

A Smart Navigation System for Public Bus Service in Dhaka City

Md. Zahid Hossain Khan	011142151
Shakhawat Hossain	011142056
Md. Junaied Hossain	011142052
Saif Ahmed Anik	011141149

A project in the Department of Computer Science and Engineering presented
in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering



United International University
Dhaka, Bangladesh

October 2018

Declaration

We, Md. Zahid Hossain Khan, Shakhawat Hossain, Md. Junaied Hossain, Saif Ahmed Anik, declare that this project titled as “**A Smart Navigation System for Public Bus Service in Dhaka City**” and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a B.Sc. degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely our own work.
- We have acknowledged all main sources of help.
- Where the project is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Md. Zahid Hossain Khan
011142151
Department of Computer Science and Engineering

Shakhawat Hossain
011142056
Department of Computer Science and Engineering

Md. Junaied Hossain
011142052
Department of Computer Science and Engineering

Saif Ahmed Anik
011141149
Department of Computer Science and Engineering

Certificate

I do hereby declare that the research works embodied in this project entitled “**A Smart Navigation System for Public Bus Service in Dhaka City**” is the outcome of an original work carried out by Md. Zahid Hossain Khan, Shakhawat Hossain, Md. Junaied Hossain, Saif Ahmed Anik under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of B.Sc. in Computer Science and Engineering.

Mohammad Mamun Elahi
Assistant Professor
Department of Computer Science and Engineering
United International University

Abstract

The contribution of public transports system could not be neglected because it plays vital role among the people. So, the people should get a well organize and an informative interface to make his journey suitable and hinder less. This proposed system is responsible to provide an informative outlook delivered to the people without less memorizing and quick responsive service. This system will provide services depending on the user's location. If the user needs to achieve the service he has to enable his location access control. Then he will get the closest bus stop based on his life feed data. By providing his destination address he will be able to receive meaningful and valid path with alternative with the direction to reach at destination. Predictive time to reach at destination will be provided, so that any farther plan doesn't get a flaw with the user's routine.

Acknowledgement

First of all, we would like to thank the almighty for giving us the strength to accomplish this task. We would like to acknowledge our supervisor Mohammad Mamun Elahi for his valuable contribution and discussion. We would also like to show our gratitude to our parents.

Table of Content

LIST OF TABLES.....	vi
LIST OF FIGURES	vii
1. Introduction.....	1
2. Background and Related Works	3
2.1 Related Works	3
2.1.1 Uber	3
2.1.2 Pathao	3
2.1.3 Google Map	3
2.1.4 Sohoj.....	4
2.1.5Ovai.....	4
2.1.6 Comparison table between different mediums	4
2.2 Public Enquires.....	5
2.2.1 Survey on general people about local transportation	6
2.2.2 Number of buses operating in the existing bus routes of Dhaka City.....	7
3. Proposed Smart Navigation System	8
3.1 Features of Smart Navigation System	8
3.2 System Architecture.....	10
3.2.1Block Diagram.....	10
3.2.2 Use Case Diagram	11
3.2.3 Activity Diagram.....	11
4. Implementation	13
4.1 Database.....	13
4.2 APIs	27
4.3 Procedures.....	32
4.4 Interface	40
5. Limitations & Future Works.....	44
5.1 Limitations.....	44
5.2 Future Works	44
6. Conclusion	47
References.....	48

LIST OF TABLES

Table 1: Benchmark analysis of different related mediums	4
Table 2: Survey on general people about local transportation	8

LIST OF FIGURES

Figure 1: Number of buses operating in the existing bus routes of Dhaka City.....	7
Figure 2: Block Diagram of Smart Navigation System.....	10
Figure 3: Use Case Diagram of Smart Navigation System.....	11
Figure 4: Activity Diagram of Smart Navigation System	12
Figure 5: Query, Attributes and Instances in MySQL Database	14
Figure 6: Joining Table Example.....	16
Figure 7: ON&WHERE example.....	17
Figure 8: Process of user sign-up	18
Figure 9: Bus stop data fetched from database and represented in Google map	19
Figure 10: Inserting data through user interface and showing shared location data from database to user interface.....	22
Figure 11: Inserting desired location information in database and showing them user interface in real-time.....	24
Figure 12: Schema diagram of Smart navigation system database	25
Figure 13: Entity relationship diagram of Smart navigation system database	26
Figure 14: GPS Communication.....	32
Figure 15: Flowchart of find user location System.....	33
Figure 16: Distance calculation between user and bus stoppages	34
Figure 15: Flowchart of Nearby Bus Stop.....	35
Figure 16: Flowchart of find nearest bus	36

Chapter1

Introduction

Public transport has become an un-severable part of life. Most of the people depend on public transport to reach from home to workplace or to schools. There are various types of vehicles which refer to public transport, like In Bangladesh the public transports are known as Rickshaw, CNG Auto-Rickshaw, Taxi, and Bus. As it is economical and has propensity to handle enormous people at a certain time, which increases the popularity of using bus. As the need and the ability of passenger's solvency the class of bus services varies. That's why the service provider divided the service of buses into three categories: local bus, sitting bus and counter service.

Local bus service is the cheapest bus service to the people of our country. Government determinates how the fair should vary with respect to the kilometer a traveler advance. As the fair varies with the type of bus service it is necessary to provide a handful information to a traveler to calculate his daily expenses that will incurred during his/her daily life journey.

The worst nightmare is that, there is no practice of using specific bus stoppage where the bus will pick-up and drop down the passengers. It allows passenger to abroad on bus anywhere from road. On the other hand, a passenger needs to memorize the name of buses along with the path to reach his/her destination. An extra headache to a person who is actually new to the city has to ask people so that he/she could reach desired location.

The need of organized view of information is necessary to a passenger to smooth his/her daily life excursions. Like, to know the nearest bus stop to get a relevant bus with the clear mention of time and fair. Allow a person to announce his/her location and the way to reach to him/her to authenticated person.

If there is no single bus service to reach at destination then we have put pressure in our mind to find a suitable way. It means have to look for alternate path maybe switching several buses, that could introduce as a riddle to a person who is relying on bus as his/her daily transport.

On the other hand, an organization wants to provide transport service to its people so that they could attend their work on time. That's why there is a need to keep an eye on the transport they provide. It is also important to the users. Cause if a person who is taking the transport service might need to find the location of the bus that is dedicated to serve him. By this fact it will be easy to know whether it is possible to catch it or not.

Our proposed system will be able to provide answer to such daily life inquiries that occur into a passenger's mind. It will show the users location that is the core of the functionality along with nearest bus stop of a user. A dedicated path based on destination input along with time and distance. Share a person's location to its friend if he/she gives the permission. Information with stoppages a bus covers during an excursion. Provides a nearest bus along with the time it will take to reach at that dedicated stoppage.

To provide a clear view of the proposed system rest of the report is organized as follows:

Chapter 2 describes background and related works, Chapter 3 propose smart navigation system which will cover all the features that are implemented. Chapter 4 is about the implementation, the process that is taken to build up this system. The limitation and future work is added in Chapter 5 and at the end conclusion Chapter 6.

Chapter 2

Background Knowledge & Related Works

2.1 Related Works

There are several apps which are actually containing the information about vehicles, specifically vehicles that can provide service to an individual figure. Conversely, our proposed system is generated to support a vast number of persons; identically the vehicles that are related to local transports like bus service. But there are several functions in common.

2.1.1 Uber

The basic operation of an Uber relies on three major aspects like request, accept and information reviews. A user request for a ride using the interface designed for users by providing basic information like source & destination along with the pickup point.

The applicant user then can see the nearest vehicles that are ready to provide a ride. It also provides estimated time and fair that can be required to complete an excursion.

By hitting the confirmation button, a request is transferred to the entire service providers that are situated in a determined range.

The first person to accept the request is responsible to provide a suitable journey to a user. At the end of a journey this app provides the actual fair that is needed to pay the service provider agent.

2.1.2 Pathao

The functionality of Pathao app is similar as Uber. It also deals with the consumer and service provider in two separated apps. The exception function is Pathao delivery system. For that it allows goods to transport from one place to another. The time and distance are calculated using Google map functionalities.

2.1.3 Google Map

Actually, it provides several numbers of useful information. Based on user's interaction it provides the information in an organized form.

If a user enters the source to destination in the specific field then this medium will show the route to reach destination point along with the alternate paths.

Preferable route with can be detected by the user's perceptual choice, but it does not contain local transport information which are related to Bangladesh.

2.1.4 Shohoz

It is actually used to buy the intercity bus tickets. It contains the information about fair, bus information, duration.

2.1.5 Obhai

Similar functionality like Uber and Pathao but has an extension version that is including CNG and Rent a Car option.

2.1.6 Comparison Table between different mediums

Features	Uber	Pathao	Google Map	Shohoz	Obhai
Real-Time Track	✓	✓	✓	✓	✓
Source-Destination	✓	✓	✓	✓	✓
Time Estimation	✓	✓	✓	✓	✓
Estimated Fair	✓	✓	✓	✓	✓
Bus Information	✗	✗	✗	✗	✗
Nearest Bus Stoppage	✗	✗	✗	✗	✗
Nearest Bus	✗	✗	✗	✗	✗
Alternate Path (Bus)	✗	✗	✗	✗	✗

Table 1.1: Benchmark analysis of different related mediums

2.2 Public Enquires

As we were asking some people who were using local transport at their daily life to reach at their destination, we found some identical information that indicates the location name, bus stoppages, bus name, and types of the services. This information is not enough to smooth a journey which is incurred by traveler. So the information we just covered based on some specific route and some frequently asked questions. Like:

- i. What is a person's most daily used route?
- ii. What are the sequential bus stoppages he/she needs to take in count in order to reach his/her workplace?
- iii. What are the related buses that travel in a route permitted track?
- iv. Preference of bus based on types of service a user likes to have?

Now, in terms of these questions answer a table (2.2) is enough to reflect them in a well-organized form. The components of the table hold specific information related to: Bus Routes, Bus Stops, Bus Services, and Types of services.

Bus Routes: The actual meaning of the topic says that it is the source to destination name which is mentioned by a user who was on his way to his/her workplace.

Bus Stops: The stoppages that incurred during excursion sequentially while the bus approaches to the destination. Say, a person is heading to location A from location B. Then if there are stoppages (C, D, and E) on the way of the route then the actual path will be A - C - D - E - B.

Bus Services: Name of the bus company that are assigned to a specific route along with the permit to provide service to users.

Types of Services: It is the pattern of a service which is related to the fair. If the fair is accepted to be high it means the passenger using the sitting service, else his either using local or ticket service.

As the number of public transports are more than five thousand in Bangladesh, that's why the authority decided to divide them into several routes. The name of the bus changes according to the route permit. The figure 2.1 illustrates the volume of the buses on a specific route along with the approximate number of buses.

2.2.1 Survey on general people about local transportation

Bus Routes	Bus Stops	Selected Bus Services	Types of the Service
Uttara-Azimpur	Airport, Mohakhali, Asad Gate, Dhanmondi 32, Kalabagan, Sciencelab, Newmarket	• BEVCO Paribahan	• Air-conditioned
		• DuldulParibahan	• Counter
		• Bus Service No. 27	• Local
Mirpur-Motijheel	Airport, Mohakhali, Asad Gate, Dhanmondi 32, Kalabagan, Sciencelab, Newmarket	•New Vision	•Sitting
		•Bihongo	•Sitting
		•Bikalpa	•Sitting
		•Satabdi	•Local
Jatrabari-Mirpur	Gulisthan, Press club, Sahbagh, Farmgate, Karowanbazar, Mirpur 10, Mirpur 12	•Choice	•Sitting
Diabari-Postoghola	Golapbagh, Maniknagar, Kamlapur, Malibagh, Rampura, Banani, Airport, Uttara,	•Ra	•Sitting
		•Salsabil	•Sitting
		•Desh Bangla	•Sitting
Motijheel-Mohammadpur	Gulisthan, Press club, Sahbagh, Science lab, Shangkar, Mohammadpur	•Dipon	•Sitting
		•Moitri	•Local
		•Midway	•Sitting
		•Malancha	•Ticket
		•Rongdhanu	•Sitting
Jatrabari-Gazipur	Golapbagh, Maniknagar, Kamlapur, Malibagh, Rampura, Banani, Airport, Uttara, Tongi, Gazipur	•Turag	• Both (Local & Sitting)

Table 2.2: Survey on general people about local transportation

2.2.2 Number of buses operating in the existing bus routes of Dhaka City

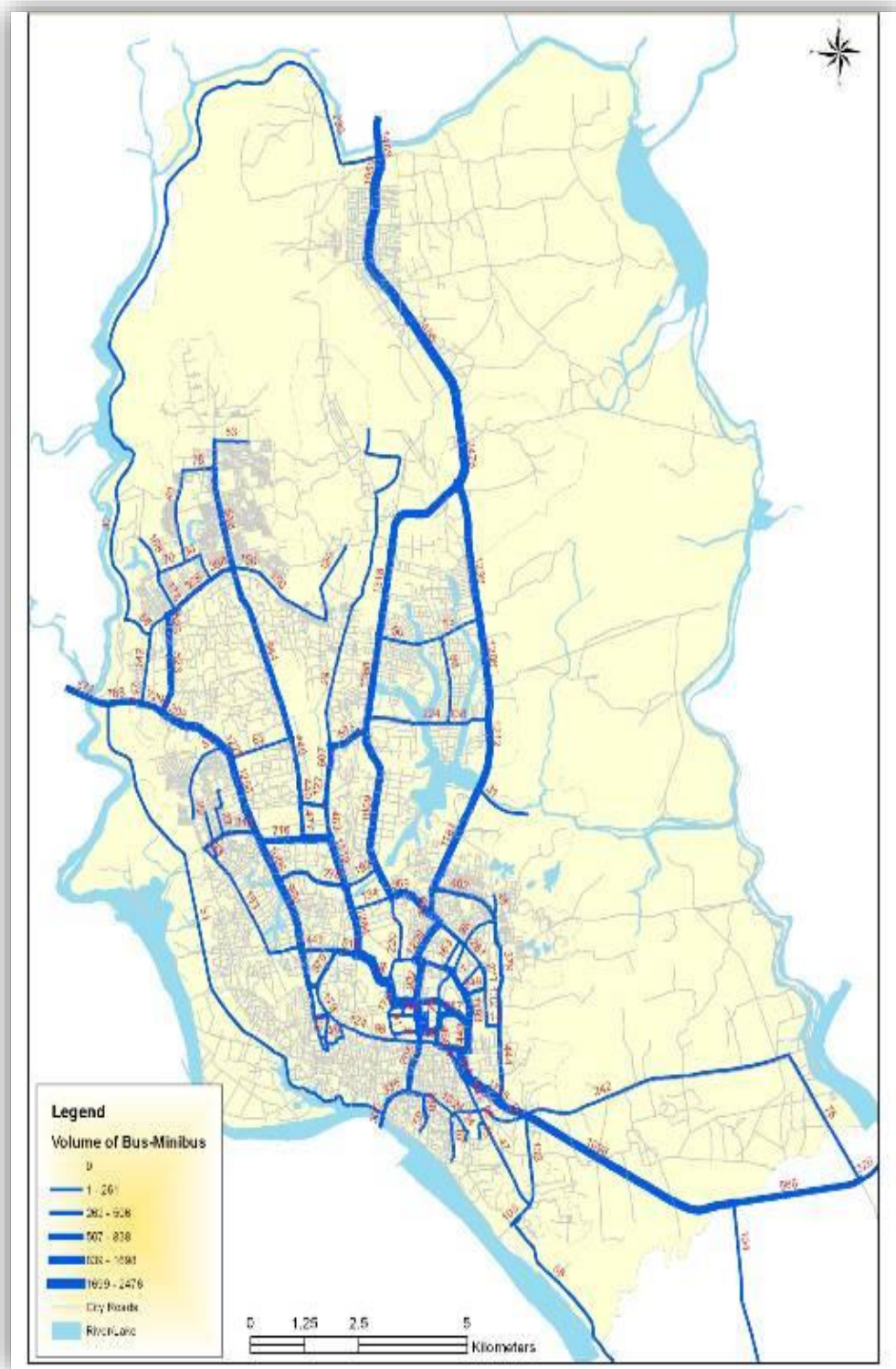


Figure 2.1: Number of buses operating in the existing bus routes of Dhaka City
Source: DTCB, 2009.

Chapter 3

Proposed Smart Navigation System

To smooth a user's excursion there is no alternative to provide organized information. The information must contain the daily necessary hints or indication of a desired route service view. Like the information of a user's nearest bus stoppage to the destination path direction. Also, the information of suitable bus that is nearest, estimated fare, time and alternate path if no direct bus service. This proposed system will provide a set of effective information which is enough to conclude a journey of a day without hassle. The proposed features of the system are:

1. Track user
2. Show bus stoppages
3. Nearby Bus stop
4. Locate nearest bus
5. Identify the possible ways
6. Share Location
7. Individual Organizational Support

3.1 Features of Smart Navigation System:

Track User:

All the operations strongly rely on this task. It will identify the user's exact location, so that a user could find all the related services that are described in the following term.

Show Bus Stoppages:

This feature will allow a user to see all the bus stoppages that are near and absorbable to its device layout. By the help of this feature a user can see all the bus stoppages that are approximately in the 5-kilometer radius.

Nearby Bus Stop:

Based on users' source to destination input this system will guide a user how to reach to its nearest bus stop. By attending there, a user will initiate its desire location base journey.

Locate Nearest Bus:

For a specific route, a user could get his/her desirable bus information. Actually, a user will get the information of a bus position that is his/her appropriate one to finish his/her desired journey. To begin this operation the system will locate the users nearest bus stop first and then find the nearest bus for the user.

Identify Possible Routes:

If there is no solitary bus service to serve a user to reach his/her destination then the system will provide a solution to use alternate paths that could be effective based on switching the bus services. The main fact is that, if there is no path from A to B using a single bus then the system will suggest user to take alternate way by choosing breakdown events, like A to C then C to B. There will be multiple ways, because the expectation of users varies from person to person, this service will provide ways depending on time, distance, number of bus switching.

Share Location:

Based on reliable communication and mutual authentication, user can share his/her location to his/her friends. So a path could be visible to one end of the user to navigate its partner in an appropriate way to reach destination. User can save his/her current location and its partner's location at that time.

Individual Organizational Support:

There are lots of companies or organizations who are trying to provide a feasible transport service to its employees. That's why the system proposed a separate option to move to another version like this system, which will provide the information which includes where the bus stoppages are, the buses locations, when the bus will arrive and departure and so on.

3.2 System Architecture

To provide a overview of the proposed system diagrams are included to make a flow of the transitional works. Such diagrams are:

- i. Block Diagram
- ii. Use Case Diagram
- iii. Activity Diagram

3.2.1 Block Diagram of Smart Navigation System:

User will input the source to destination information. Then the system will communicate with the server fetch the information. Server will receive the required data from the cloud storage to compute the result. Server will transfer the result to the interface and the user will get the information. On the other side the buses will update the required field information.

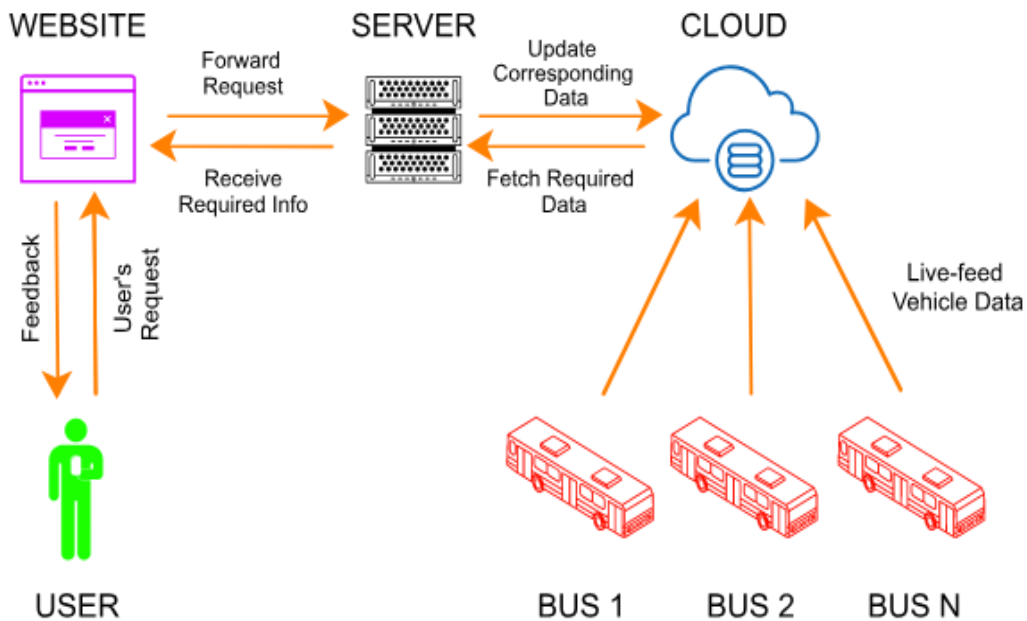


Figure 3.1: Block Diagram of proposed system.

3.2.2 Use Case Diagram of Smart Navigation System:

The primary interest holder is the passenger. He is the main persons who directly connected to the front part of this system. Can view and search in this system to obtain the proposed services. Admin is the main backbone in that system who can access any of this features and update required relevant data. On the other hand, service provider (bus service) can access specific feature and provide their instances to the admin for further update. The government can access only bus tracking service and traffic for prevent any unwanted circumstance.

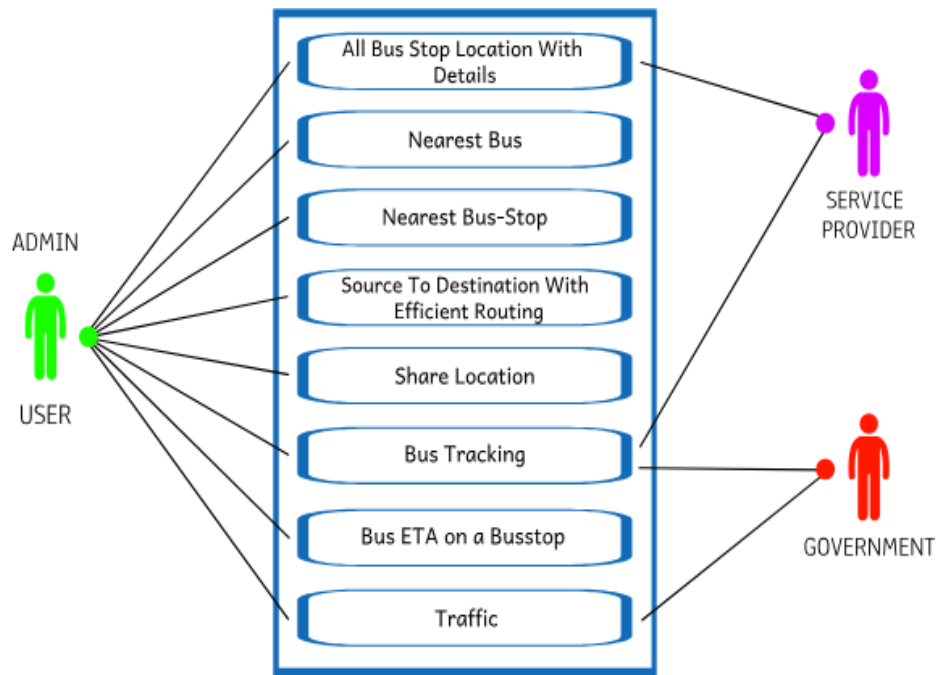


Figure 3.2: Use case diagram of proposed system

3.2.3 Activity Diagram of Smart Navigation System:

User first needsto authenticate him/her-self inorder to get all the services of the proposed system. After successful authentication, a user has to allow the system to locate him/her-self. So that the user can have the option to access the major features of the system. If user doesn't allow it then the system will throw a error message to the user and have to retry again. So, after allowing the system to locate position, a user can access two separate platforms. One is related to the user's profile and another is related to functional activities bind with the local transportation service. In the profile section, a user can build a community to share information about their own position and history. In the functional section, a user can have access to the major services such as all bus stoppages, nearby bus stop, nearest bus and so on.

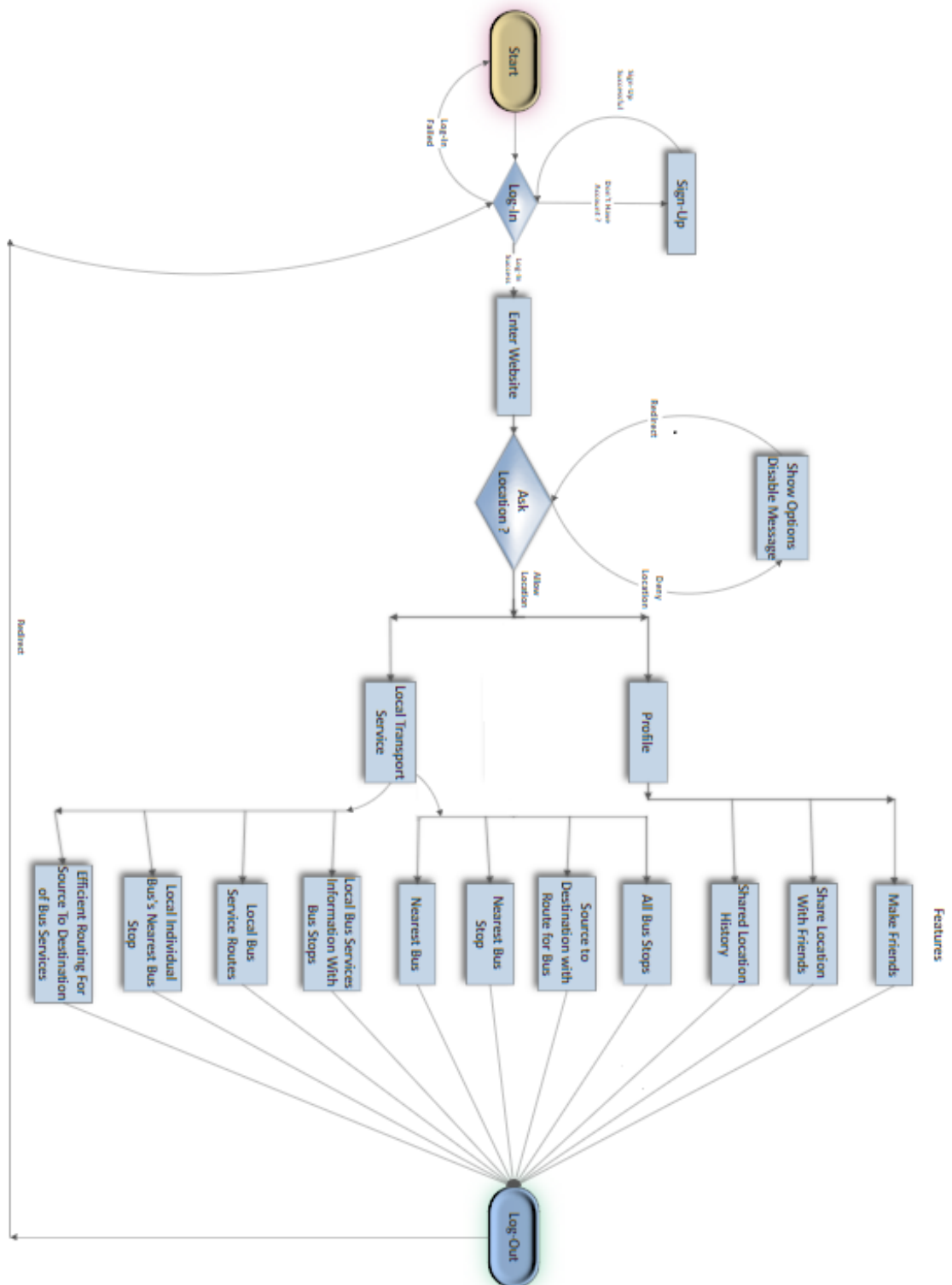


Figure 3.3: Activity diagram of proposed system

Chapter 4

Implementation

The implementation is divided into 4 sections. First of all, we need to collect data to perform operation to provide required information. The operation includes processes to find the gesture of a specific action.

4.1 Database:

Database management system is the software which is connected with end users and can be accessed through all programming languages. Popular database management system of nowadays are, Oracle, MySQL, Microsoft SQL Server, MongoDB, Maria DB etc [1].

MySQL:

MySQL is a popular database management system, which was appeared on 23 May, 1995. It was developed in C and C++ programming language [2]. As non-relational database management MySQL is highly used in development. Companies like Google, Facebook, YouTube, and Twitter are using MySQL. MySQL is slower than NoSQL, that's why NoSQL are highly used in search engines[3].

When user perform query for data, MySQL returns the data as a table. Though the programming languages we extract the data from the table. Many kinds of query can be done in MySQL. For example,

- CREATE
- DROP
- SELECT
- DELETE
- UPDATE
- RENAME
- ALTER

Keywords for grouping data,

- WHERE
- ON
- GROUP BY
- ORDER BY
- HAVING

Multiple types of joining methods,

- JOIN
- NATURAL JOIN
- FULL OUTER JOIN
- LEFT JOIN
- RIGHT JOIN
- INNER JOIN
- UNION

Keywords for doing query in programmatic way,

- AND
- OR etc.

Query, Attributes & Instances

As we know whenever we write query and execute it, MySQL returns us a table, which includes some attributes and instances. From figure 4.1, an example can be shown.

Query

`SELECT * FROM `person``

Attributes

+ Options

	id	name	password	department	type
<input type="checkbox"/> Edit Copy Delete	011141149	SAIF AHMED ANIK	1234	CSE	student
<input type="checkbox"/> Edit Copy Delete	011142141	ZAHID HOSSAIN	1234	CSE	student
<input type="checkbox"/> Edit Copy Delete	011142151	Saimoon Imran Shoccho	1234	CSE	student
<input type="checkbox"/> Edit Copy Delete	011142152	Sagor Paul	1234	CSE	student
<input type="checkbox"/> Edit Copy Delete	MHR	Mahabubur Hossain Rana	1234	CSE	faculty
<input type="checkbox"/> Edit Copy Delete	MOMN	Monirujjaman	1234	CSE	faculty
<input type="checkbox"/> Edit Copy Delete	NM	Nagib Mishkat	1234	CSE	faculty
<input type="checkbox"/> Edit Copy Delete	SS	Swakkhar Shatabda	1234	CSE	faculty
<input type="checkbox"/> Edit Copy Delete	TAR	Tamjid Al Rahat	1234	CSE	faculty

Instances

Figure 4.1: Query, Attributes and Instances in MySQL Database

In figure4.1, we can see that, there are many user's information saved in **person** table. There are two ways to insert data in database.

1. Insert by MySQL phpMyAdmin user interface
2. Insert by running query

In the **query** section, we have written "SELECT * FROM person". 'SELECT', 'FROM', '*' these are the built-in keywords in MySQL database. We shall describe few of the built-in keywords.

SELECT: 'SELECT' keyword is used to select the columns of database table. We mention the desired attribute/column names after 'SELECT' keyword.

FROM: 'FROM' keyword indicates from which table we want to get data. After 'FROM' key, we write table's name.

***:** Basically '*' is used as a shortcut. If we want to select all attributes/columns from a table, then we write '*' after the 'SELECT' keyword. We use '*', so that we don't need to write all attributes/columns in our query.
In figure4.1 we have used '*' in our query, that's why it returned a table with all attributes of **person** table.

Primary key

Every database tables must have a primary key. This primary key defines the uniqueness of instances in database table. One or multiple attributes can be defined as primary key.

Foreign key

Foreign key is basically used in relational tables. Foreign key refer data from other table.

JOIN

Depending on the query MySQL makes various kinds of tables. If we use 'JOIN' in our query, it'll merge two tables. We use 'NATURAL JOIN' when we have fully similar attributes in more than one different table. 'NATURAL JOIN' detects the similar attributes automatically. But those similar attributes must have 'PRIMARY KEY' in their own tables.

Suppose we have two tables named 'user' and 'saved location'. In 'user' table, information of user is saved and in the 'saved location' table we have information of saved location of corresponding user of 'user' table. Both tables have primary key. User has 'id' as its primary key. 'Saved location' has 'id' and 'userid' as its primary key. 'Userid' in saved location is also foreign key of 'Id' of user. Now we want to extract the information of saved location of each user.

To do **JOIN** query, we need to write table names before and after **JOIN** keyword

Two tables join

Join twotables 'a' and 'b'.

Query: SELECT * FROM a JOIN b

Three tables join

Join threetables 'a', 'b' and 'c'.

Query: SELECT * FROM a JOIN b JOIN c

JOIN example with database:

User table

id *	int(5)
name	varchar(200)
email	varchar(200)
password	varchar(15)

Saved location table

id *	int(5)
userid *	int(5)
lat	float(20,10)
lng	float(20,10)
date	date
time	time

Now we run query "SELECT * FROM user JOIN saved_location ON user.id=saved_location.userid".

<pre>SELECT * FROM user JOIN saved_location ON user.id=saved_location.userid</pre>									
id	name	email	password	id	userid	lat	lng	date	time
2	ZUNAYET HOSSAIN	zunayet@gmail.com	1234	12	2	23.7924098969	90.4351577759	2018-10-01	09:28:56
2	ZUNAYET HOSSAIN	zunayet@gmail.com	1234	13	2	23.7924098969	90.4351577759	2018-10-01	09:31:26
1	SAIF AHMED ANIK	saif@gmail.com	1234	17	1	23.7924098969	90.4351577759	2018-10-01	09:28:56
1	SAIF AHMED ANIK	saif@gmail.com	1234	18	1	23.7924098969	90.4351577759	2018-10-01	09:28:56
4	SHAKHAWAT HOSSAIN	shakhawat@gmail.com	1234	19	4	23.7924098969	90.4351577759	2018-10-01	09:28:56
4	SHAKHAWAT HOSSAIN	shakhawat@gmail.com	1234	20	4	23.7924098969	90.4351577759	2018-10-01	09:31:26
6	SHIFAT NIPA	nipa@gmail.com	1234	21	6	23.7924098969	90.4351577759	2018-10-01	09:31:26
6	SHIFAT NIPA	nipa@gmail.com	1234	22	6	23.7924098969	90.4351577759	2018-10-01	09:31:26

Figure4.2: Joining table example

In our query, we have written ‘user.id’ and ‘saved_location.userid’ to access the ‘id’ column of ‘user’ table and ‘userid’ column of ‘saved_location’ table. There may be many attributes/columns with same name in the database. That’s why we use this method, so that attributes/columns don’t get conflicted in our query.

ON & WHERE

‘ON’ and ‘WHERE’ keyword is used for making conditional logic in MySQL database. We have written ‘ON user.id=saved_location.userid’ in our query. Which mean we want our system to search for instances where user.id and saved_location.userid are same. If we would write ‘WHERE user.id=saved_location.userid’, it will work same as ‘ON user.id=saved_location.userid’.

If we want to have only user’s name and their corresponding saved locations only where user id is ‘1’, then we execute the following query,

“SELECT user.name,saved_location.lat,saved_location.lng FROM user JOIN saved_location ON user.id=saved_location.userid WHERE user.id='1'”



name	lat	lng
SAIF AHMED ANIK	23.7924098969	90.4351577759
SAIF AHMED ANIK	23.7924098969	90.4351577759

Figure4.3: ON&WHERE example

Database of Smart Navigation System

In database of our project ‘smart navigation system’, there are 9 tables. Among those there are 6 relational tables. All tables of database and description is given below.

User

id *	int(5)
name	varchar(200)
email	varchar(200)
password	varchar(15)

Information of users will be saved in **User** table. Each user will have a unique id, which increments automatically. When an user signs up in our system, then he/she will have to fill up the sign-up form in ‘Sign-up’ webpage. While user submits his/her name, email and password by clicking submit button, then in the backend, this information will be saved in PHP language variables. Then our system will execute a query, which will insert the values of those PHP variables in our database.

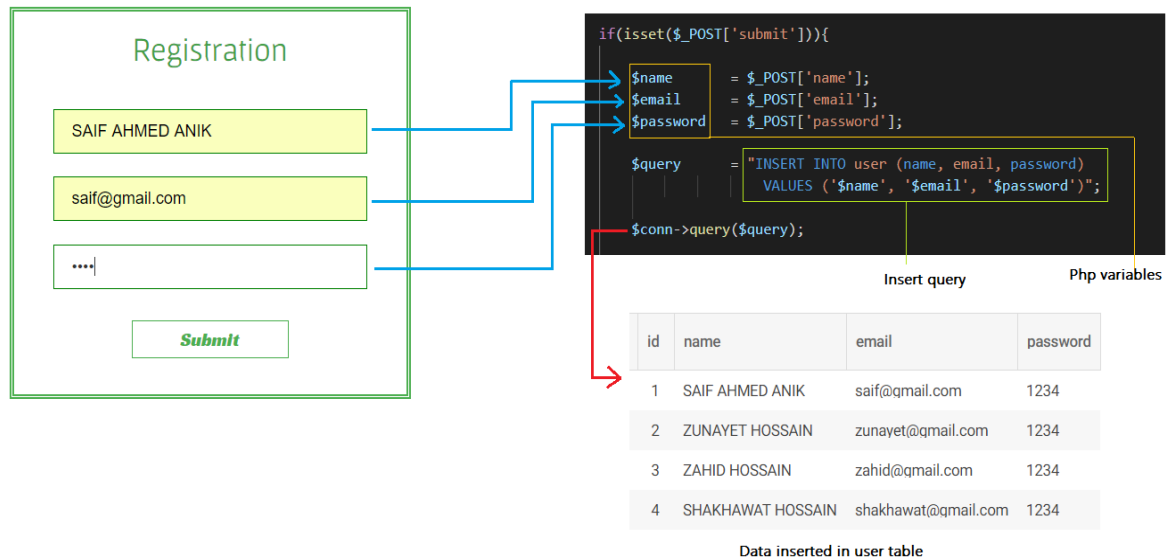


Figure4.4: Process of user sign-up

Bus

id *	int(5)
name	varchar(100)
lat	varchar(100)
lng	float(20,10)
status	tinyint(1)

Information of each bus with their corresponding latitude, longitude and driving status will be saved in bus table. Status of the bus determines that, if the bus is available or not.

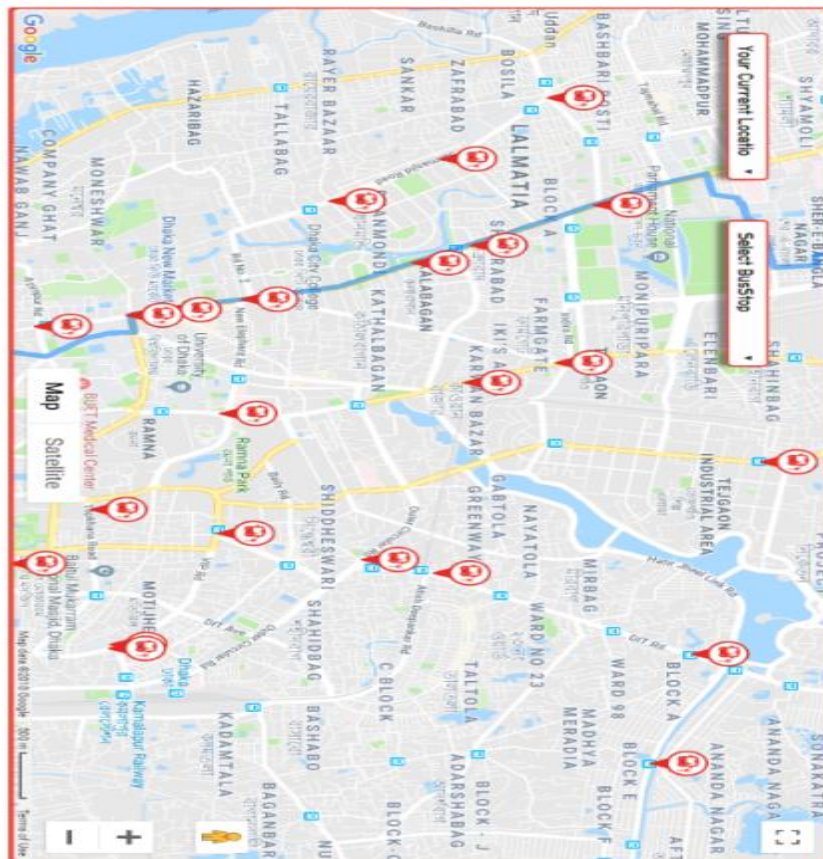
Bus stop

bus_stop_id *	int(5)
name	varchar(100)
l_lat	float(20,10)
l_lng	float(20,10)
r_lat	float(20,10)
r_lng	float(20,10)

Each bus stop's name with a unique id will be saved in "bus stop" table.

bus_stop_id	name	Lat	Long	L_Lat	L_Long
4	Kawan Bazar Bus Stop, Dhaka	23.75000807	90.392241638	23.751999014	90.392417135
8	Nabisco Bus Stop, Dhaka	23.796842468	90.400322120	23.796804321	90.401072705
10	Mohammadpur Bus Stop, Dhaka	23.757026724	90.3615188599	23.757007989	90.3616627832
11	Sarker Bus Stop, Dhaka	23.750652332	90.368164065	23.7507915497	90.368240354
12	Diamond 15 no bus stop, Dhaka	23.744052032	90.372703552	23.744286761	90.372711816
14	Mahbub Rola Gate Bus Stop, Dhaka	23.7502136230	90.4129162378	23.7501392865	90.4130172729
15	D Block Bus Stop, Bagerma Main Road	23.7631511688	90.4335708618	23.7631282806	90.4337234497

Bus stops in Database



Bus stops shown in user interface through google map

Figure4.5: Bus stop data fetched from database and represented in Google map

From the “bus stop” table, we get all information of bus stop and represent them in Google map through JavaScript and PHP.

Relationship tables

History

id *	int(5)
userid	int(5)
busid	int(5)
source	int(5)
destination	int(5)

While user will search for bus for a specific source and destination, all information of user id, bus id, source and destination will be saved in history table. The primary key **id** will be generated automatically.

Friends

from_id *	int(5)
to_id *	int(5)
status	int(1)

If user1 sends friend request to user2, then user1 id and user2 id will be saved in “from_id” column and “to_id” column respectfully. If user2 accept the friend request then status will be 1, or else status will be 0.

In our system, user has the opportunity to share his/her location only to his/her friends.

Share location

id *	int(5)
user_from	int(5)
user_to	int(5)
lat	float(20,10)
lng	float(20,10)
time	timestamp

When user1 shares location to user2, then user1 id and user2 id will be saved in user_from and user_to respectfully. The latitude and longitude will be saved in lat and lng respectfully. Instant time of sharing location will be saved in “time” column. A user can send location to another user multiple times, that’s why **id** is the primary key.

We determine the friends of the user and show those people in the ‘Share location’ page so that, user can his/her location. User can see whom he/she has shared location, or who has shared him/her which location. We obtained all these information querying from database.

From Figure 4.6, we can see that while user shares data by clicking ‘Share Now’ button in ‘Table 1’, we get user’s latitude, longitude and current time, and pass these values in INSERT query (QUERY 1).

By running query 2, we get the table of user’s friend, who were shared user’s location by user, and we showed that table in ‘Table 2’.

By running query 3, we get the table of user’s friend, who shared their location to user, and we showed that table in ‘Table 3’.

While user logs in our system, then we take user’s id in '\$userid' variable.

Query 1: "INSERT INTO share_location (user_from, user_to, lat, lng, time)
VALUES ('\$userid', '\$friend', '\$lat', '\$lng', CURRENT_TIMESTAMP)".

After having user id, friend’s id, latitude and longitude in '\$userid', '\$friend', '\$lat', '\$lng' variables respectively, we pass the variables in the query and run the query 1, so that it can create a new instance in ‘share_location’ table.

Query 2: "SELECT user.name,share_location.id FROM user JOIN share_location
ON user.id=share_location.user_to WHERE
share_location.user_from='\$userid' ORDER BY time".

If we run this query, then it will join ‘user’ and ‘share_location’, then create a table with those data where user’s id and id of that user who shared location are same and user’s id equal to logged user’s id, then selects user’s name and shared location id from the created table. Shared location id will be used for showing the user’s location in Google map.

Query 3: "SELECT user.name,share_location.id FROM user JOIN share_location
ON user.id=share_location.user_from WHERE
share_location.user_to='\$userid'".

If we run this query, then it will join ‘user’ and ‘share_location’, then create a table with those data where user’s id and id of that user who were shared location are same and user’s id equal to logged user’s id. Shared location id will be used for showing the user’s location in Google map.

Bus_busstop_relation

id *	int(5)
bus_id *	int(5)
bus_stop_id *	int(5)
serial	int(5)
crossed	varchar(10)

From this table we can know that, which bus will be covering which bus stops. Here **id**, **bus_id** and **bus_stop_id** are the primary keys. If a bus crosses a bus stops then value of cross will 1, or else 0. Suppose a bus crossed 'A' bus stop, then 'B' bus stop and then 'C' bus stop, then serial of A, B, C bus stops will be 1, 2, 3, respectfully. We are using this serial in order to obtain the distance between any two bus stops.

Busstop_distance

id *	int(10)
src_id *	int(5)
dest_id *	int(5)
distance	varchar(10)
duration	varchar(10)
heuristic	varchar(10)

Distance, travelling duration and fight path distance between each bus stop will be saved in this table. Here **id**, **src_id** and **dest_id** are the primary keys.

Saved_location

id	int(5)
userid	int(5)
lat	float(20,10)
lng	float(20,10)
date	date
time	time

If user saves any certain location, then user id, current latitude and longitude, current date and time will be saved in this table. Here **id** is the primary key and it is auto incremented.

While user clicks the 'Save current location' button, it will collect user's current position and save it in database and by using Ajax it shows user saved information in real-time. User's all saved location's information will be in 'Saved location' table.



Figure 4.7: Inserting desired location information in database and showing them user interface in real-time

Database Schema

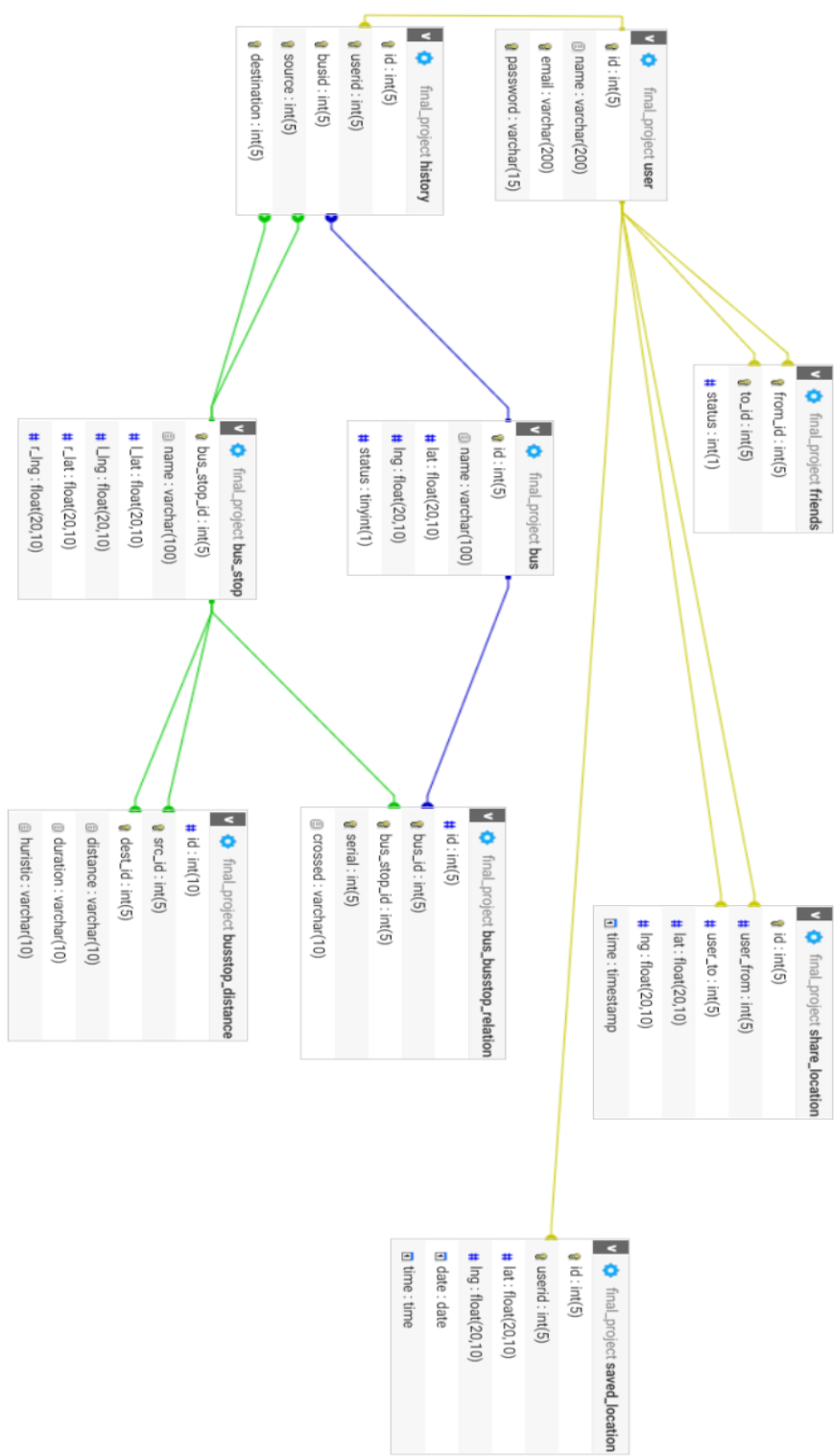


Figure4.8: Schema diagram of Smart navigation system database

Entity Relationship Diagram

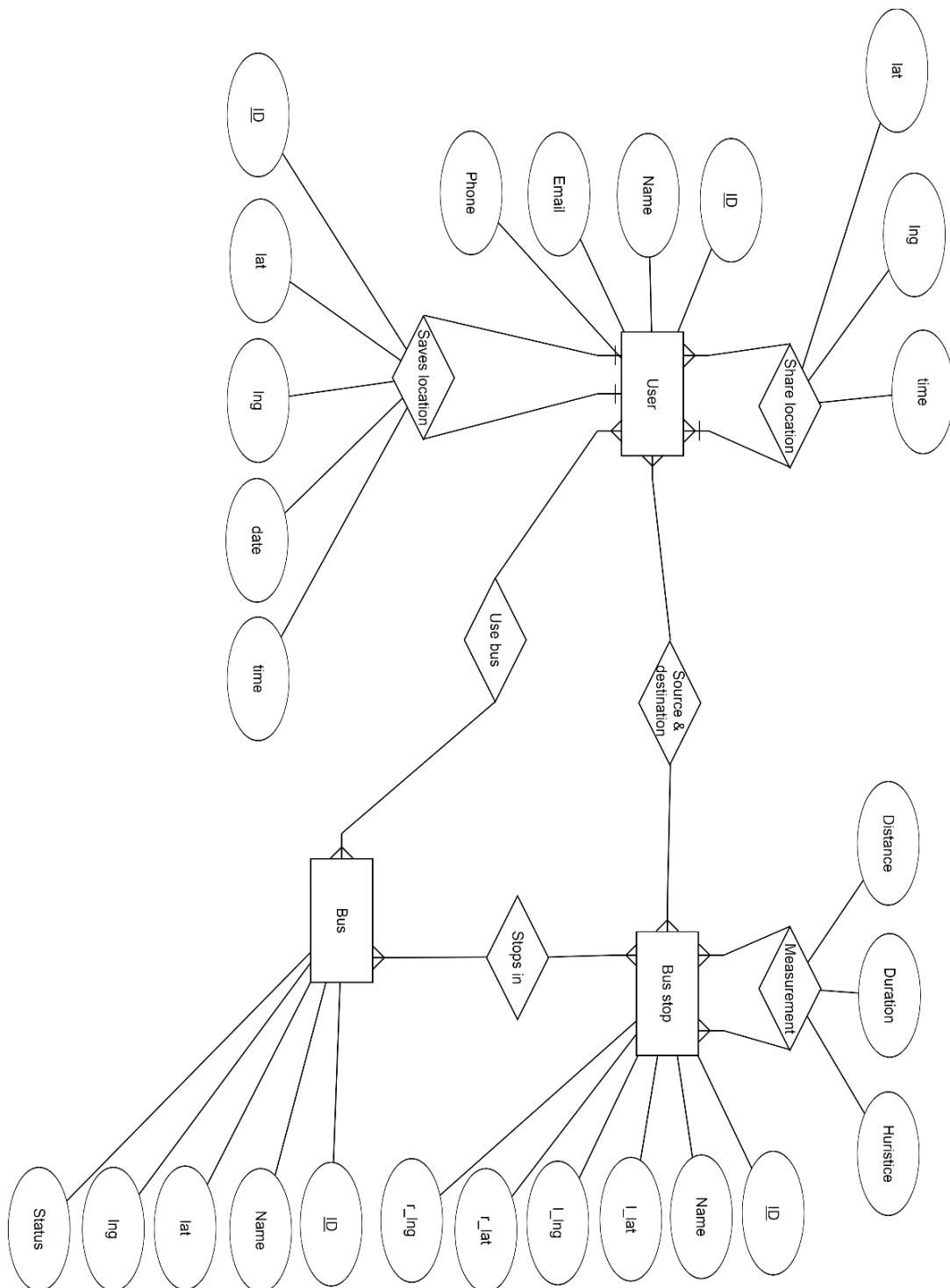


Figure4.9: Entity relationship diagram of Smart navigation system database

4.2 APIs:

What is API?

API(Application Programming Interface) is a communication protocols, routines and tool for building software for any platform. In short it's a connectivity interface to an application. Modern API maintain some standard like HTTP and REST.

It helps the developer for maintain the data.

Suppose we want build both apps and software for multi-platform but the main difficulty is platform is different and the approach is different, database should be different. But API helps us to add all platform in a certain things. API connect all the different platform to a single platform. API sends the data to the different platform through http rest api and platform based language get that data. It also allow the developer to perform create, update, delete, remove, put function. [14]

What is Google API?

Google map API allow us to show there map in our interface and they also provide lots of functionality to customize their map. [4]

Custom Google map:

Disable the default UI:

In "initmap" function there is map variable, add "disableDefaultUI: true" to disable the default UI [5].

Custom control:

It's possible to control the Google map functionality manually like "Zoom control", "Map Type control", "Fullscreen control" etc then first enable the "mapTypeControl: true", then positioning the control in "mapTypeControlOptions" like the following way:

```
mapTypeControlOptions: {  
    style: google.maps.MapTypeControlStyle.HORIZONTAL_BAR,  
    position: google.maps.ControlPosition.TOP_CENTER  
}
```

You can also add custom CSS in the Googlemap. As an example,
`controlText.style.fontFamily = 'Roboto, Arial, sans-serif';`

Note: All controls are not available in DefaultUI. So you need to bind customized control on Google map.

Example [6]:

```

{
    zoomControl: boolean, (true/false)
    mapTypeControl: boolean, (true/false)
    scaleControl: boolean, (true/false)
    streetViewControl: boolean, (true/false)
    rotateControl: boolean, (true/false)
    fullscreenControl: boolean (true/false)
}

```

Style the map:

To highlight the density of Roads, Landmarks, and Labels, one have to visit the link [13]. There are so many options you can add/adjust/create based on user's preference. After clicking the finish button, it gives you the JSON content, which needs to be copy and mark it on the map.

Info Window:

To show some info for particular location, "infowindow" will be needed. In "infowindow" "content" option information is showed, "position" option is used for showing that information "position" option which is a longitude and latitude value or particular position name as a string value [7].

Marker:

If you want to mark location on the map then the Google marker will be need. By default it has a red icon but it's possible to change that icon. There is position, map, icon, and title option. To set custom icon then put the path into to the icon option [8].

```

new google.maps.Marker({
    position: pointB,
    map: map,
    icon: "icon/bus.png",
    title: "Nearest Bus",
});

```

Note: If you want to remove marker from the map then create a marker array and put all marker into the array first. Whenever you need to remove the marker, then empty the marker array then all markers will be removed [9].

Camera Position for multiple locations:

To set the camera position to view multiple locations at a time, then bounds option is needed.

Example:

```
var bounds = new google.maps.LatLngBounds(); //initialize bounds
for(var i=0; i<location array length; i++){
    var location = new google.maps.LatLng(a[i].lat, a[i].lng);
    addMarker(location); // marker that position
    bounds.extend(location); //set the position
}

map.fitBounds(bounds); // added in the map
```

Add Layer:

In Google map there is so many layer like traffic, transit, and bicycle. You can add this in your map

Example [10]:

In initmap function,

```
var trafficLayer = new google.maps.TrafficLayer(); //TransitLayer();//BicyclingLayer();
trafficLayer.setMap(map);
```

DirectionsService:

Direction service gives you the route for source to destination. It also includes waypoints. You can customize them and define source and destination to view the desire route.

```
{
    origin: LatLng | String | google.maps.Place,
    destination: LatLng | String | google.maps.Place,
    travelMode: TravelMode,
    transitOptions: TransitOptions,
    drivingOptions: DrivingOptions,
    unitSystem: UnitSystem,
    waypoints[]: DirectionsWaypoint,
    optimizeWaypoints: Boolean,
    provideRouteAlternatives: Boolean,
    avoidFerries: Boolean,
    avoidHighways: Boolean,
    avoidTolls: Boolean,
    region: String
}
```

These are the option. If “google.maps.DirectionsStatus.OK”, and then the shortest path is showed [11].

```
directionsService.route({
    origin: pointA,//source-latlng or place name
    destination: pointB,//destination-source-latlng or place name
    provideRouteAlternatives: true,
    travelMode: google.maps.TravelMode.DRIVING,
    unitSystem: google.maps.UnitSystem.METRIC
}, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
        for (vari = 0, len = response.routes.length; i<len; i++) {
            newgoogle.maps.DirectionsRenderer({
                map: map,//initial map
                suppressMarkers: true,//
                directions: response, // google map response
                routeIndex: i,//number of path/route
                polylineOptions:{strokeColor:colors[i]} //route color});
        }
    }
});
```

```
//for customize route color
polylineOptions:
{
    strokeColor: "green", color of the route
    strokeOpacity: 2.0, // visibility
    strokeWeight: 10 // color weight
}
```

Note: If you want to add custom marker or icon for source and destination then active the “supermarkers” first.

Example:

```
var directionsDisplay = new google.maps.DirectionsRenderer({suppressMarkers: true});
```

Distance between two points:

Find distance between two points using Google predefined method `google.maps.geometry.spherical.computeDistanceBetween(pointA, pointB)`, this returns the distance in meters.

SnapTo Road:

If you want to track something that run on the road then Snap to Road will help you. Moving locations are not always 100% correct or your location always not 100% correct it depends on the position network strength. So snap to road collect your location and show on the nearest road [12].

4.3 Procedures:

Track user:

User must enable the GPS so that the system could get the current location of that device. GPS is mainly founded and controlled by the U.S Department of Defense (DOD). Despite of being designed for the military civil can use GPS without any charge and restriction []. GPS tracking system can be deployed in car, mobile or any electronically device.

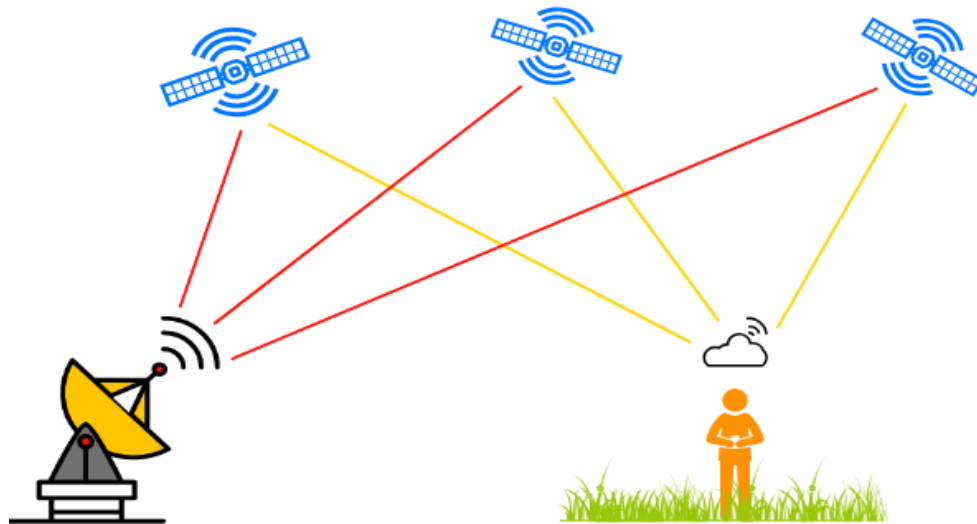


Figure 4.10: GPS Communication

Pseudo code:

Output: Users Longitude and Latitude

Method:

1. **If** user_location access is granted using geolocation, **then**
2. Navigator.geolocation performs
3. get_current_position()
4. return latlang
5. **end if**
6. **else**
7. error
8. - Browser does not support geolocation
9. - Error occurs

First the user or the device holder must allow the app to enable its devices location. Then geolocation will perform to start the operation to collect the information as an initiator. While it gets access then the navigator.geolocation performs to build a reliable connection among the device and service provider to transfer the needed data. After that the get_current_position method receives the latlang(latitude and longitude) value. Which actually indicates the position of an user's location.

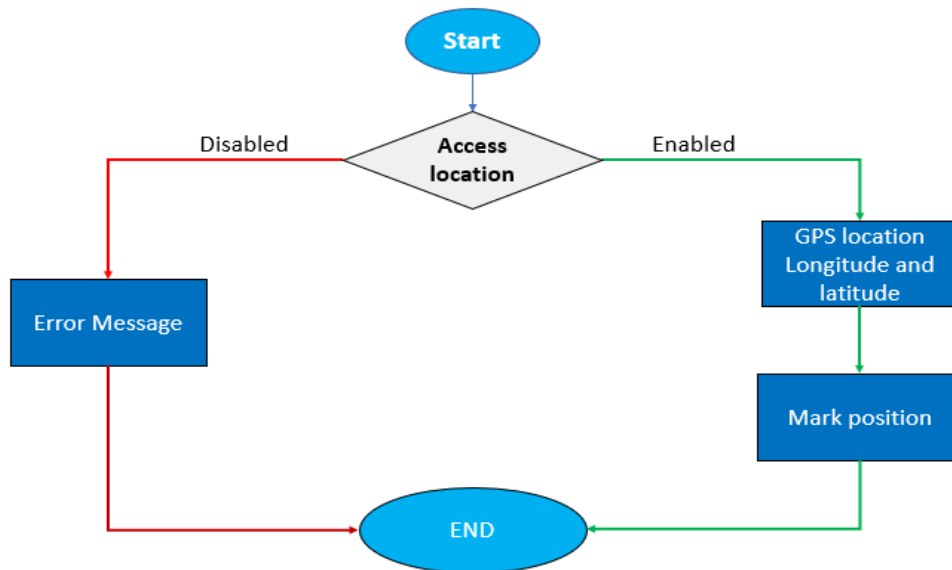


Figure 4.11: Flowchart of find user location

Show bus stoppages:

As there is no actual bus stops database available we had to assume that a dummy dataset of all the bus stoppage. If a user interacts with the service he can show all the bus stoppage at a glance based on the layout of his device. Whenever user performs a required operation to have a look of all the bus stoppage the method will take place.

This means, select all the bus stoppage in the dataset and return the longitude and latitude. Then according to the pair of data it will place a mark on the Google map. Which will indicate the bus stoppage in the database that are inserted into the table as a tuple of information.

Nearby Bus Stop:

This feature is implemented for a user to get the closest bus stop to get an appropriate bus based on the destination. Suppose a person is in a point A that could be achieved by performing action user location. Then distance function will take place to get distances from each stoppage located into a certain range respective to the user's location point. Google provides flying path distance as a building function. After that bus stoppage having minimum distance will be visible to the user. If at initial state it does not cover any available stoppage then the range will increase but does not exceed the threshold value.

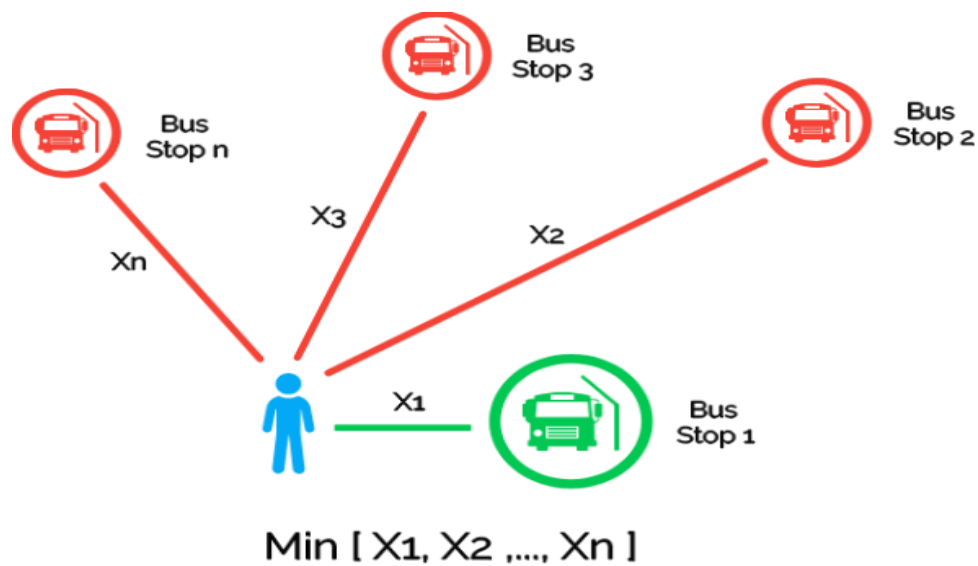


Figure 4.12: Distance calculation between user and bus stoppages.

Steps:

1. User location operation performs and returns the latlang (latitude and longitude)
2. Get all bus stoppage location latlang
3. Calculate distance between user and all bus stop location
4. Finds the minimum one
5. Returns the corresponding minimum distance bus stop latlang
6. Mark it on Google map
7. Show the route from user location to bus stop
8. Show time and distance

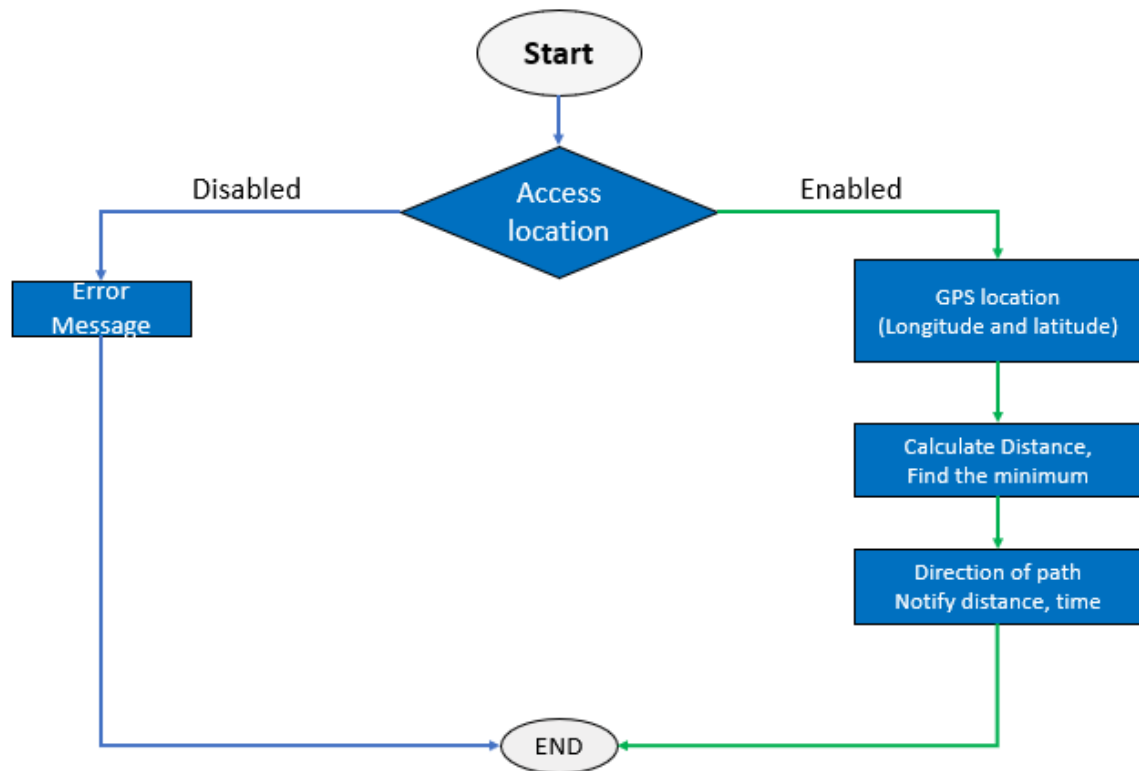


Figure 4.13: Flowchart of Nearby Bus Stop

Locate Nearest Bus:

To find the nearest bus of a certain bus stoppage first we need to find the nearest bus stop of a user according to its source to destination input. After performing the nearest bus stop function it will mark the nearest bus stop.

Then we will find the busses which did not pass that particular stoppage. From that set of buses, distance function will calculate the distance according to the bus stop. Containing the minimum distance will be the nearest bus stop.

Pseudo code

1. Performs user location method
2. Performs nearest bus stop method
3. Find those buses which did not pass the nearest bus stop
4. Calculate distance respect to the bus stop and all the buses did not pass that bus stop.
5. Find the minimum distance and return the bus's latlang
6. Mark it on the map.

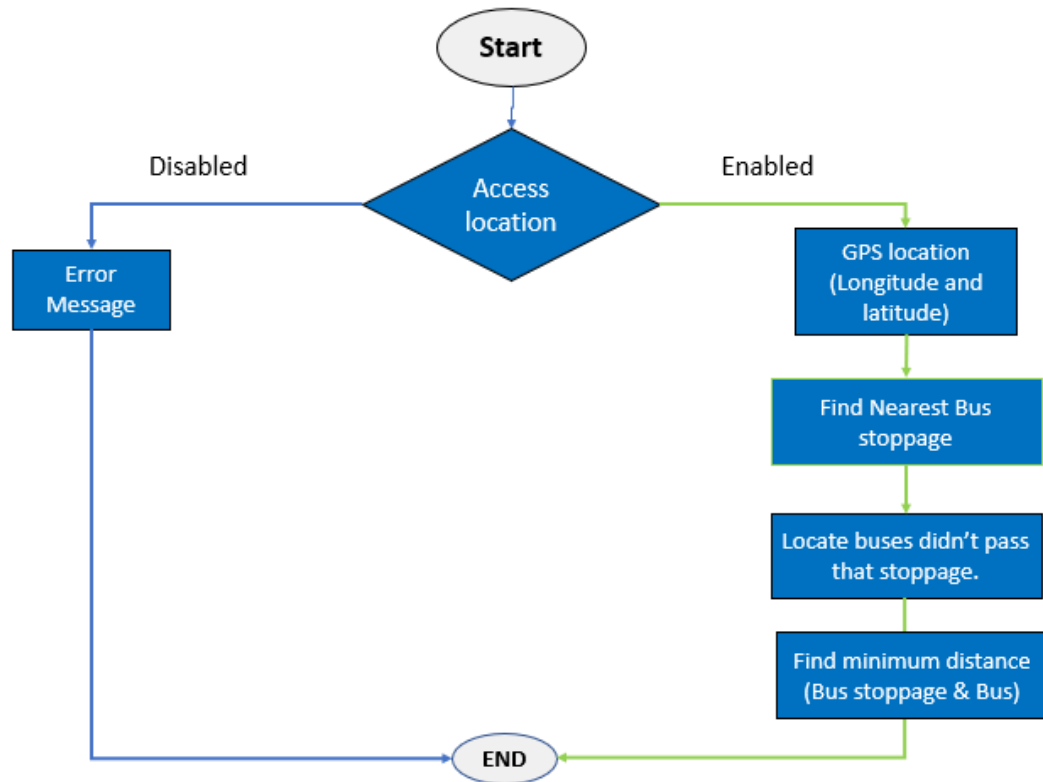


Figure 4.14: Flowchart of find nearest bus

Identify possible ways:

To implement our efficient routing feature to find the optimal path from source to destination we have to use an algorithm, which is called the A* Search algorithm. It provides an optimal and complete path from source to destination. But we are extracting all the paths using the algorithm and choosing the best three paths among those. We will provide all of the three paths to the user and will highlight a path based on user requirements (distance, time).

Pseudo code:

Input: source, destination

Output: source to destination with all paths

Method:

```
1. check_destination(n)
2. expand(n)
3. open_list(start)
4. closed_list(∅)
5. actual_cost(start) = 0
6. heuristic(start)
7. total_cost(start) = actual_cost(start) + heuristic(start)
8. if check_destination(start) == TRUE then
9.     return get_path(start, start)
10. end if
11. while open_list ≠ ∅ do
12.     ascending_sort(open_list)
13.     n == open_list.pop ()
14.     childs = expand(n)
15.     for every child ∈ childs do
16.         total_cost(child)
17.         if check_destination(child) == TRUE then
18.             path = get_path(start, child)
19.             path_list(path)
20.         end if
21.         if child ∩ closed_list == ∅ then
22.             open_list(child)
23.         end if
24.     end for
25.     closed_list(child)
26. end while
27. return path_list
```

Primarily the algorithm will take users input to execute the algorithm. As input it is taking the source and destination location from the user. The output will be all the possible paths between source and destination. To achieve the output it uses some process. At first it will insert start into an array named "open_list" and ∅ into another array named "closed_list". Then it will check whether the source is the destination. A function will be called to check this operation, which is "check_destination" function. It will take a node as input and will check whether it's the destination or not. If the start is the destination then another function will be called named "get_path". It will take two inputs as from and to. The function will then return a path between from and to parameters. If start is not the destination then it will enter into a while loop. The loop will continue until open_list is ∅. In the while loop the first thing it will do is sort the open_list in ascending order to rank the array's value with least total_cost. This cost is calculated by adding the actual_cost and heuristic of a corresponding node. As heuristic it is using Euclidean Distance.

$$Distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Here the data tuples x_1 and x_2 represents the longitude and latitude and the distance between the two co-ordinates are refers to flight path distance, which is collected from Google Map. After that it will pop the first value from the open_list and put that in a variable named “n”. This will store the node with the least total_cost into the variable “n”. Then an expand function will be called, which will explore the node and will find the available connected nodes. These connected nodes will be stored in an array named “childs”. Then the algorithm will run a for loop for every child in the “childs” array. For every child of the array it will call the total_cost function which will calculate the cost of traversing this node from the source node and will add this cost with the node. After that it will check whether the child is destination or not using “check_destination” function. If it is the destination then it will take path from source to child using “get_path” function and will store it in a variable named “path”. This will then be inserted into an array named “path_list”. After that whether it is destination or not another condition will be checked. The condition is if the child is in the “closed_list” or not. If it is not in the “closed_list” the child will be added into the “open_list”. This whole process will be done for every child. When the loop breaks the childs will be added into the “closed_list”. This will continue for all the nodes and its childs until the “open_list” is \emptyset . When the while loop breaks all the possible paths from source to destination is stored in “path_list” and the “path_list” will be returned as the output. Thus, the algorithm will stop.

Individual Organizational Support:

This feature is implemented to provide service to organizations so that they can moderate the service they are providing. The user can get a visual view of where the accessible vehicle is situated at.

1. Passenger’s location:

The driver can see the hotspot of a next stoppage whether there is anyone to be needed to get aboard.

Pseudo code:

1. If user_location access is granted using geolocation, then
2. Navigator.geolocation performs
3. getcurrentposition function incurred
4. returnlatlang
5. end if
6. else
7. error
8. - Browser does not support geolocation
9. - Error occurs

2. Vehicle's location:

It performs as user location finding, where bus device will need to allow its location showing acceptance and the confirmation of its availability. The available bus will perform a simultaneous location finding operation which will return a package of longitude and latitude which will glow onto the map.

Pseudo code:

Vehicle side:

1. **If** user_location access is granted, **then**
2. Function (get_location)
3. Navigator.geolocation performs
4. Get_current position function executed
5. Return latlang,numeric value
6. **If** Insert meaningful data available into database
7. **End if**
8. **Else**
9. Return error
10. Function end
11. Callback function(getlocation,3ms)
12. Performs get location function after 3 milliseconds sequentially.
13. **End if**

UsersSide:

1. User authentication checked
2. Perform required bus service
3. Related buses locations are picked from the database as a package
4. Package extracted and marked on map

To get this service the user must be authenticated. Then he has to select the type of service he wants to take. On the point of UIU there are several routes. So which route bus information is needed must be predefined. Then according with the preference of the vehicles position will be displayed.

4.4 Interface:

HTML:

HTML is known as Hyper Text Markup language. This language is used for creating various kinds of web pages which are displayed in the browser we see today. It has various properties, properties such as working with links, images, writing, and even sound and video [13]. The term is markup is used because HTML marks this content and describes each of this content so that the browser can show them in the web page in the correct way and in the correct order.

HTML is the skeleton of a web page, it is known as the building block of a structured web page. And it structures the web page using markup [14]. Each of the content HTML works with represented as tags. These tags can hold contents such as heading, paragraph, table, image and so on. Browser does not present these tag to the user, it shows the contents of those tags. A simple example can explain this further more:

Example [13]:

```
<!DOCTYPE html>
<html>
<head>
<title> Example Page </title>
</head>
<body>
<h1>This is a heading. </h1>
<p>This is a paragraph. </p>
</body>
</html>
```

In the example, the first tag contains DOCTYPE html it is not an HTML tag rather it is an instruction for the browser saying in which HTML version the page is written in. Secondly we see the tag <html>; this holds everything that the webpage will present. The <head> tag holds the title which is defines the title of the page or document. The head tag contains the CSS part of the corresponding webpage. The head tag can also contain <link> which is an external third party resources which can enhance and make the webpage efficient for the user to use. Lastly the <body> part holds all the tags such as <h1> - Heading, <p> - Paragraph, - Image etc which makes a webpage complete for the web browser to display in the correct way.

CSS:

CSS is known as Cascading Style Sheet. It is a one of a kind tool for describing the HTML tags, how the elements are displayed on the browser. It controls the layout of the page; it can also handle multiple layouts of multiple pages at a time [15]. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts.

This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content [16].

CSS is mainly used for the style part of elements that are in a webpage. Style for such as design of each element, layout of page, variations in displaying the page, animation etc to make a webpage look standard and user friendly for the user. CSS is also used for handling responsiveness. Using CSS a page can be showed in many platforms having many different layouts. A simple CSS example can explain more:

Example [15]:

```
<html>
<head>
<style>
    h1 {
        color: red;
        border: 1px solid red;
    }
</head>
<body>
<h1>This is an example for CSS. </h1>
</body>
</html>
```

In the example, we can see that a page containing a heading. And for that heading, in the style part of head we have a style property for heading. Property such as that the heading will have a colour red, and it will have a box border of 1px and its colour will be red and its style will be solid. CSS can also be handled by JavaScript.

JavaScript:

JavaScript is a high-level, interpreted programming language [18]. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. JavaScript is a scripting or programming language that allows handle complex things to be implemented in a web page. Using JavaScript we can show not just static page but timely content updates, user interactive page, maps, animation of 2D/3D etc.

If HTML and CSS is the first two layer of a cake then the third layer is JavaScript [19]. JavaScript can calculate, manipulate and validate data [17]. JavaScript is commonly used for client-side validation. Most commonly used app Google Map is fully handled by JavaScript. Using JavaScript we can handle HTML.

Example [17]:

```
<html>
<body>
<h1 id="header"></h1>
</body>
<script>
document.getElementById("header").innerHTML = "This is a JavaScript example.";
</script>
</html>
```

In the example using JavaScript the heading of the html page is filled with a text using DOM property of JavaScript. Using JavaScript every element of HTML can be manipulated. JavaScript can also manipulate CSS.

Example [17]:

```
<html>
<body>
<h1 id="header">This is a JavaScript example.</h1>
</body>
<script>
document.getElementById("header").style.color = "red";
</script>
</html>
```

In the example, JavaScript is changing the colour of the heading of the page using DOM property of JavaScript. JavaScript can also handle browser part.

Example [17]:

```
<script>
var width = window.innerWidth
    || document.documentElement.clientWidth
    || document.body.clientWidth;
</script>
```

In the example, JavaScript is handling the layout of webpage according to the user's device.

Bootstrap:

Bootstrap is a very commonly and famous framework of CSS. It was developed by Mark Otto and Jacob Thornton in August 2011. It's the base of CSS framework. It's faster and more efficient for developing a webpage [20].

Bootstrap provides a webpage with many important blocks such as modals, navigation, and image carousel and so on. Bootstrap also provides additional JavaScript plug-ins. Bootstrap gives the ability to make a webpage device responsive.

Why Bootstrap over other frameworks?

A framework is judged by three things [21].

- (1) Popularity
- (2) Active development
- (3) Good Documentation

If a framework has an average of above three things then that framework is judged as a good framework. There are many frameworks like bootstrap. Foundation 5, Semantic UI, Pure, Uikit, W3school etc are strong candidates like bootstrap. But among these bootstrap has gained the most popularity among web development community and became the easiest, simplest and efficient framework among all. So choosing bootstrap was a calculated decision.

Chapter 5

Limitations and Future Works

5.1 Limitations

The massive problem was that, we were no authentic data available on public transport system. So, we had to gather data by organizing survey and rely on our own travelling perceptions.

The bus companies were not familiar with the tracking device. They were not showing any kind of willingness to allow us to attach a tracking device. That's why we had to turn on a mobile device into a bus to track a bus location.

We tried to build up a mobile application. But the available service we needed to add in our component was costly. That's why we had to choose an alternate platform.

5.2 Future Works

It is possible to build up mobile based software if we get enough resource.

A secure network connection can be introduced with this system. So that, the user can get a secure connection with the server.

To pay fair using electronically medium can be added with this system. A user can pay his/her estimated fair using online service.

To provide smooth search result, meaningful pattern can be generated from a user's previous search history.

A machine learning approach could be introduced to improve the estimated time that will take a bus to reach another point. That is possible to achieve by taking 4 steps:

1. Generating related dataset for predicting arrival time
2. Selecting features from dataset
3. Initial weight of the features
4. Prepare a model.

Generating related dataset for predicting arrival time

From the original dataset we have to gather all the instances that is required specify the model. Suppose a road is connected to the bus stop A to B. Then a sub dataset will arise $D_i \in D$ with the information about that road on a specific time request. As the average speed, arrival time of a bus, departure time of a bus, weather, delay etc.

Selecting features from dataset

Ensemble extra trees classifier will be used to compute the importance of features. The chance to select a feature depends on the impact of that feature on dataset.

Suppose we have a set of importance list [10,5,6,3,2,4,5] with respect to the features $[X_1, X_2, X_3, X_4, X_5, X_6, X_7]$. If we specify a threshold value 4 that represents the acceptance of features, then the final selected features will be $[X_1, X_2, X_3, X_6, \text{ and } X_7]$.

Calculate initial weight of the features

To calculate the initial weight of the features we can use any of the method from ID3 (Quinlan, 1986), ID4 (Utgoff, 1989), ID5 (Utgoff, 1988), C4.5 (Quinlan, 1993), C5.0 (Bujlow, Riaz, & Pedersen, 2012), and CART (Breiman, Friedman, Stone, & Olshen, 1984)

ID3:

First, we need to calculate the info of the dataset. If the dataset is D and the probability of an attribute is P_i Then equation to achieve info,

$$Info(D) = - \sum_{i=1}^n k_i \log_2(k_i)$$

We will then calculate the info respect to the attribute gain

$$Info_F(D) = \sum_{j=1}^m \frac{|D_j|}{|D|} \times Info(D_j)$$

The final output we need to achieve is the gain

$$Gain(F) = Info(D) - Info_F(D)$$

C 4.5:

Gain ratio is extension information of gain.

$$SplitInfo_F(D) = - \sum_{j=1}^m \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Then we can calculate the gain ration

$$GainRatio(F) = \frac{Gain(F)}{SplitInfo(F)}$$

CART:

To explore the importance of instance the frequency is effective.

$$Gini(D) = 1 - \sum_{j=1}^n k_j^2$$

Depending on the split of dataset we can calculate the GINI using,

$$Gini_F(D) = \sum_{j=1}^n \frac{|D_j|}{|D|} \times Gain(|D_j|)$$

Prepare a model

Linear regression:

Linear regression is basically an update form of linear equation.

$$Y = mX + c$$

Where Y is the value we need to predict with respect to the feature X and weight m.

In further the equation can be derived as

$$Y = W_0 + W_1X_1 + W_2X_2 + W_3X_3 + \dots + W_nX_n$$

Then we will tune the weight $[W_0, W_1, \dots, W_n]$

If we update the value as it appears then we may not obtain at a global optimum position. That's why we will use a learning rate to update our model iteratively.

$$W_j = W - \alpha W_{j-1}$$

After finding an appropriate value for W we will multiply the test instance to find the predicted value. We have to add the delayed time with the predicted estimation time.

Chapter 6

Conclusion

The proposal Smart Navigation System is a modern solution to the problem that is related to local transport system. The covered working task is able to guide a user to find a suitable way so that he can get maximum satisfied knowledge on a bus that is required to reach his/her destination. This several steps will be able to improve the popularity of local transport system which will lead the user to an utmost level of satisfaction. From finding nearest bus stop to reach at destination, this task is divided in several separate minor operations. It is way too much easy for a user to interact with the system. Visual feedback has been introduced to rely the user about his/ her last attempt. In one word, we can say that this process is helpful in a crowded city, so that the user can get the utmost information that is needed to end their daily excursion smoothly.

References

- [1] Chong Et Al (October 11, 2018). Database [Online].
Available: <https://en.wikipedia.org/wiki/Database>. Accessed: (October 1, 2018)
- [2] MySQL AB (September 27, 2018). Overview [Online].
Available: <https://en.wikipedia.org/wiki/MySQL>. Accessed [October 1,2018]
- [3] Wikipedia (September 10, 2018). NoSQL[Online].
Available: <https://en.wikipedia.org/wiki/NoSQL>. Accessed (October 1, 2018)
- [4] Google Maps (August 13, 2018).Google Maps JavaScript v3 reference [Online].
Available: <https://developers.google.com/maps/documentation/javascript/referenc>
[e/](https://developers.google.com/maps/documentation/javascript/referenc)Accessed: (October 02, 2018)
- [5] Google Maps. (September 25, 2018). Disable The Default UI [Online].
Available:<https://developers.google.com/maps/documentation/javascript/examples/control-disableUI>.Accessed: (October 02, 2018)
- [6] Google Maps (September 25, 2018). Controls [Online].
Available:
<https://developers.google.com/maps/documentation/javascript/controls>Accessed:
(October 02, 2018)
- [7] Google Maps (September 25, 2018). Info Windows [Online].
Available:
<https://developers.google.com/maps/documentation/javascript/infowindows>
Accessed: (October 02, 2018)
- [8] Google Maps (September 25, 2018). Markers [Online].
Available:<https://developers.google.com/maps/documentation/javascript/markers>
Accessed: (October 02, 2018)
- [9] Google Maps (September 25, 2018). Marker Remove [Online].
Available:<https://developers.google.com/maps/documentation/javascript/examples/marker-remove>Accessed: (October 02, 2018)
- [10] Google Maps (September 25, 2018). KML Layers [Online].
Available:[https://developers.google.com/maps/documentation/javascript/exampl](https://developers.google.com/maps/documentation/javascript/examples/layer-kml)
[es/layer-kml](https://developers.google.com/maps/documentation/javascript/examples/layer-kml) Accessed: (October 02, 2018)
- [11] Google Maps (September 25, 2018). Direction [Online].
Available:<https://developers.google.com/maps/documentation/javascript/directio>
[ns](https://developers.google.com/maps/documentation/javascript/directio)Accessed: (October 02, 2018)
- [12] Google Maps (September 25, 2018). Roads API [Online].
Available:<https://developers.google.com/maps/documentation/roads/intro>Access
ed: (October 02, 2018)
- [13] Google Maps (September 25, 2018). Create Map Style [Online].
Available: <https://mapstyle.withgoogle.com/>Accessed: (October 02, 2018)

- [14] RestApiTutorial.com.RestApi Tutorial [Online].
Available: <https://www.restapitutorial.com/>Accessed: (October 02, 2018)
- [15] Refsnes Data (1999-2018).What is HTML [Online]
Available: https://www.w3schools.com/html/html_intro.aspAccessed: (October
1, 2018)
- [16] T. Berners-Lee (September 27, 2018). HTML [Online]
Available: <https://en.wikipedia.org/wiki/HTML>Accessed: (October 1, 2018)

- [17]Refsnes Data [1999-2018].What is CSS [Online]
Available: https://www.w3schools.com/css/css_intro.aspAccessed: (October 1, 2018)
- [18]H. Wium Lie (September 26, 2018).Cascading Style Sheets [Online]
Available: https://en.wikipedia.org/wiki/Cascading_Style_SheetsAccessed: (October 1, 2018)
- [19]Refsnes Data (1999-2018).What is JavaScript [Online]
Available: https://www.w3schools.com/whatis/whatis_js.aspAccessed: (October 1, 2018)
- [20]B. Eich (October 7, 2018).JavaScript [Online]
Available: <https://en.wikipedia.org/wiki/JavaScript>Accessed: (October 1, 2018)
- [21]MDN web docs (2005-2018).A high-level definition
Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScriptAccessed: (October 1, 2018)
- [22]Refsnes Data (1999-2018).What is bootstrap [Online]
Available:
https://www.w3schools.com/bootstrap/bootstrap_get_started.aspAccessed: (October 1, 2018)
- [23] SitePoint (2000-2018). The 5 Most Popular Front-end Framework Compared
Available: <https://www.sitepoint.com/most-popular-frontend-frameworks-compared/>Accessed: (October 1, 2018)