application. Many user and anyone who have internet access can use this system. This system must be reliable and accurate. To ensure the quality of the system we need to perform testing. Therefore, testing is important for this system. To test this system we have many strategies. I have perform some testing strategies on this system. Such as,

1. Unit testing.
2. Integration testing.
3. Functional testing.
4. Acceptance testing.
5. System testing.

4.2 Features to be tested

Feature	Priority	Description
Login	3	Use for authentication
Logout	3	Destroy session after logout
Bus Searching	3	Search bus by setting some query
Add bus	2	To create new bus information
Remove Bus	2	Remove exiting bus.
Modify Bus	2	To modify existing bus information.

Here, 1=low priority, 2=medium priority, 3=high priority.

4.3 Test Case Table:

Table 27: Login (Test Case)

Test Case #1			Test Case Name: Login		
System: Live Bus Tracking			Sub System: N/A		
Designed by: Rashedul Islam			Designed Date: 27/11/2021		
Executed by: Rashedul Islam			Executed Date: 27/11/2021		
Short Description: user must give at least 6 digit of password. Only valid credential can successfully login the user.					
Pre-Conditions: • User must be registered. • Assume that the user name is emon293 and password is “pass#123”;					
Step	Username	Password	Expected Result	Pass/Fail	Comment
1	Emon123	pass#123	These credentials do not match with record.	pass	Password is case sensitive. For this reason, Emon123 is not valid username.
2	emon123	pass#123	Trigger to login	pass	Credentials are valid so, user logged in successfully.
3	emon	Pass#123	These credentials do not match with record.	pass	Username is incorrect.
4	Emon123	pass	These credentials do not match with record.	pass	Password is incorrect and less than 6 digit.
Post-Condition: user will login by using valid credential otherwise see an error message.					

Table 28: Logout (Test Case)

Test Case #2			Test Case Name: Logout		
System: Live Bus Tracking			Sub System: N/A		
Designed by: Rashedul Islam			Designed Date: 28/11/2021		
Executed by: Rashedul Islam			Executed Date: 28/11/2021		
Short Description: if user want to logout from system user need trigger logout button and session will destroy.					
Pre-Conditions: User must be logged in.					
Step	action	response	Expected Result	Pass/Fail	Comment
1	Trigger logout button	Successfully logout	Session will destroy and back to log in page.	pass	Session will destroy and user not able to browse as a valid user.
2	Trigger back button	User will not able to go back.	User will in same page.	pass	After logout user should not able to act as a valid user.
3	Enter previous link to enter.	User will redirect to login page.	User redirect to login page.	pass	User redirect to login page.
Post-conditions: session will destroy and user unable to act as a valid user before further login.					

Table 29: Bus Searching (Test Case)

Test Case #3				Test Case Name: Bus Searching		
System: Live Bus Tracking				Sub System: N/A		
Designed by: Rashedul Islam				Designed Date: 28/11/2021		
Executed by: Rashedul Islam				Executed Date: 28/11/2021		
Short Description: user must select at least one search query.						
Pre-Conditions: User must be in home page.						
Step	Starting place	Day	Destination Place	Expected Result	Pass/Fail	Comment
1	DSC	Saturday	Dhanmondi	These values produce a valid result.	pass	Server response according to search query.
2	Uttara	Sunday		These values produce a valid result.	pass	Server response according to search query.
3	Dhanmondi			These values produce a valid result.	Pass	Server response according to search query.
4		Monday		These values produce a valid result.	pass	Server response according to search query.
5				System expect at least one value.	pass	Client side will not sent request to server.
Post-Condition: System will show a list of result.						

Table 30: Add Bus

Test Case #4			Test Case Name: Add bus		
System: Live Bus Tracking			Sub System: N/A		
Designed by: Rashedul Islam			Designed Date: 28/11/2021		
Executed by: Rashedul Islam			Executed Date: 28/11/2021		
Short Description: user must fill all input field to add bus.					
Pre-Conditions: User must be logged in.					
Step	name	type	Expected Result	Pass/Fail	Comment
1	Surjomukhi-1	staff	Added successfully	pass	Add a bus with this information.
2	Surjomukhi-2	student	Added successfully	pass	Add a bus with this information.
3	Surjomukhi-3		will not add	pass	Bus should not add without this information.
4		staff	will not add	pass	Bus should not add without this information.
5	Surjomukhi-2	staff	Will not add	pass	Bus already exist.
Post-conditions: user will see a new bus added with this information.					

Table 31: Remove Bus

Test Case #5			Test Case Name: remove bus		
System: Live Bus Tracking			Sub System: N/A		
Designed by: Rashedul Islam			Designed Date: 29/11/2021		
Executed by: Rashedul Islam			Executed Date: 29/11/2021		
Short Description: user must fill all input field to add bus.					
Pre-Conditions: User must be logged in.					
Step	action	response	Expected Result	Pass/Fail	Comment
1	Select a bus and trigger remove button	The bus will remove from database.	remove successfully	pass	Remove selected bus
Post-conditions: user will see a new bus added with this information.					

Table 32: Modify Bus Information (Test Case)

Test Case #6			Test Case Name: Modify bus information		
System: Live Bus Tracking			Sub System: N/A		
Designed by: Rashedul Islam			Designed Date: 29/11/2021		
Executed by: Rashedul Islam			Executed Date: 29/11/2021		
Short Description: user must fill all input field to add bus.					
Pre-Conditions: User must be logged in.					
Step	name	type	Expected Result	Pass/Fail	Comment
1	Surjomukhi-2	staff	Update successfully	pass	Add a bus with this information.
2	Surjomukhi-1	staff	Update successfully	pass	Add a bus with this information.
3	Surjomukhi-3		will not add	pass	Bus should not add without this information.
4	Surjomukhi-2	staff	Will not update		Bus already exist.
5		staff	will not add	pass	Bus should not add without this information.
Post-conditions: user will see updated information.					

Chapter 5: Development

5.1 Tools and Technologies:

To develop this system I have used some tools and technology.

- Client site: Flutter.
- Server site: Django, Channel, Redis
- Data storage: MySQL
- Tools: Visual Studio Code

5.2 Reason Behind choosing

- Flutter: To design UI of Application. I have used Flutter.
- Django: To make server side application.
- Channel: To make asynchronous and socket support in server site application.
- Redis: to store and deliver different kind of events.
- MySQL: To store data.
- Visual Studio code: To write program.
- Android Studio: Ide to write program.

Chapter 6: User Manual

6.1 How to Search Bus:

To search bus you must be in home page. On top of the page, you can see a search option. There are three field beside search button. You have to fill up at least one field and then click/press search button. Then result will show underneath the search option.

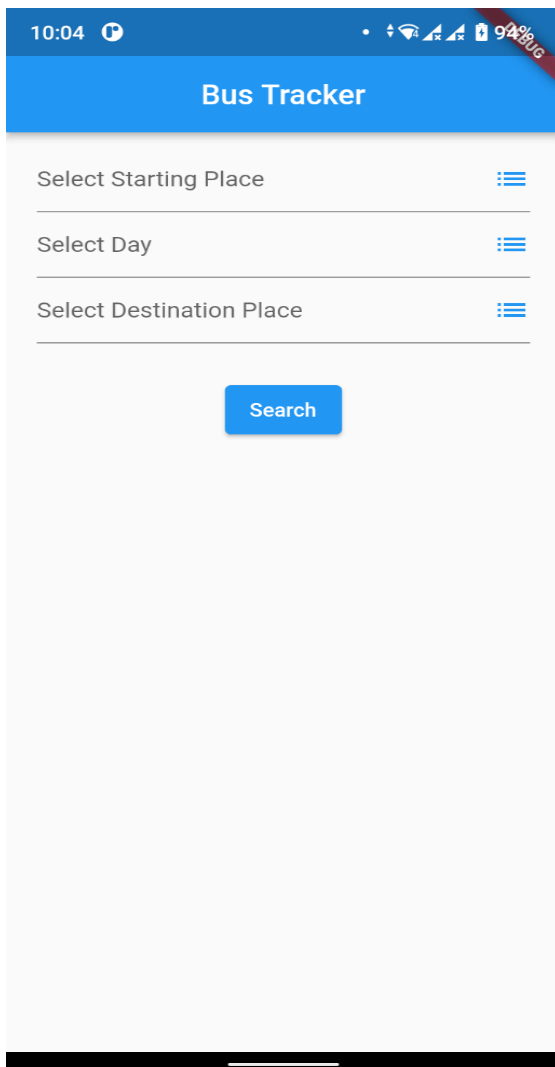


Figure 9: Bus Searching (User Manual)

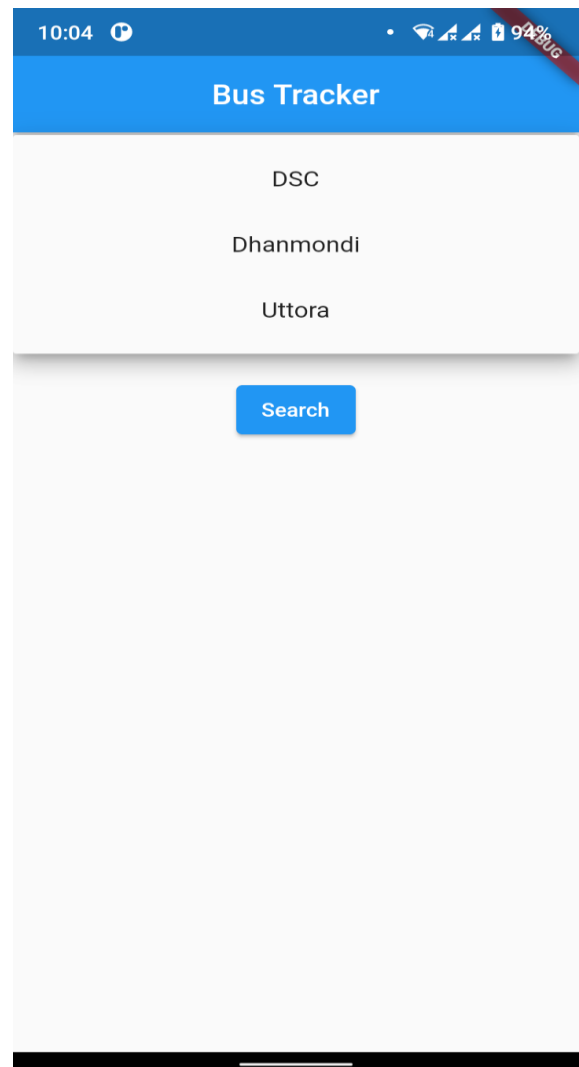


Figure 10: Place Selection (User Manual)

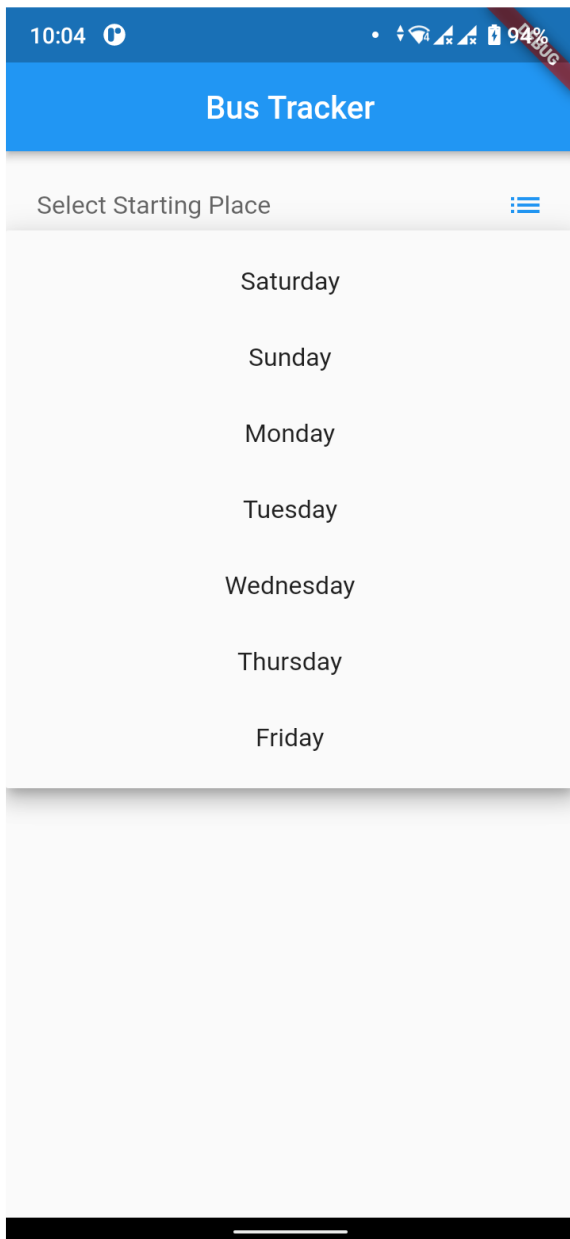


Figure 13: Day Selection (User Manual)

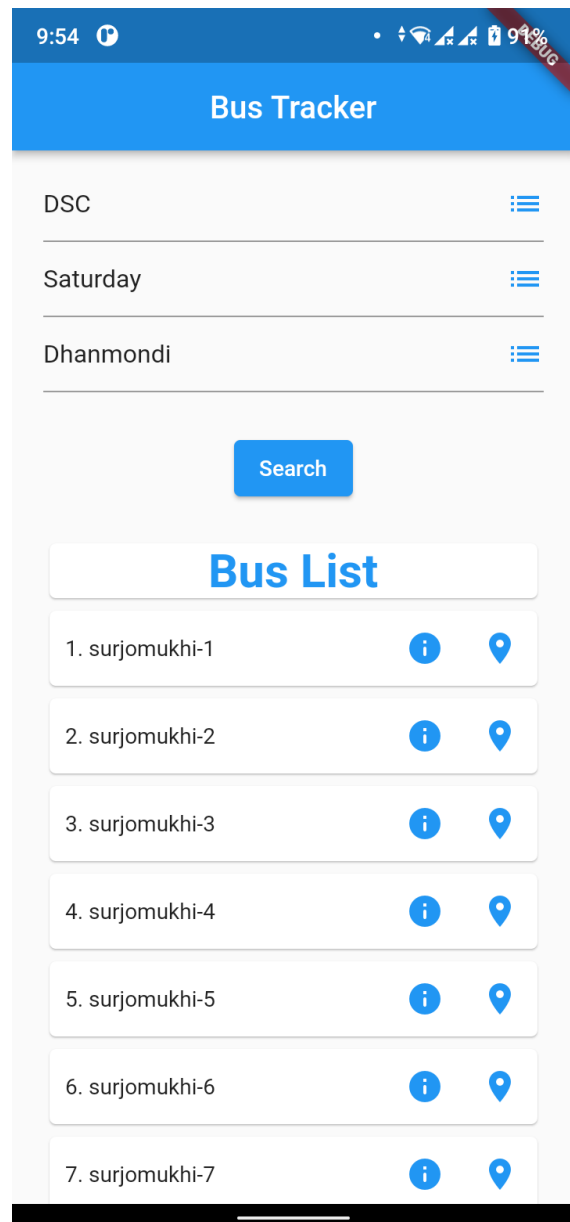
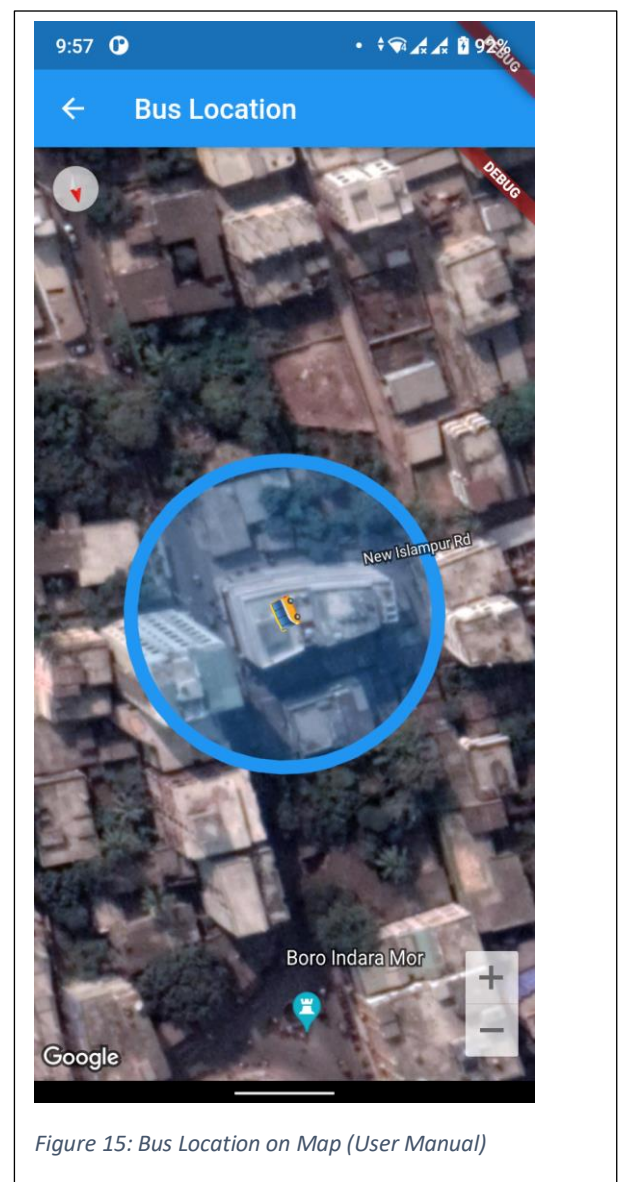
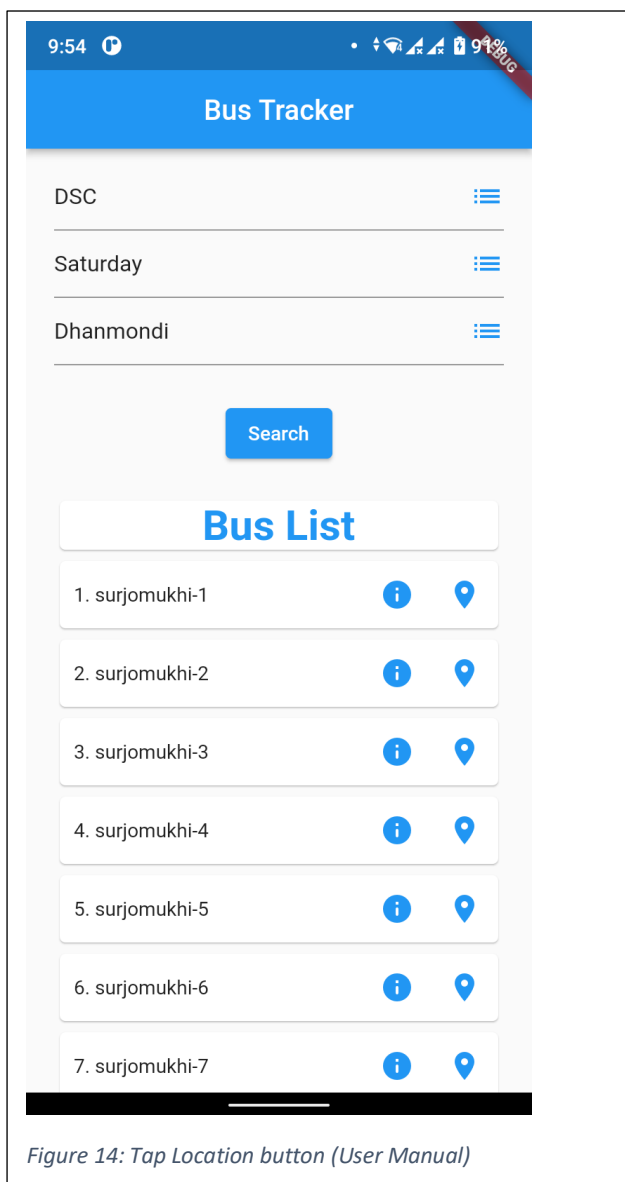


Figure 12: Bus List (User Manual)

6.2 How to see location of a bus:

After successful searching a list of bus, appear in the page. You can see active bus and inactive bus also in bus list (Figure- 14). Active bus indicating by green light and inactive bus indicating by red light. To see location you have to select one of the active bus. Beside every bus item, you can see a location icon (Figure-14). You have to press that icon to see the location. When you press the icon, you can see the location of that bus in new Page (Figure-15).



6.3 How to add to favorite list:

After getting a list of bus. You can see right side in every list item you can see information button (Figure-15). After pressing that button, you can see the details of the bus. In that page you see add button. If you tap on that bus will add to your favorite list.

6.4 How to login

Open application and you can see a login page. There are two input field one is for username and another is password. You have to fill up those fields and press login button to log into system. Then will be in home page.

6.5 how to logout

If you are a logged in user then you can see a logout button in the top right corner of page. If you press logout button then you will redirect to login page.

6.7 How operate Locator Device

First, open locator application and then tap on start button (figure-16) then app will request a permission (figure-17). If you give permission, then start button will be active and then you can press start button (Figure-16). After start service and then the stop button will be active (Figure-18).

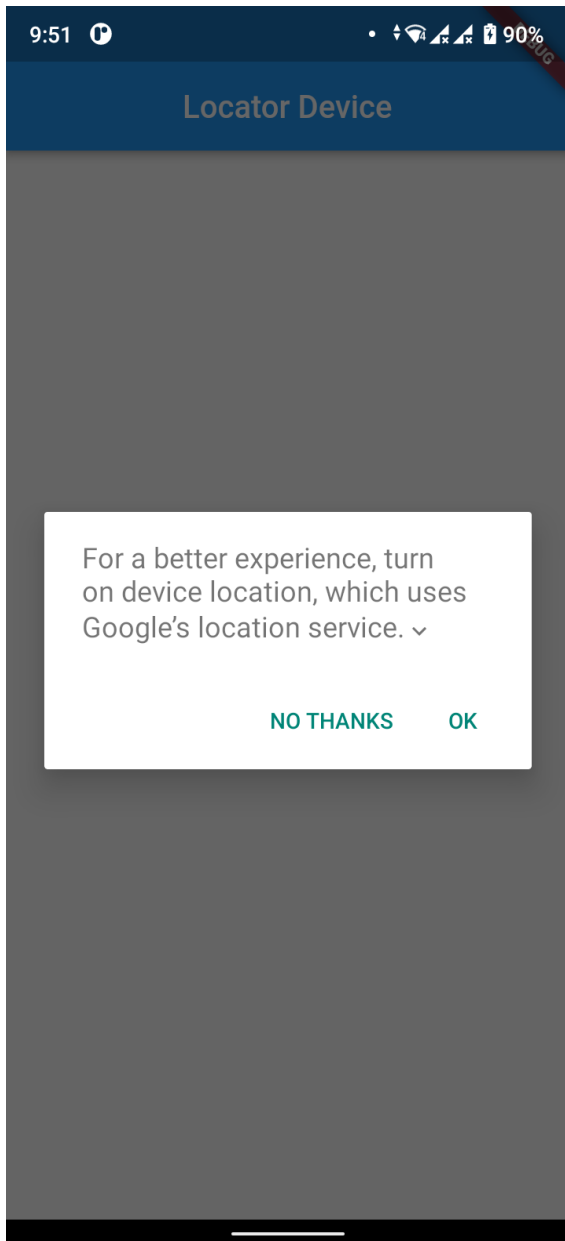


Figure 17: Location Permission Request (User Manual)

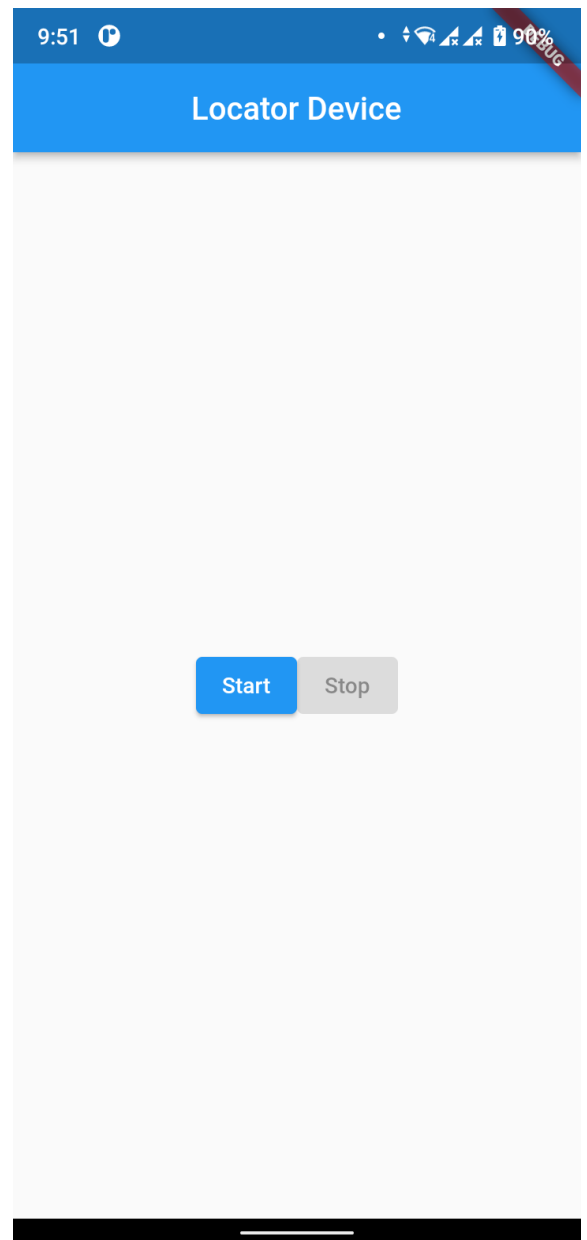


Figure 16: Start Service Button

The App is in service and you can stop by tapping on stop button (Figure-18).

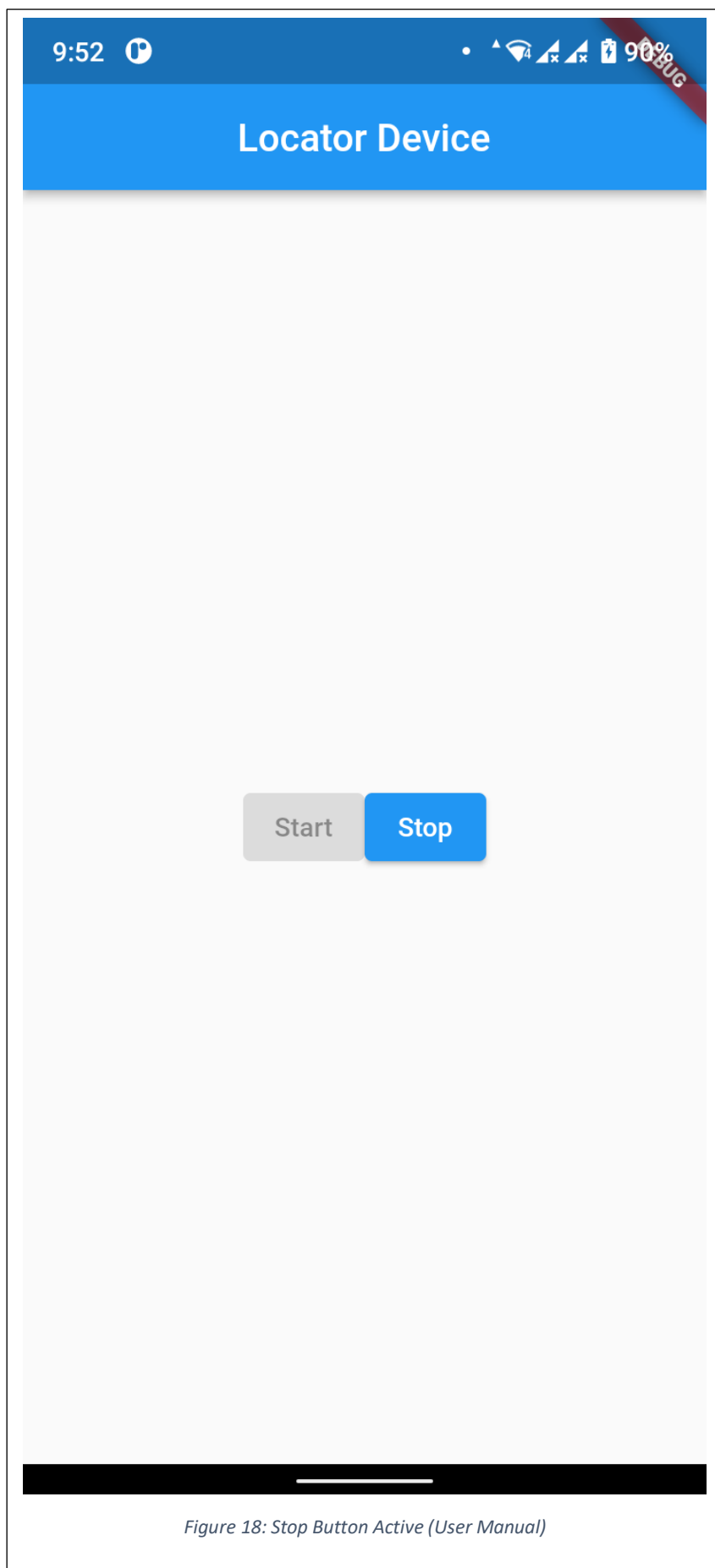


Figure 18: Stop Button Active (User Manual)

Chapter 7: Conclusion

7.1 GitHub Link:

Mobile Application:

https://github.com/RashedEmon/Bus_Tracker

Locator Application:

<https://github.com/RashedEmon/Bus Locator>

Server Application:

<https://github.com/RashedEmon/server application>

7.2 Project Summary

I have started this project in 5 September 2021. I have first select my project and discuss with my supervisor. After selection, I have brainstorming and starting to collect requirements and resources for this project. After that I have sketch my project UI. After that, I have made a list of features, functional requirements and non-functional requirements. Then I have made diagram such as, use case, ER diagram, Class Diagram. Then I have implemented the system.

7.3 Limitations

I have tried my best to make quality application and fulfill all requirements but some features were not possible to implement for some reasons.

- There is no location accuracy option to choose.
- Store favorite list in database.
- Add notifications system.

7.4 Future Scope

I have some ideas now but not included in this project for some limitations. I want to add login for traveler and want to store favorite list in database. I want to add accuracy selection option in future.