

A Mini Project Report on
ON ROAD BREAKDOWN

Submitted to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
HYDERABAD**

in partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

G.MANASA	(16BD1A054N)
K.SRAVANTHI	(16BD1A055J)
K.SWETHA	(16BD1A055K)
KSVT NAGILA KRISHNA VENI	(16BD1A055G)

Under the esteemed guidance of

Dr. K.L.S . SIRISHA
Associate Professor
Department of CSE



Department of Computer Science and Engineering
KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH
3-5-1206, Narayanaguda, Hyderabad - 500029

2019 – 2020

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTU, Hyderabad

3-5-1206, Narayanaguda, Hyderabad - 500029.



CERTIFICATE

This is to certify that the project entitled “**ON ROAD BREAKDOWN**” being submitted by **G.MANASA (16BD1A054N), K.SRAVANTHI (16BD1A055J), K.SWETHA (16BD1A055K), K.S.V.T.NAGILA KRISHNA VENI (16BD1A055G)** students of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of the requirements of the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** as a specialization is a record of bonafide work carried out by them under my guidance and supervision in the academic year 2019 – 2020

INTERNAL GUIDE

Dr. K.L.S Sirisha
Associate professor

HEAD OF THE DEPARTMENT

Dr. S. Padmaja
HOD - CSE

Submitted for the Project Viva Voice examination held on

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled “**ON ROAD BREAKDOWN**” is done in the partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

G.MANASA	(16BD1A054N)
K.SRAVANTHI	(16BD1A055J)
K.SWETHA	(16BD1A055K)
KSVT NAGILA KRISHNA VENI	(16BD1A055G)

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. Maheshwar Dutta**, B.E.M.Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project. We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Deepa Abhishek Ganu**, Academic Director for providing an excellent environment in the college.

We are also thankful to **Dr. S . Padmaja** , Head of Department for providing us with both time and amenities to make this project a success within the given schedule. We are also thankful to our guide **Dr. K.L.S Sirisha**, for her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly or indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

G.MANASA	(16BD1A054N)
K.SRAVANTHI	(16BD1A055J)
K.SWETHA	(16BD1A055K)
KSVT NAGILA KRISHNA VENI	(16BD1A055G)

CONTENTS

DESCRIPTION	PAGE NO
ABSTRACT	i
LIST OF FIGURES	ii

CHAPTER-1

1. INTRODUCTION	1-2
1.1 Purpose of the Project	1
1.2 Problem with the Existing System	1
1.3 Proposed System	2
1.4 Scope of the Project	2

CHAPTER-2

2. SOFTWARE REQUIREMENTS SPECIFICATIONS	3-6
2.1 Requirements Specification Document	3
2.1.1 Functional Requirements	4
2.1.2 Non-Functional Requirements	5
2.2 Software Requirements	6
2.3 Hardware Requirements	6

CHAPTER-3

3. LITERATURE SURVEY	7-20
-----------------------------	-------------

CHAPTER-4

4. SYSTEM DESIGN 21-26

4.1 Introduction to UML 21

4.2 UML Diagrams 21

4.2.1 Use Case diagram 21

4.2.2 Sequence diagram 23

4.2.3 Class diagram 25

CHAPTER-5

5. IMPLEMENTATION 27-34

5.1 Pseudo code 27

5.2 Code Snippets 27

CHAPTER-6

6. TESTING 35-42

6.1 Introduction to Testing 35

6.2 Testing Activities 36

6.3 Types of Testing 36

6.4 Test Plan 37

6.5 Test Cases 38

CHAPTER-7

7. SCREENSHOTS 43-52

7.1 Home Page	43
7.2 Contact Us Page	44
7.3 About Us Page	45
7.4 Review Page	46
7.5 FAQ Page	46
7.6 User Registration Page	47
7.7 User Login Page	47
7.8 Service Provider Registration Page	48
7.9 Forgot Password Page	49
7.10 User Login	49
7.11 Service Provider Details	50
7.12 Forgot Password Page	51
7.13 Service Provider Home Page	51
7.14 Service Provider Transaction Table	52

Conclusion 53

Future Enhancements 54

References 55

ABSTRACT

While travelling if your vehicle suddenly breakdown it is difficult to find mechanics in the nearby locality. Our application helps people to find mechanical assistance during such times. Customers just have to login to his/her account and provide their location and with-in few minutes our application gives the location of all possible mechanics nearby. Our application makes the process of finding mechanics not only easy but also provides service at a reasonable rate.

We have pre-bookings also where the customer can book a mechanic service on respective date and time. The application doesn't just assure a prompt service in the rare event of a car breakdown, but it also helps with the mechanical breakdown towing, fuel delivery, flat tyre change and car collision etc. The application is designed to enhance the user experience and ensure that users get immediate and hassle free service in the event of any vehicle breakdown.

LIST OF FIGURES

1. IEEE 829 test plan template	8
2. Servlet Architecture	11
3. JSP Architecture	19
4. Use case Diagram	22
5. Sequence Diagram - User	24
6. Sequence Diagram - Service Provider	25
7. Class Diagram	26
8. Black Box Testing	37
9. White Box Testing	37

1.INTRODUCTION

1.1 PURPOSE OF THE PROJECT

With the increase in demand for train and flight tickets most of us end up planning a road trip and for the others, road trips are means of getting together.

New vehicles are loaded with advanced technology designed to keep drivers and passengers safe and on the road—not in the breakdown lane. A new study by the American Automobile Association found that technology—including maintenance reminders and other safety alerts—hasn’t reduced the number of drivers stranded on the roadside. Breakdowns are actually happening more than ever.

“Vehicles today are advanced more than ever, yet are still vulnerable to breakdowns,” Cliff Ruud, AAA’s managing director of automotive solutions said in a statement. “Sleek, low profile tires are highly susceptible to damage, electronic key less ignitions can zap battery life, and despite advanced warning systems, more than half a million drivers ran out of gas last year.”

Road trips are always fun and enjoyable, but not everything goes as expected. You might also plan and take all precautions to have a safe and smooth journey. However, in the unfortunate & unforeseen event of a breakdown or road accidents what is needed is immediate help. Imagine if such a situation happens near a highway or in a place unknown to us.

Well our goal is to ensure that you get prompt assistance during such a situation.

By using our web application the user can immediately get help. The user just needs to mention the location and select the problem from the given options. We provide assistance to many problems like exchange of tyres in case of a flat tyre or a puncture, fuel services like refill of petrol, diesel, gas etc. The user will get updates regarding the service selected .

1.2 PROBLEMS WITH EXISTING SYSTEM

The assistance provided to the travelers is highly limited to the types of services that is being provided by the particular service providers, service centres or mechanic shops. Assistance through the web application is prone to unavailability of the services in case of remote locations and network availability.

The web applications provides information of the available service providers the user based on the data available in the database which changes from area to area and hence the web application does not guarantee or assure help in some circumstances.

Some areas may not have any help at all which cannot be guaranteed either. The web application uses mail system for the communication to the service providers, service centers or mechanic shops which cannot be as fast as expected (i.e. the response may be slow at times.).

1.3 PROPOSED SYSTEM

The proposed system is of a web application that is helpful when a person is in need of service centre, service providers or mechanical shops. The user will have to mention his/her details. Then the user will have to choose a location from the given list of locations and then the user will have to mention the services required by the user. Depending on the services the web application will choose the most optimal service centre service provider or the mechanical shop or any other registered service representative from the database. A list of service providers will be displayed from which the user will be able to select any service provider of his or her choice. The confirmation is taken from the user and then this service provider will be informed about the location and the service that has to be provided. The user's contact number will be mailed to the service provider by which the service provider can contact the user and enable better communication.

The web application is user friendly and has easily understandable user interface which in turn enables easy operation in the case of an emergency. Once the details have been submitted the web application connects to the database and retrieves information in the form of a query from the database.

1.4 SCOPE OF THE PROJECT

The web application filters the area and retrieves the best suitable service provider in that area and also alerts the service provider through a mailing system and also the user will be able to know the service provider who will be attending to serve the purpose.

Initially the web application consists of a web page which is dynamic in nature and takes information from the user in the form of HTML form and then with the help of a servlet program the information from the HTML page is retrieved.

The web application then will be filtering the most suitable service centre or a service provider based on the vehicle and service that has been opted for and displays the information from the available data in the database. This uses servlets and database connectivity.

This application works in some of the areas but will expand the areas in the future.

2. SYSTEM RERUIREMENT SPECIFICATIONS

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.) The SRS phase consists of two basic activities:

Problem/Requirement Analysis

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

2.1 Requirements Specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and nonfunctional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe the external behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS.

This section introduces the requirement specification document for Cancer Detection using Machine Learning which enlists functional as well as non-functional requirements.

2.1.1 Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data.

Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

The input design is the link between the information system and user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a useable form for processing can be achieved by inspecting the computer to read data from a written or printed documented or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input design considered for the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input
- Methods for preparing input validations and steps to follow when error occur.

2.1.1.1. Input design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management correct information from the computerized system.

2.1.1.2. It is achieved by creating user-friendly screens for data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulations can be performed. It also provides viewing facilities.

2.1.1.3. When the data is entered, it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

1. Enter name and location
2. Keeping records of mechanics
3. Appointments information
4. Storing feedback given by customers
5. Payment details

2.1.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

Performance:

The performance of the developed applications can be calculated by using following methods: Measuring enables you to identify how the performance of your application stands in relation to your defined performance goals and helps you to identify the bottlenecks that affect your application performance. It helps you identify whether your application is moving toward or away from your performance goals. Defining what you will measure, that is, your metrics, and defining the objectives for each metric is a critical part of your testing plan.

Performance objectives include the following:

- Response time or latency
 - Throughput
 - Resource utilization
1. Security for customer details
 2. 24*7 availability
 3. Giving locations of nearest mechanic shops within a range of 1-2 km
 4. Re-usability
 5. Reliability
 6. The system will allow notifications to be sent to users
 7. The platform will protect user data and encrypt passwords
 8. The website will be responsive

2.2 Software Requirements

- Eclipse
- MySQL
- Apache Tomcat Server
- Internet Browser

2.3 Hardware Requirements

- Processor - Intel i5 (Min 2.4 GHz)
- RAM - 8 GB
- Disk Space – 1 G

3. LITERATURE SURVEY

The process of testing a software in a well-planned and systematic way is known as software testing lifecycle (STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design
5. Test Construction and Verification
6. Test Execution and Bug Reporting
7. Final Testing and Implementation
8. Post Implementation

1. Requirements Analysis

In this phase testers analyse the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements.

It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in requirements phase rather than in future phases.

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

Usually IEEE 829 test plan template is used for test planning.

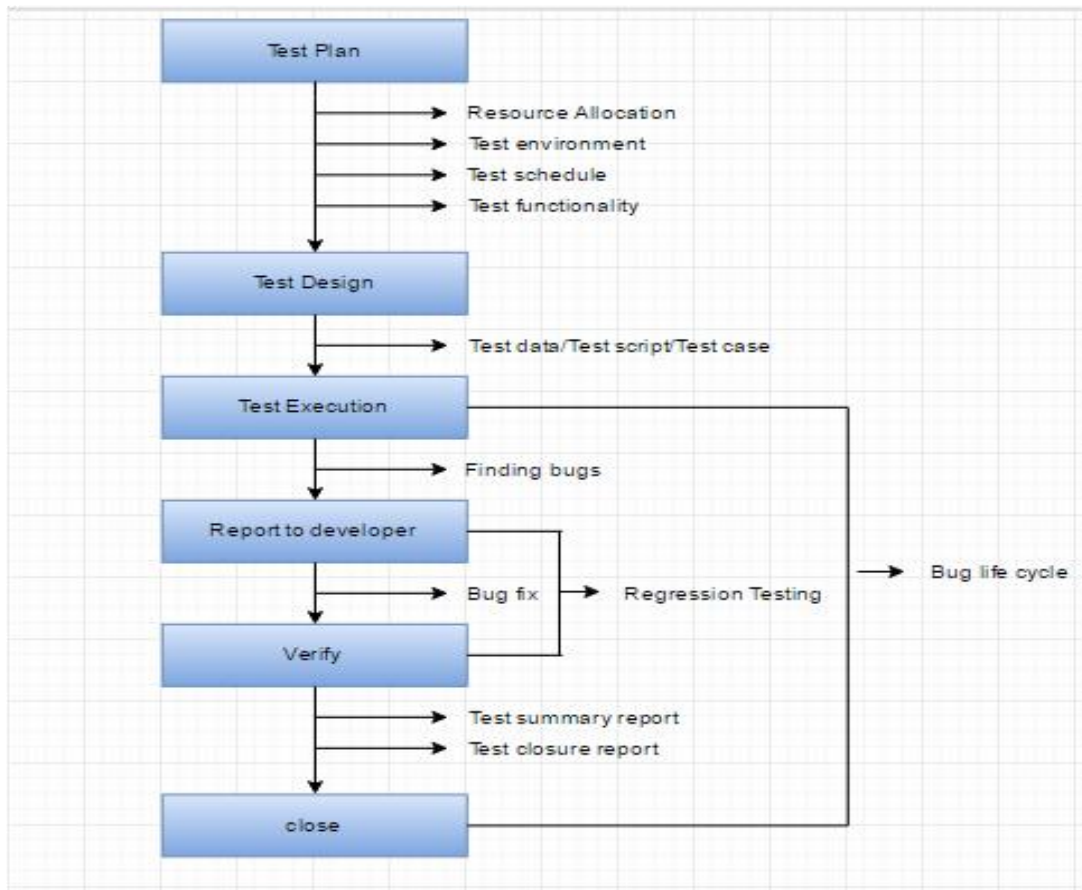


Figure 1: IEEE 829 test plan template

1. Test Analysis

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase.

Automation activities are also decided in this phase, if automation needs to be done for software product, how will the automation be done, how much time will it take to automate and which features need to be automated.

Nonfunctional testing areas (Stress and performance testing) are also analyzed and defined in this phase.

2. Test Design

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to be written in this phase.

3. Test Construction and Verification

In this phase testers prepare more test cases by keeping in mind the positive and negative scenarios, end user scenarios etc. All the test cases and automation scripts need to be completed in this phase and got reviewed by the stakeholders. The test plan document should also be finalized and verified by reviewers.

4. Test Execution and Bug Reporting

Once the unit testing is done by the developers and test team gets the test build, The test cases are executed and defects are reported in bug tracking tool, after the test execution is complete and all the defects are reported. Test execution reports are created and circulated to project stakeholders.

After developers fix the bugs raised by testers they give another build with fixes to testers, testers do re-testing and regression testing to ensure that the defect has been fixed and not affected any other areas of software.

Testing is an iterative process i.e. If defect is found and fixed, testing needs to be done after every defect fix.

After tester assures that defects have been fixed and no more critical defects remain in software the build is given for final testing.

5. Final Testing and Implementation

In this phase the final testing is done for the software, non functional testing like stress, load and performance testing are performed in this phase. The software is also verified in the production kind of environment. Final test execution reports and documents are prepared in this phase.

6. Post Implementation

In this phase the test environment is cleaned up and restored to default state, the process review meetings are done and lessons learnt are documented. A document is prepared to cope up similar problems in future releases.

Phase	Activities	Outcome
Planning	Create high level test plan	Test plan, Refined Specification
Analysis	Create detailed test plan, Functional Validation Matrix, test cases	Revised Test Plan, Functional validation matrix, test cases
Design	Test cases are revised; select which test cases to automate	Revised test cases, test data sets, sets, risk assessment sheet
Construction	Scripting of test cases to automate,	Test procedures/Scripts, Drivers, test results, Bug Reports.
Testing cycles	Complete testing cycles	Test results, Bug Reports
Final testing	Execute remaining stress and performance tests, complete documentation	Test results and different metrics on test efforts
Post implementation	Evaluate testing processes	Plan for improvement of testing process

Technologies Used

1. Java Servlets:

A Java servlet is a [Java software component](#) that extends the capabilities of a [server](#). Although servlets can respond to many types of requests, they most commonly implement [web containers](#) for hosting [web applications](#) on [web servers](#) and thus qualify as a server-side servlet [web API](#). Such web servlets are the [Java](#) counterpart to other [dynamic web content](#) technologies such as [PHP](#) and [ASP.NET](#).

Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server.

History of Java Servlets

The Java servlets API was first publicly announced at the inaugural [JavaOne](#) conference in May 1996. About two months after the announcements at the conference, the first public implementation was made available on the JavaSoft website. This was the first alpha of the Java Web Server (JWS; then known by its code name *Jeeves*) which would eventually be shipped as a product on June 5, 1997.

Importance of Servlets

- Servlet can be described in many ways, depending on the context.
- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.

Life Cycle of Servlets

Three methods are central to the life cycle of a servlet. These are `init()`, `service()` and `destroy()`. They are implemented by every servlet and are invoked at specific times by the server.

- During initialization stage of the servlet life cycle, the web container initializes the servlet instance by calling the `init()` method, passing an object implementing the `java.servlet.servletConfig` interface. This configuration object allows the servlet to access name-value initialization parameters from the web application.
- After initialization, the servlet instance can service client requests. Each request is serviced in its own separate thread. The web container calls the `service()` method of the servlet for every request. The `service()` method determines the kind of request being made and dispatches it to an appropriate method to handle the request. The developer of the servlet must provide an implementation for these methods. If a request is made for a method that is not implemented by the servlet, the method of the parent class is called, typically resulting in an error being returned to the requester.
- Finally, the web container calls the `destroy()` method that takes the servlet out of service. The `destroy()` method, like `init()`, is called only once in the lifecycle of a servlet.

Servlet Architecture

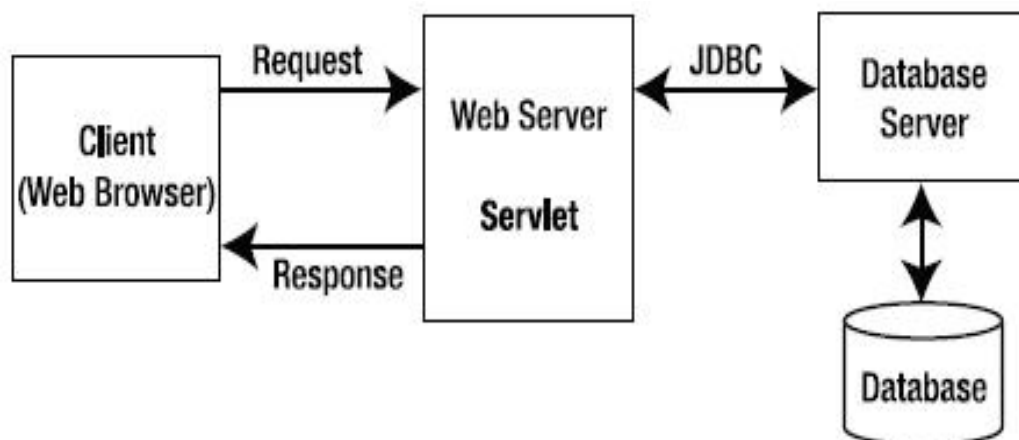


Figure 2. Servlet Architecture

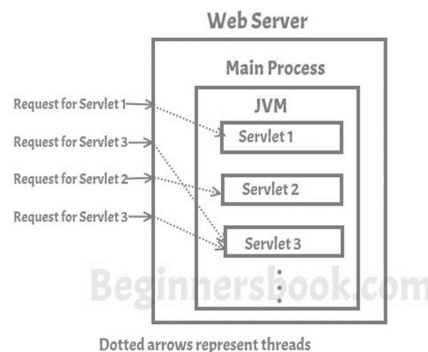
Similar Technology

CGI programs are handled by a new process every time a new request has been made. Unlike CGI, the servlet programs are handled by separate threads that can run concurrently more efficiently.

CGI program can be written in any programming language that makes it mostly platform dependent as not all programming languages are platform independent. Servlet only uses Java as programming language that makes it platform independent and portable. Another benefit of using java is that the servlet can take advantage of the object oriented programming features of java.

How Servlet Works

As I mentioned above that concurrent requests to the server are handled by threads, here is the graphical representation of the same –



Features of Servlets

1. Portable:

As I mentioned above that Servlet uses Java as a programming language, Since java is platform independent, the same holds true for servlets. For example, you can create a servlet on Windows operating system that uses GlassFish as web server and later run it on any other operating system like Unix, Linux with Apache tomcat web server, this feature makes servlet portable and this is the main advantage servlet has over CGI.

2. Efficient:

Once a servlet is deployed and loaded on a web server, it can instantly start fulfilling request of clients. The web server invokes servlet using a lightweight thread so multiple client requests can be fulfilled by servlet at the same time using the multithreading feature of Java. Compared to CGI where the server has to initiate a new process for every client request, the servlet is truly efficient and scalable.

3. Robust:

By inheriting the top features of Java (such as Garbage collection, Exception handling, Java Security Manager etc.) the servlet is less prone to memory management issues and memory leaks. This makes development of web application in servlets secure and less error prone.

Session Tracking in Servlets

Session simply means a particular interval of time. Session Tracking is a way to maintain state (data) of an user. It is also known as session management in servlet.

Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests to the server, server treats the request as the new request. So we need to maintain the state of an user to recognize to particular user. HTTP is stateless that means each request is considered as the new request.

Session Tracking Techniques

There are four techniques used in Session tracking:

1. **Cookies**
2. **Hidden Form Field**
3. **URL Rewriting**
4. **HttpSession**

Cookies in Servlet

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

Types of Cookies

1)SessionCookies:

Session cookies do not have expiration time. It lives in the browser memory. As soon as the web browser is closed this cookie gets destroyed.

2)PersistentCookies:

Unlike Session cookies they have expiration time, they are stored in the user hard drive and gets destroyed based on the expiry time.

JDBC Connection in Java

JDBC is an acronym for Java Database Connectivity. It's an advancement for ODBC (Open Database Connectivity). JDBC is an standard API specification developed in order to move data from frontend to backend. This API consists of classes and interfaces written in Java. It basically acts as an interface (not the one we use in Java) or channel between your Java program and databases i.e it establishes a link between the two so that a programmer could send data from Java code and store it in the database for future use.

Steps for connectivity between Java program and database

1. Loading the Driver

To begin with, you first need load the driver or register it before using it in the program . Registration is to be done once in your program.

2. Create the connections

3. Create a statement

Once a connection is established you can interact with the database. The JDBC Statement, Callable Statement, and Prepared Statement interfaces define the methods that enable you to send SQL commands and receive data from your database.

4. Execute the query

Now comes the most important part i.e executing the query. Query here is an SQL Query . Now we know we can have multiple types of queries. Some of them are as follows:

- Query for updating / inserting table in a database.
- Query for retrieving data.

5. Close the connections

So finally we have sent the data to the specified location and now we are at the verge of completion of our task .

By closing connection, objects of Statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

2. MySQL

MySQL is an Oracle-backed open source relational database management system ([RDBMS](#)) based on Structured Query Language ([SQL](#)). MySQL runs on virtually all platforms, including [Linux](#), [UNIX](#) and [Windows](#). Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

- MySQL is a database management system.
A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.
- MySQL databases are relational.
A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.
Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.
- The MySQL Database Server is very fast, reliable, scalable, and easy to use.
If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- MySQL Server works in client/server or embedded systems.
The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

- A large amount of contributed MySQL software is available.
MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

How MySQL works

MySQL is based on a [client-server](#) model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into separate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer. MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

Reasons to choose MySQL:

- **Secure Money Transactions**
MySQL transactions work as a single unit, which means unless and until every individual operational stage is successfully completed, the transaction is not cleared. So, if an operation fails at any stage, the entire transaction happening within that group fails. MySQL ensures that financial transactions have data integrity, so customers can make worry-free transactions online. The money is not debited until the entire process is completed and in case of failure, every process is reverted to the previous stage.
- **On-Demand Scalability**
MySQL comes with the advantage of unmatched flexibility that facilitates efficient management of deeply embedded applications, even in gigantic data centers that stack tremendous amounts of mission-critical information. It enables complete customization to cater to the unique requirements of eCommerce businesses with a much smaller footprint. MySQL provides ultimate platform flexibility to enterprises who need additional features and functionalities for their database servers.
- **High Availability**
Consistent availability is the stalwart feature of MySQL – enterprises that deploy it can enjoy round-the-clock uptime. MySQL comes with a wide variety of cluster servers and master-slave replication configurations that enable instant failover for uninterrupted access. Whether you run an eCommerce website or a high-speed processing system, MySQL is designed to process millions of queries and thousands of transactions while ensuring unique memory caches, full-text indexes and optimum speed.
- **Rock-Solid Reliability**
Protecting sensitive business information is the primary concern of every enterprise. MySQL ensures data security with exceptional data protection features. Powerful data encryption prevents unauthorized viewing of data and SSH and SSL supports ensure safer connections. It also features a powerful mechanism that restricts server access to authorized users and has the ability to block users even at the man-machine level. Finally, the data backup feature facilitates point-in-time recovery.

- **Quick-Start Capability**

You can go from software download to complete installation in just 15 minutes. MySQL is exceptionally quick, regardless of the underlying platform. It features self-management capabilities like auto restart, space expansion and automatic configuration changes for ease of management. It also comes with a comprehensive set of migration tools and a fully loaded graphical management suite. MySQL enables real-time performance monitoring for timely troubleshooting of operational issues from a single workstation.

- For all of these reasons, organizations are using MySQL to instantly develop and launch apps. From retail and finance, to healthcare and manufacturing, many industries are capitalizing on the cost-effectiveness, efficiency and reliability of MySQL to deliver seamless services and boost their revenue.

Apache Tomcat Server

Apache Tomcat is an open source Web server tool developed by the Apache Software Foundation (ASF). It is one of many Apache-related open source products used by IT professionals for various tasks and objectives.

Apache Tomcat allows the implementation of Java Servlets and JavaServer Pages (JSP) to promote an effective Java server environment. Users can also access resources for configuration and use extensible markup language (XML) to configure projects. Successive versions of Apache Tomcat have solved different problems by applying software patches and other solutions. Some experts characterize Apache Tomcat as a product offering a runtime shell for Java Servlets. Users can also set up Java virtual machines (JVM) to configure virtual hosting.

Tomcat is an [application server](#) from the Apache Software Foundation that executes Java [servlets](#) and renders Web pages that include [Java Server Page](#) coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the [Apache](#) Web site in both binary and source versions. Tomcat can be used as either a standalone product with its own internal [Web server](#) or together with other Web servers, including [Apache](#), Netscape Enterprise Server, Microsoft Internet Information Server ([IIS](#)), and Microsoft [Personal Web Server](#). Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later.

Tomcat is one of several open source collaborations that are collectively known as Jakarta. Born out of the Apache Jakarta Project, Tomcat is an application server designed to execute Java servlets and render web pages that use Java Server page coding. Accessible as either a binary or a source code version, Tomcat's been used to power a wide range of applications and websites across the Internet. At the time of writing, it's definitely one of the more popular servlet containers available.

Reasons to choose Tomcat Server

- **It's Incredibly Lightweight**

Even with JavaEE certification, Tomcat is an incredibly lightweight application. It offers only the most basic functionality necessary to run a server, meaning it provides relatively quick load and redeploy times compared to many of its peers, which are bogged down with far too many bells and whistles. This lightweight nature also allows it to enjoy a significantly faster development cycle.

- **Its an Open Sorce**
Open-source always counts as a win. Tomcat's free, and the source code for the server is readily available to anyone who'd care to download it. What this means is that – assuming you're willing to tinker with the moving parts of your server – you've got an incredible degree of freedom insofar as what you want to do with a Tomcat installation.
- **It's Highly Flexible**
Thanks to its lightweight nature and a suite of extensive, built-in customization options, Tomcat is quite flexible. You can run it in virtually any fashion you choose, and it'll still work as intended. The fact that it's open-source helps as well, since you can tweak it to fit your needs, provided you've the knowledge to do so.
- **Your Server Will Be More Stable**

Tomcat is an extremely stable platform to build on – and using it to run your applications will contribute to your server's stability, as well. This is because Tomcat runs independently of your Apache installation – even if a significant failure in Tomcat caused it to stop working, the rest of your server would run just fine.
- **It Offers An Extra Level Of Security**
Many organizations choose to position their Tomcat installation behind an extra firewall, accessible only from the Apache installation. In short, depending on how you implement your Tomcat installation, it can add an extra layer of security to your server – which is never a bad thing.

Similar Technology

GlassFish is the Open Source reference implementation for a Java EE application server. In addition to being an open source reference implementation of Java EE application server; GlassFish comes packed with core Java EE technologies such as: Servlets, Enterprise Java Beans (EJBs), Java Persistence API (JPA), JavaServer Faces (JSF), Java Message Service (JMS) as well as the default Java EE SDK.

Furthermore, in addition to being a Java EE application server, GlassFish handles EJB requests thus is also an EJB Container. Also, essentially it has its own web container (a derivative of Tomcat) and thus shares the same Catalina servlet container with Tomcat.

3. JSP

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

JSP is a server side technology of programming which is used in order to create the dynamic web pages. Like the servlets, JSP also have an access to different APIs of Java. In a java server page, the java code is inserted inside HTML content. JSP is an extended form of servlet. The page of JSP consists of the HTML tags, JSP tags and scripting elements. The JSP page is able to return the type of content including static and dynamic as the requirement in response to the request.

The static content can be in the form of HTML, XML and text.

The dynamic content can be in the form of custom tags, java beans and code of java. There is a container in the web server that is required in order to process the pages of JSP. The container of JSP basically works with the web server in order to provide the related services which this type of page requires.

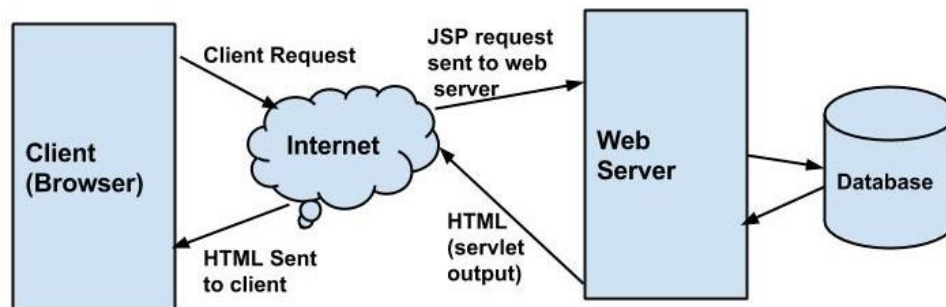


Figure 3. JSP Architecture

Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

4) Less code than Servlet

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

The Lifecycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes `jspInit()` method).
- Request processing (the container invokes `_jspService()` method).
- Destroy (the container invokes `jspDestroy()` method).

Importance:

The servlet gets invoked for each request and the full page of JSP can be translated once into a servlet in an easy manner. The pages of JSP are translated to servlets and the servlets runs when the request comes and client is not able to see anything related to JSP. In order to understand the working behind JSP, there is a need to first have a complete understanding about servlets.

The main advantage of JSP is that the static and dynamic content gets separated i.e., the static content includes the design of a webpage and the dynamic content includes the business logic which helps in proper understanding and the page can be maintained in an easy manner. JSP is easy to program and it basically provides the facility in order to develop the dynamic web pages and uses the Java programming language and it consists of scripting and different tags.

How do Java and Servlets work together?

JSP has two phases - Translation Phase and Execution Phase. Technically, a JSP is indeed converted as a Servlet at the backend during the translation phase.

JSP is used as a front end technology for displaying the content/ output to the user with its ease of use syntax mixed with HTML. A Servlet is mostly used as a controller - which as a central component decides the routing of requests - more like a receptionist picking up the phone for incoming calls and routing to the concerned person whom the actual caller wants to reach out to.

There are ways in which you can mix and match these two. Means, you can produce the HTML output in Servlet and write the business/control logic in JSP itself. However this is discouraged due to the fact the maintenance (or any changes in future) is very difficult. Hence, a better approach is suggested by using MVC - Model View Controller Architecture. It is also a design pattern which helps you to achieve a clear separation between your business logic and the presentation so that any change happened in one layer does not affect much on the other layers.

4. SYSTEM DESIGN

4.1 Introduction to UML

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

1. User Model View
 - i. This view represents the system from the users' perspective.
 - ii. The analysis representation describes a usage scenario from the end-users' perspective.
2. Structural Model View
 - i. In this model, the data and functionality are arrived from inside the system.
 - ii. This model view models the static structures.
3. Behavioural Model View It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.
4. Implementation Model View In this view, the structural and behavioural as parts of the system are represented as they are to be built.
5. Environmental Model View In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

4.2 UML Diagrams

4.2.1 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A

single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

In brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

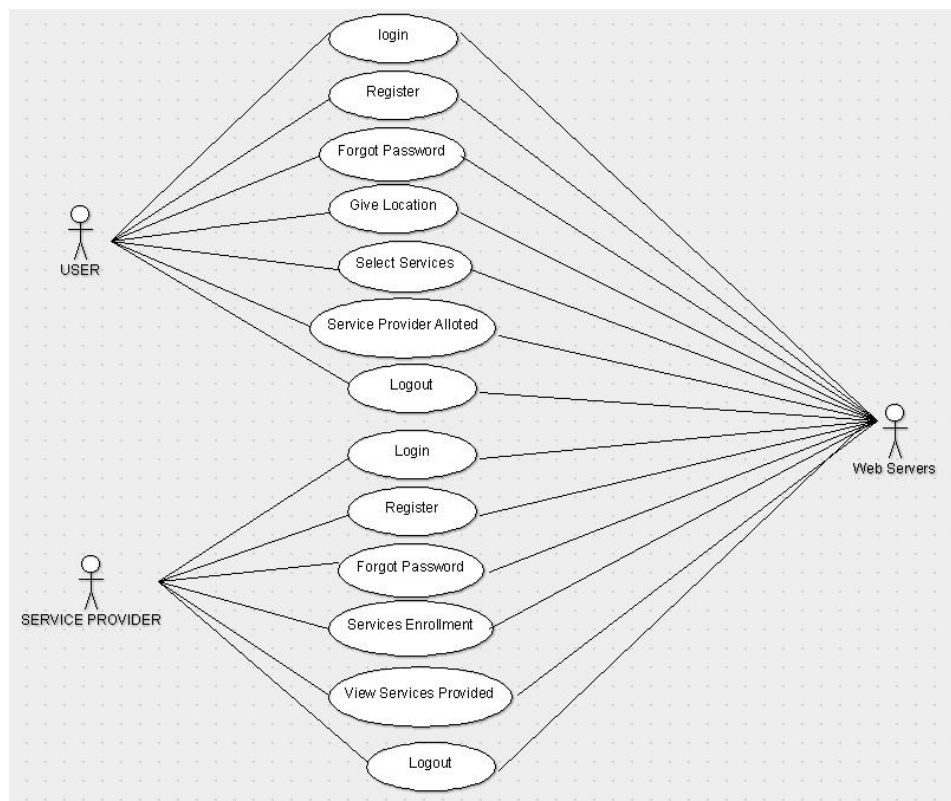


Figure 4: Use case Diagram

4.2.2 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

Basic Sequence Diagram Notations

- **Class Roles or Participants**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

- **Activation or Execution Occurrence**

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

- **Messages**

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

- **Lifelines**

Lifelines are vertical dashed lines that indicate the object's presence over time.

- **Destroying Objects**

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

- **Loops**

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

● Guards

When modelling object interactions, there will be times when a condition must be met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.

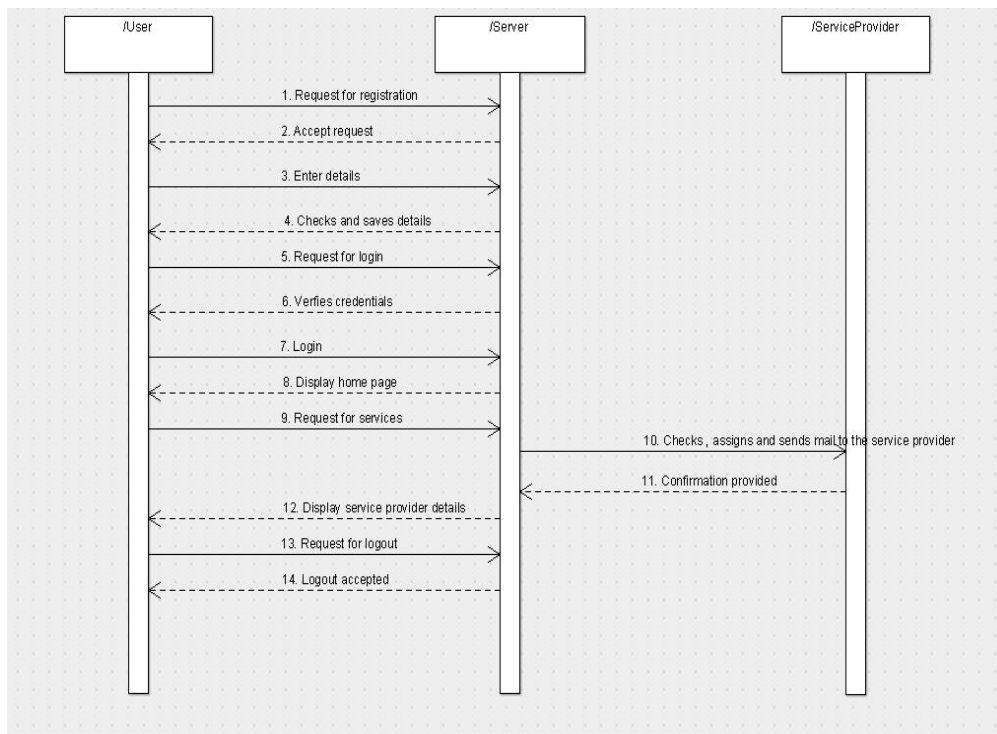


Figure 5 : Sequence Diagram -User

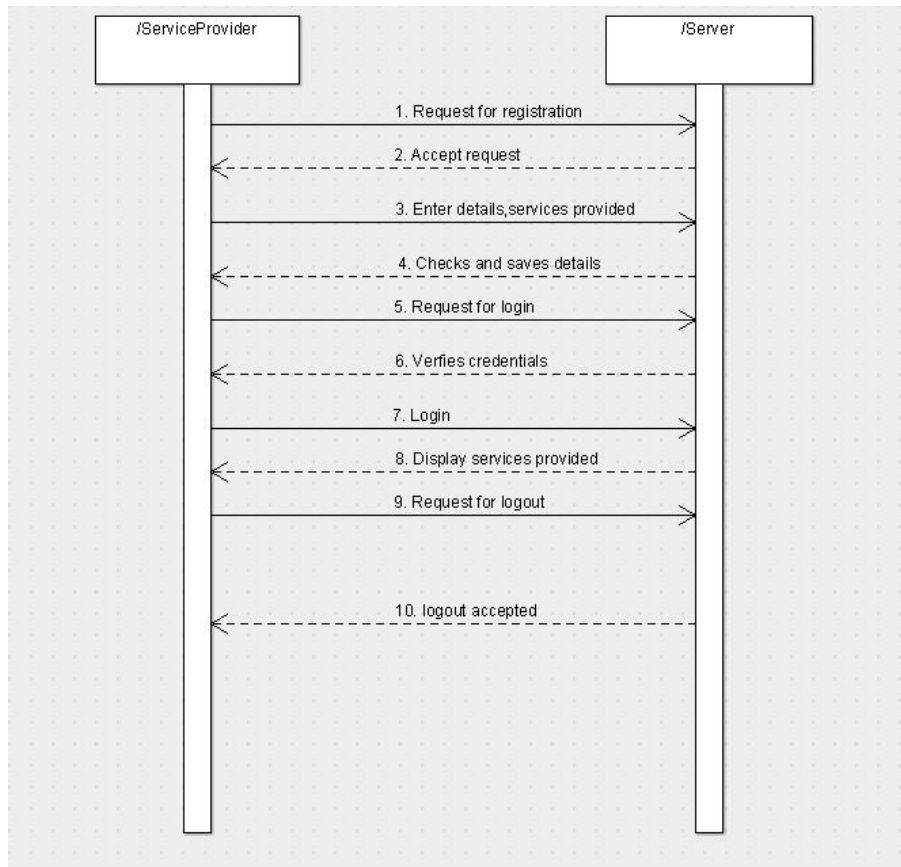


Figure 6 : Sequence Diagram - Service Provider

4.2.3 Class Diagram

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

- This is the only UML which can appropriately depict various aspects of OOPs concept.
- Proper design and analysis of application can be faster and efficient.
- It is base for deployment and component diagram.

Each class is represented by a rectangle having a subdivision of three compartments name, attributes and operation.

There are three types of modifiers which are used to decide the visibility of attributes and operations.

- + is used for public visibility (for everyone)
- # is used for protected visibility (for friend and derived)
- is used for private visibility (for only me)

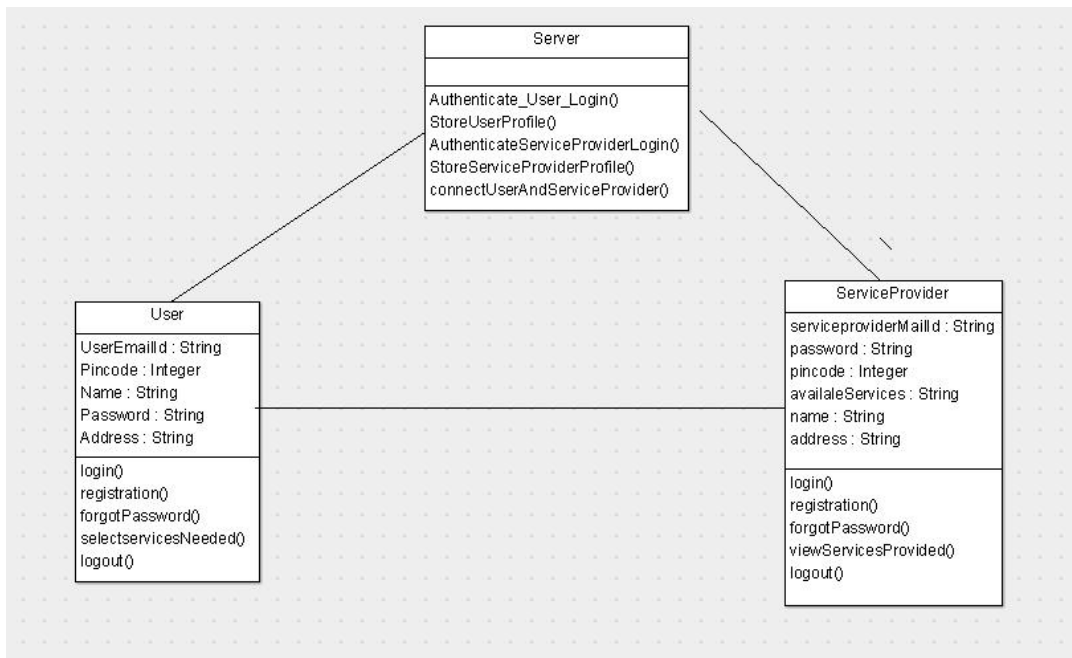


Figure 7: Class Diagram

5. IMPLEMENTATION

5.1 Pseudo Code

Step1:

In the home page, the user login, registration as well as Service provider login, registration hyper links are provided. After click on the required Field.

Step2:

Suppose If user login is selected

The required details are retrieved and authenticated with the database details, which are stored while registration process.

Step3:

- If the user Mail-Id is not found in the database, We redirect him to the User Registration Page.
- Else if the Password is incorrect we redirect the User to the Login Page.
- Else if user is correctly authenticated then we redirect user to the Home Page where he can choose what services are needed by him.

Step4:

After User has given the services needed by him. We search the Service Provider depending on the Pin-code entered by the User while logging in and also based on the services requested by the User.

Step5:

- We display the selected Service Provider details to the user.
- An E-Mail is sent to the User – Thanking him for choosing our services.
- An E-Mail is sent to the Service Provider providing the user details and the services requested by the user.

Step6:

Logout option is provided to the user

After clicking logout button, he is redirected to the Home Page.

5.2 Code Snippets

MySQL:

The Database and Tables are created.

User Table is Created as follows:

```
create database OnRoadBreakDown;
use OnRoadBreakDown;
create table user(
userId int auto_increment,
userName varchar(20),
userPswd varchar(20),
userAddress varchar(100),
userMobile varchar(20),
userMail varchar(30),
userPinCode varchar(6),
primary key(userId)
);
```

Service Provider Table is Created as follows:

```
create table serviceProvider(  
  serviceProviderId int auto_increment,  
  serviceProviderName varchar(20),  
  serviceProviderPswd varchar(20),  
  serviceProviderAddress varchar(100),  
  serviceProviderMobile varchar(20),  
  serviceProviderMail varchar(30),  
  serviceProviderPinCode varchar(6),  
  MechAsst varchar(5),  
  BatRepl varchar(5),  
  TyreRepl varchar(5),  
  VechTow varchar(5),  
  FuelRef varchar(5),  
  Servicing varchar(5),  
  primary key(serviceProviderId)  
);|
```

Transaction Table is Created as follows:

```
create table transaction(  
  transId int auto_increment,  
  serviceProviderId int,  
  serviceProviderName varchar(20),  
  userId int,  
  userName varchar(20),  
  userPincode varchar(20),  
  userMobile varchar(20),  
  primary key(transId)  
);
```

Sending Mail using mail-api dependency :

Java Class:

```
public class SendingMail {
    String otp;
    Properties emailProperties;
    Session mailSession;
    MimeMessage emailMessage;
    static {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException enfe) {
            System.out.println(enfe);
        }
    }

    private Connection getConnection() throws SQLException {
        return DriverManager.getConnection("jdbc:mysql://localhost:3306/onRoadBreakDown", "root", "root");
    }

    private void closeConnection(Connection con) {
        if (con == null) {
            return;
        }
        try {
            con.close();
        } catch (SQLException sqle) {
            sqle.printStackTrace();
        }
    }

    public void setOTP(String OTP) {
        this.otp = OTP;
    }

    public String getOTP() {
        return otp;
    }

    public void setMailServerProperties() {
        String emailPort = "587"; // gmail's smtp port
        emailProperties = System.getProperties();
        emailProperties.put("mail.smtp.port", emailPort);
        emailProperties.put("mail.smtp.auth", "true");
        emailProperties.put("mail.smtp.starttls.enable", "true");
    }

    public void createEmailMessage(String toEmail, String otp) throws AddressException, MessagingException {
        String emailSubject = "Verify this OTP";
        String emailBody = "Thank you for choosing us. This is an email sent by OnRoadBreakdown Assistance ." + otp;
        mailSession = Session.getDefaultInstance(emailProperties, null);
        emailMessage = new MimeMessage(mailSession);
        emailMessage.setSubject(emailSubject);
        emailMessage.setContent(emailBody, "text/html");
    }

    public void sendEmail() throws AddressException, MessagingException {
        String emailHost = "smtp.gmail.com";
        String fromUser = "minimajorp";
        String fromUserEmailPassword = "Kmit123$";
        Transport transport = mailSession.getTransport("smtp");
        transport.connect(emailHost, fromUser, fromUserEmailPassword);
        transport.sendMessage(emailMessage, emailMessage.getAllRecipients());
        transport.close();
        System.out.println("Email sent successfully.");
    }

    public void getMail(String mail) {
        System.out.println(mail);
    }
}
```

Login Page:

- HTML Page:

```
<head> <title>Insert title here</title></head>
<body>
    <div>
        <ul><li><a href="home.html">Home</a></li>
        <li><a href="aboutus.html">About Us</a></li>
        <li><a href="contactus.html">Contact Us</a></li>
        <li><a href="reviews.html">Reviews</a></li>
        <li><a href="faqs.html">FAQ's</a></li></ul>
        <h2>User Login :</h2>
        <form method="get" action="UserLogin">
            <div class="imgcontainer">
                
            </div>
            <div class="container">
                <center> Mail ID:<input type="text" name="userMail" placeholder="Enter Mail"></center><br>
                <center> Password:<input type="password" name="userPswd" placeholder="Enter Password"></center><br>
                <center> PinCode: <input type="text" name="userGivenPinCode" placeholder="Enter Pincode where service is needed"><br><br><br><
                <center> <button type="submit" > LOGIN</button></center>
            </div>
        </form>
        <div>
            <center> <a href="UserForgotPswd.html">Forgot Password?</a></center>
        </div>
    </div>
    <div>
        <center>Not Registered Yet!!
        <a href="UserRegistration.html">Register Here</a></center>
    </div>
</div>
</div>
</body>
</html>
```

- Servlet Code Snippet:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String userMail = request.getParameter("userMail");
    String userPswd = request.getParameter("userPswd");
    String userGivenPinCode = request.getParameter("userGivenPinCode");
    System.out.println(userMail+" "+userPswd);
    PrintWriter out = response.getWriter();
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/onRoadBreakDown", "root", "root");
        PreparedStatement pstmt = con.prepareStatement("select * from user where userMail=?");
        pstmt.setString(1, userMail);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next() == false) {
            System.out.println("First Register");
            RequestDispatcher rd = request.getRequestDispatcher("UserRegistration.html");
            rd.include(request, response);
        } else {
            String pswd = "";
            String userMobile = "";
            String userName = "";
            String userId = "";
            String userPinCode = "";
            do {
                pswd = rs.getString("userPswd");
                userMobile = rs.getString("userMobile");
                userName = rs.getString("userName");
                userId = rs.getString("userId");
                userPinCode = rs.getString("userPinCode");
            } while (rs.next());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



```

    } while (rs.next());
    if (userPswd.equals(pswd)) {
        System.out.println("Login Successful");
        HttpSession session = request.getSession();
        session.setAttribute("userMail", userMail);
        session.setAttribute("userMobile", userMobile);
        session.setAttribute("userName", userName);
        session.setAttribute("userId", userId);
        if (userGivenPinCode == null) {
            userGivenPinCode = userPinCode;
        }
        session.setAttribute("userPinCode", userGivenPinCode);
        System.out.println("pincode for searching: " + userGivenPinCode);
        RequestDispatcher rd = request.getRequestDispatcher("UserHomePage.html");
        rd.include(request, response);
    } else {
        out.println("<html><body><h2> You have entered Wrong Password, Please Login Again </body></html></h2>");
        RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
        rd.include(request, response);
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

Registration Pages:

- HTML Code:

```

<label> <b> Mail Id : </b><input type="text" name="userMail" placeholder="Enter Mail Id" id="userMail"
onkeyup="validateEmail(document.getElementById('userMail').value)" required /> </la
&nbsp;<span style="color:red;font-size:smaller" id="mailValidation"></span> <br>

<label> <b> Name : </b><input type="text" name="userName" placeholder="Enter Name" required></label> <br>
<label> <b>Phone Number : </b><input type="text" name="userMobile" placeholder="Enter phone number" required></label> <br>
<label> <b> Address : </b><textarea type="text" name="userAddress" placeholder="Enter Address" style="height:60px;width:200px" required></t

<label> <b> Pin Code : </b><input type="text" name="userPinCode" placeholder="Enter Pin Code" id="userPinCode"
onkeyup="validatePinCode(document.getElementById('userPinCode').value)" required></label>
&nbsp;<span style="color:red;font-size:smaller" id="pinCodeValidation"></span> <br>

<label> <b>Set Password : </b><input type="password" name="userPswd" placeholder="Enter Password" id="userPswd"
onkeyup="validatePassword(document.getElementById('userPswd').value)" required></label>
&nbsp;<span style="color:red;font-size:smaller" id="pswdValidation"></span> <br>

<label> <b> Re-Enter Password : </b><input type="password" name="userPswd" placeholder="Re-Enter Password" id="userPswd2" required></label>
<input style="color:blanchedalmond;font-size:large;height:40px;background-color:#0099cc;align-content:center" type="submit" name="submit"

```

- Servlet Code:

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    String userMail = request.getParameter("userMail");
    String userPswd = request.getParameter("userPswd");
    String userPswd2 = request.getParameter("userPswd2");
    String userName = request.getParameter("userName");
    String userAddress = request.getParameter("userAddress");
    String userPinCode = request.getParameter("userPinCode");
    String userMobile = request.getParameter("userMobile");
    SendingMail sm = new SendingMail();
    String generatedotp = sm.getRandom();
    System.out.println("Generated otp: " + generatedotp);
    System.out.println("Before including otp: ");
    HttpSession session = request.getSession();
    session.setAttribute("generatedotp", generatedotp);
    session.setAttribute("userMail", userMail);
    session.setAttribute("userMobile", userMobile);
    session.setAttribute("userName", userName);
    session.setAttribute("userPinCode", userPinCode);
    session.setAttribute("userPswd", userPswd);
    session.setAttribute("userAddress", userAddress);
    session.setAttribute("generatedotp", generatedotp);
    try {
        sm.setMailServerProperties();
        sm.createEmailMessage(userMail, "Your OTP is:" + generatedotp);
        sm.sendEmail();
    } catch (AddressException e) {e.printStackTrace();} catch (MessagingException e) {e.printStackTrace();}
    RequestDispatcher rd = request.getRequestDispatcher("EnterOtp.html");
    rd.forward(request, response);
}

```

OTP Verification Servlet Code:

```
String enteredotp = request.getParameter("otp");
HttpSession session = request.getSession();
String userPinCode = (String) session.getAttribute("userPinCode");
String userName = (String) session.getAttribute("userName");
String userMobile = (String) session.getAttribute("userMobile");
String userMail = (String) session.getAttribute("userMail");
String userPswd = (String) session.getAttribute("userPswd");
String userAddress = (String) session.getAttribute("userAddress");
String generatedotp = (String) session.getAttribute("generatedotp");
System.out.println(userName + " " + userMobile + " " + userMail + " " + userPswd + " " + userAddress + " " + userPinCode);
if (enteredotp.equals(generatedotp)) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/onRoadBreakDown", "root", "root");
        PreparedStatement pstmt = con.prepareStatement("insert into user(userName,userPswd,userAddress,userMobile,userMail,userPinCode) values(?,?,?,?,?,?)");
        pstmt.setString(1, userName); pstmt.setString(2, userPswd); pstmt.setString(3, userAddress);
        pstmt.setString(4, userMobile); pstmt.setString(5, userMail); pstmt.setString(6, userPinCode);
        int rs = pstmt.executeUpdate();
        if (rs > 0) {
            out.println("Registered Successfully");
            RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
            rd.forward(request, response);
        } else {
            System.out.print("Error");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    System.out.println("OTP verified - Registered Successfully");
    RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
    rd.forward(request, response);
} else {
    System.out.println("Invalid OTP");
}
```

Request Processing: User

HTML Code:

```
<center><h1>Select the services needed: </h1></center><br>
<form method="get" action="UserRequestProcessing" align="center">
<div class="margin">
<center>
<div class="b">
<h2><input type="checkbox" name="MechAsst" value="MechAsst" style="width:20px;height:20px;"> Mechanical Assistance</h2>
<h2><input type="checkbox" name="BatRepl" value="BatRepl" style="width:20px;height:20px;"> Battery Replacement</h2>
<h2><input type="checkbox" name="TyreRepl" value="TyreRepl" style="width:20px;height:20px;"> Tyre Replacement</h2>
<h2><input type="checkbox" name="VechTow" value="VechTow" style="width:20px;height:20px;"> Vehicle Towing</h2>
<h2><input type="checkbox" name="FuelRef" value="FuelRef" style="width:20px;height:20px;"> Fuel Refill</h2>
<h2><input type="checkbox" name="Servicing" value="Servicing" style="width:20px;height:20px;"> Servicing</h2><br><br>
<center> <input style="color:blanchedalmond;font-size:large;height:40px;background-color:#0099cc;align-content:center" type="submit" value="Submit" />
</div>
</center></div>
</form>
```

Servlet Code:

```
List<String> availableServiceProviders = new ArrayList<>();
while (rs.next()) {
    int flag = 0;
    for (String service : servicesSelected) {
        if (!rs.getString(service).equals("YES")) {
            flag = 1;
            break;
        }
    }
    if (flag == 0) {
        availableServiceProviders.add(rs.getInt("serviceProviderId") + "");
    }
}
```

On Road Breakdown

```
Random random = new Random();
int sizeOfList = availableServiceProviders.size();
if(sizeOfList>0) {
    int selectedSp = random.nextInt(sizeOfList);
    int selectedServiceProvider = Integer.parseInt(availableServiceProviders.get(selectedSp));
    String mailBody="Sorry for Inconvenience. We couldn't provide Service Providers near you.";
    PreparedStatement pstmt2 = con.prepareStatement("select * from serviceProvider where serviceProviderId=" + selectedServiceProvider +
    ResultSet rs2 = pstmt2.executeQuery();
    if (rs2.next()) {
        PreparedStatement pstmt3 = con.prepareStatement(
            "insert into transaction(serviceProviderId,serviceProviderName,userId,userName,userPinCode,userMobile) values(?,?,?,?,?,?)
        mailBody= "Service Provider details are: "+rs2.getString("serviceProviderName")+ " "+rs2.getString("serviceProviderMobile")+
        " "+rs2.getString("serviceProviderAddress")+ " "+rs2.getString("serviceProviderMail");

        int rs3 = pstmt3.executeUpdate();
        if(rs3>0) {
            SendingMail sm=new SendingMail();
            sm.setMailServerProperties();
            sm.createEmailMessage(userMail, mailBody);
            sm.sendEmail();
            System.out.println("Entered into Transaction table successfully");
        }
    }
}
```

Logout:

```
request.getRequestDispatcher("home.html").include(request, response);
HttpSession session=request.getSession();
session.invalidate();
out.close();
```

Service Provider Home Page:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/onRoadBreakDown", "root",
    "root");

HttpSession session = request.getSession();
int serviceProviderId = Integer.parseInt((String) session.getAttribute("serviceProviderId"));
PreparedStatement pstmt = con.prepareStatement("select * from transaction where serviceProviderId=?");
pstmt.setInt(1, serviceProviderId);
ResultSet rs = pstmt.executeQuery();
out.println("<html>");
out.println("<head>");
out.println("<title>Details of the Services Provided By You: ");
out.println("</title>");
```

JavaScript code for validating the fields:

```
<script language="Javascript">
    function validateEmail(x) {
        var ret = false;
        var error = "Please enter a valid e-mail address ";
        if (/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/ .test(x)) {
            error = " ";
            ret = true;
        }
        document.getElementById("mailValidation").innerHTML = error;
        console.log(error);
        return ret; }
</script>
```


- Validating Pin-code:

```
function validatePinCode(pincod) {  
    var len = parseInt(pincod.length);  
    var ret = false;  
    var error = "Please enter a valid PinCode ";  
    if (len == 6 && isNaN(pincod) == false) {  
        error = " ";  
        ret = true;  
    }  
    document.getElementById("pinCodeValidation").innerHTML = error;  
    return ret;  
}
```

- Validating Password:

```
function validatePassword(str) {  
    var ret = false;  
    var res = "Password Not valid( Should have 1 lowercase, 1 Uppercase, 1 special character,1 number, Min 8 characters";  
    if (str.match(/[a-z]/g) && str.match(/[A-Z]/g) && str.match(/[0-9]/g) && str.match(/^[a-zA-Z\d]/g) && str.length >= 8) {  
        res = " ";  
        ret = true;  
    }  
    document.getElementById("pswdValidation").innerHTML = res;  
    return ret;  
}
```

- Verifying Re-Entered Password:

```
function validatePasswords(pswd1, pswd2) {  
    if (pswd1 != pswd2) {  
        alert("Passwords are not matching");  
    }  
}
```

- Validating Form:

```
function validateForm() {  
    if (!validatePassword(document.getElementById("userPswd").value)  
        || !validateEmail(document.getElementById('userMail').value) ||  
        !validatePinCode(document.getElementById('userPinCode').value)) {  
        alert("Please Enter Valid Details");  
    }  
}
```

6.TESTING

6.1 Introduction to testing

The purpose of testing is to detect error. Testing is the system of seeking to observe each possible fault or weak point in a piece product. It presents a technique to determine the functionality of accessories, sub-assemblies, assemblies and/or a finished product.

It is the approach of exercising program with the intent of ensuring that the application system meets its standards and consumer expectations and does now not fail in an unacceptable method. There are quite a lot of types of test. Each experiment style addresses a precise trying out requirement.

Testing is one of the principal phases within the application development pastime. In program development lifestyles cycle (SDLC), the predominant aim of Testing process is the satisfactory; the progress application is demonstrated against achieving the specified performance and efficiency.

For the duration of the trying out procedure the program is worked with some detailed scan cases and the output of the test circumstances are analyzed whether the application is working in step with the expectations or not. The success of the Testing system in settling on the errors in general relies on the experiment case standards, for trying out any program we have to have an outline of the anticipated habits of the method and process of settling on whether or not the found habits confirmed to the expected conduct.

LEVELS OF TESTING

When you consider that the mistakes within the application can be injured at any stage. So, we have got to carry out the testing process at specific stages for the duration of the development. The elemental stages of trying out are unit, Integration, method and acceptance trying out.

The unit testing is carried out on coding. Here exclusive modules are proven against the specification produced during design for modules.

In case of Integration Testing exceptional verified modules are combined into sub systems and established in case of the procedure Testing the entire application is demonstrated and within the subsequent level of trying out the method is established with person requirement file prepared for the duration of SRS.

There are two basic approaches for testing. They are,

Functional Testing

In functional testing circumstances are decided solely and the groundwork of necessities of the application or module and the Internals of the software or modules should not regarded for determination of experiment cases. That is often known as black box testing.

Structural Testing

In structural Testing scan circumstances are generated on actual code of the application or module to be demonstrated. This is called white field Testing.

6.2 TESTING ACTIVITIES

Different levels of testing are used in the testing process, each level of testing aims to test different aspects of the system. The basic levels are:

I. Unit Testing:

Unit testing entails the design of test circumstances that validate that the internal application common sense is functioning correctly, and that application inputs produce legitimate outputs. All decision branches and glide of inner code should be validated. It is the Testing of individual software items of the applying its carried out after the completion of an character unit earlier than integration.

It is a kind of structural Testing which depends on advantage of its construction and is invasive. Unit assessments perform normal checks at element stage and experiment a certain industry process, utility, and/or procedure configuration. Unit assessments ensure that every certain direction of a industry system performs properly to the documented requisites and contains certainly outlined inputs and expected outcome.

II. Integration Testing:

Integration exams are designed to experiment integrated application components to assess if they in reality run as one software. Testing is occasion driven and is more worried with the elemental outcome of screens or fields. Integration exams show that even though the accessories have been individually delight, as proven by means of efficaciously unit testing, the blend of accessories is right and consistent.

Integration testing is chiefly aimed at exposing the problems that arise from the combination of add-ons.

III. System Testing:

In system testing the entire software is tested. The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

IV .Acceptance Testing:

User Acceptance Testing out is a primary section of any undertaking and requires big participation with the aid of the end consumer. It also ensures that the approach meets the functional specifications.

6.3 TYPES OF TESTING

Black Box Testing

A software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

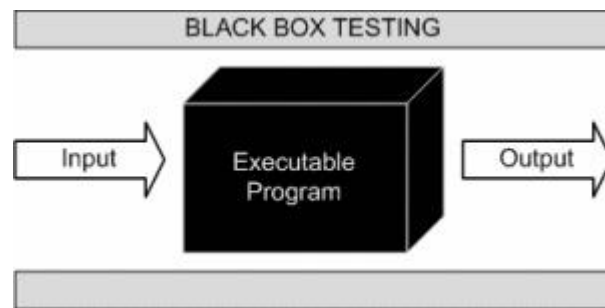


Figure 8:Black Box Testing

White Box Testing

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

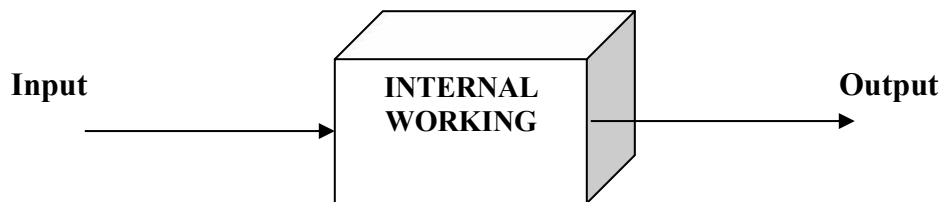


Figure 9: White Box Testing

6.4 TEST PLAN

Experiment plan is common file for entire task, which defines the scope, method to be taken and the individual accountable for extraordinary routine of testing. The inputs for forming scan aircraft are

Challenge plan

Requirement file

Process design

Test case specification: although there is one scan plan for whole mission scan cases have to be exact separately for every test case. Scan case specification gives for each item to be confirmed all scan instances and outputs expected for those scan cases.

Testing Process

A quantity of routine ought to be carried out for Testing software. Trying out begins with experiment plan. Experiment plan identifies all testing associated hobbies that needed to be performed together with the agenda and consultant traces for testing.

The plan also specifies the phases of testing that have got to be completed, with the aid of deciding upon the exceptional trying out units. For each unit specified within the plan first the scan instances and studies are produced.

Test strategy and approach

Field testing can be performed manually and sensible assessments will likely be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
-

Features to be tested

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

6.5 TEST CASES

A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Types of test cases

There are two types of test cases as mentioned below:

1. **Formal test cases:** Formal test cases are those test cases which are authored as per the test case format. It has all the information like preconditions, input data, output data, post conditions, etc. It has a defined set of inputs which will provide the expected output.
2. **Informal test cases:** Informal test cases are authored for such requirements where the exact input and output are not known. In order to test them the formal test cases are not authored but the activities done and the outcomes are reported once the tests are run.

Test Cases Examples:

◆ **Module: User Login**

◆ **File Name: Login.java**

Test Case	Input	Expected Output	Actual Output	Description
Valid login	username, password, type	Success	Success	Login successful.
Invalid Login	username, password, type	Failed	Failed	Login unsuccessful. Try again.

◆ **Module: User registration**

◆ **Filename:Register.java**

Test Case	Input	Expected Output	Actual Output	Description
Register new User	User Info.	Register successful.	Register successful.	User registered.
Register new User	User Info.	Failed.	Failed.	Invalid Data.

VALIDATING THE FIELDS WHILE REGISTERING

Mail Id :

Name :

Phone Number :

Address :

Pin Code :

Please enter a valid PinCode

Set Password :

have 1 lowercase, 1 Uppercase, 1 special character, 1 number, Min 8 characters

Mail Id :

Name :

Phone Number :

Address :

Pin Code :

Set Password :

Password Not valid(Should have 1 lowercase, 1 Uppercase, 1 special character, 1 number, Min 8 characters


Re-Enter Password :

On Road Breakdown

localhost:8080 says

Passwords are not matching

OK



REGISTRATION

Mail Id :

Name :

Phone Number :


Address :

Pin Code :

Set Password :

Re-Enter Password :

Register Here!!!



REGISTRATION

Mail Id :

Name :

Phone Number :

Address :

Pin Code :

Set Password :

Re-Enter Password :

On Road Breakdown

Mail Id :

Please enter a valid e-mail address

Name :

Phone Number :

Address :

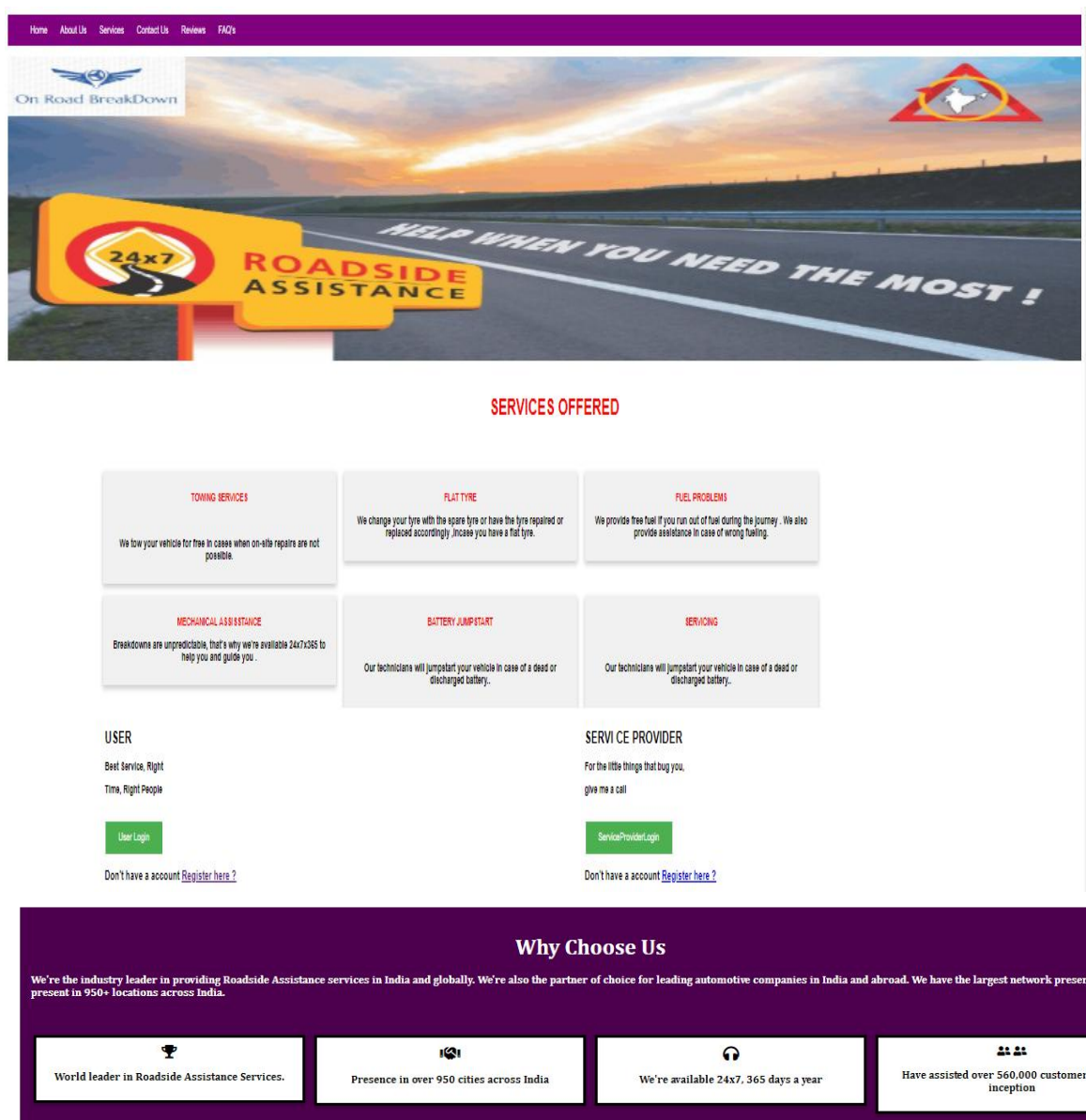
Pin Code :

Set Password :

Re-Enter Password :


7. SCREENSHOTS

7.1 Home Page



7.2 Contact Us Page

[Home](#) [About Us](#) [Services](#) [Contact Us](#) [Reviews](#) [FAQs](#)



Need Road Assistance? Give Us A Call

Location Information

Our Address
121,NAGOLE "X" Road
NAGOLE, Near SUNSHINE HOSPITAL
HYDERABAD
tel: +91-8970 6750
tel: +91-7763 8890
Email: admin@onroadbreakdown.com

Send us your Feedback

First Name

Last Name

Contact

Area code

Tel. number

Email

May We contact you?

Tel

Your Feedback

Our Address
121,NAGOLE "X" Road
NAGOLE, Near SUNSHINE HOSPITAL
HYDERABAD
tel: +91-8970 6750
tel: +91-4321 8765
Email: admin@onroadbreakdown.com

7.3 About Us Page

[Home](#) [About Us](#) [Contact Us](#) [Reviews](#) [FAQ's](#)



With you on every road

OVERVIEW

On Road BreakDown commenced operations in Hyderabad in 2017. We provide services to a number of customers and have many service providers serving hundreds of beneficiaries.

Helping people anytime, anywhere is the essence that drives us and has always remained at the heart of it.

In India, while Road Side Assistance is one of the core business, with service available 24/7, besides offering direct services to consumer, they can also customize them. The company assists in creating and implementing tailor-made programs to maximise customer satisfaction and loyalty. Committed to our partners, we focus on building customer relationships that continue to enhance their brand value.

relationships that continue to enhance their brand value.

OUR HISTORY



Our story of helping people began 2 years ago and has never ceased. It all began in 2017 in Hyderabad when the 4 of us were on a road trip to Goa and on the way we had a BreakDown. We looked out for help but couldn't find any. At the very moment we thought of doing something to overcome such problems in the future. As soon as we came back from the trip we started working on it. The result of which is "ON ROAD BREAKDOWN". As the world and people's needs within the world is evolving, our ways of helping people have also transformed. Today, we reach out to hundreds of beneficiaries.

BreakDown. We looked out for help but couldn't find any. At the very moment we thought of doing something to overcome such problems in the future. As soon as we came back from the trip we started working on it. The result of which is "ON ROAD BREAKDOWN". As the world and people's needs within the world is evolving, our ways of helping people have also transformed. Today, we reach out to hundreds of beneficiaries.

SERVICE INFRASTRUCTURE



We have proven credibility in managing complex networks of high quality service providers to deliver the best services each and every time. We connect them with customers in need, wherever they are. We stand unparalleled with a huge fleet of trucks that are dedicated to get you out of the worst conditions. We're the partner of choice for thousands of people when it comes to providing roadside assistance services. As veterans in the service, we understand consumers and respond intuitively providing apt assistance, thus

7.4 Review Page

[Home](#) [About Us](#) [Services](#) [Contact Us](#) [Reviews](#) [FAQ's](#)



WHAT OUR CUSTOMER THINKS


 MARTIN

Very pleased.
Mechanic arrived sooner than expected. Very pleased with the service I received. Would definitely recommend...

 MISS J PYM

7.5 FAQ Page

[Home](#) [About Us](#) [Services](#) [Contact Us](#) [Reviews](#) [FAQ's](#)



What is Roadside Assistance and other FAQs

***What is Roadside Assistance?**

Roadside assistance is a service provided by Allianz Global Assistance which assists you in situations where you need help on the road with your vehicle. The service offers many benefits from getting your vehicle fixed on the spot to towing or taxi service to help you reach your destination.


***Would you be providing assistance even if some other member of the family is driving the car?**

We provide roadside assistance to any person who is driving the car, as long as the car is covered. There may be an authorisation request call for service to the owner for confirmation in case help is requested by another member.

7.6 User Registration Page

[Home](#) [About Us](#) [Services](#) [Contact Us](#) [Reviews](#) [FAQs](#)

Register Here!!!



Mail Id :

Name :

Phone Number :


Address :

Pin Code :

Set Password :

Re-Enter Password :

7.7 User Login Page



Mail ID:

Password:

PinCode Where service is Needed:

[Forgot Password?](#)

Not Registered Yet!! [Register Here](#)

7.8 Service Provider Register Page

Mail Id :

Name :

Phone Number :

Address :

Pin Code :

Set Password :

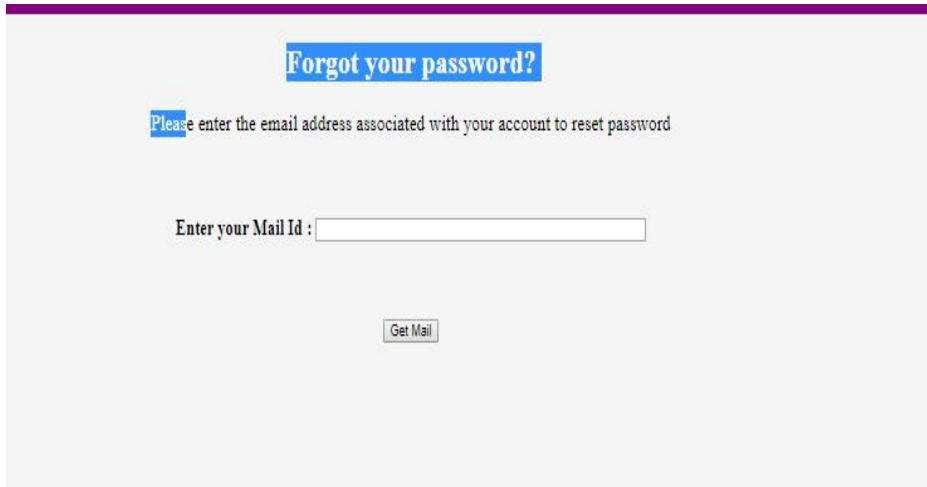
Re-Enter Password :

Services You Provide

- ☐ Mechanical Assistance
- ☐ Battery Replacement
- ☐ Tyre Replacement
- ☐ Vechicel Towing
- ☐ Fuel Refill
- ☐ Servicing

Register

7.9 Forgot Password Page

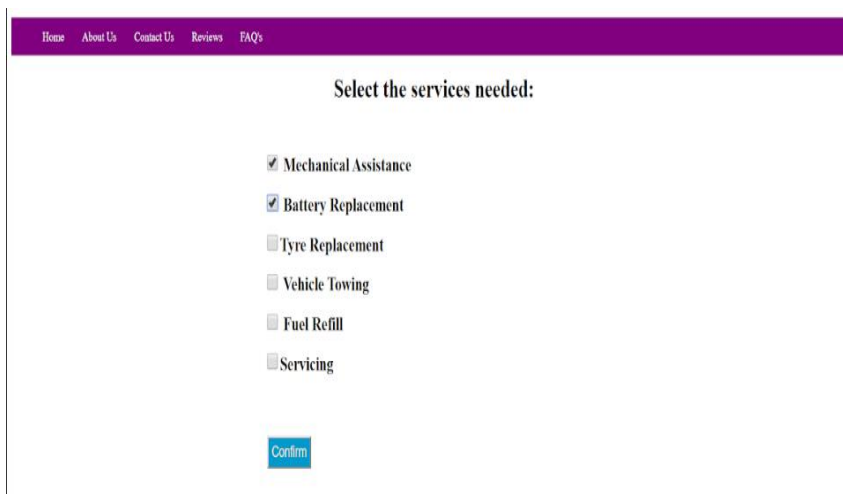


Forgot your password?

Please enter the email address associated with your account to reset password

Enter your Mail Id :

7.10 User Login



Home About Us Contact Us Reviews FAQs

Select the services needed:

- ☒ Mechanical Assistance
- ☒ Battery Replacement
- ☐ Tyre Replacement
- ☐ Vehicle Towing
- ☐ Fuel Refill
- ☐ Servicing

7.11 Service Provider Details

[Home](#) [About Us](#) [Contact Us](#) [Reviews](#) [FAQ's](#)

The Details of the Requested Service Provider

Name : Manasa

Mail id : sonyreddy2114@gmail.com


Mobile Number : 87481658951

Pin Code : 500068

Address : Nagole

[Logout](#)

ServiceProvider Login :



Mail ID:

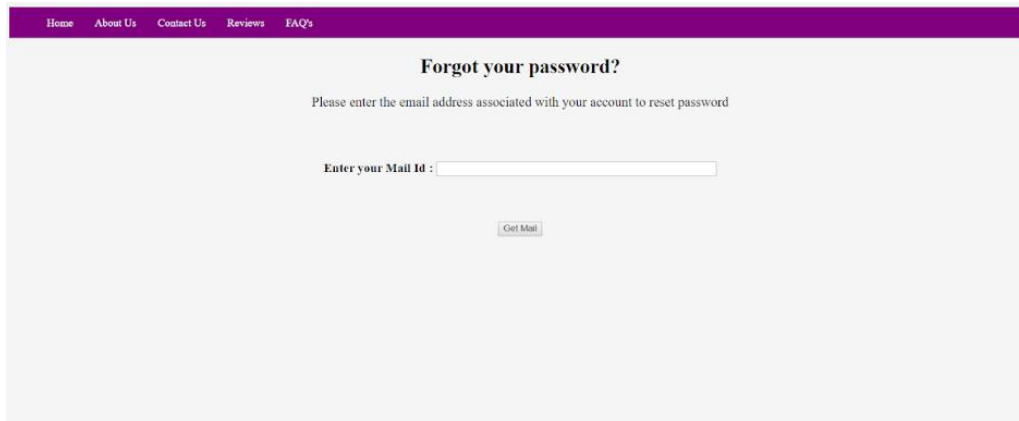
Password:

[LOGIN](#)

[Forgot Password?](#)

Not Registered Yet!! [Register Here](#)

7.12 Forgot Password Page



Home About Us Contact Us Reviews FAQ's

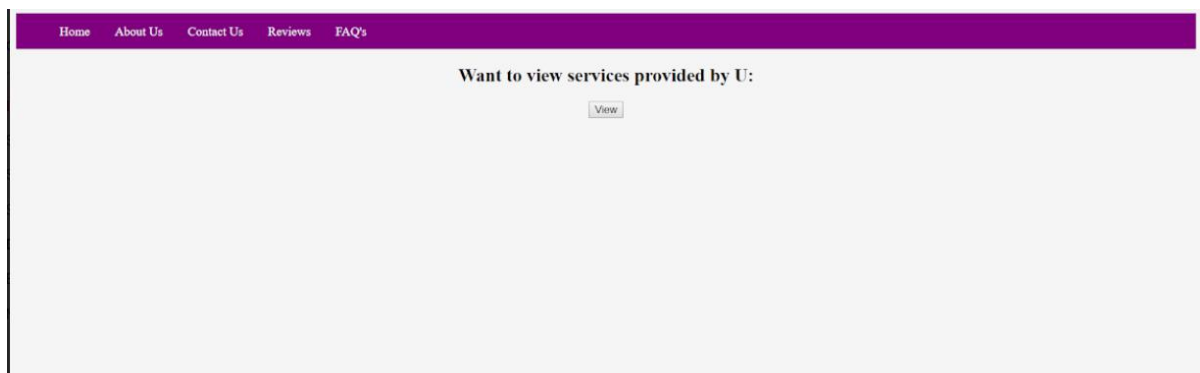
Forgot your password?

Please enter the email address associated with your account to reset password

Enter your Mail Id :

[Get Mail](#)

7.13 Service Provider Home Page



Home About Us Contact Us Reviews FAQ's

Want to view services provided by U:

[View](#)

7.14 Service Provider Transaction Table

[Home](#)
[About Us](#)
[Contact Us](#)
[Reviews](#)
[FAQ's](#)

Details of the Services Provided By You:

User Name	User Mobile Number	Pin Code Of the area
manasa	9490297756	500068
manasa	9490297756	500068
manasa	9490297756	500068

Logout

CONCLUSION

Breakdown of cars or any vehicle can be troublesome and can create a situation of panic any time. As such situations cannot be avoided completely, measures can be taken to act when a situation like that arises. One such attempt is this web application.

The web application can be helpful to a great extent whenever such a situation arises by providing a information without any research done by the user and thus avoiding a situation of panic. The web application is helpful whenever a user is in search of data regarding service providers and also can come in contact with the service provider. Since the web application verifies the identity and authenticity of the service provider and the user as well, both the parties (the user and the service provider) have authentic information and hence they avoid any sort of time waste which arises due to wrong data or contact details.

Normally when such a situation arises the user goes to the web and googles the information of the various service providers in that particular area but having applications like the On Road Break Down web application reduces the work of the user and avoids any possible confusion.

FUTURE ENHANCEMENTS

The web application currently provides an interface between the user and the service provider wherein the user is suggested a service provider depending on the location information (i.e. the pin code) and the services needed by the user. The web application is currently capable of selecting a service provider for the user, displaying the users for a particular service provider, sending mails to the service provider and password recovery.

In the future we would like to make the web application capable of taking the location of the user with the help of a GPS provide the service providers in that area depending and the services selected. We also intend to make communication between the user and the service provider stronger by enabling an SMS system where the details of the user will be messaged to the service provider.

We also intend to develop an android application with the above mentioned improvements for easier usage and convenience of the user.

REFERENCES

1. <https://www.javatpoint.com/servlet-tutorial>
2. <https://www.geeksforgeeks.org/introduction-java-servlets/>
3. <https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html>
4. https://www.w3schools.com/html/html_css.asp
5. <https://www.w3.org/Style/Examples/011/firstcss.en.html>
6. <https://www.tutorialspoint.com/jdbc/jdbc-db-connections.html>
7. <https://www.codejava.net/java-ee/jsp/sending-e-mail-with-jsp-servlet-and-javamail###targetText=Code%20Java%20Servlet%20for%20sending%20e%2Dmail&targetRead%20SMTP%20server%20settings%20from,a%20response%20to%20the%20user.>
8. https://docstore.mik.ua/orelly/java-ent/servlet/ch13_02.html
9. <https://www.mysql.com/>
10. <https://www.tutorialspoint.com/mysql/index.html>
11. <https://stackoverflow.com/>
12. https://en.wikipedia.org/wiki/Roadside_assistance