TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A REPORT ON

ROUTEFINDER

(MINOR PROJECT)

AUGUST 27, 2013

SUBMITTED BY

BIPIN PANDEY	067BCT511
NAVIN KHADKA	067BCT524
PRAKASH ARYAL	067BCT528
SANJAY RAUT	067BCT536

Acknowledgement

This project is done as a minor project as a part of computer engineering course in Institute of Engineering, Pulchowk Campus. We would like to take this opportunity to express our profound gratitude and deep regards to the Head of Department, Department of electronics and Computer Engineering, Dr. Arun Timalsina for giving us this project. We also thank him for giving this opportunity to explore into the real world project and learn some valuable knowledge which will always be useful in our future.

We also take this opportunity to express a deep sense of gratitude to our teacher Anil Verma for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. Without his guidance and support this project would not be successful.

We heartily thank our friends and seniors for their help to solve various difficulties occurred in due course of development of this project.

Bipin Pandey

Prakash Aryal

Navin Khadka

Sanjay Raut

Abstract

Public transportation system in Kathmandu valley is presently organized in the form of different numbered routes. Generally people do not have a clear idea about the routes and they are in dilemma about which route to follow when they have to visit a new place. There is a need for a proper guiding system that readily suggests the route number of the vehicle that takes the users to their destination.

This report details our attempt at limiting this confusion by providing a system that takes source and destination place as input from user, calculates the proper route and suggests the vehicle number that runs in that route.

List of Tables

List of figures

List of abbreviations

1. OSM : Open Street Map
2. GeoJSON : Geographic JavaScript Object Notation
3. HTML : Hyper Text Markup Language
4. CSS : Cascading Style Sheet
5. API : Application Programming Interface
6. ERD : Entity Relationship Diagram
7. XML : Extensible Markup Language
8. UML : Unified Modeling Language
9. Latlng : Latitude and Longitude

Table of Contents

Chapter 1

Introduction

1.1. Problem statement

Kathmandu valley has one of the most developed public transport system in the country. Roads reach almost every corners of the city and there are a good number of vehicles that take people to their destinations. These public vehicles commuting in a fixed route are assigned a unique route number. The number of new roads is only increasing and this fact is complemented by an increasing number of bus-routes. Now there are very less places, even in the farthest parts of the valley, where there is no public vehicle reach.

This improvement in public transit has greatly facilitated the common people in their daily lives. But this improvement has also introduced a difficulty in the form of complex route-numbering system. It may be feasible for people to remember the routes which they commonly use. But when people have to go to some place new, the problem is more apparent. Also, for many foreign and national tourists entering the valley, there is not an existing route recommendation system. People almost always have to rely on consulting other people to find out what public vehicle they need to take.

As there is no existing system that solves this problem, this project is intended to build a handy android application that attempts to present a good reference.

1.2. Background

There exists a web based vehicle routing system in which the main stations are pre plotted in the map and the users are allowed to check for the routes being confined within those stations.

Unfortunately this system is not so reliable. This system sometimes shows the results that are misleading. It shows the possible paths but does not show the shortest possible paths. If the users follow the instruction they end up wasting a lots of time to reach their destination, where they could have reached in less time through shortest path. Understanding and using this routing guide is a laborious process. And there is also no proper system to add new routes in this system which seems to neglect the fact that transpiration system of Kathmandu valley is ever increasing. This prohibits the easy future expansion of this system to accommodate the increasing new routes of the valley.

1.3. Challenges

The availability of large volume of data, often continually increasing has greatly challenged the ability to accommodate all the routes in valley. If the system is to be extended to a large territory the challenge becomes more intense. As there are no proper open data revealing this system we are forced to input all the data manually which is a tedious and time consuming task. As we also don’t know the complete ideas about routes on every place in the valley we had to consult many people from different areas to add new routes. This problem will be more intense with increasing area coverage so we have decided to use crowd sourcing in future to add new routes. Another big challenges is the incompetence of open street map. The places inside Ring road in Kathmandu are well mapped but we do not find much information on the map as we move farther from the Ring road area. So to locate the stops in these area was a big challenge for us.

1.4 Objectives

The initial objectives of our project is to produce a system that provides the user with the following facilities

1. Provide appropriate vehicles number to take to reach their destination.
2. The vehicle number suggested follows the shortest possible path to reach the destination among the alternatives.
3. The system should be very handy, for this we not only develop a web based application but also develop an android application.
4. Develop a feature to allow the users to add new routes, so that the system can accommodate the increasing transportation network.

As well as these functional requirements we want to produce a user friendly system to allow the user to readily access the information they are entitled to.

1.5. Scope of work

This system is expected to find its application in wide range among the general people who use public transportation to travel between the places. It will be proven a useful tool to find out the proper vehicle to take to reach their desired destination. This system can solve the users’ dilemma and help them to choose correct vehicles shaving their time and effort.

This system is most expected to be used by the tourists visiting the Kathmandu valley. Nowadays tourists are seen to be travelling carrying maps and guide books. They are obliged to ask people about the roads and vehicles and they often have problem due to language. This application can be highly useful for them. As this application can be available as an android application too, it is most likely to be used by the tourists on their mobile phones or other devices.

This system can also be used by the programmers to study the existing systems in vehicle routing system. They can study the drawbacks of our system and consider them to develop a better system in future.

1.6. Outline of report

The report is organized in five chapters:

The first chapter discusses about the subject matter, description of the problem, a brief background and challenges and scope of the project.

The second chapter is about the Planning and management of the project, it discusses how the requirement analysis and documentation was done during the development of the project. It also discusses how we divided the work among the team members and how we divided time for various phases of project development.

Third chapter is the literature review which describes the basic theories used in the development of the system.
It discusses about the data sources for the map, and route information. It discusses how we worked on map in our system. It includes the methods used to calculate the distance in the map and algorithm used to calculate the shortest route which in this case is Dijkstra’s algorithm. The method used to draw a graph is discussed in this section. We further include the main features of the system in this section.

The fourth chapter is about methodology and technical discription of the system. It discusses the various diagrams of the system, tools and techniques used and the method of their implementation. It also discusses the output of the system.

The fifth chapter is the concluding part which shows the limitations of the system and the future enhancements that can be applied in the system to make it more effective.

Chapter 2

Planning and Management

2.1. Requirement Planning

We first focused ourselves on the brainstorming of requirements; analyzing the limitations of the current system and speculating what additional feature may benefit the end user. We studied the maps and algorithms that can be used in the system to make it efficient one. We were aware that the set of requirements were more likely to change as we became familiar with the system.

2.2. Documentation and Management

Given the nature of the project we decided tousle an iterative design process, with ongoing documentation of each iteration detailing the changing specifications as limitations were discovered and the scope fully comprehended. At the beginning of each iteration an initial specifications plan was drafted to provide framework for the direction of development and as a rough measure of progress, which outlined specific features that we aimed to incorporate into our project and to ensure all group members were aware of the intricacies of the current system, its restrictions and what it implied to the project. A short evaluation was produced at the end of each iteration highlighting the main accomplishments, setbacks and limitations and to what extent the initial plan was adhered to. This also served as a basis for the specifications for the next iteration as ideas were adapted given the changing course of development.

2.3. Work division

The work was divided among the team members and time to time discussion was carried out on a regular basis. First the work was divided to work on map and on algorithms and on every discussion the works were merged. Then after the web based system was made then the work for mobile application was done. Documentation was done and updated after every discussion.

2.4. Time management

The time for the project was managed as follows

Action	Time duration
Research and data collection	2 weeks
Prototype design	2 weeks
Coding	3 weeks
Debugging and testing	2 weeks

Table 2.4. Time management table

Chapter 3
Literature Review

3.1. Working with maps

In our system we used Open Street Maps to show users location, stations and the suggested route. To display the map we used leaflet JavaScript library. Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps. We used leaflet library to locate and display the current position of the user. When clicked at each position in the map it provides the latitude and longitude of that position. This latitude and longitude is calculated by latlng function. This function returns the value of latitude and longitude of that place. We wrote a function which displays a pop up when clicked at a point on the map. This pop up allows admin to enter name of that location and bus route number that passes through that route. This function parses the latitude and longitude values of that location and stores them in the database along with name of station and vehicle route number.
Another library GeoJSON is used to locate the point from the database the map .GeoJSON is an open computer file format for encoding collections of simple geographical features along with their non-spatial attributes using JavaScript Object Notation. The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types. It draws an indication on defined latitude and longitude from the database.

3.2. Finding the route

This system is initiated by the user by inputting the source and destination places in the system. Then the system works in the back end to calculate the possible routes between the source and destination and returns the shortest one from the list of possible routes. To find out the shortest path first all the possible paths are calculated, the process is described below.

3.2.1 Calculation of all possible routes

To find all the possible routes between the entered source and destination first of all the nodes(stations) between the source and the destinations are determined. Each nodes are pre assigned with the bus route number which are stored in the data base. The vehicle number of all the nodes are compared, if all the nodes ends up with the same vehicle number then that route is a possible route. Else the node are traversed till the same vehicle number continues and when the route num changes the new route number is traced to reach the destination. In this way all the possible routes is checked and a list of possible routes is returned to determine the shortest route among them.

3.2.2 Calculation of shortest route.

Then for each possible route the distance between the consecutive nodes is calculated and plotted in a graph. Then we use the Dijkstra’s algorithm to calculate the shortest path among the alternatives.

3.2.2.1 Graph

A drawing of a graph or network diagram is a pictorial representation of the vertices and edges of a graph. The vertices are points in the plane and are connected by edges whose weights are distances. Think of the vertices as cities and the edges as roads connected to them. Then we applied Dijkstra’s algorithm in the graph to find the shortest path.

3.2.2.2 Dijkstra’s Algorithm

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph search algorithm that solves the single source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices 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.
In our system Dijkstra’s algorithm is used to find the shortest route among the possible routes.

3.2.3 Calculation of distance between two nodes

The latitude and longitude values parsed in the database are used in two cases:

a) To point the location on a map.
b) To calculate the distance between two places.

For the calculation of distance a function named dist is defined which takes the latitudes and longitudes of the source and destination as input parameter and returns distance in kilometers.

3.3. Platform Development

Route Finder is mobile app cum web based application working on cross-platforms such as Android phones, iPhones and Windows Phones. To make this application compatible on mobile devices we have used Phonegap. Phonegap is a web-based mobile development framework, based on the open-source Cordova project. Phonegap allows you to use standard web technologies such as HTML5, CSS3 and JavaScript for cross-platform development, avoiding each mobile platforms' native development language. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's sensors, data and network status.

3.4. Data source

The different source of data required for our system are as follows:

1. the information about the routes were obtained from nepal2day.com
2. The information about the latitude and longitude of a place were obtained from Open Street Map.
3. The results were also displayed using open street maps.

3.5. Features of program

The final system has the following features:

1. It is provided with both web based and mobile application.
2. The system can be divided into two parts: administrative control part and user interface part
    2.1.1. At the user interface:

2.1.2. The users are allowed to input source and destination which initiates the system.

2.1.3. The program executes at back end and suggests the route number to be followed to reach destination.

2.1.4. The vehicle to be taken is suggested in a popup message.

2.1.5. The route is shown in the map.

2.1.6. If the user does not finds the route then he/she is provided with the facility to report the missing route which will later be added by the admin to the system.

2.2. At the administrative control:

2.2.1. The administrator can add new routes as follows:

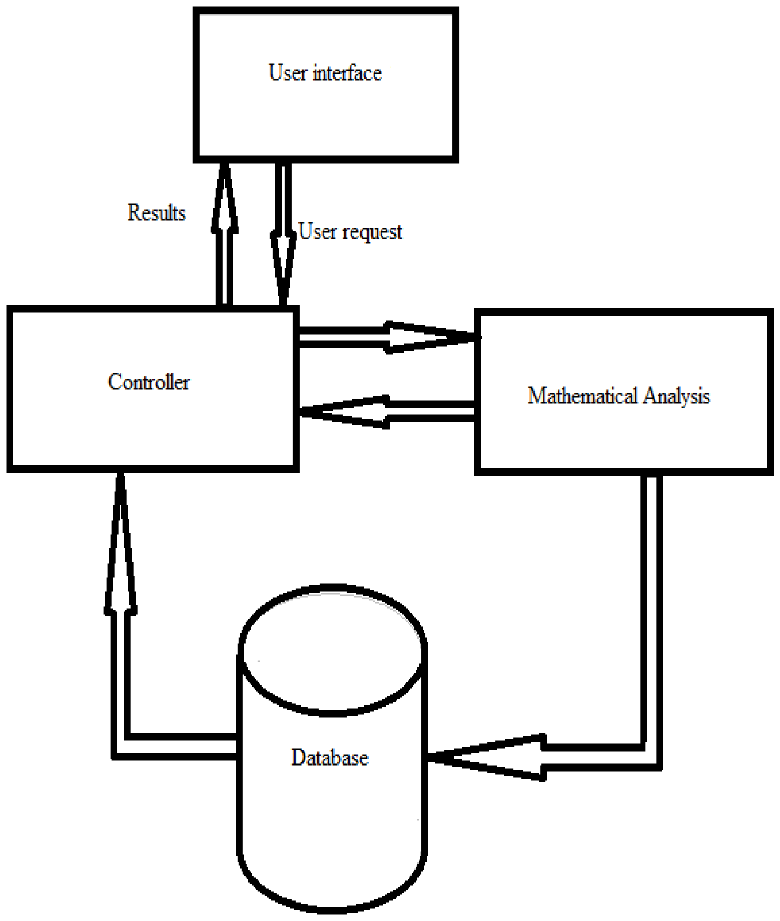
The stops can be defined by clicking on the map which display a pop up allowing user to enter name of the place and route number desired to be added. The stops with common route number sums up to make a new route.

Chapter 4

Methodology

4.1. Diagram

4.1.1.System Flow Diagram



System Flow Diagram shows how the processes occur in a functional system. In this system there are four main components:

1. User interface

2. Controller
3. Mathematical calculations
4. Database

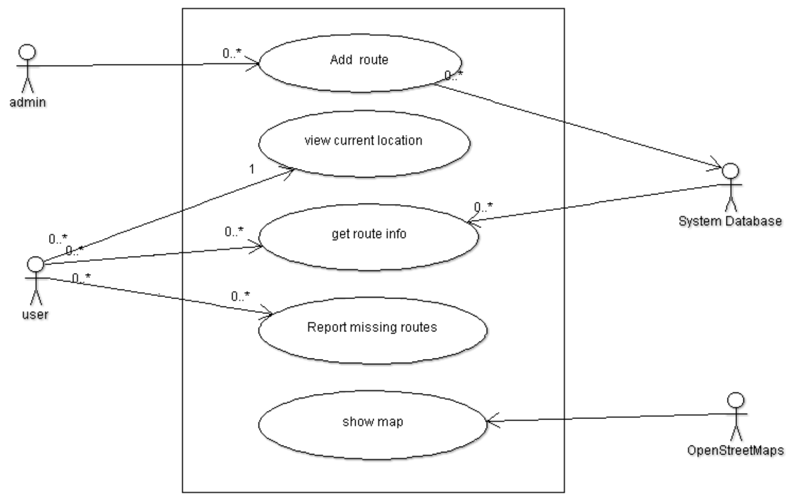
User interface is the part through which the user interacts to the system, The user can input source and destination and view maps and description of routes through user interface.

The request of user by inputting the source and destination is forwaded to mathematical calculation by controller. And the result is displayed on the screen by this portion.

The mathematical calculation part consists of functions to calculate possible routes, distance between stations and shortest path. They used datas from database. Admin uses this part to calculate and store datas in database. This part is used to calculate distance between stations and draw graph to implement dijkstra’s algorithm.

Database is the important part of every system. Ours system has database that keeps records of stations, their name, their latitude and longitudional positions, distance of nearest station from that station, route of bus that run in through the system. The mathematical components continuously interact with database for its calculations.

4.1.2.Use case Diagram



The use case diagram shows that the system has five main functions:

1. Add route
2. View current location
3. Get route info
4. Report missing routes
5. Show map

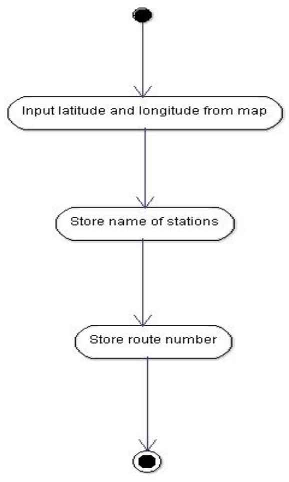
Only Admin is provided with the right to add new route. This is done by adding stations in a route and providing same vehicle number whose route is being added. The user can view his current location, get route information and report missing route to the admin if the route of his interest is not available in the system.
The map is extracted from openstreetmap and displayed.
When new route is added it is saved in the systems database which is later used to calculate the shortest route. The route info is also accessed from database and displayed to user.

4.1.3. Activity Diagram

There are two main actors on the system:

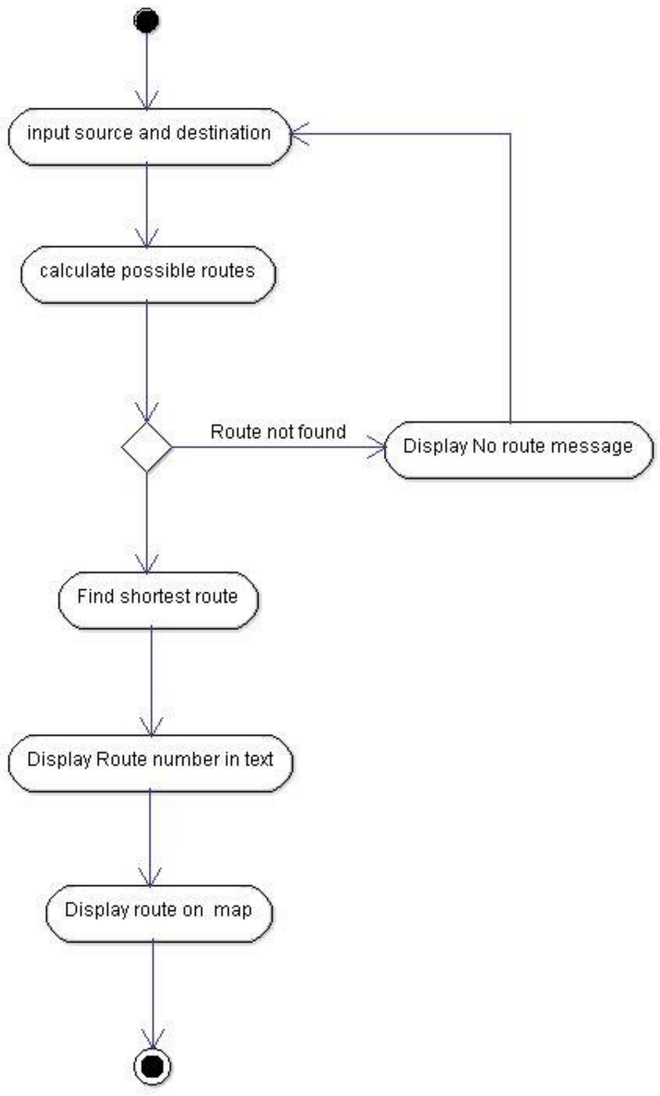
1. Administrator
2. User

The activity diagram for administrator is as follows



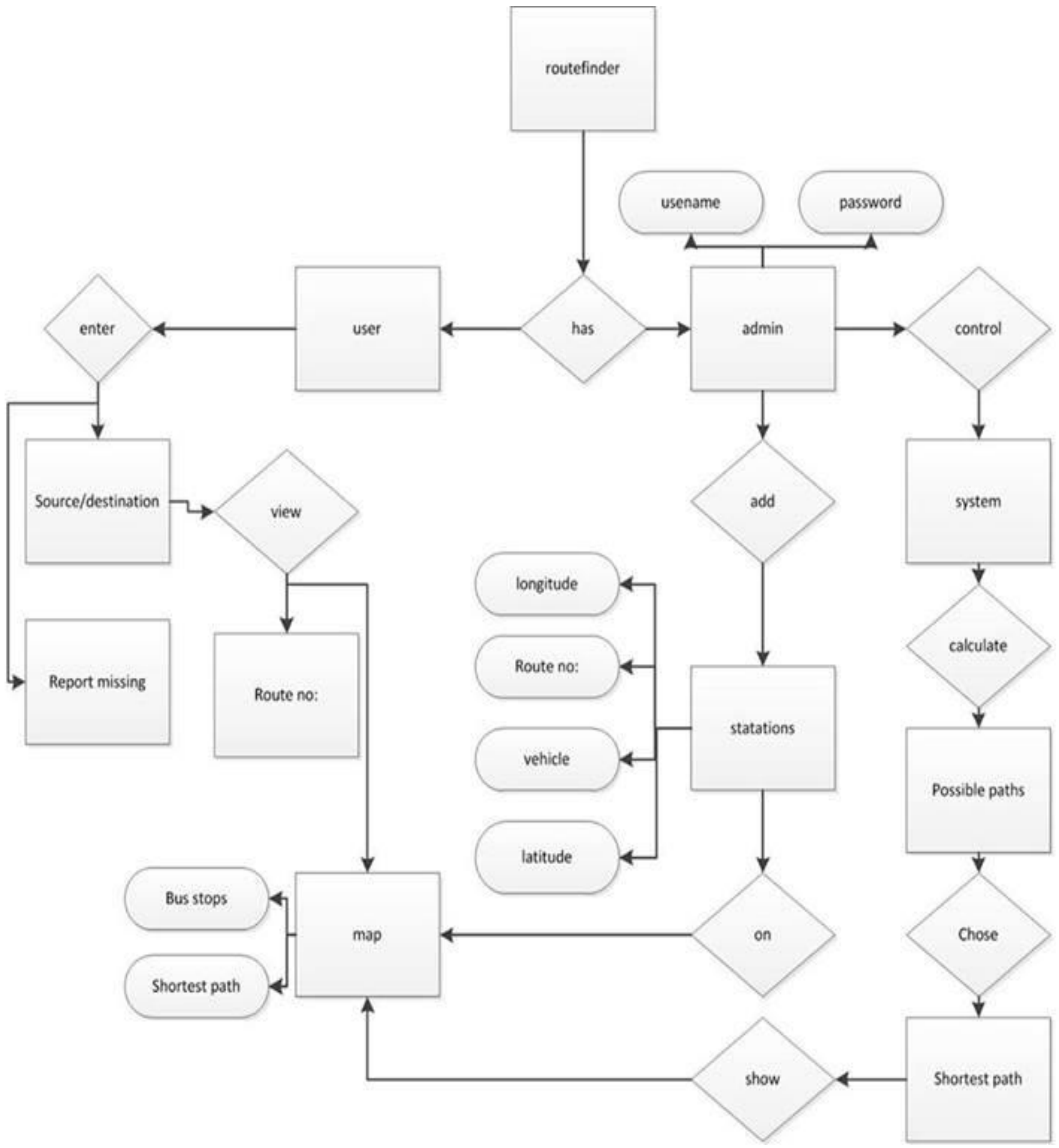
The administrator while adding the new route extracts the latitude and longitude of the station from the map. Then he adds the name of stations and vehicle number through the station in the route.

The activity diagram for the user is as follows:


input source and destination
calculate possible routes
Route not found
Display No route message
Find shortest route
Display Route number in text
Display route on map

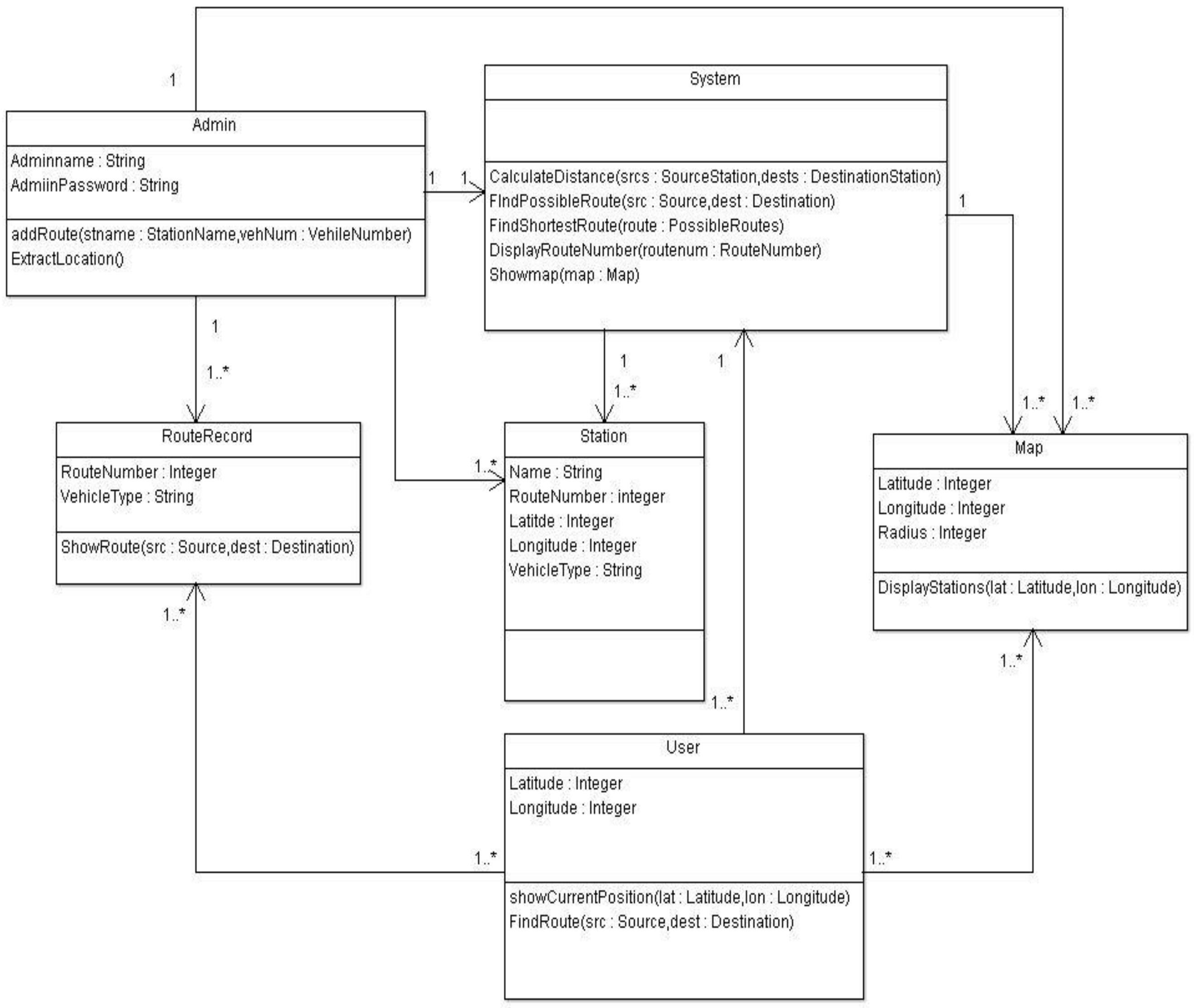
The user enters the source and destination. The system calculates possible routes, if no routes are found then it displays no routes found message. If route is found then it calculates shortest path and display route number and route in map.

4.1.4. Entity Relationship Diagram



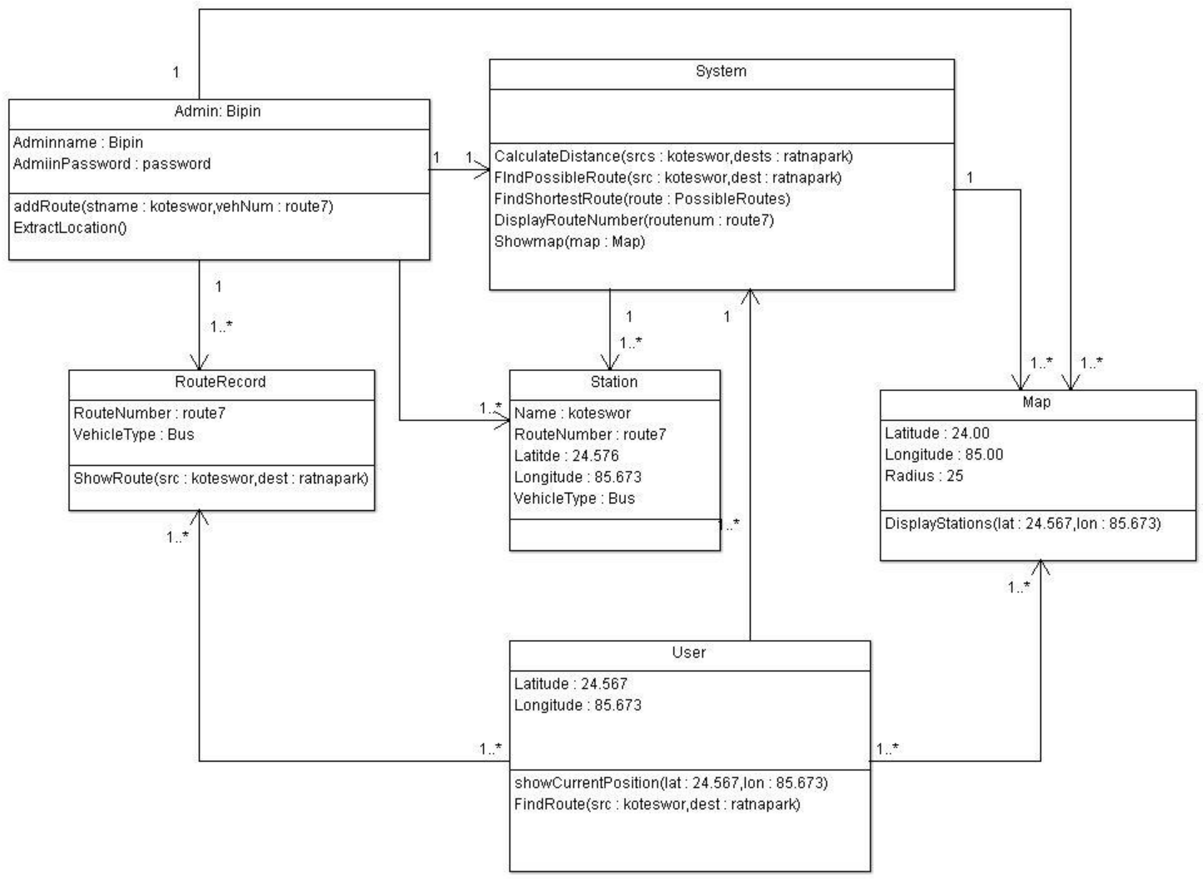
Entity–relationship model (ER model) is a data model for describing a database in an abstract way.

4.1.5. Class Diagram


System
CalculateDistance(srcs : SourceStation,dests : DestinationStation)
FIndPossibleRoute(src : Source,dest : Destination)
FindShortestRoute(route : PossibleRoutes)
DisplayRouteNumber(routenum : RouteNumber)
Showmap(map : Map)
Admin
Adminname : String
AdmiinPassword : String
addRoute(stname : StationName,vehNum : VehileNumber)
ExtractLocation()
RouteRecord
RouteNumber : Integer
VehicleType : String
ShowRoute(src : Source,dest : Destination)
Station
Name : String
RouteNumber : integer
Latitde : Integer
Longitude : Integer
VehicleType : String
Map
Latitude : Integer
Longitude : Integer
Radius : Integer
DisplayStations(lat : Latitude,lon : Longitude)
User
Latitude : Integer
Longitude : Integer
showCurrentPosition(lat : Latitude,lon : Longitude)
FindRoute(src : Source,dest : Destination)
1
1..*

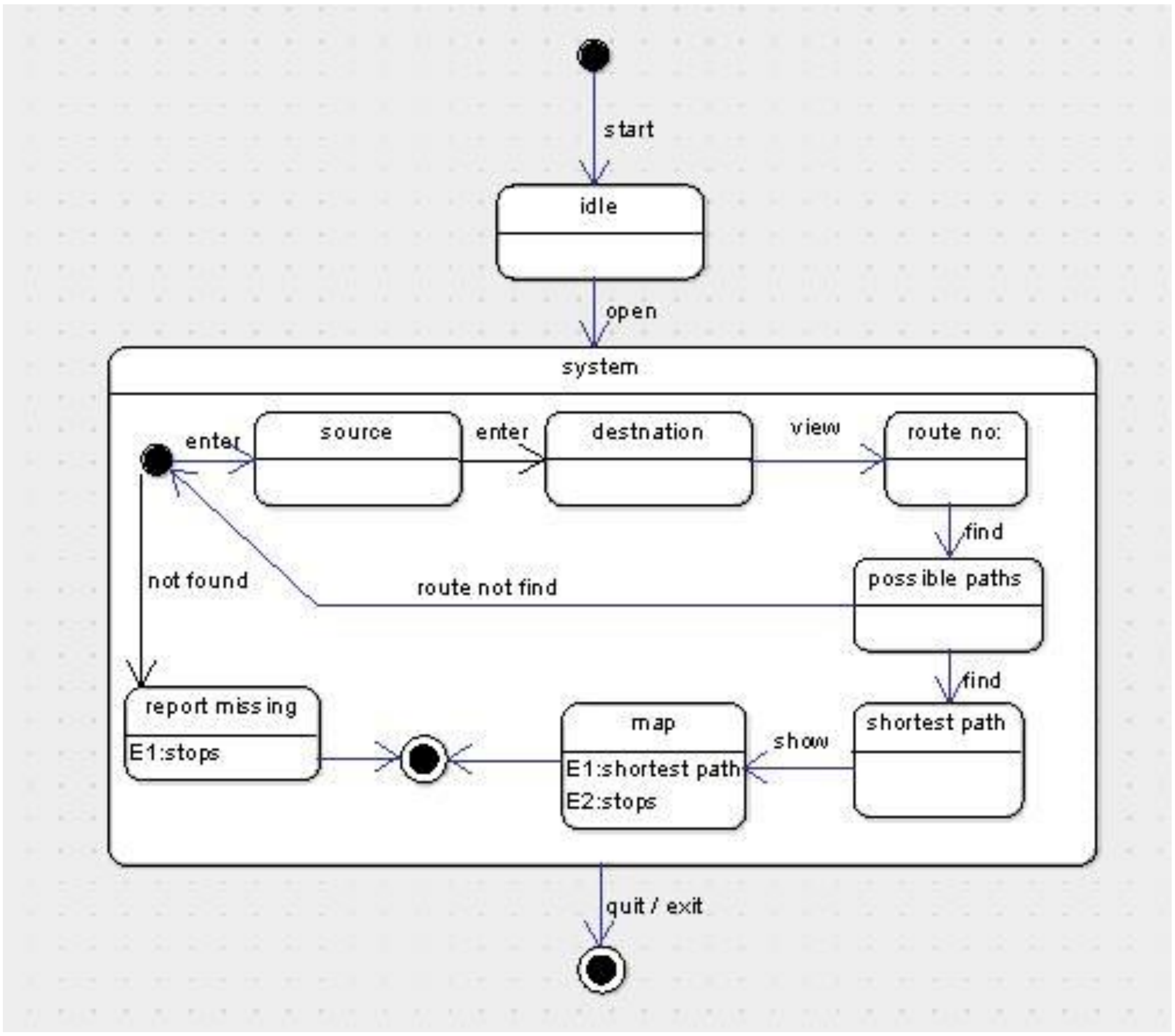
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. the above class diagram has six classes and it shows their attributes and relationships among themselves.

4.1.6. Object Diagram


Admin: Bipin
Adminname : Bipin
AdmiinPassword : password
addRoute(stname : koteswor,vehNum : route7)
ExtractLocation()
System
CalculateDistance(srcs : koteswor,dests : ratnapark)
FIndPossibleRoute(src : koteswor,dest : ratnapark)
FindShortestRoute(route : PossibleRoutes)
DisplayRouteNumber(routenum : route7)
Showmap(map : Map)
RouteRecord
RouteNumber : route7
VehicleType : Bus
ShowRoute(src : koteswor,dest : ratnapark)
Station
Name : koteswor
RouteNumber : route7
Latitde : 24.576
Longitude : 85.673
VehicleType : Bus
Map
Latitude : 24.00
Longitude : 85.00
Radius : 25
DisplayStations(lat : 24.567,lon : 85.673)
User
Latitude : 24.567
Longitude : 85.673
showCurrentPosition(lat : 24.567,lon : 85.673)
FindRoute(src : koteswor,dest : ratnapark)
1
1..*

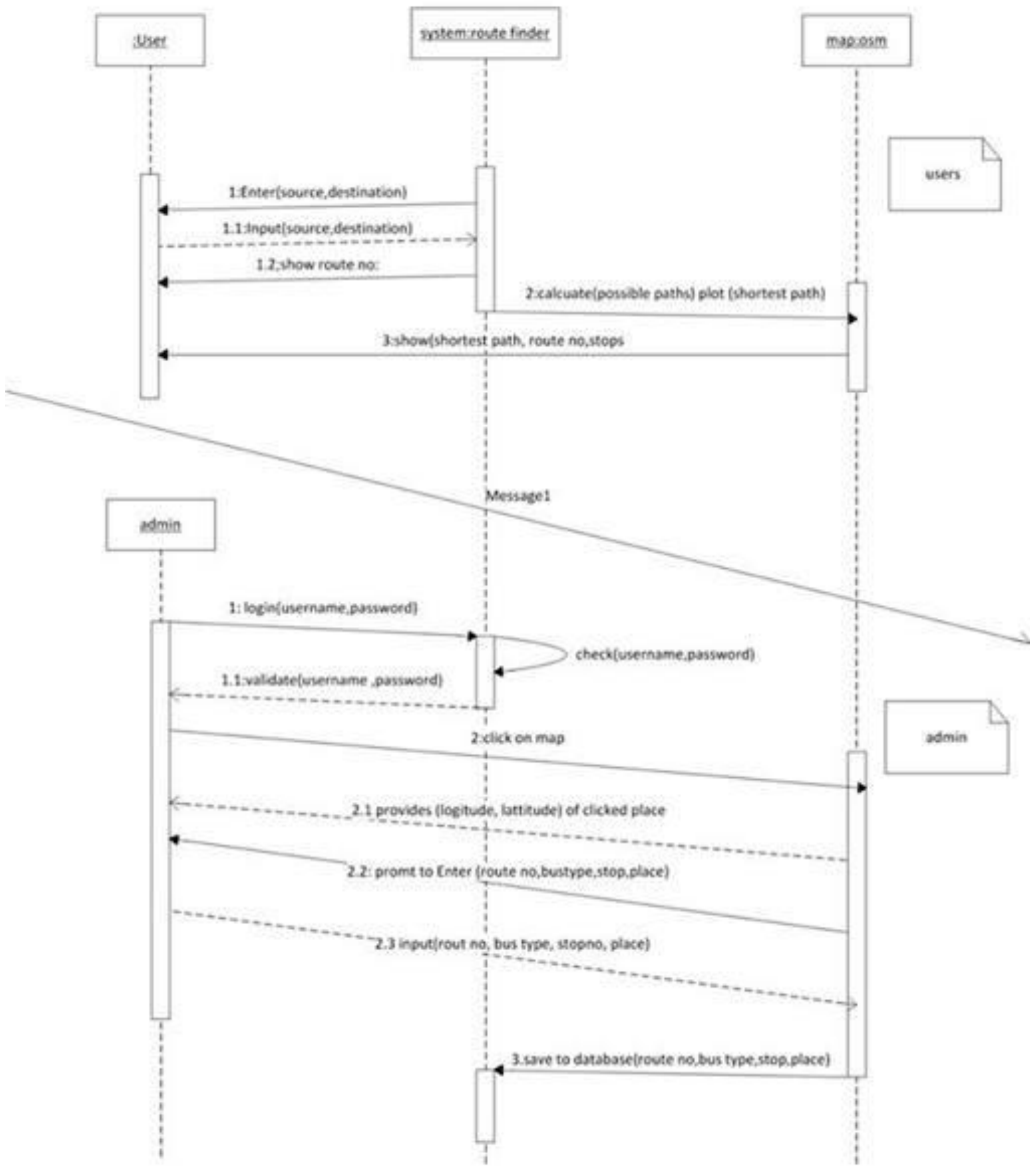
An object diagram in the Unified Modeling Language (UML), is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time. It focuses on some particular set of objects and attributes, and the links between these instances. The above object diagram shows the working on a real data at a instance

4.1.7.State Chart Diagram



A state chart diagram describes different states of a component in a system. The above state chart describes the states of the system.

4.1.8. Sequence Diagram


:User
system:route finder
map:osm
users
1:Enter(source,destination)
1.1:Input(source,destination)
1.2;show route no:
2:calcuate(possible paths) plot (shortest path)
3:show(shortest path, route no,stops
Message1
admin
1: login(username,password)
check(username,password)
1.1:validate(username ,password)
admin
2:click on map
2.1 provides (logitude, lattitude) of clicked place
2.2: promt to Enter (route no,bustype,stop,place)
2.3 input(rout no, bus type, stopno, place)
3.save to database(route no,bus type,stop,place)

4.2. Technical Description

Route Finder is a mobile device cum web based platform to find the public transportation route inside the Kathmandu Valley. Being built with latest web technologies such as HTML, CSS and JavaScript, it has provided flexibility to make the app compatible to mobile devices. Describing the technical details, the application consists of two aspects: Administrative and User interactive.

In the administrative part, different public vehicle transportation routes around Kathmandu Valley can be added. For this we have used JavaScript functions to extract the position of the vehicle stops on clicking on the Open Street Map. The administrator is then asked to submit the information about the stops and vehicle route to MySQL database. We have used PHP to interact with MySQL database. When a route is completed, the administrator can now convert the MySQL database to Geo JSON data which is used to display the stops on the map. Open Street Map is implemented with the help of Leaflet JavaScript Library.
Meanwhile in the user interactive section the user will be able to view his current location in the OSM. User can input the source and destination in the dialogue box appearing on the application. We have used JavaScript to display the route suggestions of the vehicle he needs to follow after calculating in the back end. The results are then displayed on the map with points representing the stops, the user needs to follow to reach the destination. The required points are extracted from Geo JSON data and traced on the map. To calculate the distance between two points we implemented have sine Formula in JavaScript which takes the location of the points in latitude and longitude and returns the distance between the points.
For calculating the possible route, at first the vehicle routes are scanned for source and destination. If the source and destination lie on same route, then the same vehicle route is suggested to the user. If not, the intersecting node or the common vehicle stops for source and destination routes are determined. If there is only one common point, then the user is suggested to change the vehicle from that point. If multiple points are found then the shortest path is to be calculated. The distance between consecutive nodes is calculated till the destination and total dictation is determined adding them. The user is then suggested with the shortest calculated route to follow to destination.

For cross-platform development of application, we have used Phonegap. As our application is first built for web based framework, a configuration XML file is written to describe the application for respective mobile frameworks. API bindings are referenced to use the device features such as connection and geolocation.

4.3. Tools and Techniques

Name of Tools/Technology	Type/Purpose
Javascript	Programming language
WEB STANDARDS	Presentation over Browser
Openstreetmap	Map
Js Objects and Arrays	Database

Table 4.3 Tools and Techniques used table

4.4. Result and discussion

We have been succeeded in developing a system, both web based and mobile application which helps the users to find the correct routes to reach the place of their desire. Our system allows the user to input the source and destination. Then the system calculates the shortest route and the vehicle number that should be taken to reach the destination and displays the output in text and map. We are very satisfied with the outputs generated by our system. This system is simple, easy to use, handy and expected to be used widely as a mobile application for this system is also developed.

Chapter 5

Conclusion and Future enhancements

5.1. Conclusion

We feel that the system developed has been able to address its problem statement. The complex transport network of Kathmandu valley and the confusion it has created to the general public is the problem. This system solves this problem by suggesting the proper route to follow to travel between the places. This makes travelling within the valley via public transport system easier. Though there are the rooms for improvement we are satisfied with the level of system we have developed. This system will be further enhanced to make it effective and user friendly.

5.2. Limitations

The output we have achieved has limitations which can be corrected to enhance the system functionality in future. These limitations includes.

1. The route is displayed by highlighting the point stations instead of road.
2. The user cannot add the new route, only the admin can add it, so it is a overload job for admin to manage all the routes.
3. The system is limited only for the Kathmandu valley.

5.3. Future enhancements

Though we have developed a software that generates satisfactory output, there are lots of rooms of improvement that can be considered in future to make this system better and reliable. The future enhancement that can be applied to this system are as follows:

1. Along with the route number the information regarding the bus fare can be displayed.
2. It can be further enhanced into voice based dynamic routing system which can be used in private vehicles to show directions.
3. By collaborating with traffic control system we can display the information about traffic jams in the map so that the users get predefined about the jams and choose alternative ways to reach their destinations.

References/Bibliography

1. www.openstreetmaps.org
2. www.geojson.org
3. www.phonegap.com
4. www.leafletjs.com
5. www.cloudmade.com