ROUTE SUGGESTION SYSTEM (RSS)
USING DIJKSTRA'S ALGORITHM

NURUL ATIQAH SABRI

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE DEVELOPMENT)
UNIVERSITI SULTAN ZAINAL ABIDIN

2017

ROUTE SUGGESTION SYSTEM (RSS) USING DIJKSTRA'S ALGORITHM

NURUL ATIQAH SABRI

Bachelor of Computer Science (Software Development)

Faculty of Informatics and Computing

Universiti Sultan Zainal Abidin, Terengganu, Malaysia

MAY 2017

DECLARATION

I hereby declare that this report is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Sultan Zainal Abidin or other institutions.

_____

Name: Nurul Atiqah Bt Ahmad Sabri

Date : 11/5/2017

CONFIRMATION

This project report entitled Route Suggestion System is prepared and submitted by Nurul Atiqah Naquiyyah Bt Ahmad Sabri , matric number 930726055264. I hereby certify that this report has met its condition and requirements to qualify for Bachelor of Computer Science (Software Development) in Universiti Sultan Zainal Abidin (UniSZA)

_____

Name : Dr Wan Suryani bt Wan Awang

Date : 11/5/2017

ACKNOWLEDGEMENT

Alhamdulillah, all praises to Allah for the strength and His blessings that I managed to complete this project. Special appreciation goes to my supervisor, Dr Wan Suryani Wan Awang for her supervision and support. Her invaluable help and suggestions throughout the project have contributed a lot. Not forgotten, my appreciation to the panels and lecturers.

Sincere thanks to all my friends and seniors for their kindness and moral support during my study. My deepest gratitude goes to my beloved parents, Mr. Ahmad Sabri Awang and Mrs. Norhiza Mohd Salleh for their prayers and encouragement. To those who are indirectly contributing, your kindness means a lot to me. Thank you

ABSTRACT

The Route Suggestion System is a web based system which would enable users to choose the shortest route they will take. In this project travelers may choose the best route to choose. There are two types of user which is admin and travelers. Travelers can only check the shortest route.

This project is basically targeted to travelers. This system will save the travelling time and cost and definitely will make travel arrangement easy and efficient. This project uses Dijkstra's algorithm to solve the problem.

CONTENTS

LIST OF ABBREVIATIONS / TERMS / SYMBOLS

CD	Context Diagram
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
FYP	Final year project
GA	Genetic algorithm
HCI	Human computer interface

CHAPTER I

INTRODUCTION

1.1 Background

This chapter aims to describe the project background, problem statement, objectives and scopes. The aim of this is system to help any traveller around the world to find the shortest route.

In Malaysia, public transportation normally focusing on local people to travel around the Malaysia. The travelling information are for international traveller are still lacking. This information include the shortcut route via city from one destination to another destination.

Route Suggestion System (RSS) enables user to choose the shortest route they will take. Travellers, will have an information in advance. This kind of online system is a new system because it doesn't exist in Malaysia based. Currently, travellers doesn't have any choice to choose different routes to take when going from one location to another location. The Route Suggestion System enables travelers to choose from one location to another location via different route (city). This project will be used PHP language and MySQL to develop the RSS. It is hoped that the development of this system can solve the problem.

1.2 PROBLEM STATEMENT

Practically, often confuse travelers lack of information include the shortest route via city from the origin to the destination. Travelers usually have less option to choose what route (city) to take from one location to another location. Normally, travelers have fix time to travel, for example, they have only one week to travel due to this traveler need to find the shortest route in order for them shorten the travelling cost and save cost

1.3 OBJECTIVES

Generally, the objective to develop Route Suggestion System (RSS) is to help the travelers around the world to find the shortest route trip from one location to location.

i. To design a Route Suggestion System (RSS) that can give information about the route (city) to choose for travelling from the origin location to the destination

ii. To implement the Java and Java Server Page (JSP) technique in the Route Suggestion System (RSS)

iii. To test the functionality of the system, so travelers will have an option to choose the shortest route (city) to take from one location to another location

1.4 SCOPE

Route Suggestion is a web based system. The scope of this system to help the travelers around the world to find the best route.

In this Route Suggestion system, travelers, and the other one are the admin is the user of this system. Travelers need to make registration or login before travelers can choose the route (city). Travelers will enter the destination they want to go then the system will give an option the best route (city) to choose. Travelers have the option to choose from one location to another location via a different route (city). They will choose based on that. Admin's scope, the admin will check a monthly report, manage user and arrange a database.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

A literature review is a critical and in depth evaluation report of previous research related to the selected area. The review must be described, evaluate and clarified into a summary or synopsis of a particular area of research. It can also be a precursor to introduce the research or an overview of a research. The literature review provides background for the topics of previous research to anybody reading the paper. The literature review in this chapter will discuss and review about the current system, route suggestion system, methods in Dijkstra's algorithm and the implementation of the system.

2.2 DIJKSTRA'S ALGORITHM

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. The algorithm exists in many variants, Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree. For a given source node in the graph, the algorithm finds the shortest path between that node and node other. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities.

2.3 Study and Analysis of Shortest Path Algorithms

2.3.1 Definition

Shortest path problem is a fundamental technique in computer networking for route discovery. Utilizing the shortest path algorithms overall costs of setting the network is reduced. Shortest path trees are made when shortest paths of remaining nodes are calculated from on a single root node. The algorithms determining the shortest path that is summarized in the following subsections have been classified under an efficiency analysis. Dijkstra's algorithm calculates the shortest path between a given initial node to a single destination node and all other nodes of the graph when all edges have positive weight values. The algorithm selects any node I as a starting node. Then the algorithm builds a tree T that ultimately spans all vertices reachable from S1. Vertices are inserted into a tree T according to the distances, i.e., first S1, then the vertex closest to S1 and so on. Initially the distance between each node is set to infinity. At the beginning of the algorithm the source node i is set as the current node. From this node the tentative distances of all unvisited neighbour nodes are calculated that are connected by edge. The current node is then marked as the visited node and the next current node is selected which has the minimum tentative distance from the previous current node. The algorithm is terminated when all nodes have been visited and the set of unvisited nodes is empty. This time can be reduced by using different variants of Dijkstra's algorithm.

2.3.2 Steps of Dijkstra algorithm's operation

In Figure 2.1 to Figure 2.6 shows the network graph of steps taken in Djikstra's algorithm operation.

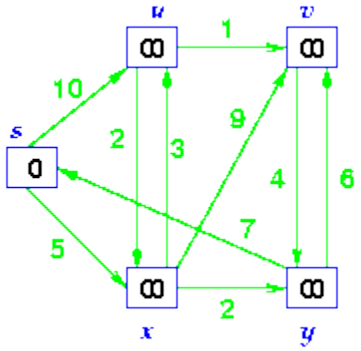

Figure 2.1

Step 1. Given initial graph G=(V, E). All nodes have an infinite cost except the source node, s, which has 0 cost.

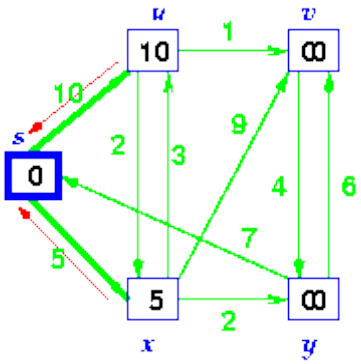

Figure 2.2

Step 2. Choose the node, which is closest to the source node, s. We initialized [s] to 0. Add it to S. Relax all nodes adjacent to source, s. Update predecessor for all nodes updated.

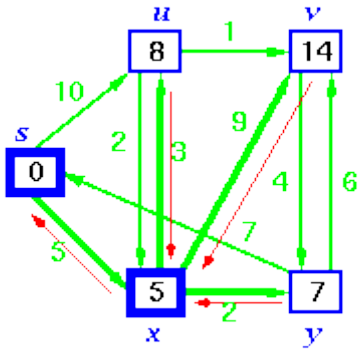

Figure 2.3

Step 3. Choose the closest node, x. Relax all nodes adjacent to node x. Update predecessors for nodes u, v and y.

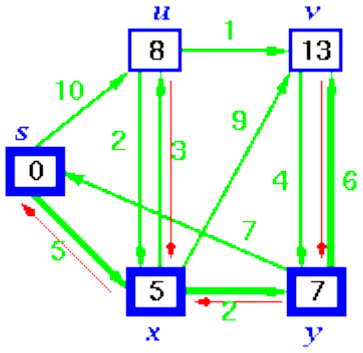

Figure 2.4

Step 4. Node y is the closest node, so add it to S. Relax node v and adjust its predecessor

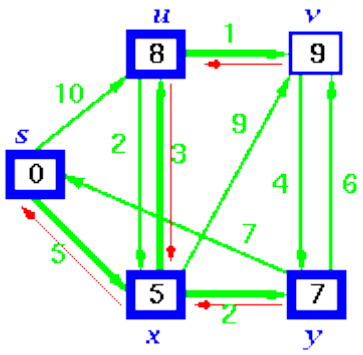

Figure 2.5

Step 5. Node u that are closest. Choose this node and adjust its neighbour node v.

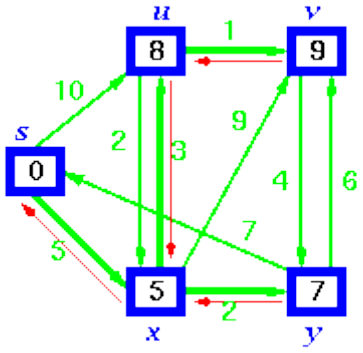

Figure 2.6

Step 6. Add node v. The predecessor list now defines the shortest path from each node to the source node, s.

2.4 Comparison between Methods

No	Author/Year	Method	Advantage
1.	R.J Elliott and M.E Lesk 1982	Breadth-first search	-If there is a solution, BFS will definitely find it out. -. If there is more than one solution, then BFS can find the minimal one that requires less number of steps.
2.	Chandler Burfield 2013	Floyd-Warshall Algorithm	-Floyd-Warshall to build shortest paths from smaller shortest paths, in the classic dynamic programming way.
3.	Rabia Arshad, Danista Khan, Muhamamd Arslan Shahid, Syed Hammad Hussain Shah 2016	Bellman–Ford algorithm	-Negative edge weights are found in various applications of graphs, hence the usefulness of this algorithm -The Bellman–Ford algorithm can detect negative cycles and report their existence
4.	Robin 2016	Breadth-First Search	-Breadth-first search always finds a shortest path to a goal. -Breadth first search will never get trapped exploring the useless path forever.
5.	Kai Gutenschwager, Axel Radtke, Sven Volker, Georg Zeller 2012	Dijkstra's Algorithm	-The simplest implementation of Dijkstra's algorithm stores nodes in a linked list or an array, and the operation to find the minimum value in list Dist is a linear search through all nodes in Dist -two-side Dijkstra is based on the idea

			that finding the shortest path from start node i to destination node j might be carried out faster
6.	Alberto Colorni, Marco Dorigo, Vittorio Maniezzo	Ant Algorithm	-Least partially shared by members of this class of algorithms, are the use of a natural metaphor, the inherent parallelism, the stochastic nature and adaptivity, the use of positive feedback.
7.	F.Benjamin Zhan 1998	Shortest Path Algorithms	-Shortest paths from one source node to a subset of the nodes on a network can be defined as one-to-some shortest paths
8.	Anu Pradhan, M.ASCE; and G. (Kumar) Mahinthakumar, M.ASCE 2013	Floyd-Warshall Algorithm	-For solving these problems on a moderate number of symmetric multiprocessor (SMP) nodes.

Table 2.1 Comparison between methods

2.4 Comparison between methods

There are several methods that can be used in selection. Such as:

2.4.1 Route Finding in Street Maps by Computers and People (Breadth-first search)

Breadth-first search is an alternative method to find the shortest path.

If there is more than one solution, then BFS can find the minimal one that requires less number of steps.

2.4.2 Study and Analysis of Shortest Path Algorithms (Bellman–Ford algorithm)

Bellman Ford computes shortest paths between a single source node to all other nodes in a weighted digraph. It is more versatile, but slower than Dijkstra's Algorithm since it can compute the shortest path of the graphs that have negative edge which makes it useful in the applications having negative edge graphs. It works on relaxation procedure by taking two nodes and edge connecting them.

2.4.3 An investigation of some properties of an "Ant algorithm"

To test the Ant system we used the travelling salesman problem**.**

In Ant-cycle an ant is an agent that visits only towns that were not visited by it in the preceding steps, this property implemented to force the ants to make legal tours, holds until a tour is completed. Then the "ant memory" is reset and the ant is free again.

CHAPTER III

PROJECT METHODOLOGY

3.1 Project Methodology

The research methodology is essential to ensure the research objectives can be achieved. There is a variety of research methodology to be used should be compatible with the system which is being developed. The methodology can be divided in two which are Research Methodology and System Methodology. Route Suggestion System (RSS) using Dijkstra's Algorithm will be carried out using a model of Feature driven development (FDD) methodology prototype because the system development environment and system development need to develop is compatible with most of the characteristics (FDD). This chapter will explain about the use of development method, strategy, approach used to develop system and details for every phase that involve in building this system.

3.2 Research methodology

- Feasibility study
- Data and information gathering
- Framework design
- Development
- Integration and evaluation

3.2.1 System methodology

Feature driven development (FDD) is an iterative and incremental software development process that follows the principles of agile. The purpose is to develop high level features, scope, domain object model and being used to plan, design, develop and test the specific requirement and task based on the feature that they belong to and being able to estimates, scheduling and report on the status of the project at every level.

Besides that, in FDD the requirement for the project do not have the codified up front, but they are prioritize and scheduling for each iteration. The requirement consists of requirements and planning that can be scheduled into a particular release and iteration differences between agile and waterfall is the approach to quality and testing. In waterfall model the time take for the specification and specification and implementation are longer comparable to agile. The ability to respond quickly to change for agile is higher compare to waterfall model.

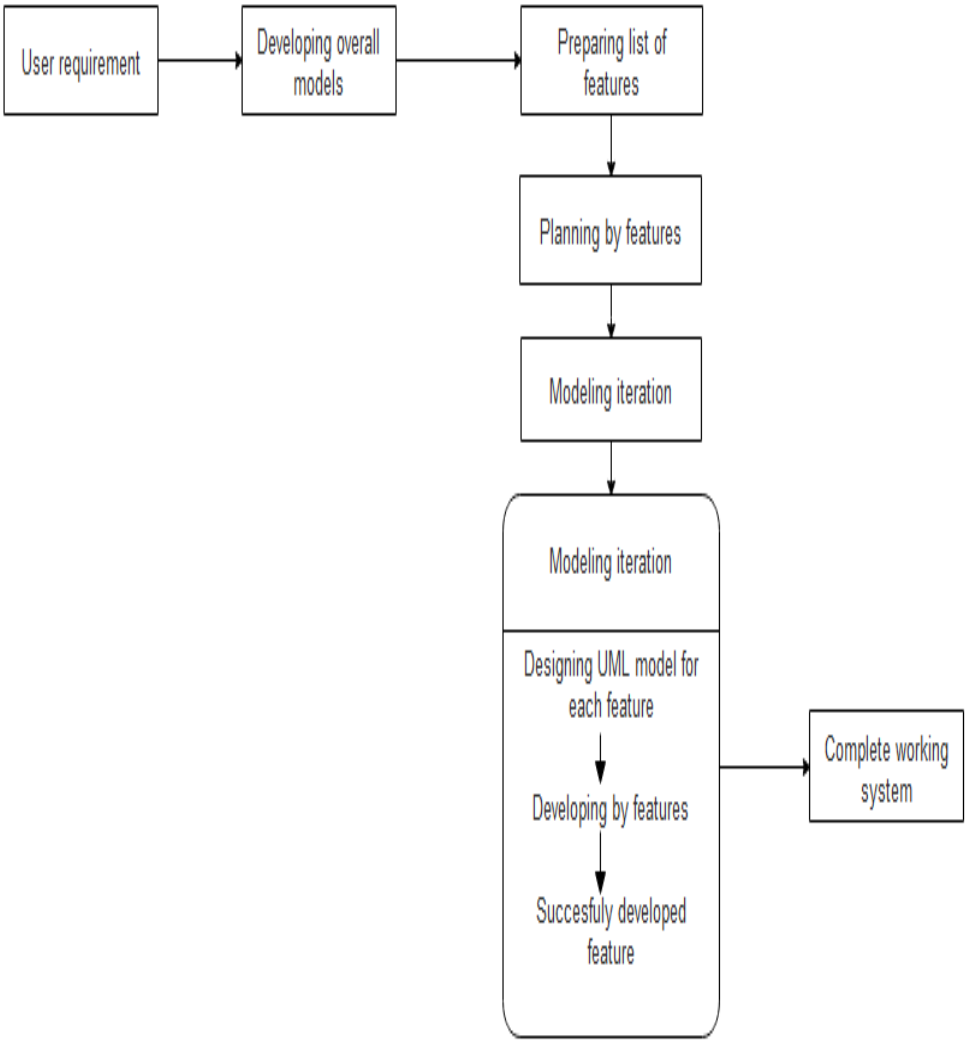

FIGURE 3.1 Feature Driven Development

3.2.3 Planning Phase

In the planning phase cost of building the system, benefit analysis, risk management plan and feasibility study are included.It has also included the development of the project planning such as schedules, Gantt Chart that function to plan, organize the report progress for the system within the system environment and system management plan.

The most important in this phase is to develop an overall model while improving the initial model before continuing to the next phase, an analysis phase.

3.2.4 Requirement an Analysis phase

This phase includes the task that goes determining the needs and condition require in order to build the system and to recognize the stakeholders (user) in building the system. Analysis of user need and requirement the needs and requirement while determining the main functionality which include creating a detailed of RSS functional and non-functional requirement of the system.

Collection of data and information for the system is done in this phase. Data gathering about information of finding the shortest route, djikstra's algorithm that will be applied in this system either from journals, articles, books white or research paper. SWOT analysis also will be done as an early precaution to recognize the weakness of the system, thus figuring a way to overcome it. All details gather are grouped into sets and course areas and being broken down as at ask or model for iteration phase.

3.2.5 Iteration Phase

Every design iteration will be designed in this phase and being built according to the iteration as planned. Besides that, at each development iteration some testing and validation will be done in order to identify whether each development iteration are in sync with the guidelines set in an earlier phase (planning phase).

Design: The first sketch (rough sketch) of iteration is the main challenge in this phase, it's also affecting the future of the whole system. The next iteration it will focus more on the improvement and adoption of technology that are being used by the software feature in which each iteration requires to produce a good model that give the best result for the end product.

Development: This phase is done after the design phase and after all the design and inspection for each feature are done successfully. Production of a complete function

Testing and Evaluation: After the code are successfully being built each of its will be test and evaluate to ensure that all the development meets the requirement of the software specification on the compatibility and suitability of the architecture on the technology that has been designed to be implemented.

3.2.6 Deployment Phase

This phase includes the implementation preparation of the system into the production environment and resolution of problems identified in the integration. Once the system passes the acceptance test, a documentation for describing task on how to function and maintain the system in production environment includes post implementation and in process reviews are done.

3.3 Software and Hardware Requirement

There are several requirements that take part in the system deployment environment which include the architecture specification. The list of all software and hardware involve in the development process are:

3.3.1 Software Requirement

Software type	Description/ Version
Web Server	Apache 2.2 X
Database Server	Workbench MySql, PHP MyAdmin
Operating System	Windows 10
Web Browser	Google Chrome, Mozilla Firefox
Presentation and Word Processing	Microsoft Word 2013, Micrososft PowerPoint 2013
Graphic Editor	GIMP
Code Editing	Netbeans, Notepad++
Diagram Production	E-draw

3.3.2 Hardware Requirement

Item Name	Description / Version
Laptop	Development Specification • Intel Inside i5 • Memory 4 GB
Printer and Scanner	Documentation Purpose

3.4 SYSTEM DESIGN

3.4.1 Framework

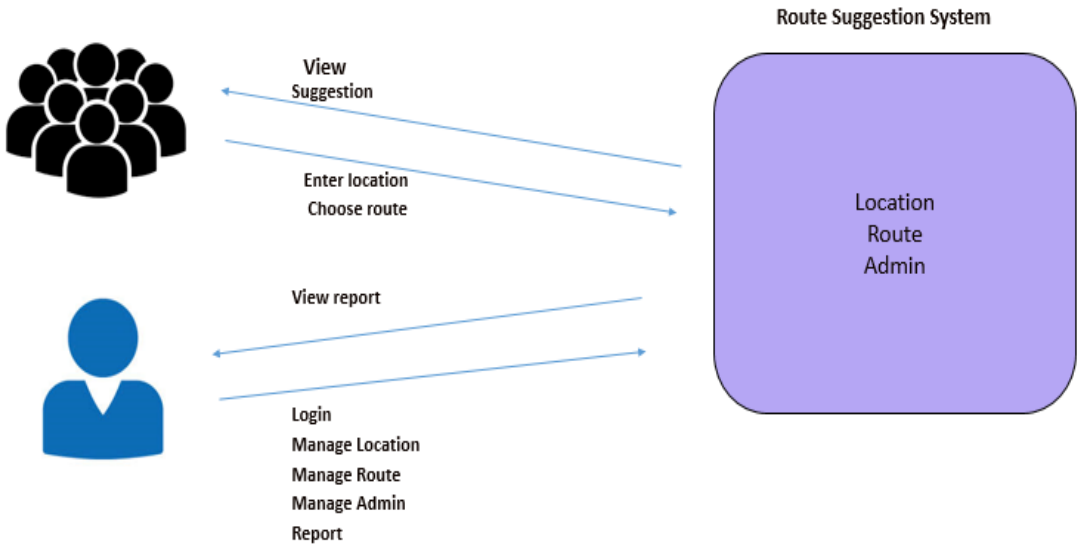

FIGURE 3.2 Framework Route Suggestion System

In this, Route Suggestion System, the traveler will be the user of this system. They can enter the origin of location details and the system will display the suggestion of the routes. Admin will be able to login, manage the location details, route details, admin details and report details and the system will display the report details.

3.4.2 Context diagram

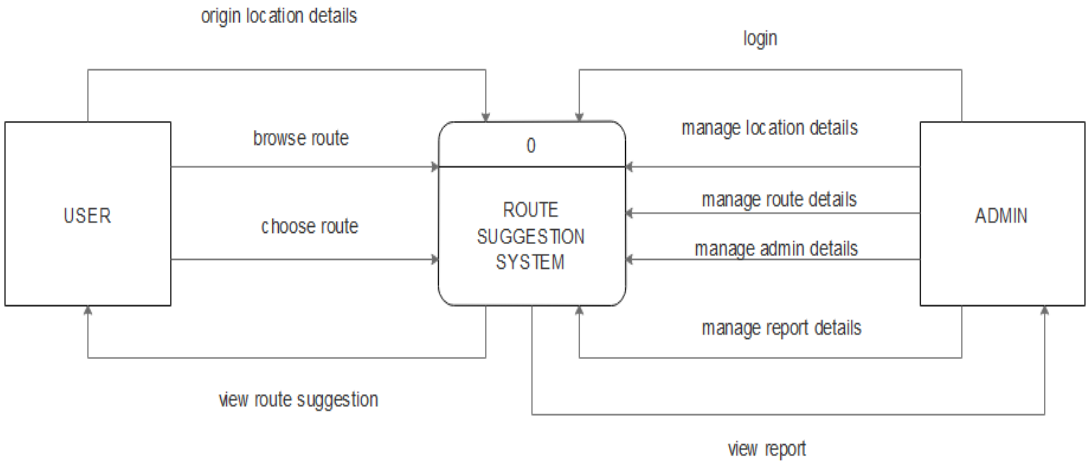

FIGURE 3.2 Route Suggestion System Context Diagram

Admin will login to the system as an admin then, will manage the location details, route details, admin details, and report details. The user will be entering the origin location details and will be received by the system. The user will choose the route suggested by the system.

3.4.3 Data Flow Diagram Level 0

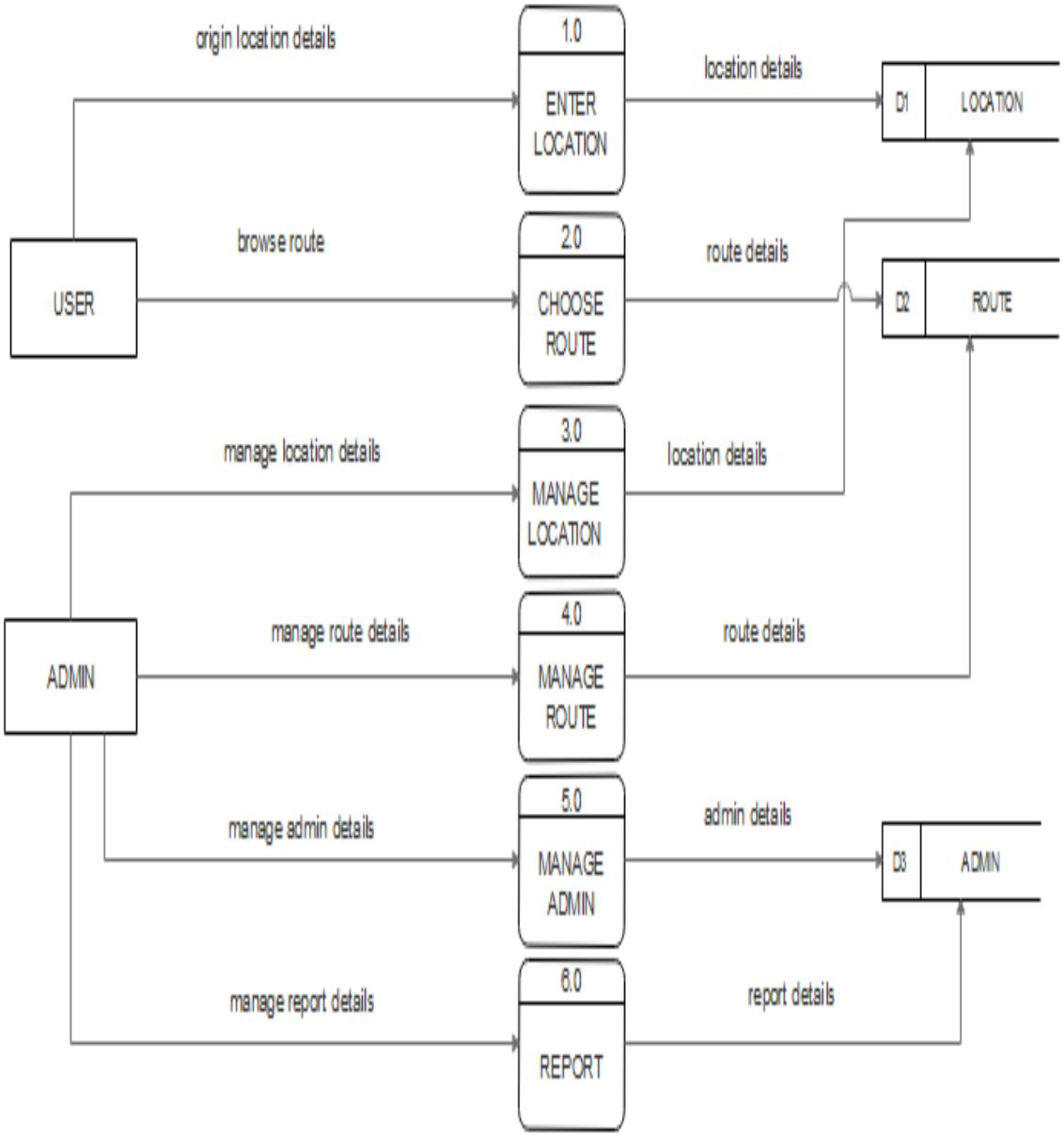

Figure 3.3 Route Suggestion System DFD Level 0

The DFD LEVEL 0 of RSS has two entities which are USER and ADMIN. ADMIN.ENTER LOCATION, CHOOSE ROUTE, MANAGE LOCATION, MANAGE ROUTE, MANAGE ADMIN, REPORT is the processes in RSS. The user will enter location and choose a route. The location details will be saved in location data store meanwhile, route details will be saved in route data store. Admin can manage location details through managing location process. The process then sends location details to location data store. Admin can manage route details through the managing route process. The process then sends route details to route data store. Admin can manage admin details through the managing admin process. The process then sends admin details to admin data store. Admin can manage report details through report process. The process then sends report details to report data store.

3.4.3.1 DATA FLOW DIAGRAM LEVEL 1

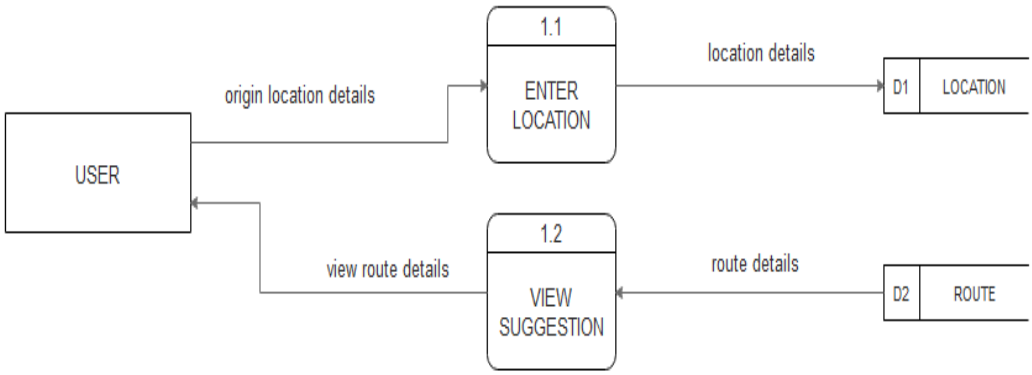

FIGURE 3.4 Data Flow Diagram Level 1 for User (Location)

Figure 3.4 shows the Data Flow Diagram Level 1 for Location. User needs to enter the location that is processed in the ENTER LOCATION process and the process, then sends location details into location data store. The route data store sends back route details to the user through the VIEW SUGGESTION process.

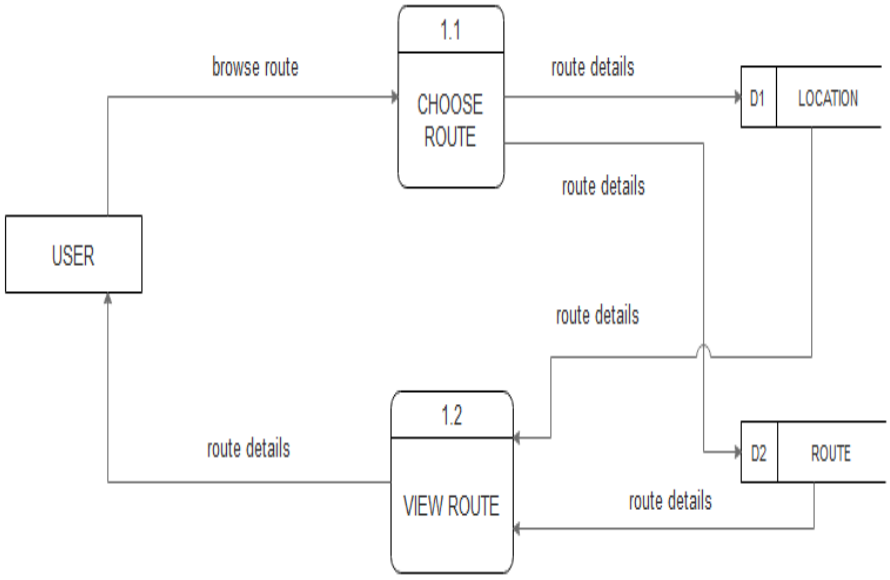

FIGURE 3.5 Data Flow Diagram Level 1 User (Choose Route)

Figure 3.5 shows the Data Flow Diagram Level 1 for Route. User needs to choose the route that is processed in the CHOOSE ROUTE process and the process, then sends route details into route data store and location data store. The route and location data store sends back route details to the user through VIEW ROUTE process.

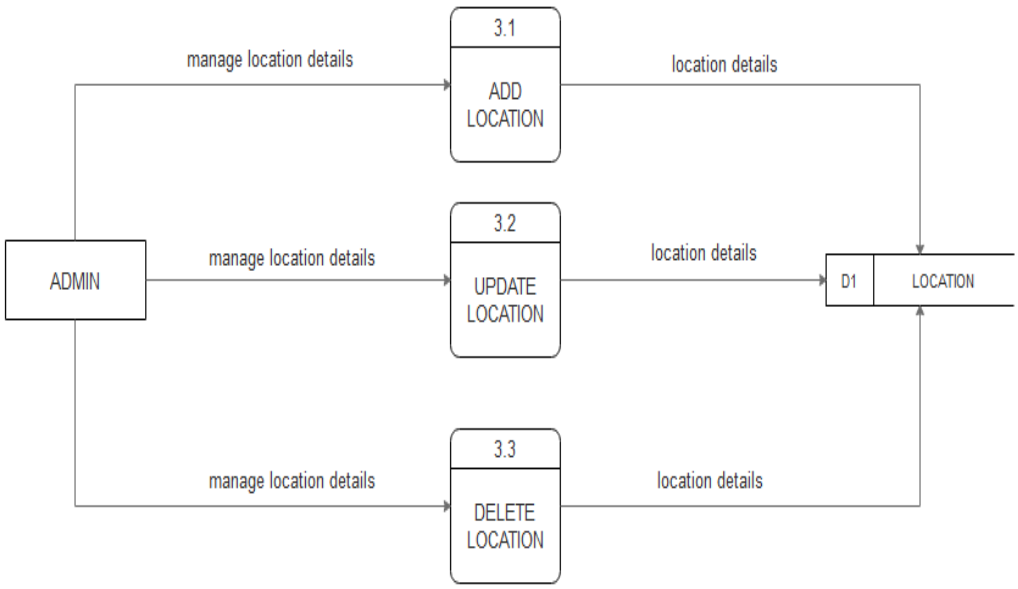

FIGURE 3.6 Data Flow Diagram Level 1 Admin (Manage Location)

Figure 3.6 shows the Data Flow Diagram Level 1 Admin (Manage Location). Admin can manage location details through the ADD LOCATION process. The process sends location details to location data store. Admin keys in location details. The UPDATE LOCATION process inputs location details into location data store. Next, admin keys in location details which create DELETE LOCATION process. The process sends location details to location data store.

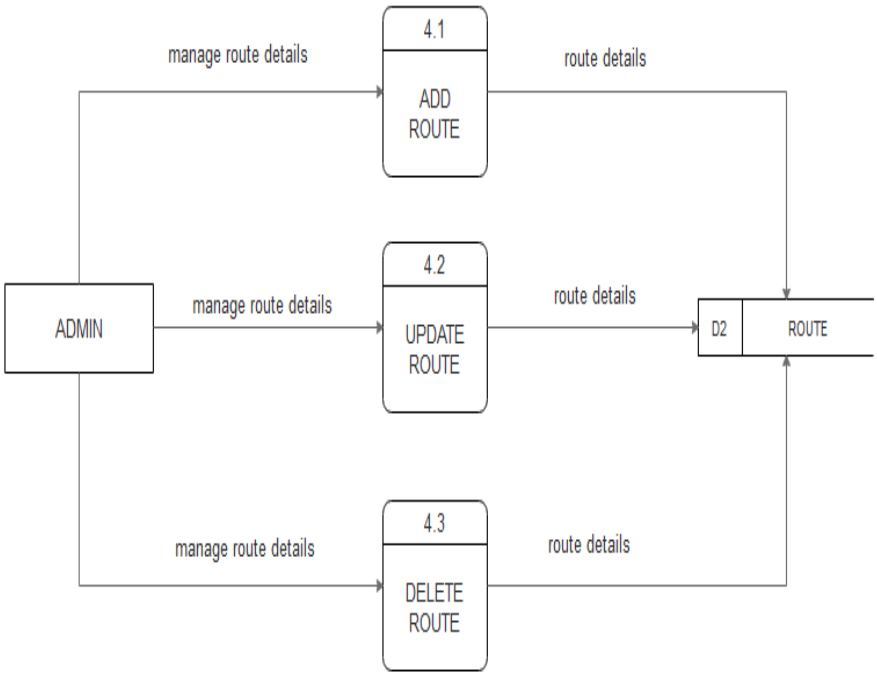

FIGURE 3.7 Data Flow Diagram Level 1 Admin (Manage Route)

Figure 3.7 shows the Data Flow Diagram Level 1 Admin (Manage Route). Admin can manage route details through the ADD ROUTE process. The process sends route details to route data store. Admin keys in route details. The UPDATE ROUTE process inputs route details into the route data store. Next, admin keys in route details which create DELETE ROUTE process. The process sends route details to route data store.

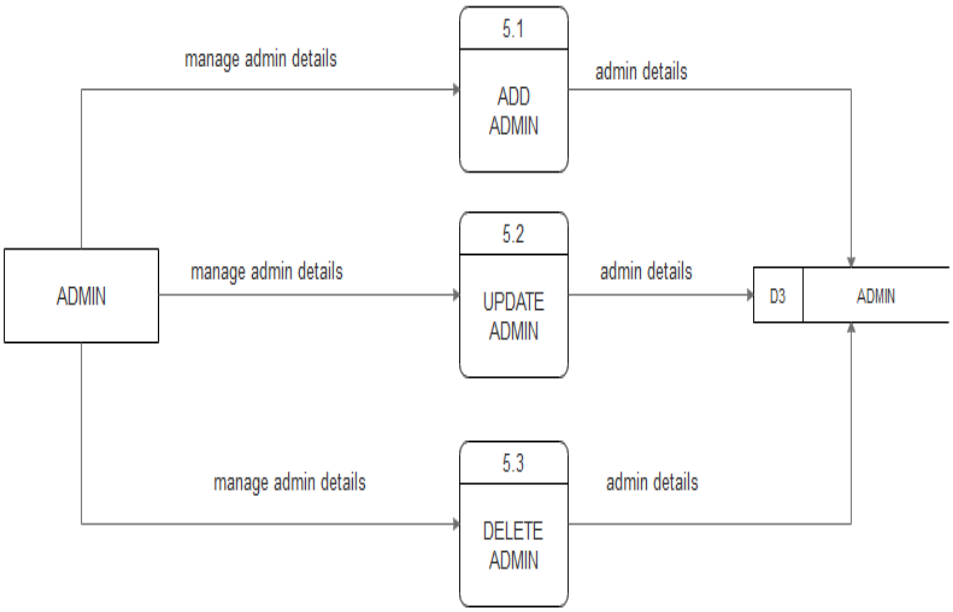

FIGURE 3.8 Data Flow Diagram Level 1 Admin (Manage Admin)

Figure 3.8 shows the Data Flow Diagram Level 1 Admin (Manage Admin). Admin can manage admin details through the ADD ADMIN process. The process sends admin details to admin data store. Admin keys in admin details. The UPDATE ADMIN process inputs admin details into admin data store. Next, admin keys in admin details which create DELETE ADMIN process. The process sends admin details to admin data store.

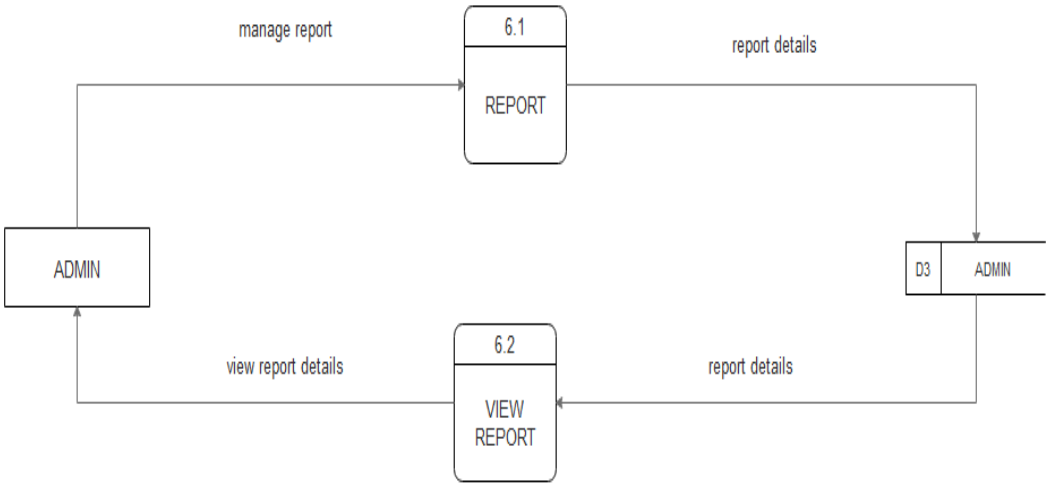

FIGURE 3.9 Data Flow Diagram Level 1 Admin (Report)

Figure 3.9 shows the Data Flow Diagram Level 1 for Admin (Report). Admin need to manage report details that are processed in REPORT process and the process sends report details into admin data store. The admin data store sends back report details to the admin through VIEW REPORT process.

3.4.4 Entity Relationship Diagram

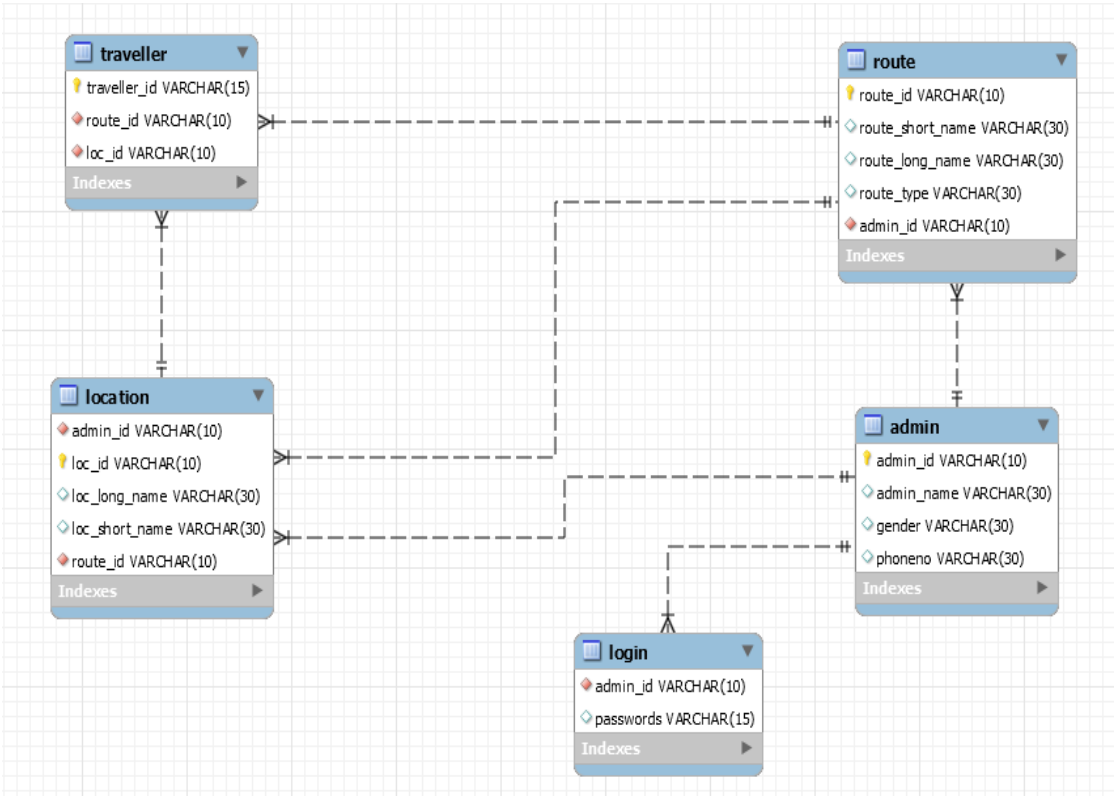

Figure 3.10 Route Suggestion System ERD

The ERD represents the relationship between tables in the database. For this Route Suggestion System (RSS). There are five tables which are admin, route, location, traveler, login. The primary keys are route_id, admin_id, loc_id, traveller_id. Each route has many travelers. Each location has many travelers. Each route has many locations. Each admin has many login.

3.4.5 Database Design

Table Name	Attribute Name	Type	Contents	PK/ FK	Refer ence
admin	admin_id	Varchar (10)	Admin's Id	PK	
	admin_name	Varchar (30)	Admin's name		
	gender	Varchar (30)	Admin's gender		
	phoneno	Varchar (30)	Admin's phone number		
route	route_id	Varchar (10)	Route Id	PK	
	route_short_name	Varchar (30)	Short name of route		
	route_long_name	Varchar (30)	Long name of route		
	route_type	Varchar (30)	Type of route		
	admin_id	Varchar (30)	Admin's Id		FK
location	loc_id	Varchar (10)	Location's Id	PK	
	loc_long_name	Varchar (30)	Long name of location		
	loc_short_name	Varchar (30)	Short name of location		

	route_id	Varchar (30)	Route's Id		FK
login	admin_id	Varchar (10)	Admin's Id		FK
	passwords	Varchar (30)	Passwords		
traveller	traveller_id	Varchar(15)	Traveller's Id	PK	
	route_id	Varchar(10)	Route's Id		FK
	loc_id	Varchar(10)	Location's Id		FK

Table 3.1 Data Dictionary

CHAPTER IV

SYSTEM DESIGN AND IMPLEMENTATION

4.0 Introduction

Testing and implementation is about to construct the system as specific design that has been developed in previous phases, methodically verified to ensure that they are error-free and fully meets user requirements. This phase should be done before a system is fully utilized. This chapter will cover the correct mechanism for testing the system based on white box and black box testing.

4.1 INTERFACE DESIGN

4.1.1 Homepage

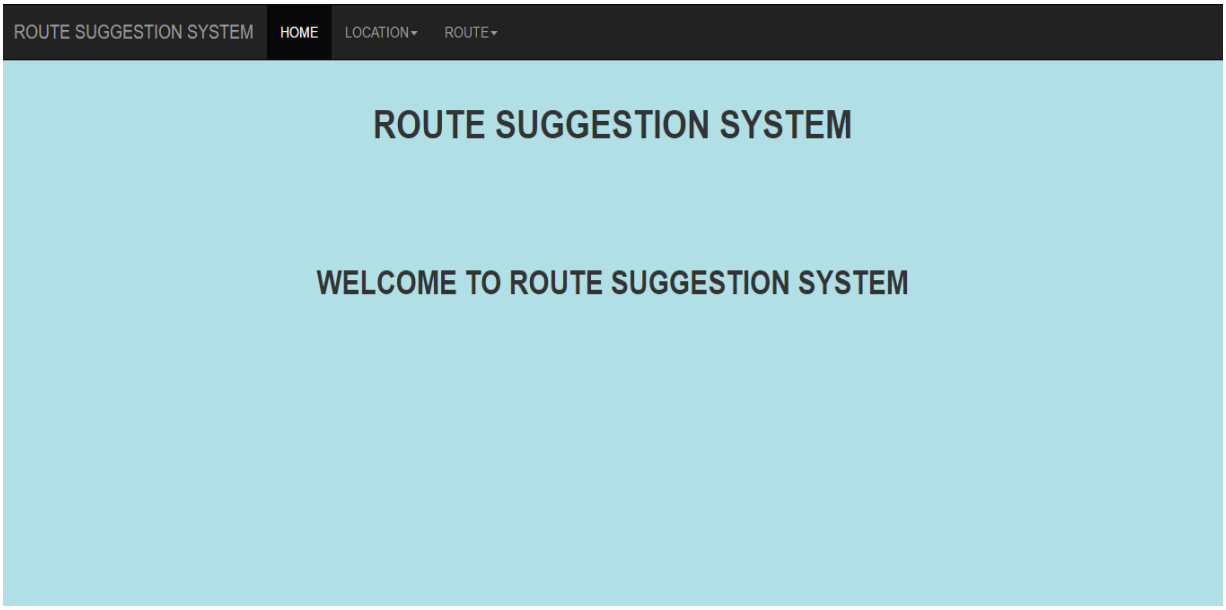

Figure 4.1 Homepage

Figure 4.1 shows the homepage of Route Suggestion System. User can choose to go to another page by click on the drop down button

4.1.2 Origin location

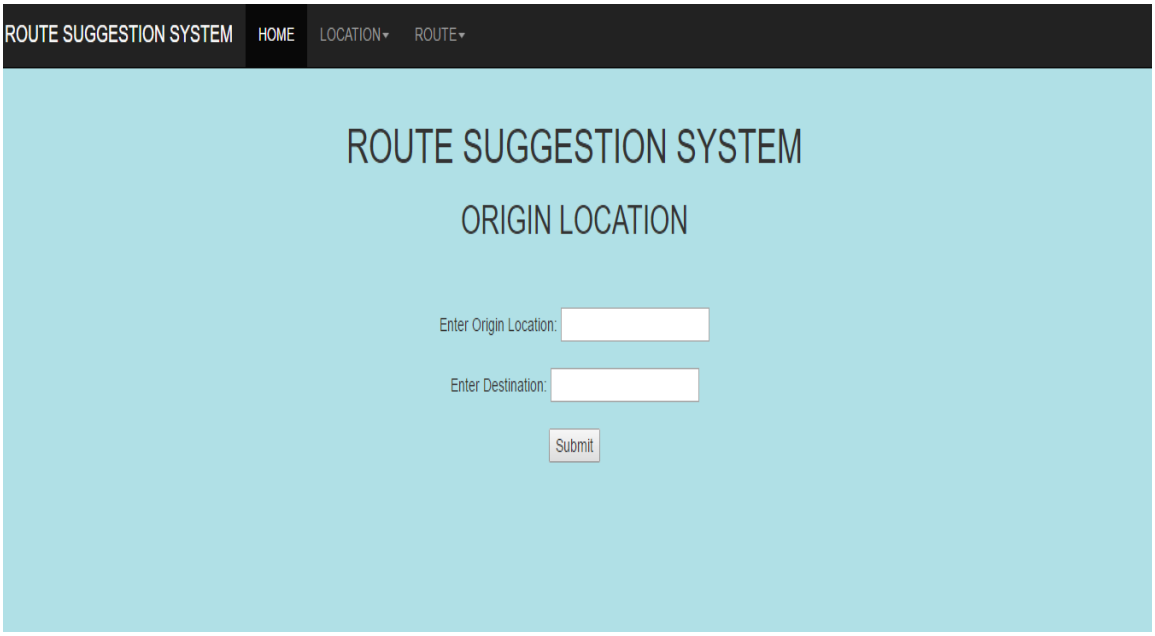

Figure 4.2 Origin location page

Figure 4.2 shows origin location page. On this page user need to fill in the origin location and the destination text box.

4.1.3 Choose Route

Figure 4.3 Choose Route page

Figure 4.3 shows choose route page. On this page, the user just needs to click on the drop down button and will able to choose the route they prefer.

REFERENCES

1) https://pdfs.semanticscholar.org/2c63/17d37edfae7af682910aacedbb6be6a80395.pdf

2) http://ieeexplore.ieee.org/document/6465023/

3) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.5166&rep=rep1&type=pdf

4) http://gauss.cs.ucsb.edu/~aydin/ChapterBFS2015.pdf