# 1. INTRODUCTION

**1.1 ABOUT PROJECT:**

Location-based services (LBS) enable mobile users to query points-of-interest (e.g., restaurants, cafes) on various features (e.g., price, quality, variety). In addition, users require accurate query results with up-to-date travel times. Lacking the monitoring infrastructure for road traffic, the LBS may obtain live travel times of routes from online route APIs in order to offer accurate results. Our goal is to reduce the number of requests issued by the LBS significantly while preserving accurate query results. First, we propose to exploit recent routes requested from route APIs to answer queries accurately. Then, we design effective lower/upper bounding techniques and ordering techniques to process queries efficiently. Also, we study parallel route requests to further reduce the query response time. Our experimental evaluation shows that our solution is three times more efficient than a competitor, and yet achieves high result accuracy (above 98 percent).

**1.2 OBJECTIVE:**

The project Route-Saver: Leveraging Route APIs for Accurate and Efficient Query Processing at Location-Based Services is a real-time application in the domain of JAVA. This project is used to hide and also transfer the files to the users.

**1.3 SCOPE:**

We propose an application where confidentiality and privacy is entirely held by the clients so guaranteeing them privacy. We designed the application such a way that the storage provider/database only provides simple data access and does not contain any other information related to the file sharing. In this project, we present a secure file sharing application for both LAN and in web.

# 2. LITERATURE SURVEY

The availability of GPS-equipped Smartphone leads to a huge demand of location-based services (LBSs), like city guides, restaurant rating, and shop recommendation websites,

e.g.,OpenTable, Hotels, UrbanSpoon.1 They manage points-of-interest (POIs) specific to their applications, and enable mobile users to query for POIs that match with their preferences and time constraints. As an example, consider a restaurant rating website that manages a data set of restaurants P (see Fig. 1a) with various attributes like: location, food type, quality, price, etc. Via the LBS (website), a mobile user q could query restaurants based on these attributes as well as travel times on road network to reach them. Here are examples for a range query and a KNN query, based on travel times on road network. Successful LBS must fulfill two essential requirements: (R1) accurate query results, and (R2) reasonable response time. Query results with inaccurate travel times may disrupt the users’ schedules, cause their dissatisfaction, and eventually risk the LBS losing its users and advertisement revenues. Similarly, high response time may drive users away from the LBS. Observe that the live travel times from user q to POIs vary dynamically due to road traffic and factors like rush hours, congestions, road accidents. As a case study, we used Google Maps to measure the live travel times for three pairs of locations in Brisbane, Singapore, and Tokyo, on two days. Even on the same weekday (Wednesday), the travel times exhibit different trends. Thus, historical traffic data may not provide accurate estimates of live travel times. Unfortunately, if the LBS estimates travel times based on only local information (distances of POIs from user q), then query results (for range and KNN) would have low accuracy (50 percent for NoAPI, see Fig. 7). Typical LBS lacks the infrastructure and resources (e.g., road-side sensors, cameras) for monitoring road traffic and computing live travel times. To meet the accuracy requirement (R1), the framework SMashQ is proposed for the LBS to answer KNN queries accurately by retrieving live travel times (and routes) from online route APIs (e.g., Google Directions API, Bing Maps API), which have live traffic information. Given a query q, the LBS first filter POIs by local attributes in P. Next, the LBS calls a route API to Obtain the routes (and live travel times) from q to each remaining POI, and then determines accurate query results for the user. As a remark, online maps (e.g., Google Maps, Bing Maps), on the other hand, cannot process queries for the LBS either, because those queries may involve specific attributes (e.g., quality, price, facility) that are only maintained by the LBS. Using online route APIs raises challenges for the LBS in meeting the response time requirement (R2). It is important for LBS to reduce the number of route requests for answering queries because a route request incurs considerable time (0.1-0.3 s) which is high compared to CPU time at LBS. SMashQ obtains the latest travel times for queries from online route API. Though it guarantees accurate query results, it may still incur a considerable number of route requests.

**2.1EXISTING SYSTEM:**

* To meet the accuracy requirement, the framework SMashQ is used for the LBS to answer KNN queries accurately by retrieving live travel times (and routes) from online route APIs (e.g., Google Directions API, Bing Maps API, which have live traffic information.
* Indexing on road networks have been extensively studied in the literature. Various shortest path indices have been developed to support shortest path search efficiently.
* Papadias et al. study how to process range queries and KNN queries over points-of-interest, with respect to shortest path distances on a road network.
* Thomsen et al. study the caching of shortest paths obtained from online route APIs. They exploit the optimal subpath property on cached paths to answer shortest path queries.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Query results with inaccurate travel times may disrupt the users’ schedules, cause their dissatisfaction, and eventually risk the LBS losing its users and advertisement revenues.
* Similarly, high response time may drive users away from the LBS.
* As a remark, online maps (e.g., Google Maps, Bing Maps), on the other hand, cannot process queries for the LBS either, because those queries may involve specific attributes (e.g., quality, price, facility) that are only maintained by the LBS.
* SMashQ does not utilize route log to derive exact travel times nor lower/upper bounds to boost the query performance of the LBS.

**2.2PROPOSED SYSTEM:**

* In this paper, we exploit an observation namely that travel times change smoothly within a short duration. Routes recently obtained from online route APIs may still provide accurate travel times to answer current queries. This property enables us to design a more efficient solution for processing range and KNN queries.
* Specifically, our method Route-Saver keeps at the LBS the routes which were obtained in the past d minutes (from an online route API), where d is the expiry time parameter. These recent routes are then utilized to derive lower/upper bounding travel times to reduce the number of route requests for answering range and KNN queries.
* To reduce the number of route requests while providing accurate results, we combine information across multiple routes in the log to derive tight lower/upper bounding travel times. We also propose effective techniques to compute such bounds efficiently. Moreover, we examine the effect of different orderings for issuing route requests on saving route requests. And we study how to parallelize route requests in order to reduce the query response time further.

**ADVANTAGES OF PROPOSED SYSTEM:**

* Our experiments show that our solution is three times more efficient than SMashQ, and yet achieves high result accuracy (above 98 percent).
* Combine information across multiple routes in the log to derive lower/upper bounding travel times, which support efficient and accurate range and KNN search.
* Develop heuristics to parallelize route requests for reducing the query response time further.
* Evaluate our solutions on a real route API and also on a simulated route API for scalability tests.

# 3.REQUIREMENTS SPECIFICATION

1. **1.FUNCTIONAL REQUIREMENTS (Modules):**

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

* + - The Admin has to login by using valid user name and password.After login successful he can do some operations such add contents, view all content details, list of all search history, List All User and documents ranking for both query level and document level search, List all documents users, Auto recommend the documents based on the other user recommendations, Measure the Expectation loss if the content is not matched and logout.
    - Add the document, If the admin want add the new document, then he will enter document name, enter a document title, domain, sub domain, browse the document then submit and that data will stored in data base.
    - The admin can view the all registered user and also view the list of all users and document.
    - The admin can view the document details i.e, document name, document title, domain, sub domain, file name, document content, related images.
    - Admin can view list of all users. Here all register users are stored with the details such as user name, DOB, e-mail, mobile, location and user images.
    - Admin can view all search comparisons. Here all users search history are stored with the details such as user name, document ID, document name, document title, domain, sub domain, date and time and view details.
    - The ranking details of each document will be displayed such as document rank, document name, document title, domain, and sub domain.
    - Auto recommended document detail contains no of times document used, domain name, document title, domain, and sub domain, user name and view document.
    - The android User should register before processing operations with web servers. After registration, he has to login by using authorized user name and password.
    - Login successful he will do some operations like Query Search on doc titles, Query Search on domain, sub domain, Query search based on Top k Query and scanning type of document and contents to check whether the document contains mailware.
    - If documents are malware related then those documents will be scanned and never takes to view in the android mobile.
    - Expectation measurement details will be displayed, i.e, user name, matched documents, expected results, expectation loss, date and time.

**3.1.2. Functionality:**

**User Registration:**

**Purpose:** This project is reduce the number of route requests while providing accurate results, we combine information across multiple routes in the log to derive tight lower/upper bounding travel times.

**Inputs:** The Admin has to login by using valid user name and password; The android User should register before processing operations with web servers. After registration, he has to login by using authorized user name and password.

**Processing**  users search history are stored with the details such as user name, document ID, document name, document title, domain, sub domain, date and time and view details.

**Output:** After the registration the user will be directed to the main home.

**3.2.NON FUNCTIONAL REQUIREMENTS:**

Non – Functional requirements, as the name suggests, are those requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability response time and store occupancy. Alternatively, they may define constraints on the system such as the capability of the Input Output devices and the data representations used in system interfaces. Many non-functional requirements relate to the system as whole rather than to individual system features. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention.

**The key non-functional requirements are:**

* + - * Security: The system should allow a secured communication between server, Admin and users.
      * Energy Efficiency: The Energy consumed by the Users to receive the File information from the server and admin.
      * Reliability: The system should be reliable and must not degrade the performance of the existing system and should not lead to the hanging of the system.

**3.4 Feasibility study:**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

**Operational Feasibility**

Question that going to be asked are

* + - Will the system be used if it developed and implemented.
    - If there was sufficient support for the project from the management and from the users.  Have the users been involved in planning and development of the Project.
    - Will the system produce poorer result in any respect or area?

This system can be implemented in the organization because there is adequate support from management and users. Being developed in Java so that the necessary operations are carried out automatically.

**Technical feasibility**

* + - Does the necessary technology exist to do what is been suggested
    - Does the proposed equipment have the technical capacity for using the new system?
    - Are there technical guarantees of accuracy, reliability and data security?
    - The project is developed on Pentium III with 128 MB RAM.
    - The environment required in the development of system is any windows platform.
    - The observer pattern along with factory pattern will update the results eventually.
    - The language used in the development is JAVA J2SDK1.4.0, Servlets, JDBC, Tomcat Server and database as Oracle 8i.

**Financial and Economical Feasibility**

The system developed and installed will be good benefit to the organization. The system will be developed and operated in the existing hardware and software infrastructure. So there is no need of additional hardware and software for the system. The System developed can reduce the cost overheads of the organization in providing the internet services for establishing the communication channel between the organization staff in conducting the organizational works in smoother and time effective manner.

**3.3.SOFTWARE & HARDWARE REQUIREMENTS:**

**Hardware Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| • | System |  | : Pentium IV 3.4 GHz. |
| • | Hard Disk |  | : 40 GB. |
| • | Monitor |  | : 14’ Colour Monitor. |
| • | Mouse |  | : Optical Mouse. |
| • | Ram |  | : 1 GB. |

**Software Requirements:**

* Operating system : Windows Family.
* Coding Language : J2EE (JSP,Servlet,Java Bean) • Data Base : MY Sql Server.
* IDE : Eclipse Juno
* Web Server : Tomcat 6.0

# 4.TECHNOLOGY USED

4.1 **Technologies :**

Technology : java

Web Technologies : HTML,Servlets,JavaScript

**SOFTWARE OVERVIEW:**

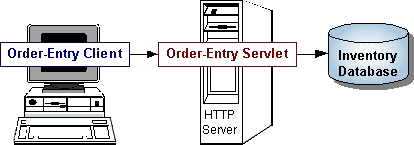
**Tomcat Server:**

Apache tomcat,often refered to as Tomcat ,is a open source web server developed by the Apache Software Foundation(ASF).Tomcat implements several java EE specifications including Java Servlet

,Java Server Pages(JSP),Java EL, and Web Socket ,and provides a “pure java” HTTP web server environment for Java code to run in. Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation ,released under the Apache License 2.0 license and is open source software.tomcat server running on: [http://localhost:8000/.](http://localhost:8000/)

**Servlets:**

Servlets are modules that extend request/response-oriented servers, such as Javaenabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.



## Fig no. 4.1 servlets overview

Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface.

Servlets can be embedded in many different servers because the servlet API, which you use to write servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the Servlet API.

**The ServletRequest Interface:**

The ServletRequest interface allows the servlet access to:

* Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it.
* The input stream, ServletInputStream. Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and PUT methods.

Interfaces that extend ServletRequest interface allow the servlet to retrieve more protocol-specific data. For example, the HttpServletRequest interface contains methods for accessing HTTPspecific header information.

**The ServletResponse Interface:**

The ServletResponse interface gives the servlet methods for replying to the client. It:

* Allows the servlet to set the content length and MIME type of the reply.
* Provides an output stream, ServletOutputStream, and a Writer through which the servlet can send the reply data.

Interfaces that extend the ServletResponse interface give the servlet more protocol-specific capabilities. For example, the HttpServletResponse interface contains methods that allow the servlet to manipulate HTTP-specific header information

**About Html:**

HTML (hyper text markup language) is a language used to create hypertext documents that have hyper links embedded in them. It consists of tags embedded in the text of a document with HTML. We can build web pages or web document s. it is basically a formatting language and not a programming language. The browser reading the document interprets mark up tags to help format the document for subsequent display to a reader. HTML is a language for describing structured documents. HTML is a platform independent. WWW (world wide web) pages are written using HTML. HTML tags control in part the representation of the WWW page when view with web browser. The browser interprets HTML tags in the web document and displays it. Different browsers show data differently. Examples of browsers used to be web pages include:

* Netscape
* Internet Explorer

<html>

<head><title>Welcome To Ezee Mail System</title></head>

<body bgcolor=#40004>

<pre>

<center><h1><font color=white size=+3 face=mscomic><img src="wel1.gif"></font></center></h1>

<center><h1><font color=white size=+3 face=mscomic>Ezee Mail System</font></center></h1> <center><img src="email.gif" border=0></center>

<center><h1><u><a href="loginsc.html"><font color=yellow>-------Enter into the WebPage-------

</h1></a></center></font>

</pre>

</body>

</html>

**Java And Internet:**

Java is strongly associated with Internet and known as Internet programming language. Internet users can use java to create applet programs and run them locally using java enabled browser search as hot java. Applets can be downloaded from remote machine via Internet and run it on local machine.

**Java And World Wide Web:**

World wide web is an open-ended information retrieval system designed for the use in the distributed environment. This system contains web pages that provide both information and controls. We can navigate to a new web page in any direction. This is made possible with HTML. Java was meant to be used in the distributed environment such as Internet. So java could be easily incorporated into the web system and is capable of supporting animation graphics, games and other special effect. The web has become more dynamic and interactive with support of java. We can run a java program on remote machine over Internet with the support of web.

**All About JDBC (Java Database Connectivity):**

The first thing you need to do is check that you are set up properly. This involves the following steps:

**Install Java and JDBC on your machine:**

To install both the Java *tm* platform and the JDBC API, simply follow the instructions for downloading the latest release of the JDK *tm* (Java Development Kit *tm* ). When you download the JDK, you will get JDBC as well. The sample code demonstrating the JDBC 1.0 API was written for JDK1.1 and will run on any version of the Java platform that is compatible with JDK1.1, including JDK1.2. Note that the sample code illustrating the JDBC 2.0 API requires JDK1.2 and will not run on JDK1.1.

**Install a driver on your machine:**

Your driver should include instructions for installing it. For JDBC drivers written for specific DBMS, installation consists of just copying the driver onto your machine; there is no special configuration needed. The JDBC-ODBC Bridge driver is not quite as easy to set up. If you download either the Solaris or Windows versions of JDK1.1, you will automatically get the JDBC-ODBC Bridge driver, which does not itself require any special configuration. ODBC, however, does. If you do not already have ODBC on your machine, you will need to see your ODBC driver vendor for information on installation and configuration.

**Install your DBMS if needed:**

If you do not already have a DBMS installed, you will need to follow the vendor's instructions for installation. Most users will have a DBMS installed and will be working with an established database.

**Setting Up a Database:**

We will assume that the database Mailing System already exists. (Creating a database is not at all difficult, but it requires special permissions and is normally done by a database administrator.) When you create the tables used as examples in this tutorial, they will be in the default database. We purposely kept the size and number of tables small to keep things manageable.

Suppose that our sample database is being used by the proprietor of a small coffee house called “The Coffee Break”, where coffee beans are sold for the pound and brewed coffee is served in the cup. To keep things simple, also suppose that the proprietor needs only two tables, one for types of coffee and one for coffee suppliers.

First we will show you how to open a connection with your DBMS, and then, since what JDBC does is to send your SQL code to your DBMS, we will demonstrate some SQL code. After that, we will show you how easy it is to use JDBC to pass these SQL statements to your DBMS and process the results that are returned. This code has been tested on most of the major DBMS products. However, you may encounter some compatibility problems using it with older ODBC drivers with the JDBC-ODBC Bridge.

**Establishing a Connection :**

The first thing you need to do is establish a connection with the DBMS you want to use. This involves two steps: (1) loading the driver and (2) making the connection.

**Loading Drivers:**

Loading the driver or drivers you want to use is very simple and involves just one line of code.

If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Your driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverXYZ , you would load the driver with the following line of code:

Class.forName("jdbc.DriverXYZ");

You do not need to create an instance of a driver and register it with the DriverManager because calling Class.forName will do that for you automatically. If you were to create your own instance, you would be creating an unnecessary duplicate, but it would do no harm. When you have loaded a driver, it is available for making a connection with a DBMS.

**Making the Connection:**

The second step in establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

Connection con =DriverManager.getConnection(url,"myLogin", "myPassword");

This step is also simple, with the hardest thing being what to supply for url . If you are using the JDBC-ODBC Bridge driver, the JDBC URL will start with jdbc:odbc: . The rest of the URL is generally your data source name or database system. So, if you are using ODBC to access an ODBC data source called " Fred, " for example, your JDBC URL could be jdbc:odbc:Fred . In place of " myLogin " you put the name you use to log in to the DBMS; in place of " myPassword " you put your password for the DBMS. So if you log in to your DBMS with a login name of " Fernanda " and a password of " J8, " just these two lines of code will establish a connection:

String url = "jdbc:odbc:Fred";

Connection con = DriverManager.getConnection(url, "Fernanda", "J8");

If you are using a JDBC driver developed by a third party, the documentation will tell you what subprotocol to use, that is, what to put after jdbc: in the JDBC URL. For example, if the driver developer has registered the name acme as the subprotocol, the first and second parts of the JDBC

URL will be jdbc:acme: . The driver documentation will also give you guidelines for the rest of the JDBC URL. This last part of the JDBC URL supplies information for identifying the data source.

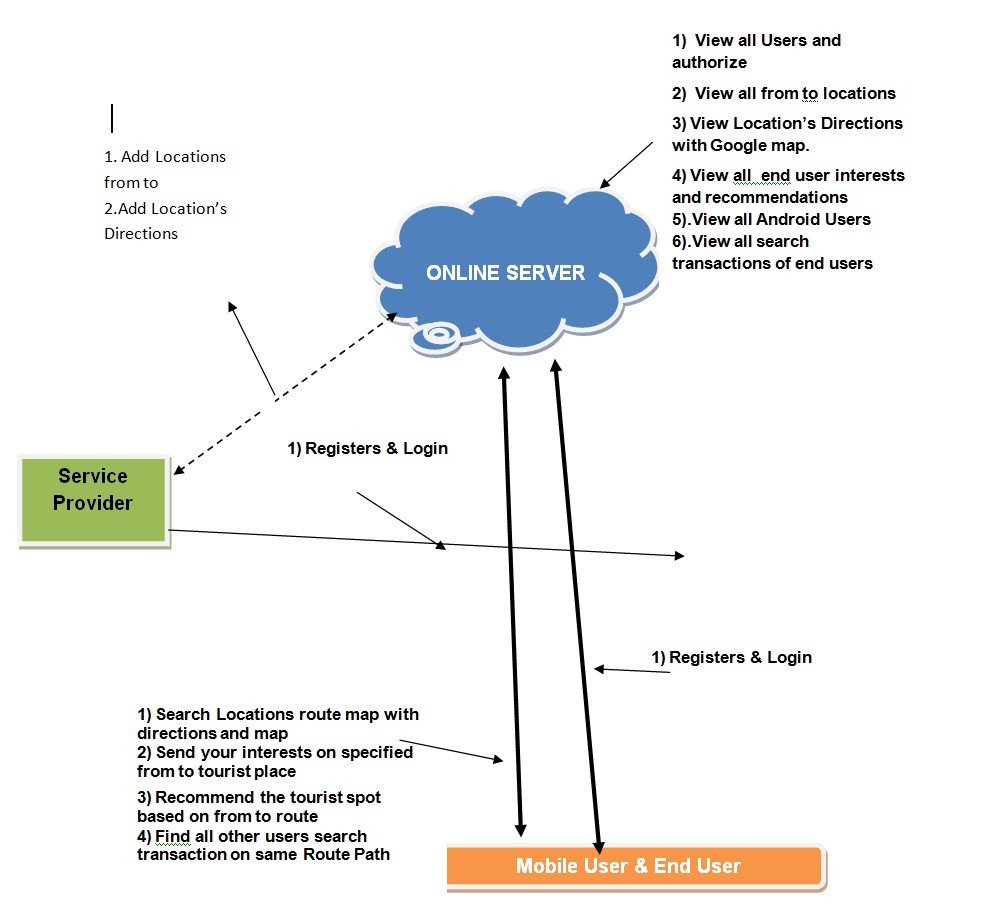
If one of the drivers you loaded recognizes the JDBC URL supplied to the method

DriverManager.getConnection , that driver will establish a connection to the DBMS specified in the JDBC URL. The DriverManager class, true to its name, manages all of the details of establishing the connection for you behind the scenes. Unless you are writing a driver, you will probably never use any of the methods in the interface Driver, and the only DriverManager method you really need to know is DriverManager.getConnection.

# 5.SYSTEM ARCHITECTURE DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

**5.1SYSTEM ARCHITECTURE:**



## Figure 5.1 System Architecture

**5.2UML:**

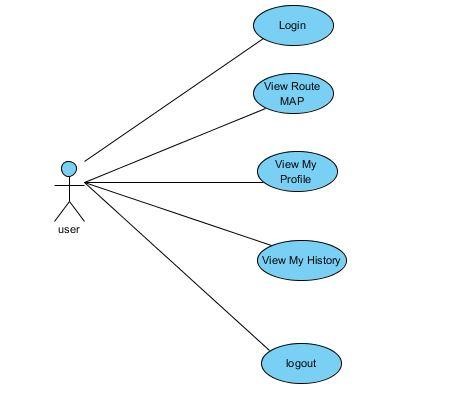
The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system.

Modeling yields an understanding of a system.

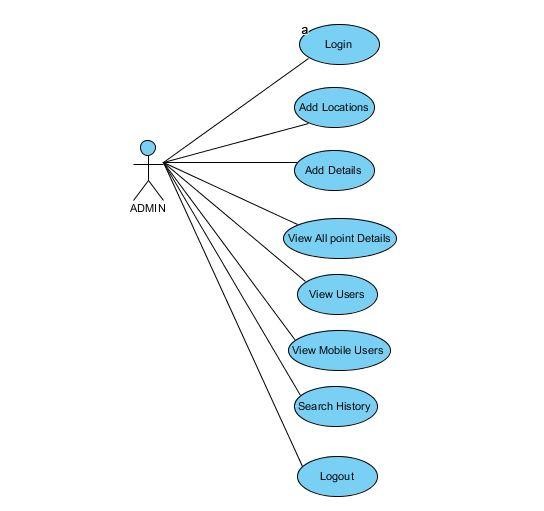
**Diagrams:**

**5.2.1USE CASE DIAGRAM:**

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality.



## Figure 5.2.1: Use case diagram for data user



**Figure 5.2.2: Use case diagram for admin**

**5.2.2: CLASS DIAGRAM:**

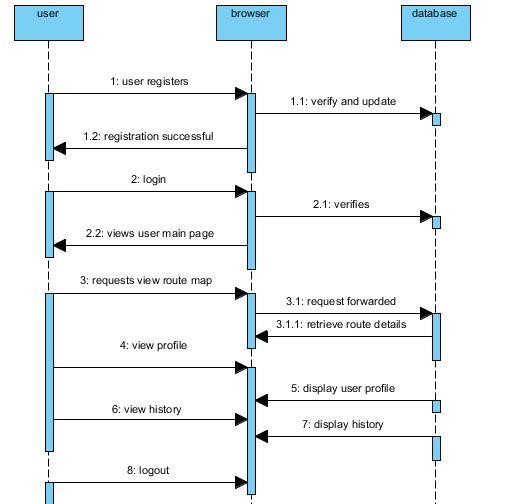
Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. The Class diagram is one of the most widely used diagrams from the UML specification.



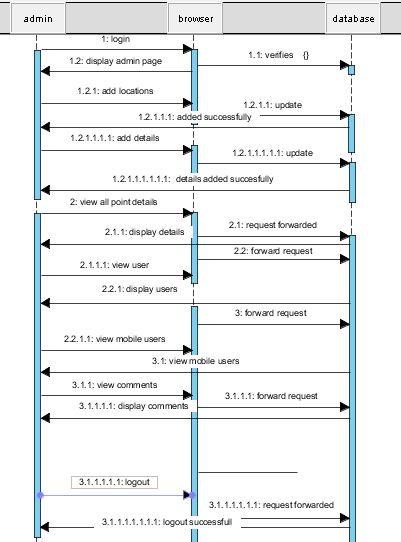
## Figure 5.2.3: Class Diagram

**5.2.3: SEQUENCE DIAGRAM:**

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.



## Figure 5.2.4: Sequence Diagram for user



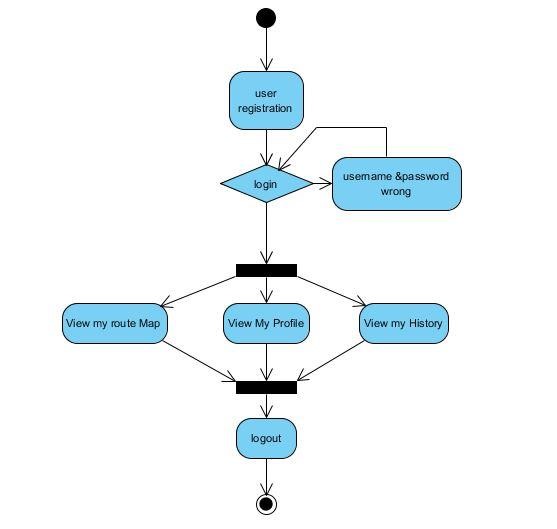
**Figure 5.2.4: Sequence Diagram for user**

**5.2.4: ACTIVITY DIAGRAM:**

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.



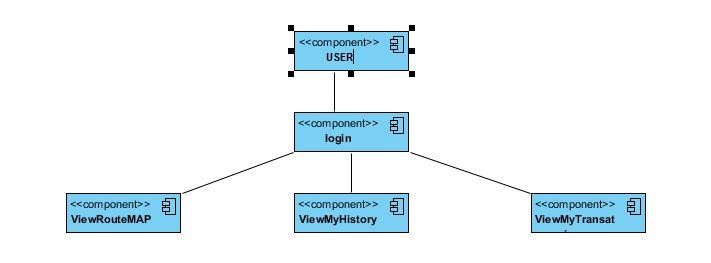
## Figure 5.2.5: Activity diagram for Data admin



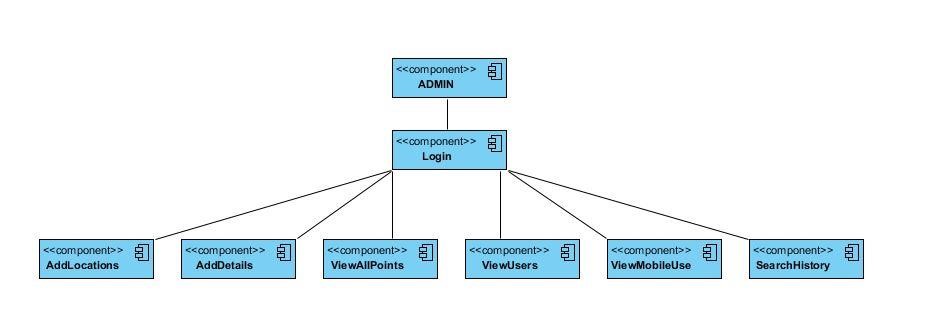
## Figure 5.2.6: Activity diagram for user

**5.2.5: COMPONENT DIAGRAM:**

Component diagram is a special kind of diagram in UML. The purpose is also different from describes the components used to make those functionalities. So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.



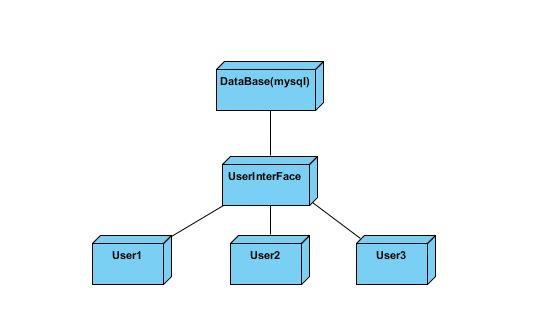
## Figure 5.2.7: Component diagram for user



## Figure 5.2.8 :Component diagram for admin

**5.2.6: DEPLOYMENT DIAGRAM :**

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components. So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.



**Figure 5.2.9: Deployment diagram**

# 6.TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

**6.1TESTING OBJECTIVES:**

1. To ensure that during operation the system will perform as per specification.
2. To make sure that system meets the user requirements during operation.
3. To make sure that during the operation, incorrect input, processing and output will be detected
4. To see that when correct inputs are fed to the system the outputs are correct.
5. To verify that the controls incorporated in the same system as intended.
6. Testing is a process of executing a program with the intent of finding an error.
7. A good test case is one that has a high probability of finding an as yet undiscovered error.

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

**Test Case Design:**

**White box testing:**

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the user is given a chance to enter a valid password. After which it opens the main page.

**Black Box Testing:**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

**Unit Testing:**

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

**Integration Testing:**

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

**Validation Testing**:

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

**System Testing:**

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

1. Implementing the system in a simulated production environment and testing it.
2. Introducing errors and testing for error handling.

**6.2TEST CASES:**

**Test case 1: Home Page**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Scenario** | **Preconditions** | **Test step** | **Test case** | **Test data** | **Experimental**  **results** | **Actual result** | **Pass/fail** |
| Build deployment check.        Build deployment check. | Deploy the build into server.        Deploy the build into server. | Deploy the code folder into server.      Deploy the code folder into server. Start the server. Open  the browser and send a request to server. | Check response on entering the valid URL.    Check response on entering the valid URL. | Home page with all  required component        Home page with all  required component | Home page must be displayed with all the menu items and project title.    Home page must be displayed with all the menu items and project title. | Home page is  not received as response from server. Received home page as response from server. | Fail            Pass |

**Test case 2: Admin login check**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenario** | **Preconditions** | **Test step** | **Test case** | **Test data** | **Experimental results** | **Actual result** | **Pass**  **/fail** |
| Check admin login functionality.              Check admin login functionality. | Deploy the source code into the server.              Deploy the source code into the server.        Open the browser and send a valid URL request to server. | Click on admin login  menu item.        Click on admin login  menu item.    Enter valid username.    Enter valid password. Click on login button. | Check response on entering valid username and password.    Check response on entering valid username and password | Admin username and password.            Admin username and password. | Login must be successful.                Login must be successful. | Login unsucce ssful.              Login success ful. | Fail                  Pass |

**Test case 3: User Registration**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Scenario** | **Preconditions** | **Test step** | **Test**  **Case** | **Test data** | **Experime nt results** | **Actual results** | **Pass/fail** |
| Check user  registration functionality.        Check user  registration functionality. | Deploy the source code into the server.        Deploy the source code into the server.        Open the browser and send a valid URL request to server. | Click on user login  menu item.    Click on user login  menu item.    Click on register now link. Enter the valid text into all the text compone nts. | Check response on entering  all the fields with valid inputs  such as  password length, email, date of birth and contact number.    Again do the same test case. | Password length. Valid email id.  Date of  birth format. Contact number length.    Again check the password length,  date of  birth format and contact no. | All the  inputs  must be  mandatory and valid.                  All the  inputs  must be  mandatory and valid. | Registration un- successful                      Registration successful | Fail                          Pass |

**Test case 4: validate user query**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test scenari o** | **Precondition**  **s** |  | **Test step** | **Test**  **Case** | **Test data** | **Experimenta**  **l results** | **Actual result** | **Pass/fai**  **l** |
| Check user query details.          Check user query details. | Deploy source into server.          Deploy source into server.  Open browser send a valid user id and password.    Register/login to the account. | the code the the code the  the and    user | Click on user login menu item. Enter valid user id and  password  .  Click on user login menu item. Enter valid user id and  password    Click on send query link. | Check respons  -e on  entering all the  fields with valid input.    Check respons  -e on  entering all the  fields with valid input. | Input  Query.    User name.    Addres s field.    Valid email id.    Valid length contact number  . | All the inputs must be  mandatory and valid.              All the inputs must be  mandatory  and valid | Validatio  n un-  successful                Validatio n successful | Fail                    Pass |

**TEST CASES:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case** | **Check**  **Item** | **Test Case**  **Objective** | **Test Data**  **/Input** | **Experimental Result** | **Actual**  **Result** | **Result** |
| TC-001 | Login  Page | Leave all the fields as blank and click login button |  | By leaving all the fields as blank and on click login button then mandatory symbol(\*) should appearin front of username and password. |  | Fail |
| TC-002 | Login  Page | Leave all the fields as blank and click login button |  | By leaving all the fields as blank and on click login button then mandatory symbol(\*) should appearin front of username and password. |  | Pass |
| TC-003 | Username | Enter invalid username | Username:  Deepthi | By entering invalid username then an error message should appear as “Please Enter Valid Username”. |  | Fail |
| TC-004 | Username | Enter invalid username | Username:  Deepthi | By entering invalid username then an error message should appear as “Please Enter Valid Username”. |  | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC-005 | Username | Enter valid username | Username:  Deepthi | It should allow the user to proceed |  | Fail |
| TC-006 | Username | Enter valid username | Username:  Deepthi | It should allow the user to proceed |  | Pass |
| TC-007 | Password |  |  | The password field should display the encrypted format of the text typed as (\*\*\*). |  | Fail |
| TC-008 | Password |  |  | The password field should display the encrypted format of the text typed as (\*\*\*). |  | Pass |
| TC-009 | Password | Enter wrong password | Password:\*\*\* | By entering the invalid password then an error message should appear  as “Please Enter  Correct Password”. |  | Fail |
| TC-010 | Password | Enter wrong password | Password:\*\*\* | By entering the invalid password then an error message should appear  as “Please Enter  Correct Password”. |  | Pass |
| TC-011 | Password | Enter correct password | Password:  \*\*\*\*\*\*\*\*\* | It should allow the user to proceed. |  | Fail |
| TC-012 | Password | Enter correct password | Password:  \*\*\*\*\*\*\*\*\* | It should allow the user to proceed |  | Pass |
| TC-013 | Login  Button | Correct inputs |  | It should lead the user to the respect page. |  | Fail |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC-014 | Login  Button | Correct inputs |  | It should lead the user to the respect page. |  | Pass |
| TC-015 | Forgot password | Check hyperlink on forgot password table |  | While mouse over of the label an hand icon should display. |  | Fail |
| TC-016 | Forgot password | Check hyperlink on forgot password table |  | While mouse over of the label an hand icon should display. |  | Pass |
| TC-017 | Forgot password |  |  | User can recover the password using the  “Forgot Password” link page. |  | Fail |
| TC-018 | Forgot password |  |  | User can recover the password using the  “Forgot Password” link page. |  | Pass |
| TC-019 | Registration | Check hyperlink on registration table |  | While mouse over of the label an hand icon should display. |  | Fail |
| TC-020 | Registration | Check hyperlink on registration table |  | While mouse over of the label an hand icon should display. |  | Pass |
| TC-021 | Registration |  |  | On click  “Registration” page should redirect to the User Registration page. |  | Pass |

# 7. SCREEN SHOTS

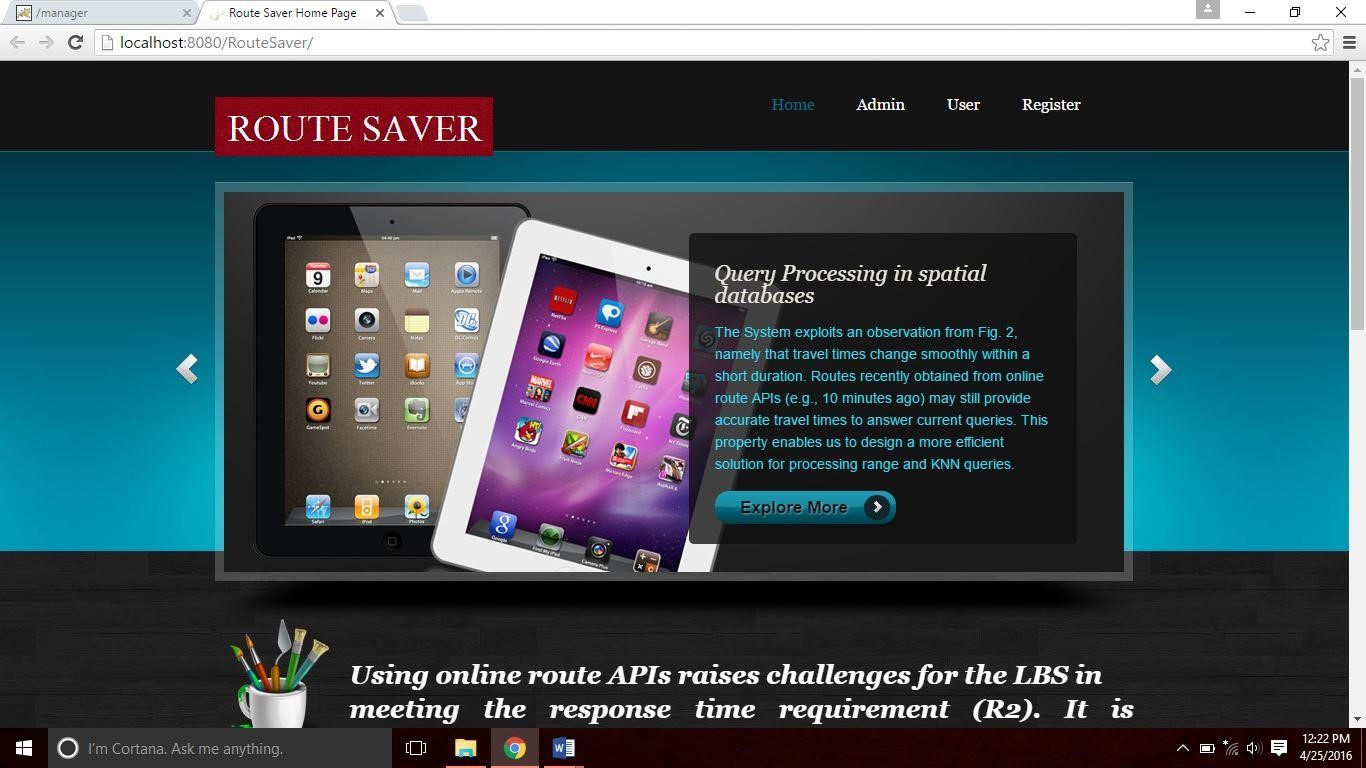
**Figure**

**7.**

**1**

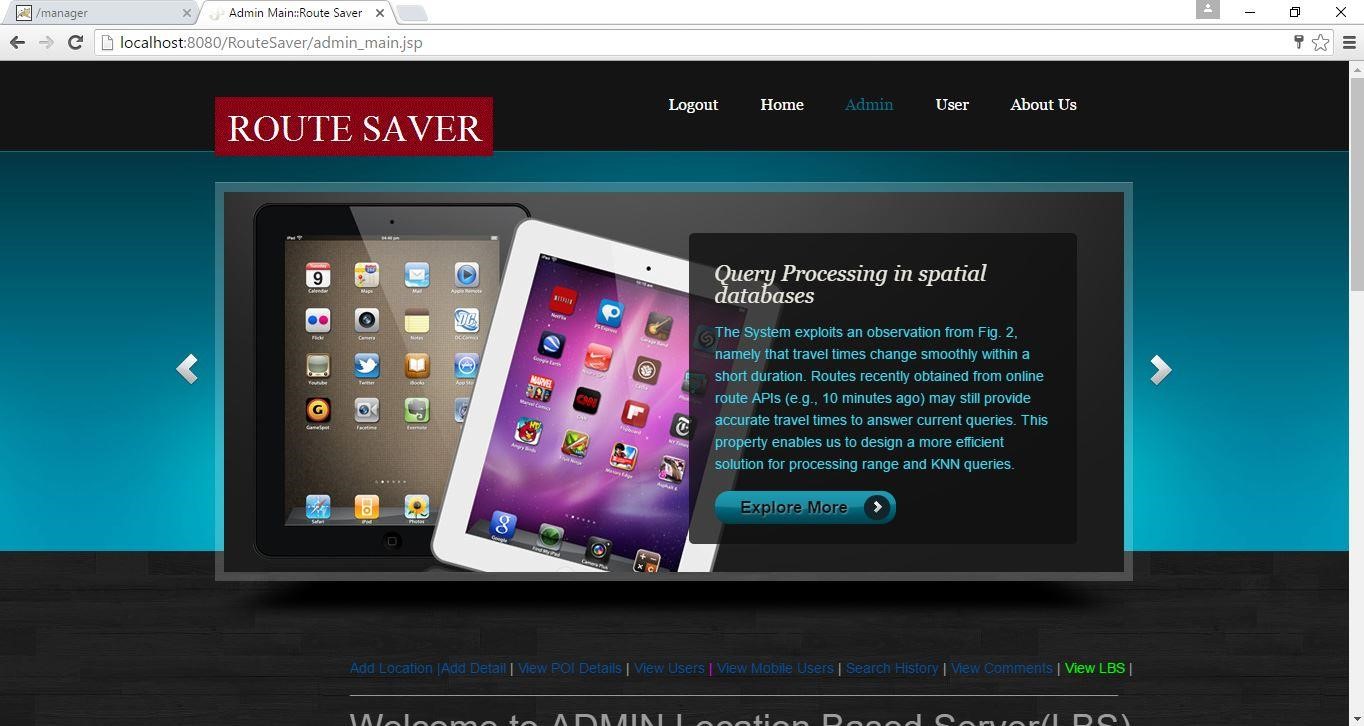
**:**

**Home page**

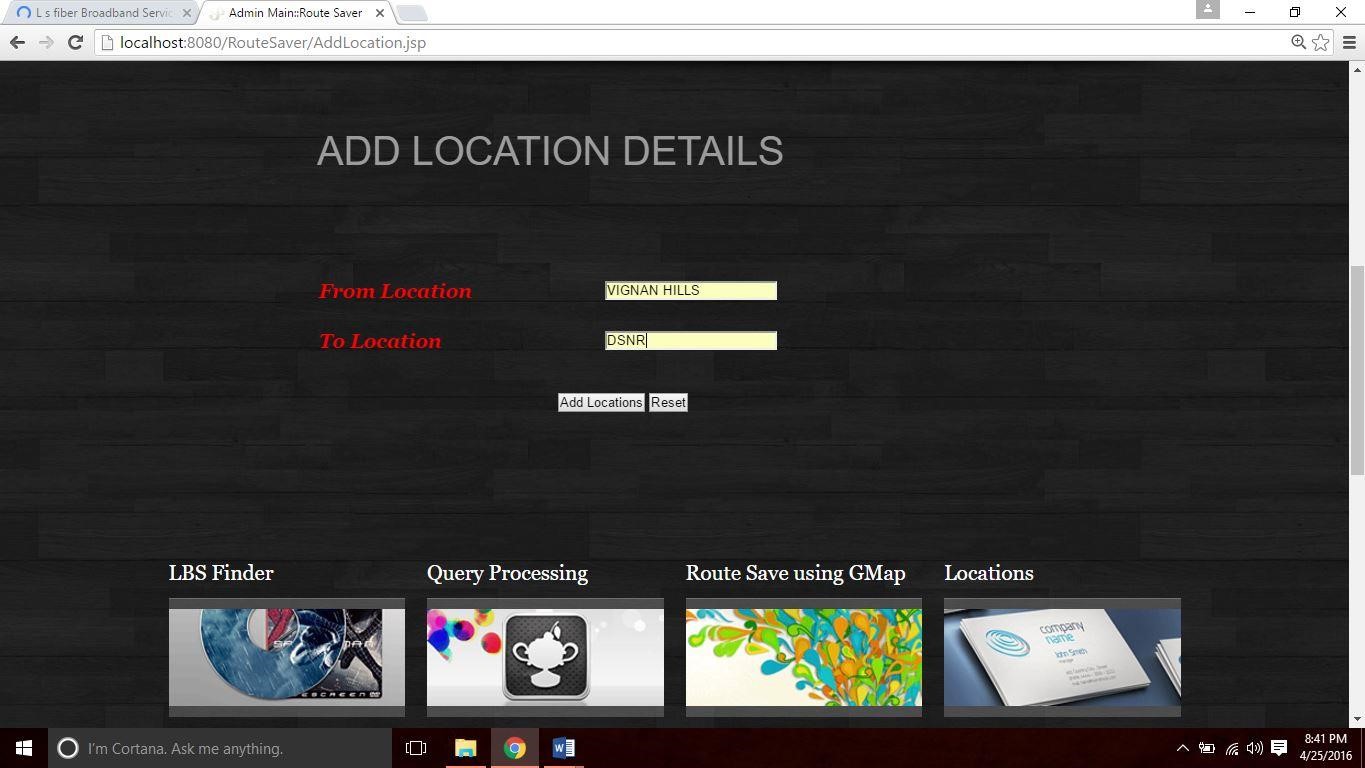




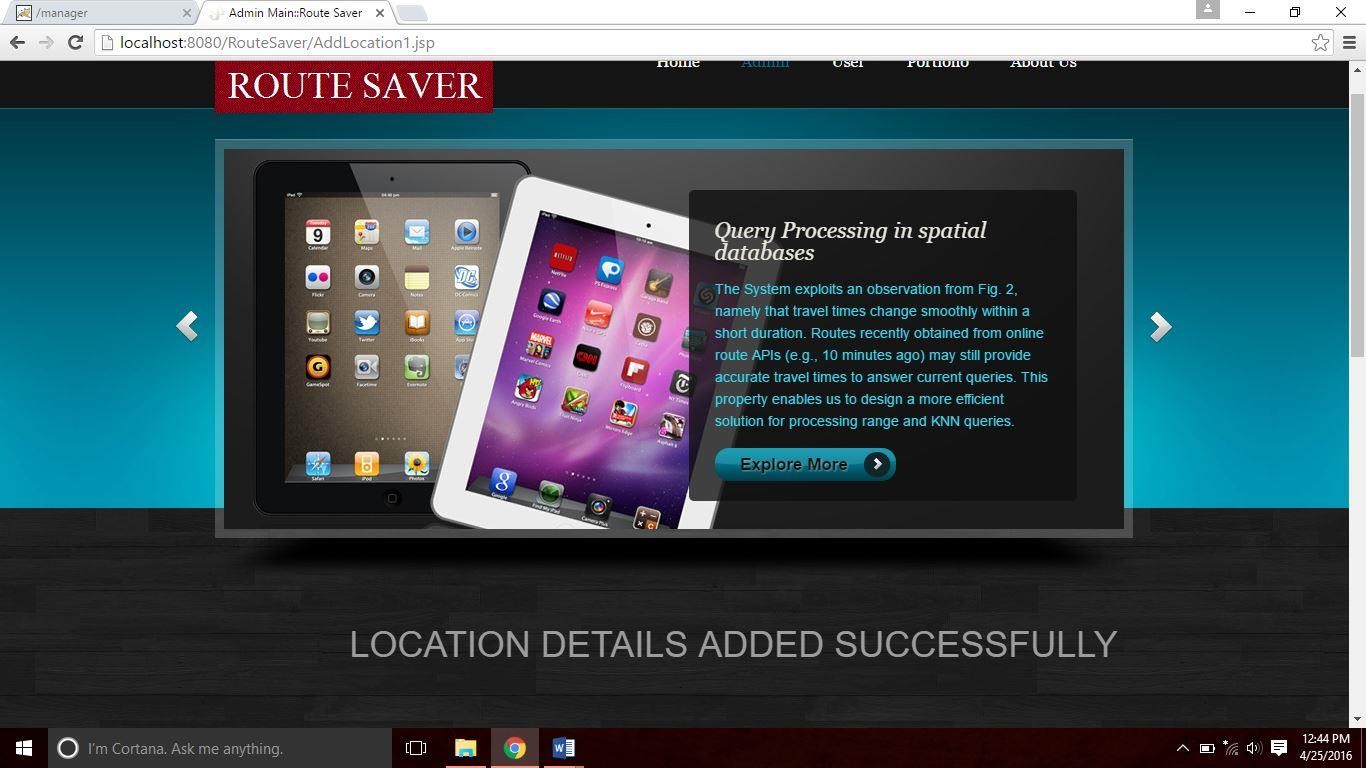
**Figure 7.2:Admin login Page**



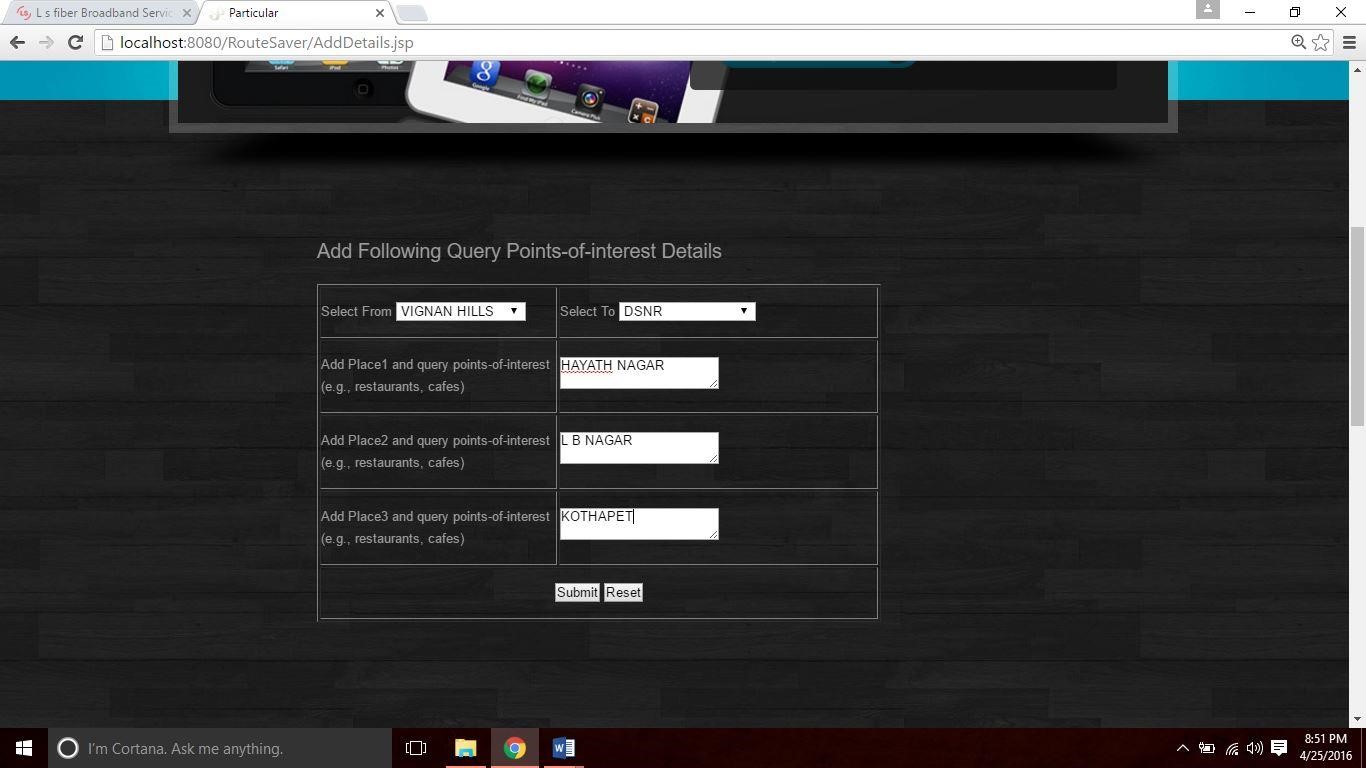
**Figure 7.3:Admin Page**



**Figure 7.4:AddLocation Page**



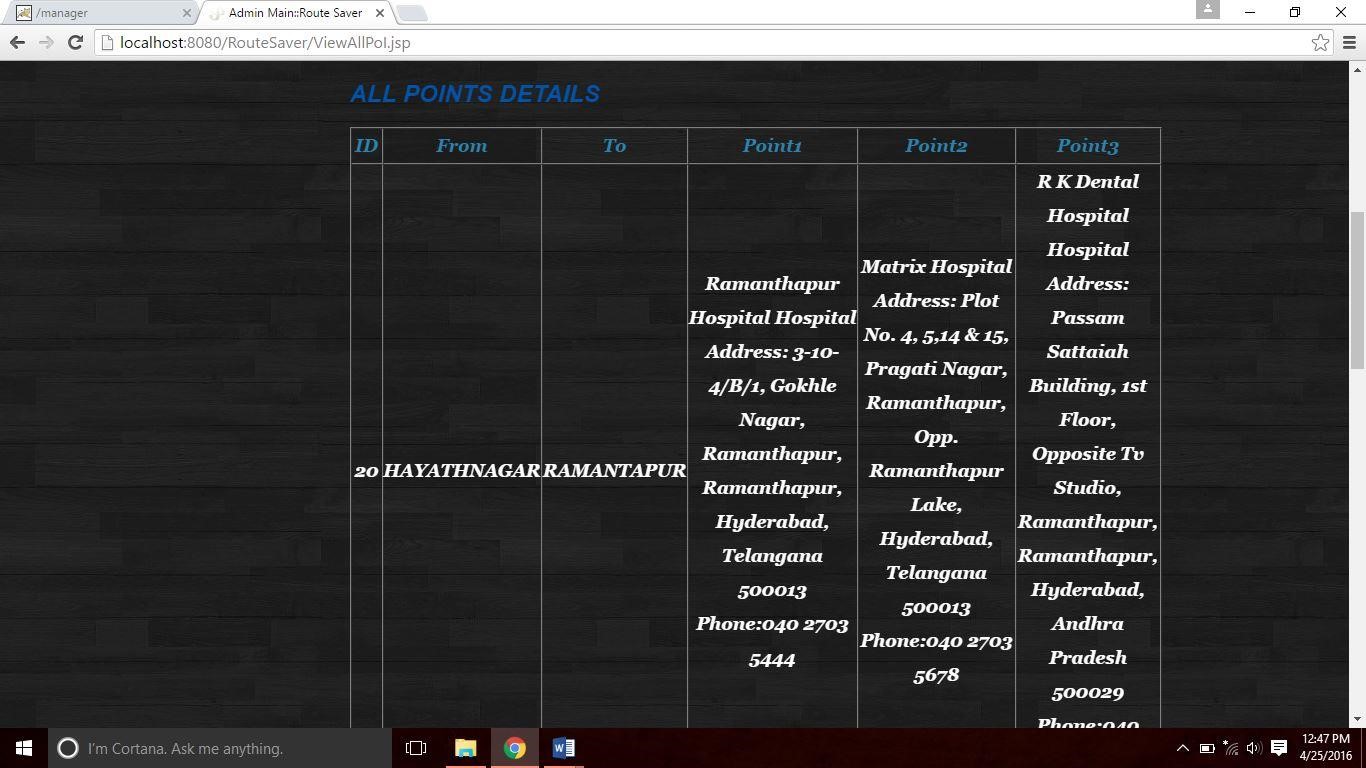
**Figure 7.5:Location Added Page**



**Figure 7.6:AddDetails Page**



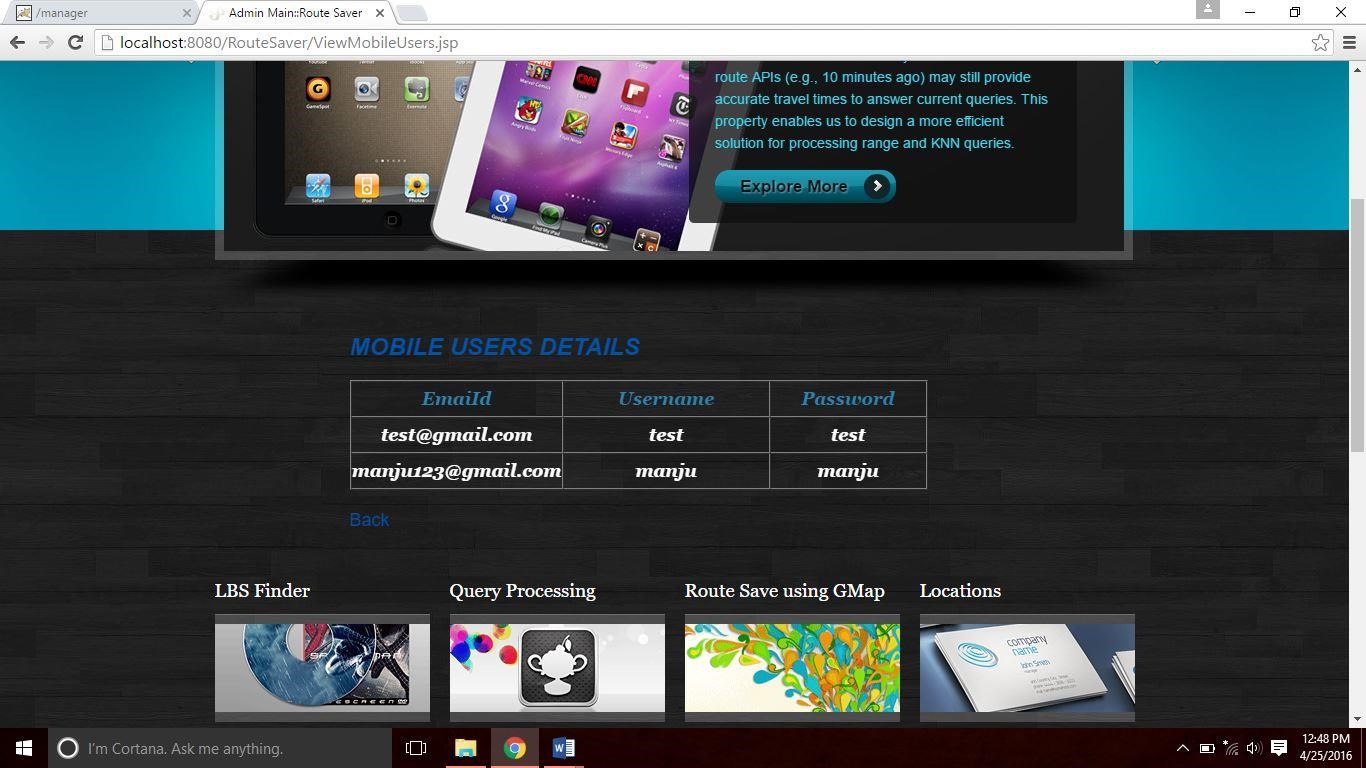
**Figure 7.7 :AddDetails Successfully Page**



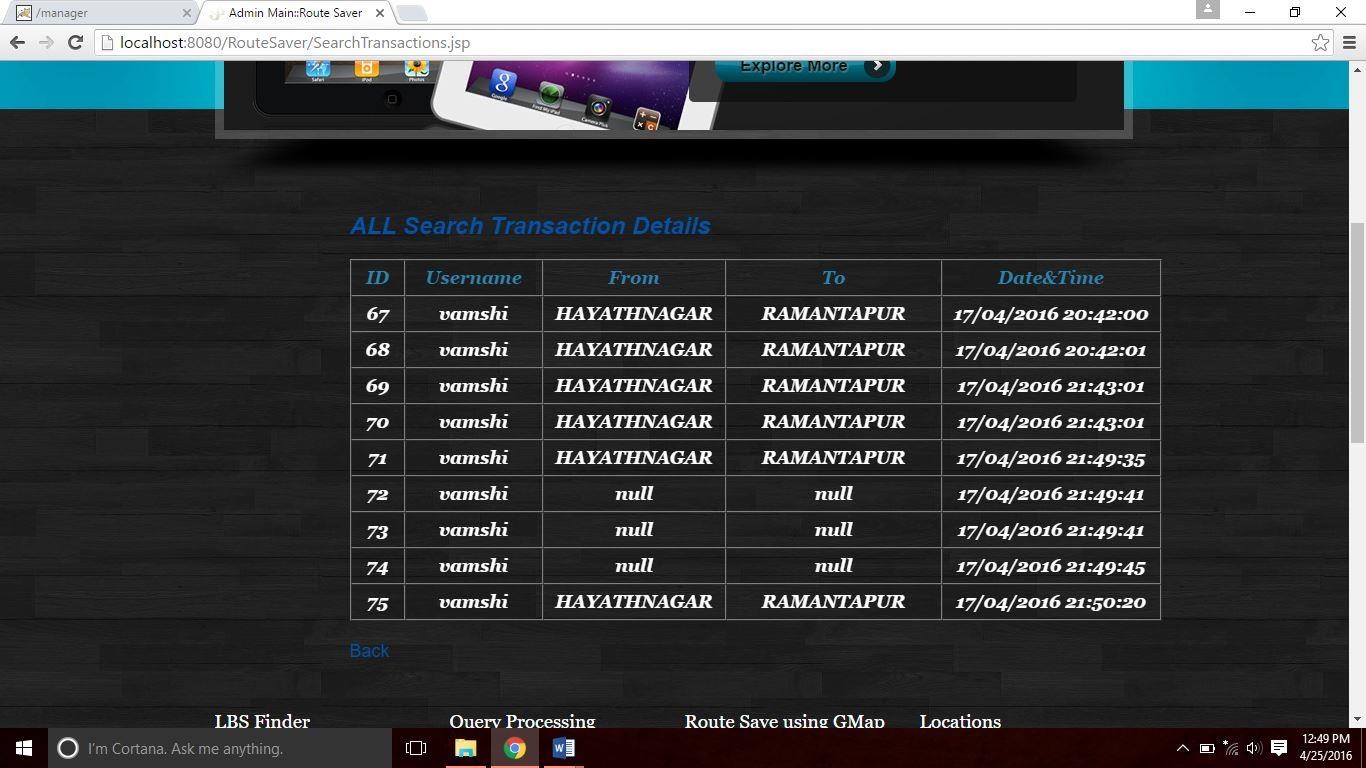
**Figure 7.8:View All Points page**



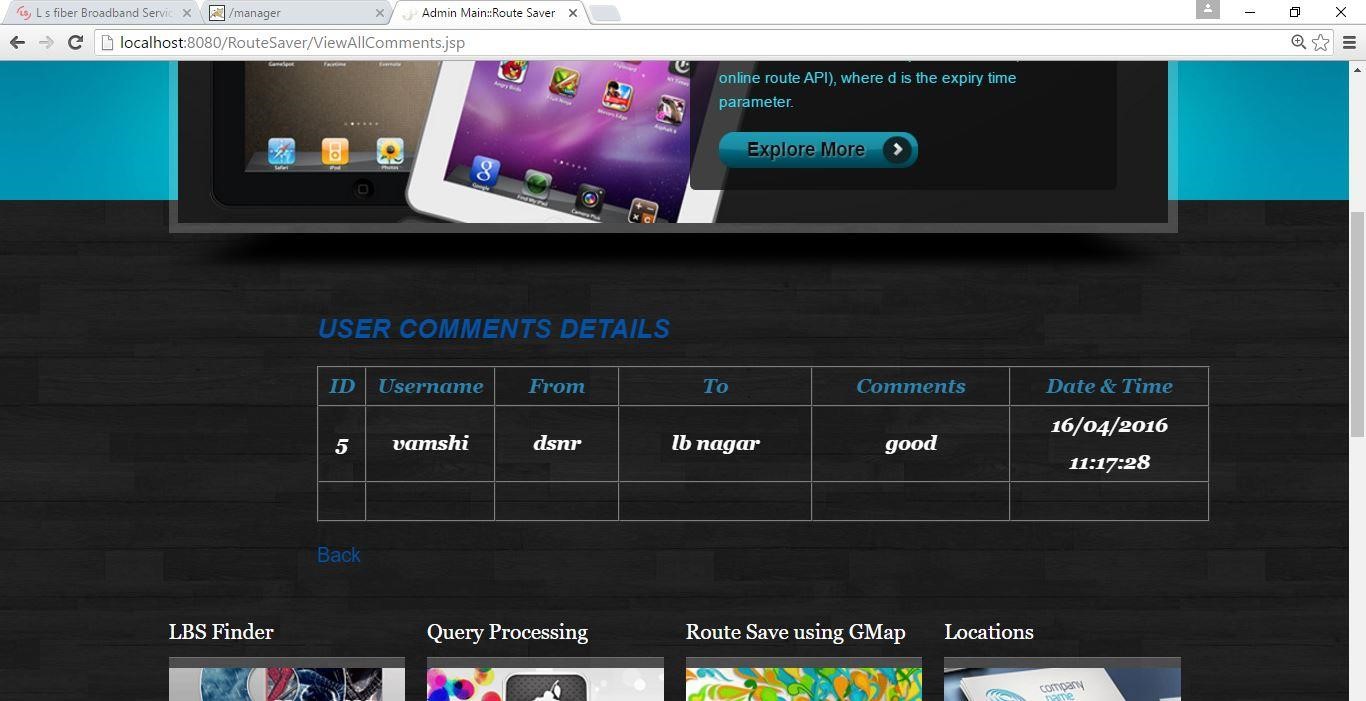
**Figure 7.9 :End Users Page**



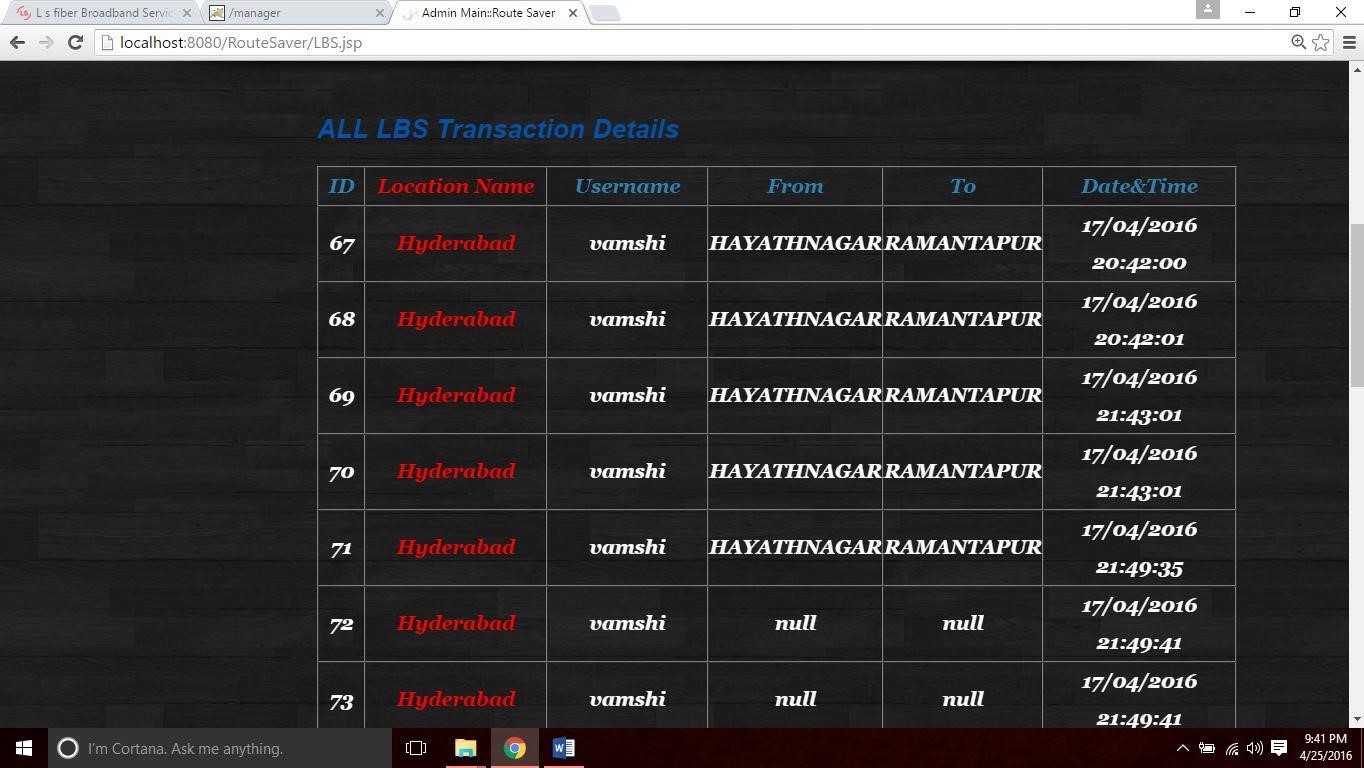
**Figure 7.10 :Mobile Users Page**



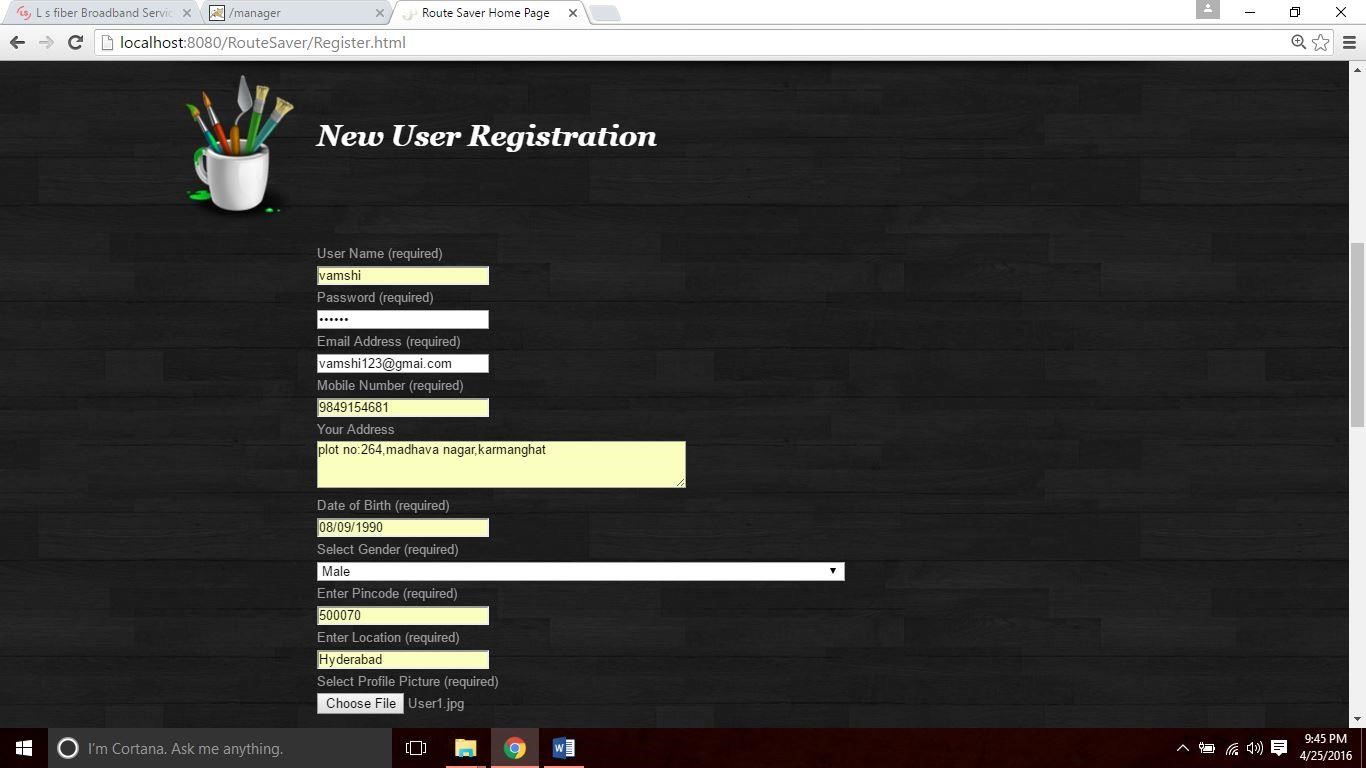
**Figure 7.11 :Search History Page**



**Figure 7.12 : Comments Page**



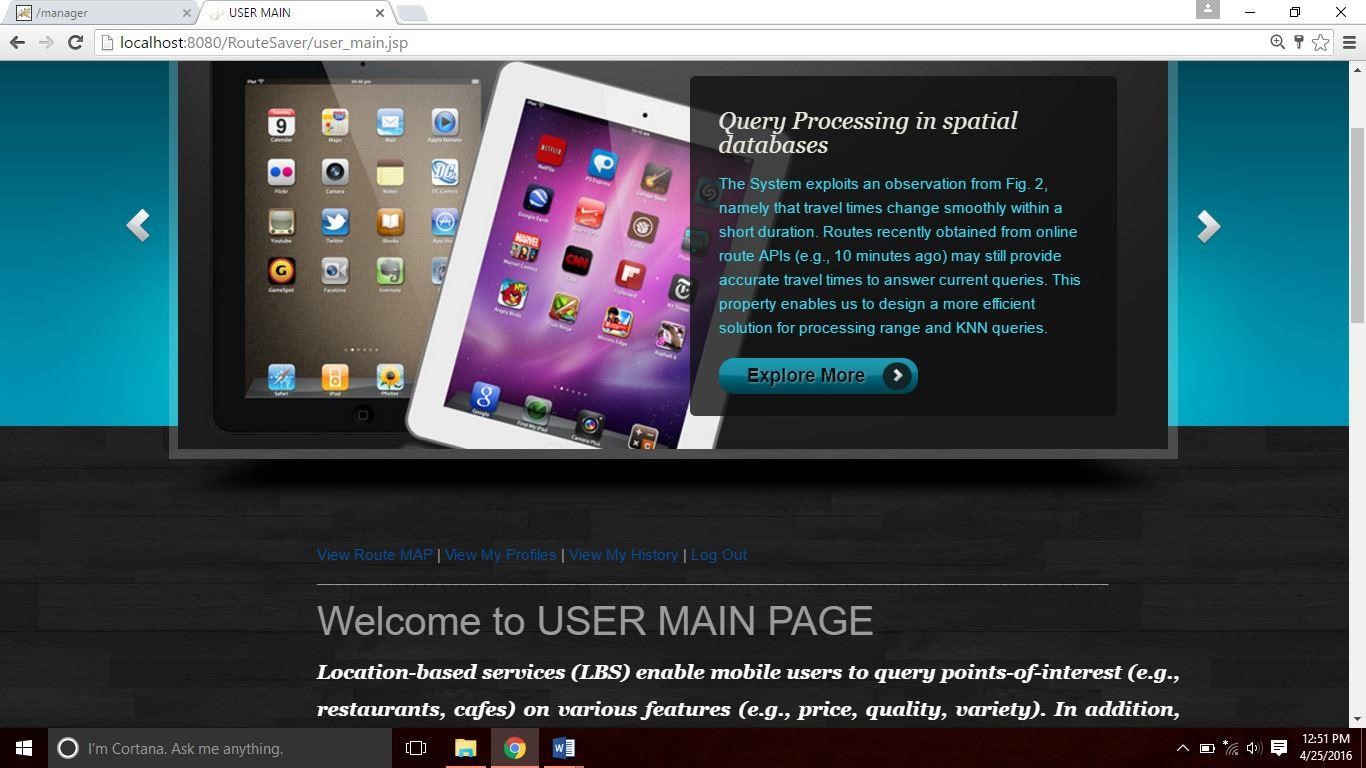
**Figure 7.13 :All LBS Transaction Details Page**



**Figure 7.14 :Registration Page**



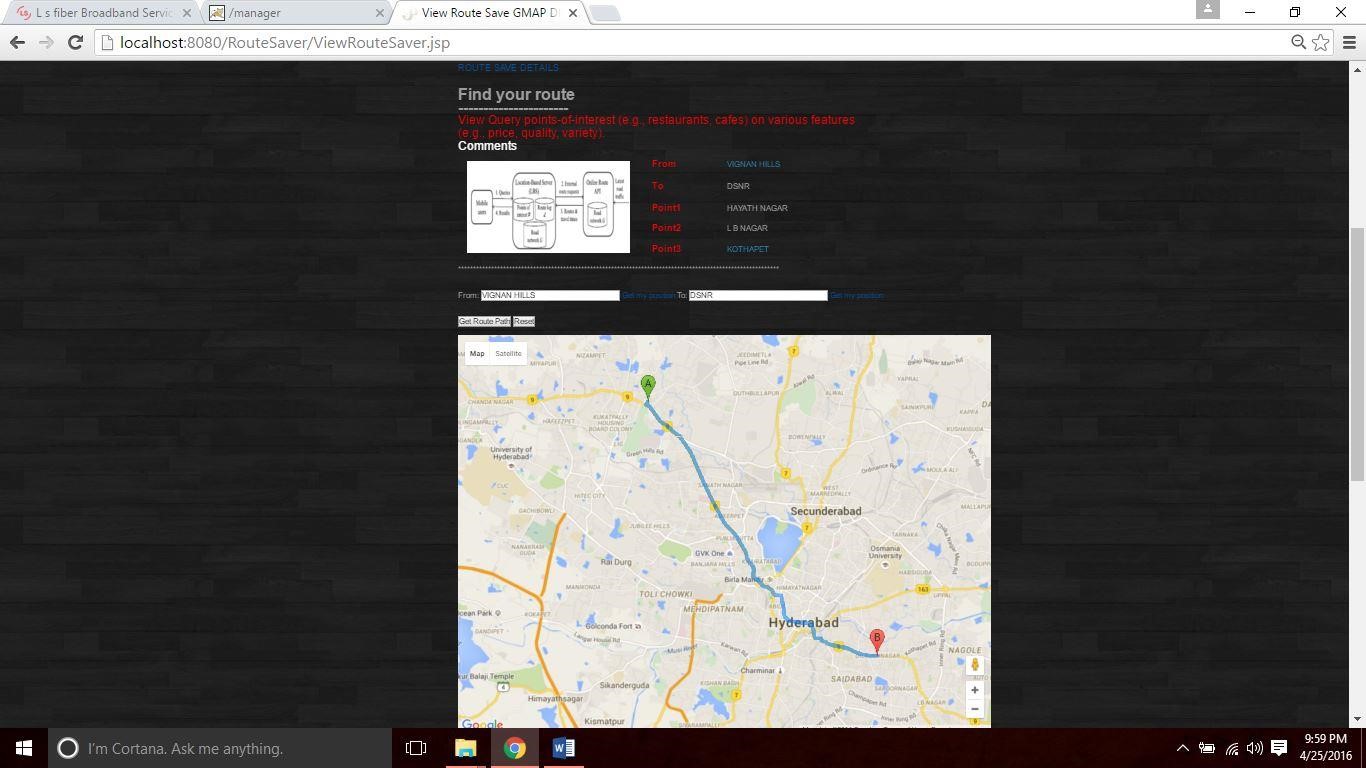
**Figure 7.15 :User Login Page**



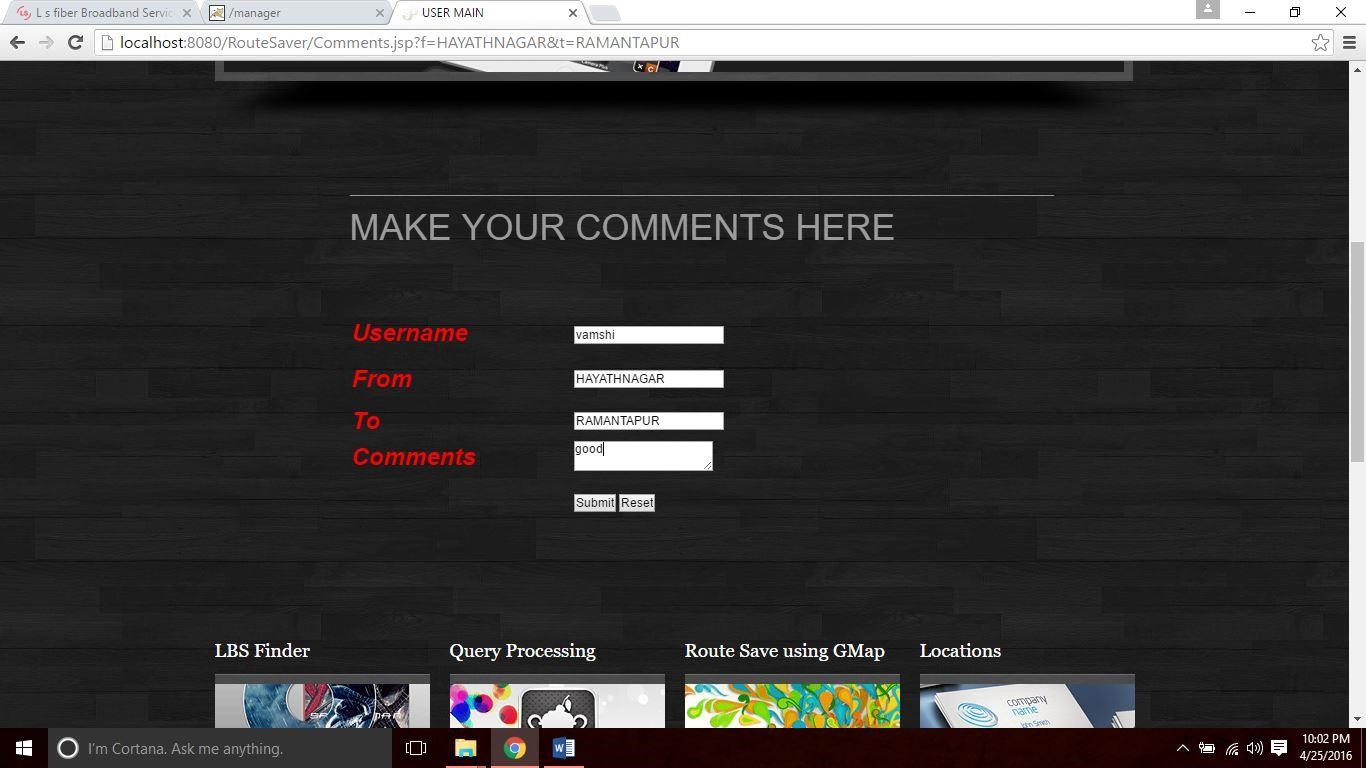
**Figure 7.16 :User Main Page**



**Figure 7.17 :Select From & To Details Page**



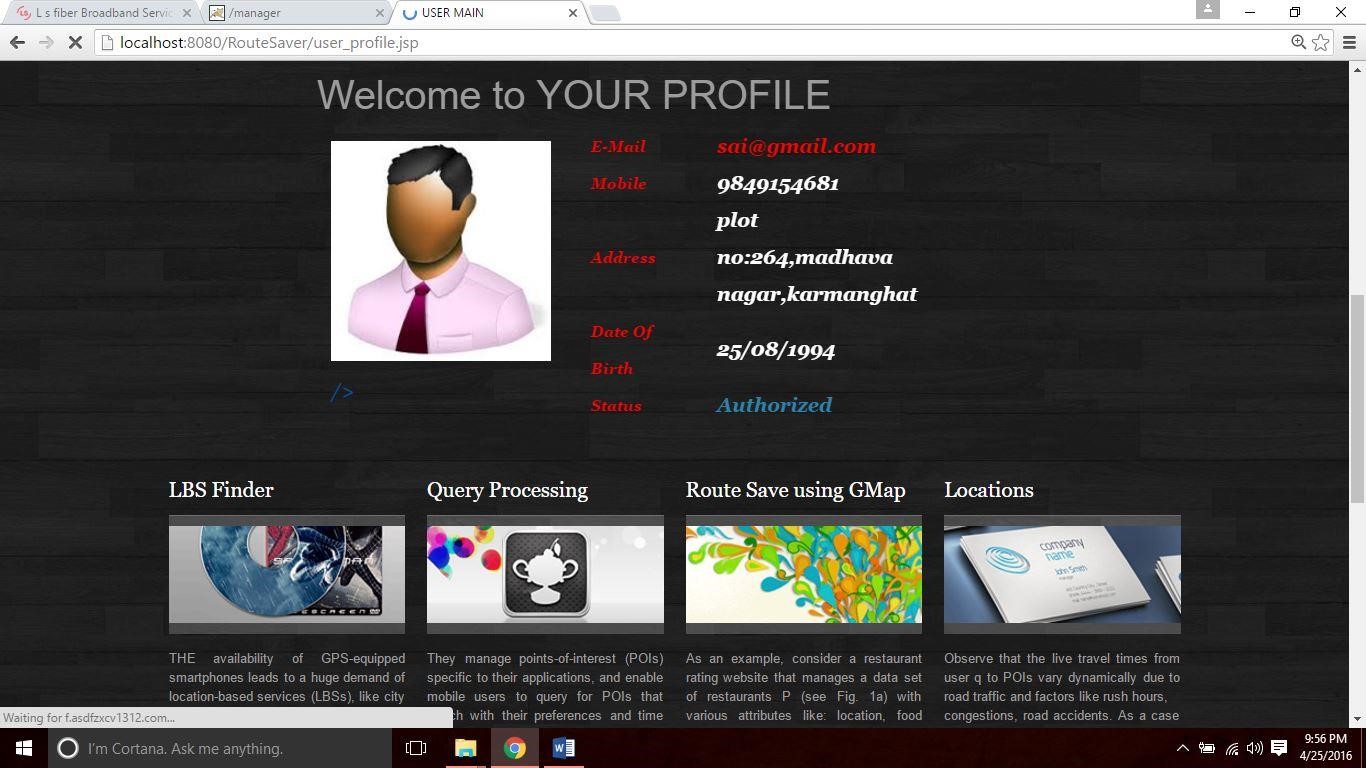
**Figure 7.18 :Select From & To Route MAP and Points**



**Figure 7.19:user comment page**



**Figure 7.20:user commets made successfully Page**



**Figure 7.21:User Profile Page**



**Figure 7.22:User History**

# 8.CONCLUSION

**8.1. CONCLUSION :**

In this paper, we propose a solution for the LBS to process range/KNN queries such that the query results have accurate travel times and the LBS incurs few number of route requests. Our solution Route-Saver collects recent routes obtained from an online route API (wit hind minutes). During query processing, it exploits those routes to derive effective lower-upper bounds for saving route requests, and examines the candidates for queries in an effective order. We have also studied the parallelization of route requests to further reduce query response time. Our experimental evaluation shows that Route-Saver is 3 times more efficient than a competitor, and yet achieves high result accuracy (above 98 percent). In future, we plan to investigate automatic tuning the expiry time d based on a given accuracy requirement. This would help the LBS guarantee its accuracy and improve their users’ satisfaction.

**10.BIBLIOGRAPHY REFERENCES :**

1. 2011 Census TIGER/Line Shapefiles. (2011).[Online].Available: [http://www.census.gov/cgibin/geo/shapefiles2011/main](http://www.census.gov/cgi-bin/geo/shapefiles2011/main)

1. 9th DIMACS Implementation Challenge on Shortest Paths. (2013). [Online]. Available:

<http://www.dis.uniroma1.it/challenge9/data/tiger/>

1. Bing Data Suppliers. (2013). [Online]. Available: [http://windows.](http://windows/) microsoft.com/enHK/windows-live/about-bing-data-supplie rs/

1. Bing Maps API. (2013). [Online]. Available: [http://www.](http://www/) microsoft.com/maps/developers/web.aspx

1. Bing Maps Licensing and Pricing Information. (2013). [Online]. Available:

<http://www.microsoft.com/maps/product/licensing.aspx>

1. Google Directions & Bing Maps: Live Traffic Information. (2013). [Online]. Available:

<http://support.google.com/maps/bin/answer.py?hl=en&answer=2549020&topic=1687356&><http://msdn.microsoft.com/en-us/library/aa907680.aspx>

1. Google Directions API. (2013). [Online]. Available:

https:// developers.google.com/maps/documentation/directions/

1. Google Directions API Usage Limits. (2013). [Online]. Available:

<https://developers.google.com/maps/faq#usagelimits>

1. Google Map Maker Data Download. (2013). [Online]. Available:

<https://services.google.com/fb/forms/mapmakerdatadownload/>

1. OpenStreetMap. (2013). [Online]. Available: http://www.openstreetmap.org/
2. Statistics of Usage. (2013). [Online]. Available: [http://www.quantcast.com](http://www.quantcast.com/)

1. US Maps from Government. (2013). [Online]. Available: <http://www.usgs.gov/pubprod/>

1. N. Bruno, S. Chaudhuri, and L. Gravano, “STHoles: A multidimensionalworkload-aware histogram,” in Proc. ACM SIGMODInt. Conf. Manage. Data, 2001, pp. 211–222.

1. E. P. F. Chan and Y. Yang, “Shortest path tree computation indynamic graphs,” IEEETrans. Comput., vol. 58, no. 4, pp. 541–557,Apr. 2009.

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introductionto Algorithms.Cambridge, MA, USA: MIT Press, 2009.

1. U. Demiryurek, F. B. Kashani, C. Shahabi, and A. Ranganathan,“Online computation offastest path in time-dependent spatialnetworks,” in Proc. 12th Int. Symp. Adv. SpatialTemporal Databases,2011, pp. 92–111.

1. A. Dingle and T. Partl, “Web cache coherence,” Comput. Netw., vol. 28, pp. 907–920, 1996.