**1.INTRODUCTION**

* 1. **Purpose**

The purpose of this document is to systematically capture requirements for the project and the system to be developed. The document also captures the functional requirements and serves as an input for the scope of project.

* 1. **Scope**

This product is a powerful web aggregation engine is a core enabling technology of Cognizant solutions and is a fundamental tool for improving the management of collections and deductions. This product is a great start to an efficiency initiative

* 1. **Aim**
* Project is about the mechanic and customer login in which there would be two user using the web.
* This project is about On-Road Assistance where the customer end would login to share the problems related to him/her like tyre puncture, need of petrol, accident or some basic needs related to the vehicle. And the customer would send request to the local mechanics present over that arena.
* Similarly, there would be mechanic end login where the mechanic would either accept or reject the request send by the customer.
* This accretion or rejection by the mechanic would be further notified to the customer end.
  1. **Objectives**
* Customer and Mechanic registration and credential authentication.
* Requests for service by customer
* Location based searching of mechanics.
* Accepting requests made by customer and providing service.
* Customer will also provide rating to the mechanic on the basis of the service.
* Real time navigation.
  1. **Methodology**

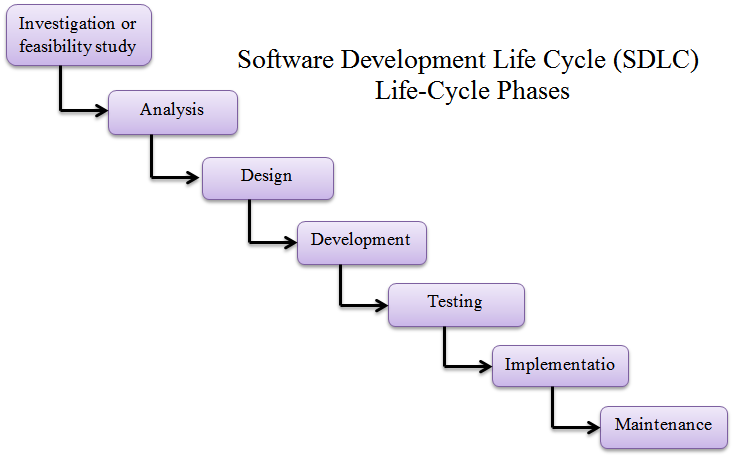
**SDLC**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software development process is the process of dividing [software development](https://en.wikipedia.org/wiki/Software_development) work into distinct phases to improve [design](https://en.wikipedia.org/wiki/Software_design), [product management](https://en.wikipedia.org/wiki/Software_product_management), and [project management](https://en.wikipedia.org/wiki/Software_project_management). It is also known as a software development life cycle.

The methodology may include the pre-definition of specific [deliverables](https://en.wikipedia.org/wiki/Deliverable) and artifacts that are created and completed by a project team to develop or maintain an application.

Most modern development processes can be vaguely described as [agile](https://en.wikipedia.org/wiki/Agile_software_development). Other methodologies include [waterfall](https://en.wikipedia.org/wiki/Waterfall_model), [prototyping](https://en.wikipedia.org/wiki/Software_prototyping), [iterative and incremental development](https://en.wikipedia.org/wiki/Iterative_and_incremental_development), [spiral development](https://en.wikipedia.org/wiki/Spiral_development), [rapid application development](https://en.wikipedia.org/wiki/Rapid_application_development), and [extreme programming](https://en.wikipedia.org/wiki/Extreme_programming).

Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization.



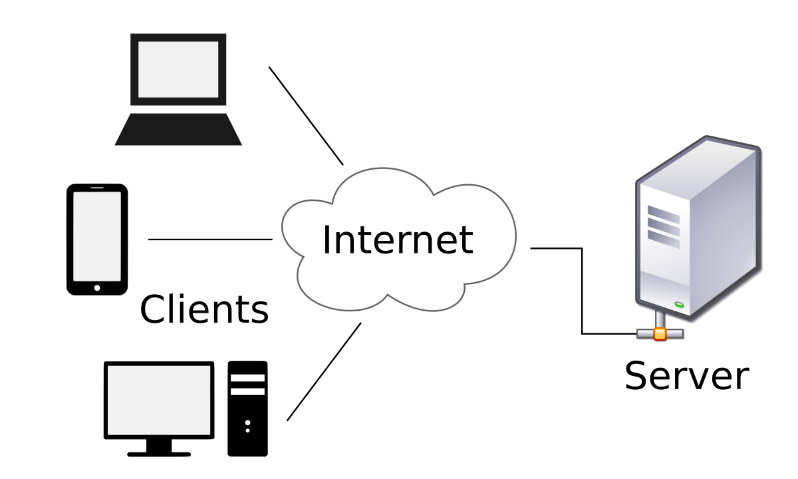
**Fig1: SDLC Life-Cycle Phases**

**Client-Server Model**

Client–server model is a [distributed application](https://en.wikipedia.org/wiki/Distributed_application) structure that partitions tasks or workloads between the providers of a resource or service, called [servers](https://en.wikipedia.org/wiki/Server_(computing)), and service requesters, called [clients](https://en.wikipedia.org/wiki/Client_(computing)).

Often clients and servers communicate over a [computer network](https://en.wikipedia.org/wiki/Computer_network) on separate hardware, but both client and server may reside in the same system.

A server [host](https://en.wikipedia.org/wiki/Host_(network)) runs one or more server programs which share their resources with clients.A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.



**Fig2: Client-Server Architecture**

**DAO Design Pattern**

Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services. Following are the participants in Data Access Object Pattern.

* Data Access Object Interface - This interface defines the standard operations to be performed on a model object(s).
* Data Access Object concrete class - This class implements above interface. This class is responsible to get data from a data source which can be database / xml or any other storage mechanism.
* Model Object or Value Object - This object is simple POJO containing get/set methods to store data retrieved using DAO class.

**Advantages of DAO pattern**

There are many advantages for using DAO pattern. Let’s state some of them here:

1. While changing a persistence mechanism, service layer doesn’t even have to know where the data comes from. For example, if you’re thinking of shifting from using MySQL to MongoDB, all changes are needed to be done in the DAO layer only.

2. DAO pattern emphasis on the low coupling between different components of an application. So, the View layer have no dependency on DAO layer and only Service layer depends on it, even that with the interfaces and not from concrete implementation.

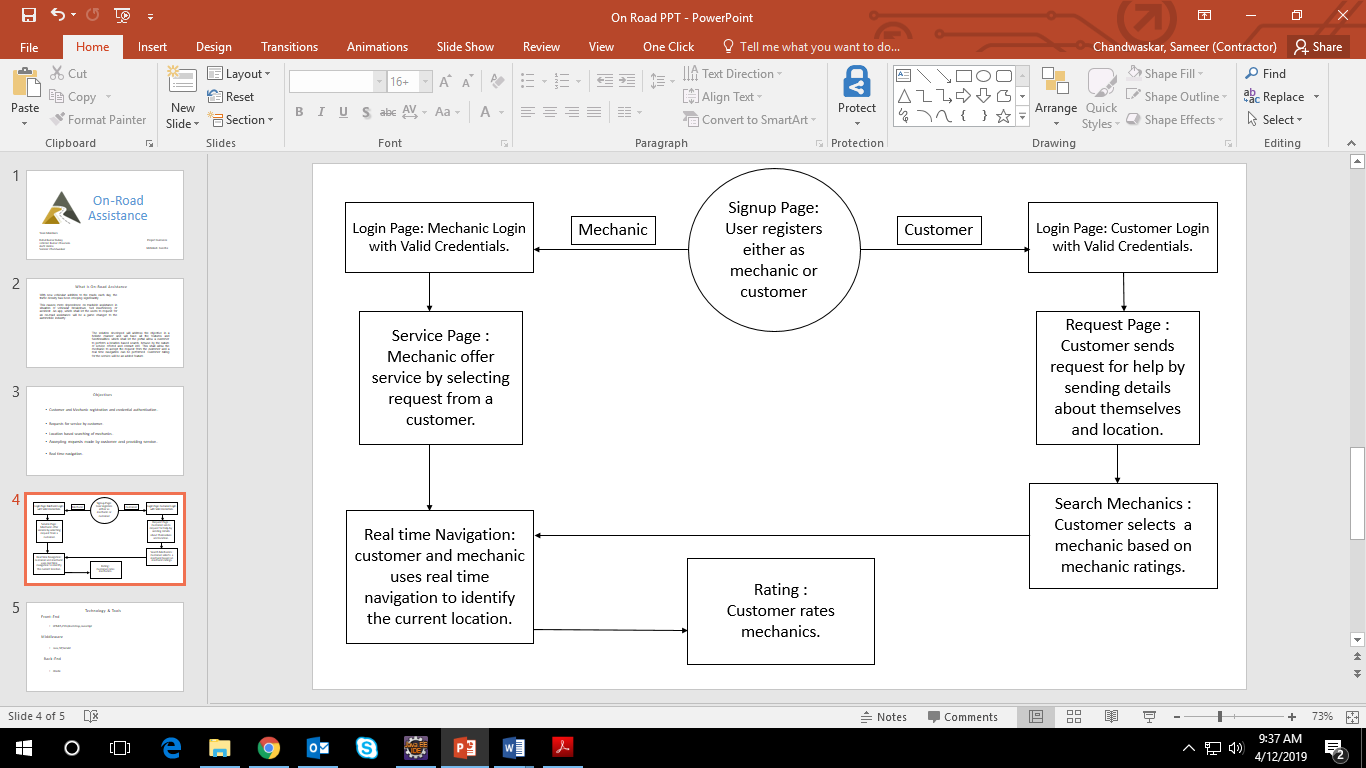
3. As the persistence logic is completely separate, it is much easier to write Unit tests for individual components. For example, if you’re using JUnit and Mockito for testing frameworks, it will be easy to mock the individual components of your application.

4. As we work with interfaces in DAO pattern, it also emphasizes the style of “work with interfaces instead of implementation” which is an excellent [OOPs](https://www.journaldev.com/12496/oops-concepts-java-example) style of programming.

**2.REQUIREMENTS**

**2.1 Process Architecture**

The below diagram shows the architecture of process involved in smooth execution of application



**Fig3: Process Architecture Diagram**

**2.2 Business Requirements**

Primary focus is to complete developing the critical requirements and then to proceed with the remaining requirements.

**2.2.1 High Level Requirements**

|  |  |  |
| --- | --- | --- |
| **Req\_ID** | **Short Description** | **Detail** |
| 1 | Customer and Mechanic Registration | Ability of the system to procure the fundamental details of the customer and mechanic |
| 2 | Customer and Mechanic Authentication | Ability of the system to authenticate the credentials of the registered customer and mechanic. |
| 3 | Customer Requests | Ability of system to submit requests to mechanic. |
| 4 | Accepting Requests  ( Mechanic Module ) | Ability of system to enable mechanics to accepts requests from customer |
| 5 | Real time navigation | Ability of system to enable mechanic and customer to reach out through real time navigation. |
| 6 | Mechanic Rating | Ability of system to enable customer to rate mechanic. |

**T1: High Level Requirement**

**2.2.2 Detailed Functional Requirements**

Additionally, the following elements are captured for each business requirement in the table provided below: -

\* Req. Type = (F Core Functionality, E Exception, UI User Interface, R Reporting)

\*\* Priority of Requirement = (1=Base Functionality, 2=Advanced Functionality,

3=Additional Opportunities)

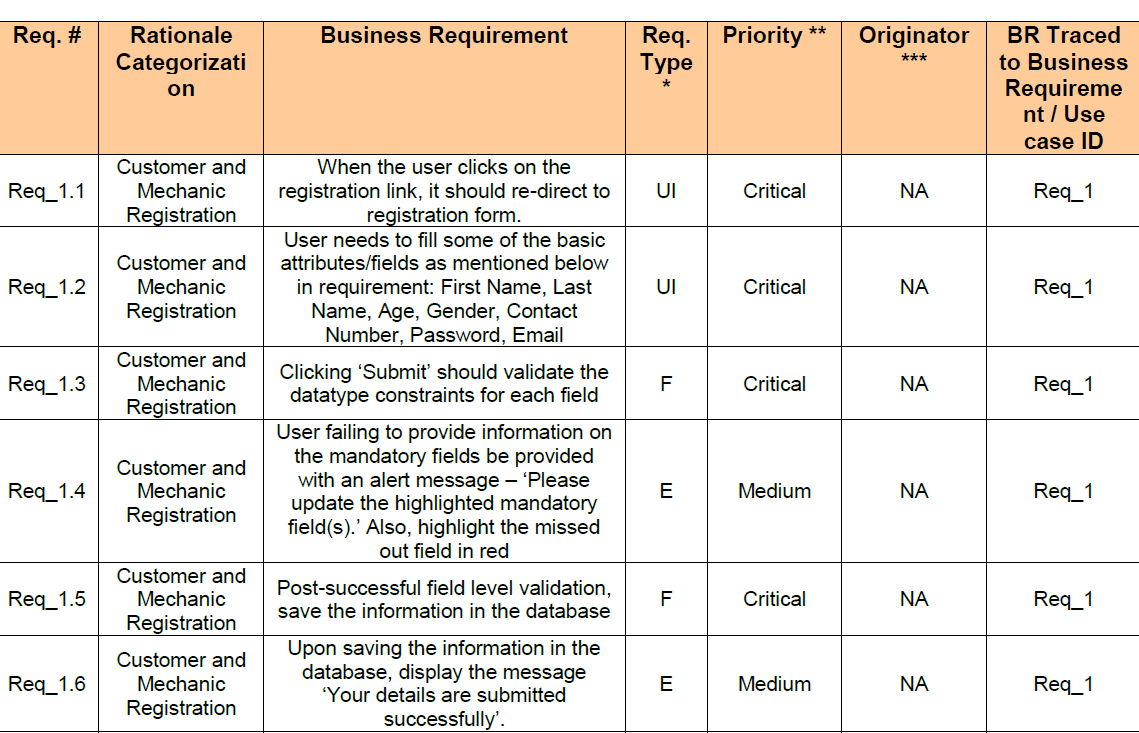
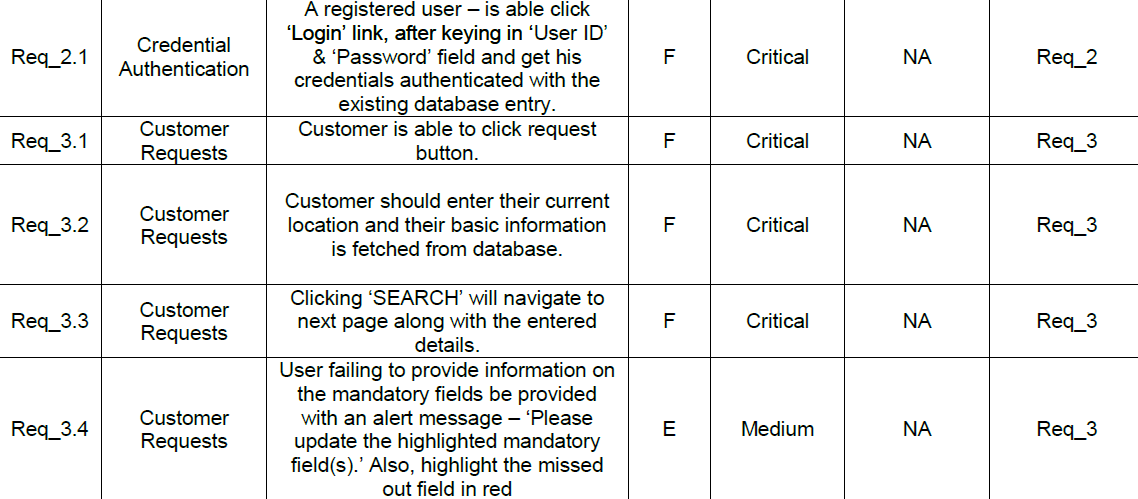
\*\* Originator = (Name of the business process of the system/ department or function

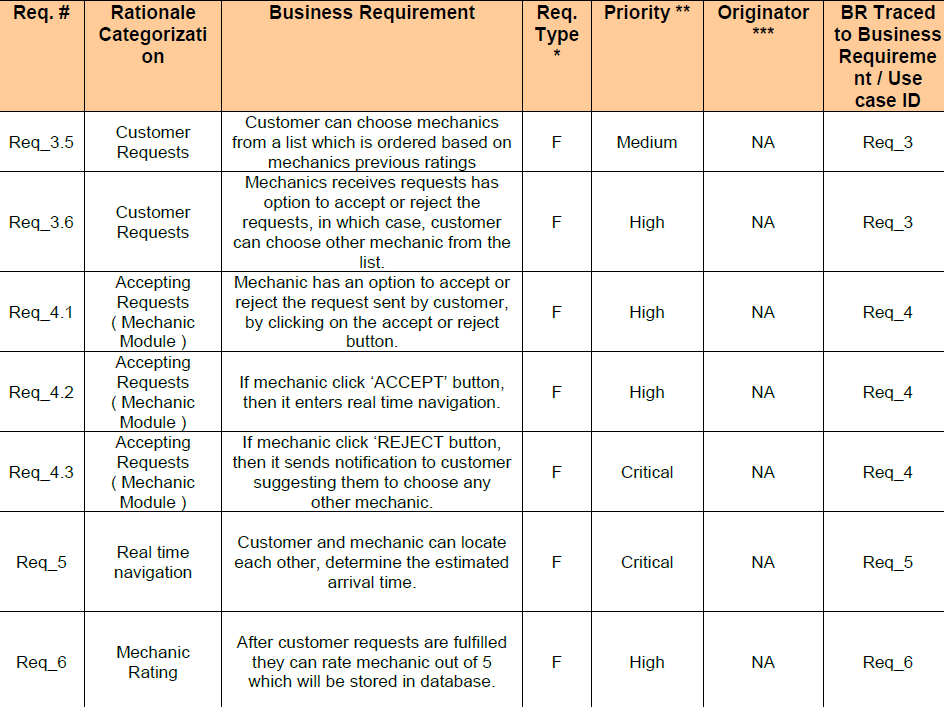
name in the customer organization)

The Requirements in this document are prioritized as follows:

|  |  |  |
| --- | --- | --- |
| **Value** | **Rating** | **Description** |
| 1 | Critical | This requirement is critical to the success of the project. The project will not be possible without this requirement. |
| 2 | High | This requirement is high priority, but the project can be implemented at a bare minimum without this requirement. |
| 3 | Medium | This requirement is somewhat important, as it provides some value but the project can proceed without it. |
| 4 | Low | This is a low priority requirement, or a “nice to have” feature, if time and cost allow it. |
| 5 | Future | This requirement is out of scope for this project, and has been included here for a possible future release. |

**T2: High Level Requirement Priority**

**Requirements Priority**



**T4: Detailed Requirements Priority**

**2.3 Database Requirements**

**Customer Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Customer Name | Text(50) | Alphabetic | Yes | |
| Password | Text(50) | Alphabetic | Yes | |
| Gender | Numeric(1) | Numeric | Yes | Male, Female |
| Date of Birth | Date | NA | Yes | yyyy-MM-dd |
| Contact Number | Text(10) | Numeric | Yes | 10 digits |
| Email ID | Text(30) | Alphanumeric | No | |
| Cust ID | Numeric(10) | Numeric | YES | |

**Mechanic Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Mechanic Name | Text(50) | Alphabetic | Yes | |
| Password | Text(50) | Alphabetic | Yes | |
| Gender | Numeric(1) | Numeric | Yes | Male, Female |
| Date of Birth | Date | NA | Yes | yyyy-MM-dd |
| Contact Number | Text(10) | Numeric | Yes | 10 digits |
| Email ID | Text(30) | Alphanumeric | No | |
| Latitude | Double | Numeric | Yes | For navigation |
| Longitude | Double | Numeric | Yes | For navigation |

**Login Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| User ID | Numeric(10) | Numeric | Yes | |
| Password | Text(50) | Alphabetic | Yes | |
| Type (Customer/Mechanic) | Text(1) | Alphabetic | Yes | C or M |

**Request Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Customer ID | Text(50) | Alphabetic | Yes | |
| Contact Number | Text(10) | Numeric | Yes | 10 digits |
| Email ID | Text(30) | Alphanumeric | No | |
| Location | Text(30) | Alphabet | Yes | |
| Latitude | Double | Numeric | No | For navigation |
| Longitude | Double | Numeric | No | For navigation |

**Rating Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Customer ID | Text(50) | Alphabetic | Yes | |
| Contact Number | Text(10) | Numeric | Yes | 10 digits |
| Email ID | Text(30) | Alphanumeric | No | |
| Location | Text(30) | Alphabet | Yes | |
| Latitude | Double | Numeric | No | For navigation |
| Longitude | Double | Numeric | No | For navigation |

**2.4 Technology & Tools**

**Technologies Used**

* Front-End: - HTML5, CSS3, Bootstrap, JavaScript
* Middleware: Java, JSP, Servlet
* Back-End: Oracle 11g express edition

**Tools Used**

* Eclipse oxygen
* Visual Studio Code
* SQL Developer 18.4
* Apache Tomcat 8.5

**HTML**

HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
* Browsers do not display the HTML tags, but use them to render the content of the page

**CSS**

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once

**JavaScript**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript. JavaScript is a lightweight, interpreted programming language.

* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform

Advantages

* Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* Immediate feedback to the visitors − They don't have to wait for a page reload to see if they have forgotten to enter something.
* Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations

We cannot treat JavaScript as a full-fledged programming language.JavaScript doesn't have any multi-threading or multiprocessor capabilities.

**Bootstrap**

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

* Bootstrap is a free front-end framework for faster and easier web development
* Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
* Bootstrap also gives you the ability to easily create responsive designs
* Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.
* **In June 2014 Bootstrap was the No.1 project on GitHub!**

**JAVA**

Java is a popular programming language, created in 1995.It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

* Mobile applications (especially Android apps), Desktop applications
* Web applications, Web servers and application servers
* Database connection
* Games and much, much more!

**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.

Advantages

1)Extension to Servlet

2)Easy to Maintain

3)Fast Development: No need to recompile and redeploy

**Servlet**

**Servlet** technology is used to create a web application (resides at server side and generates a dynamic web page).**Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language.

* Servlet is an API that provides many interfaces and classes including documentation.
* Servlet is a web component that is deployed on the server to create a dynamic web page.

Advantages

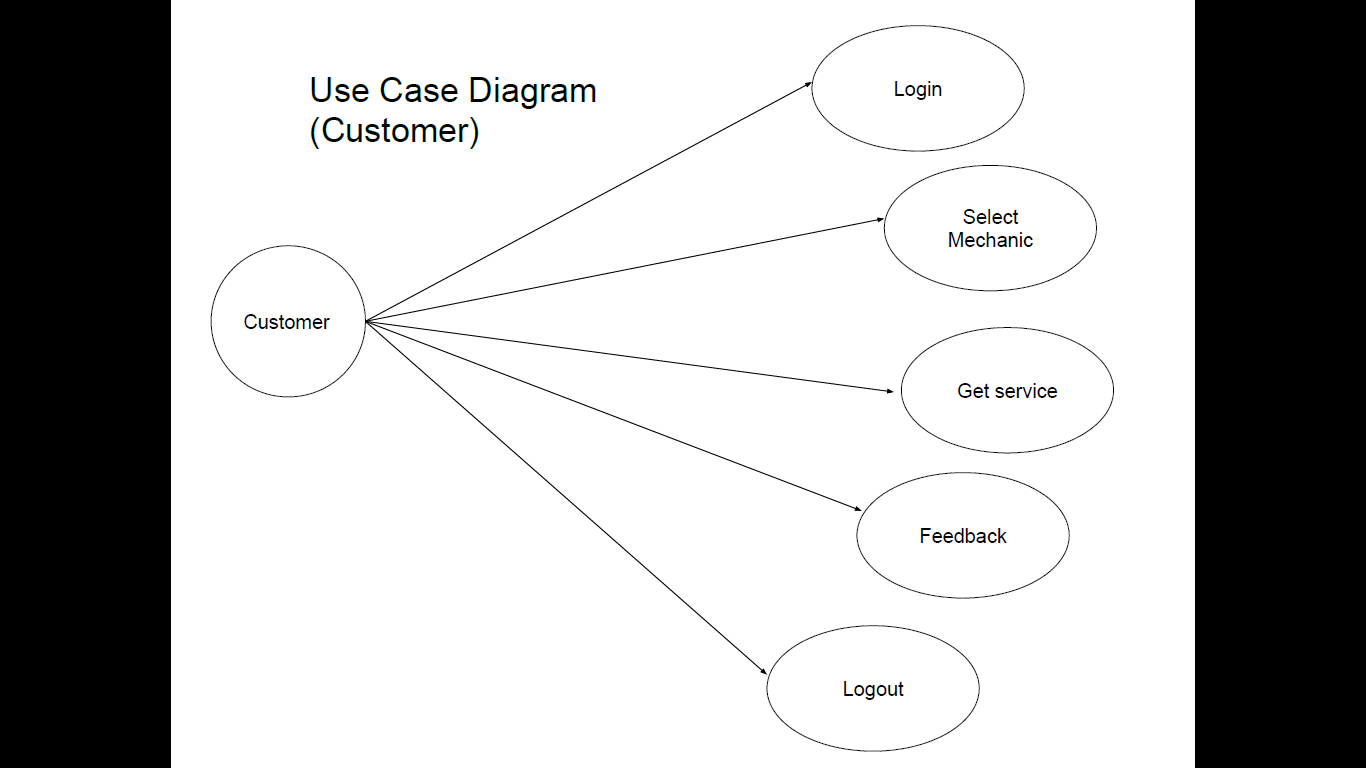
1. Better performance: because it creates a thread for each request, not process.
2. Portability: because it uses Java language.
3. Robust: JVM manages Servlets, so we don't need to worry about the memory leak, garbage collection, etc.
4. Secure: because it uses java language.

**Oracle 11g express edition**

Like other RDBMS software, Oracle Database is built on top of [SQL](https://searchsqlserver.techtarget.com/definition/SQL), a standardized programming language that database administrators, data analysts and other IT professionals use to manage databases and query the data stored in them. The Oracle software is tied to [PL/SQL](https://searchoracle.techtarget.com/definition/PL/SQL), an implementation developed by Oracle that adds a set of proprietary programming extensions to standard SQL -- a common practice among RDBMS vendors. Oracle Database also supports programming in [Java](https://www.theserverside.com/definition/Java), and programs written in PL/SQL or Java can be called from the other language.

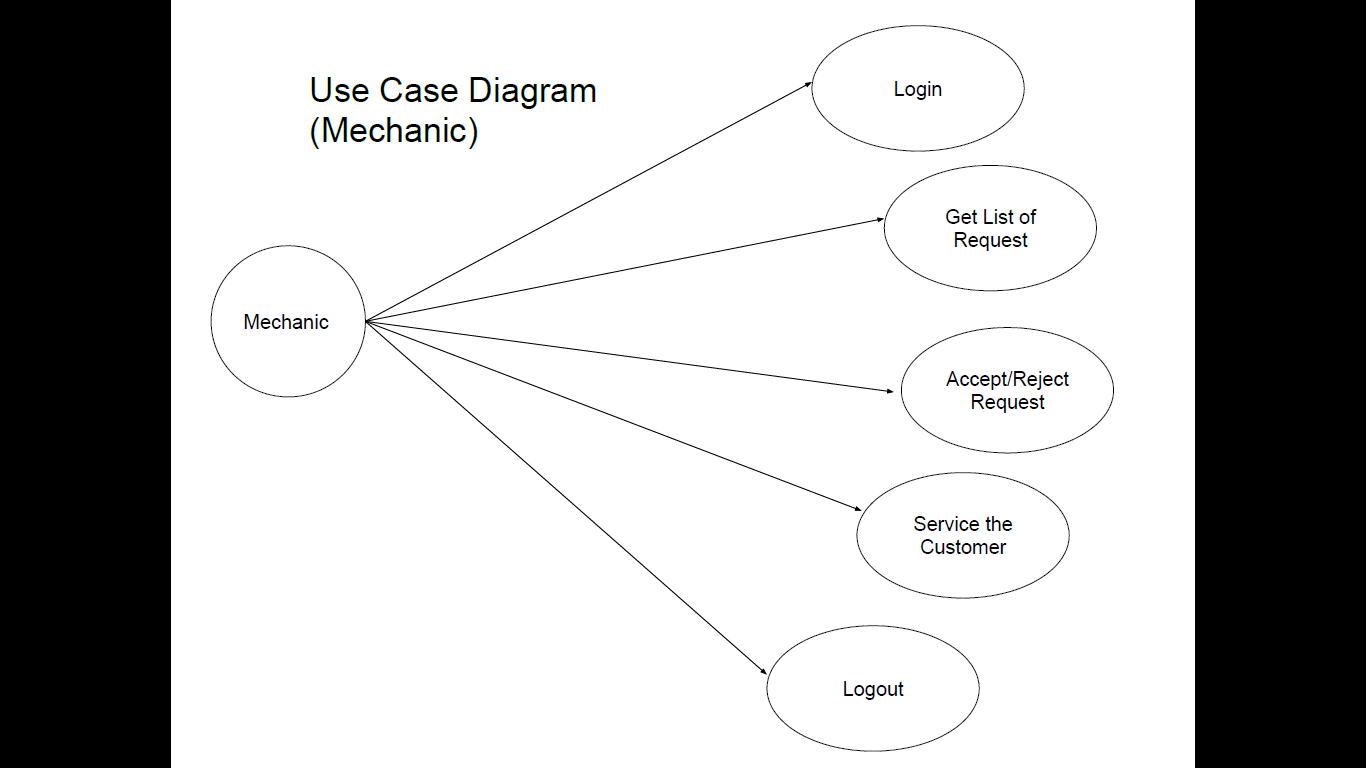
Oracle Database 11g Express Edition (Oracle Database XE) is an entry-level, small-footprint database based on the Oracle Database 11g Release 2 code base.  It's free to develop, deploy, and distribute; fast to download; and simple to administer.

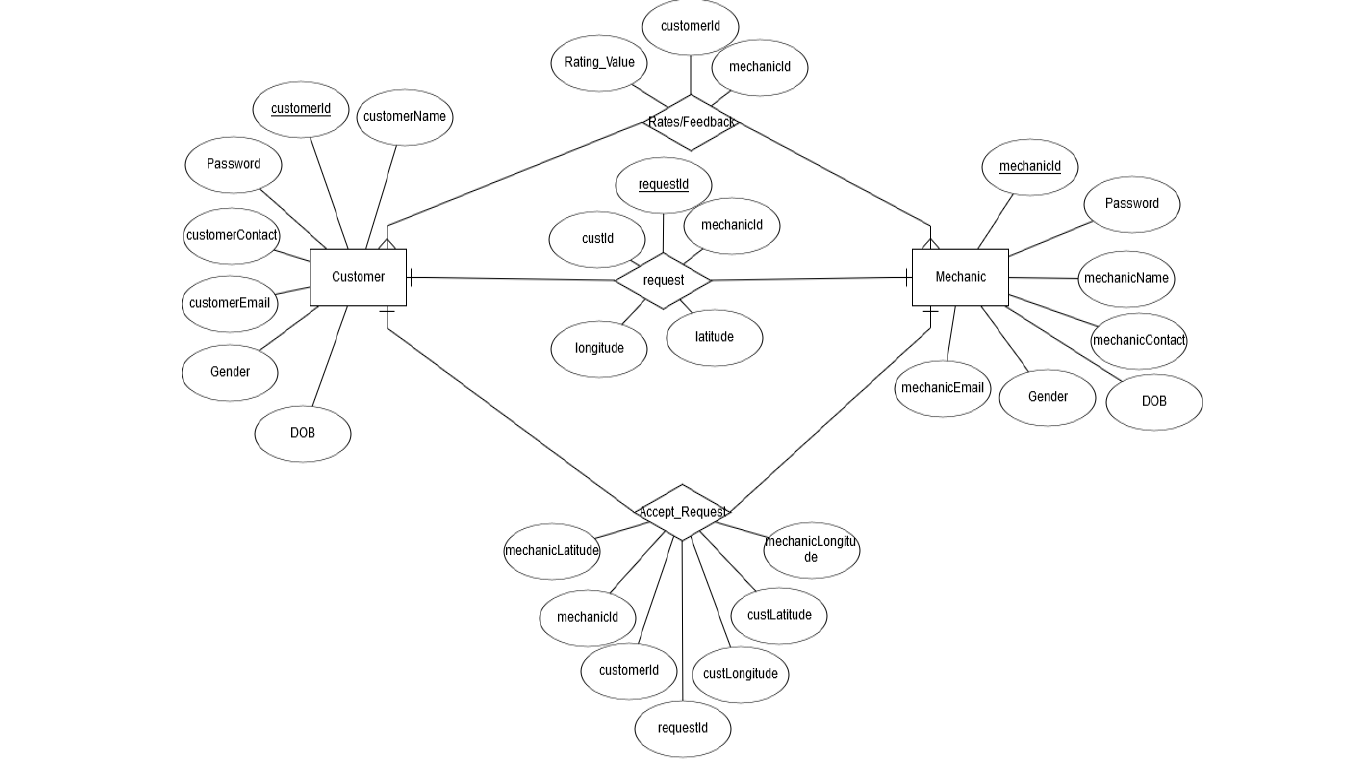
**2.4 Use-Case Diagram**



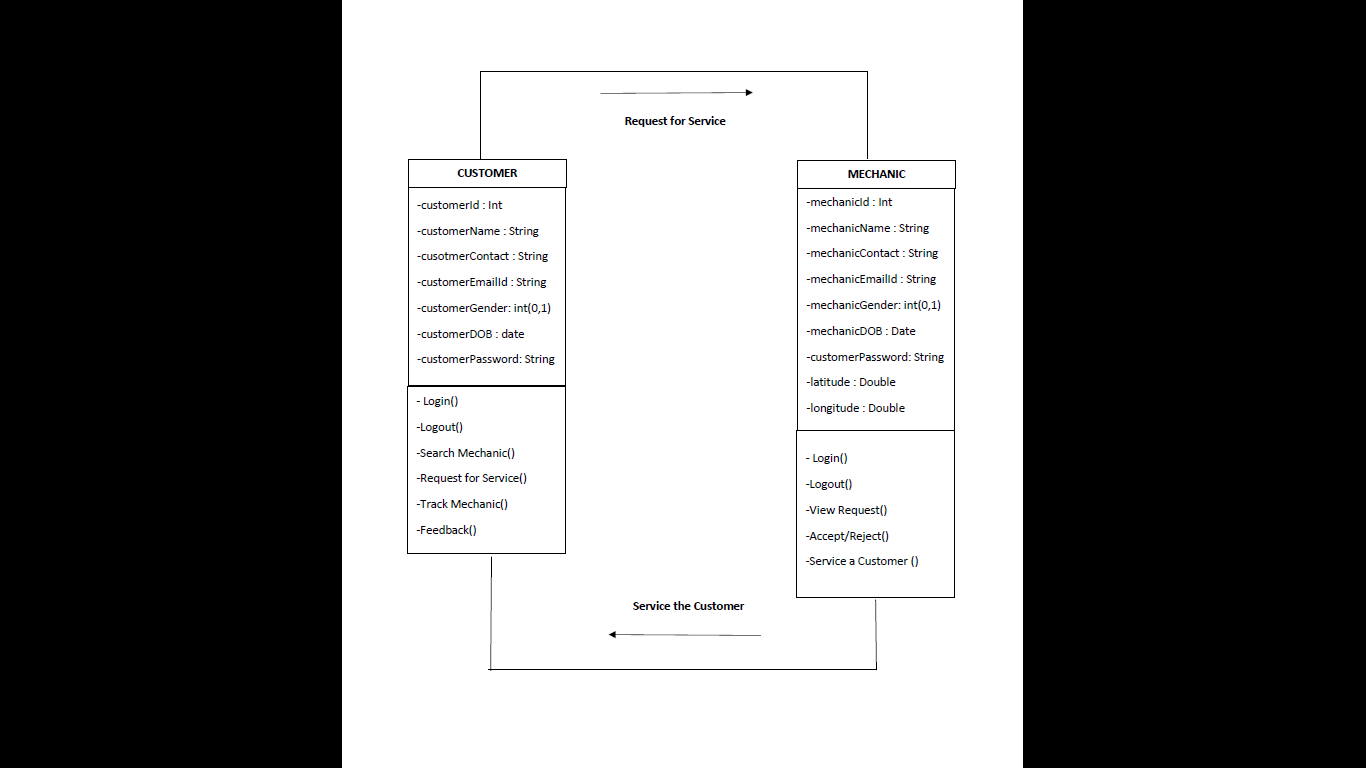
**Fig4: Use Case(Customer)**

**Fig5: Use Case(Mechanic)**

**3.DESIGN**

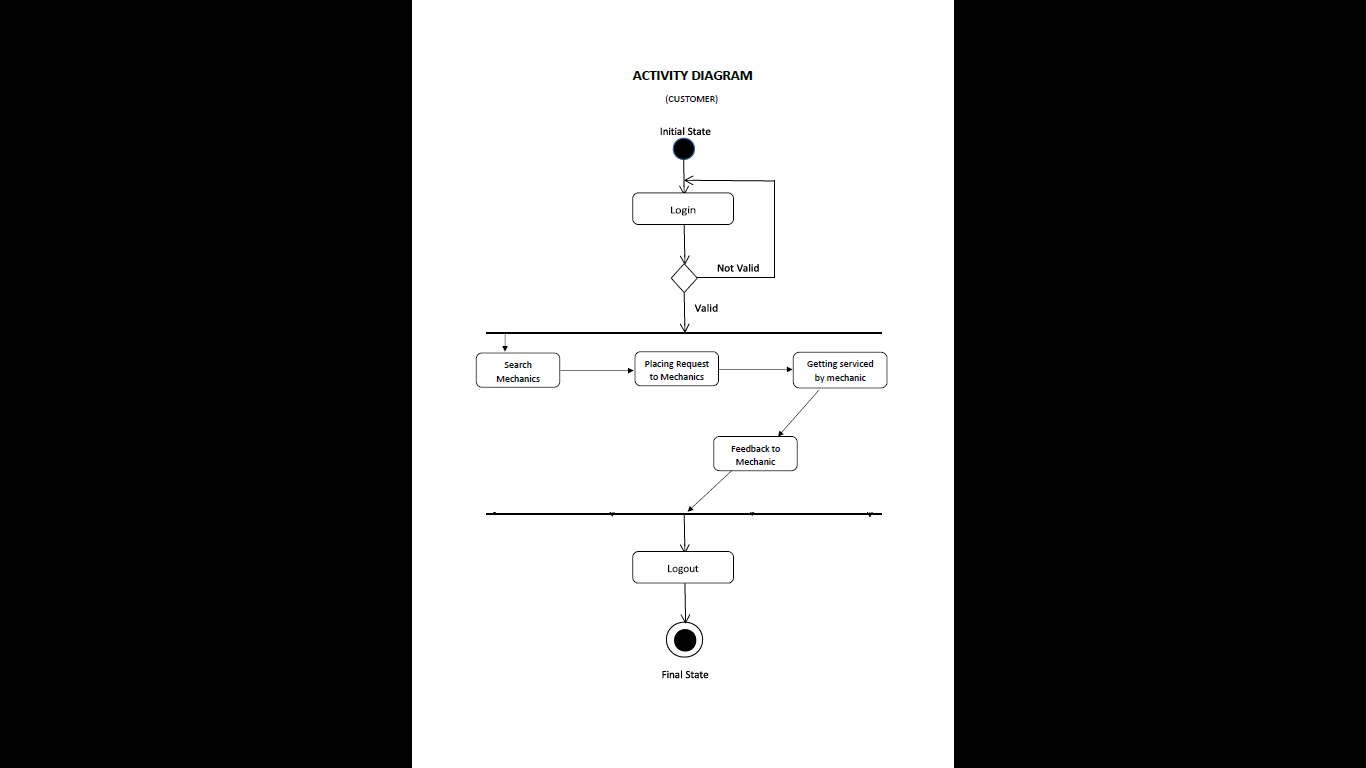
**3.1 ER-Diagram**

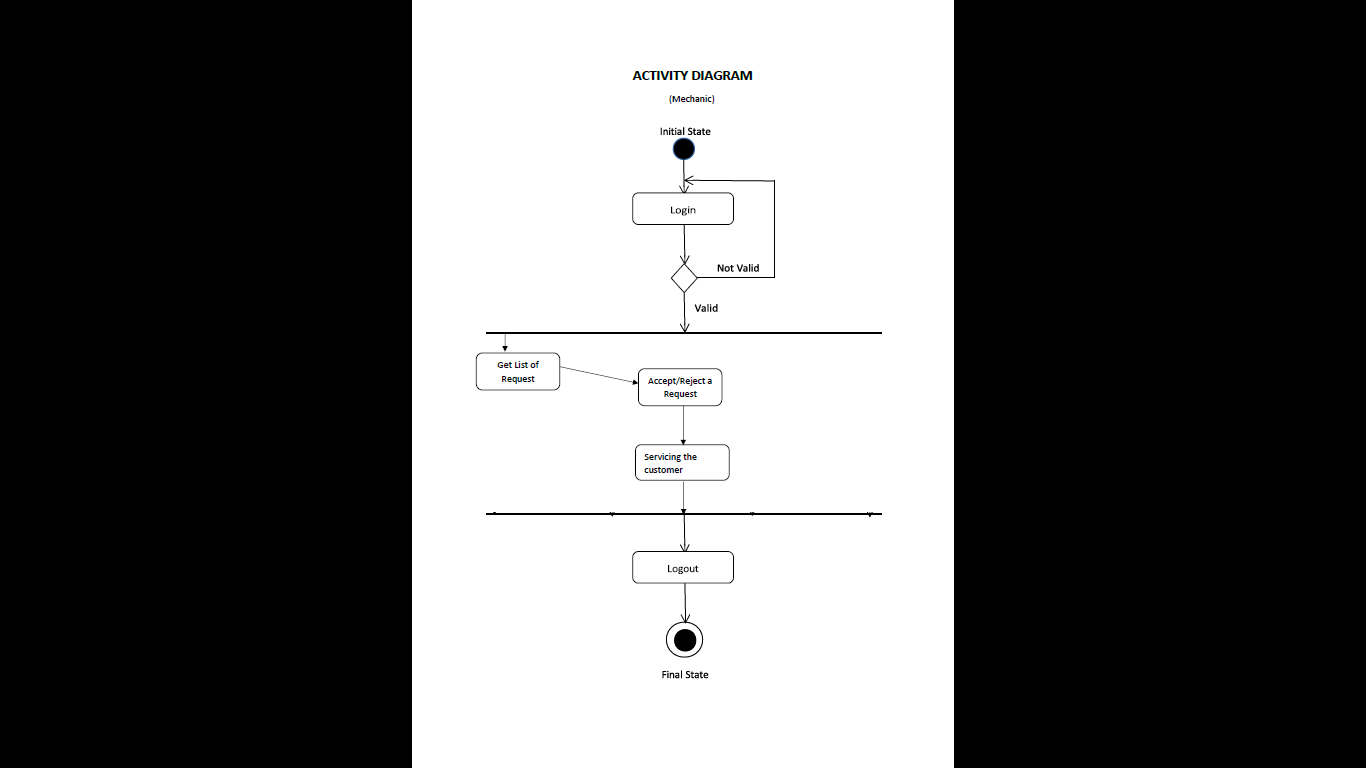
**Fig6: ER- Diagram**

**3.2 Class Diagram**

**Fig7: Class Diagram**

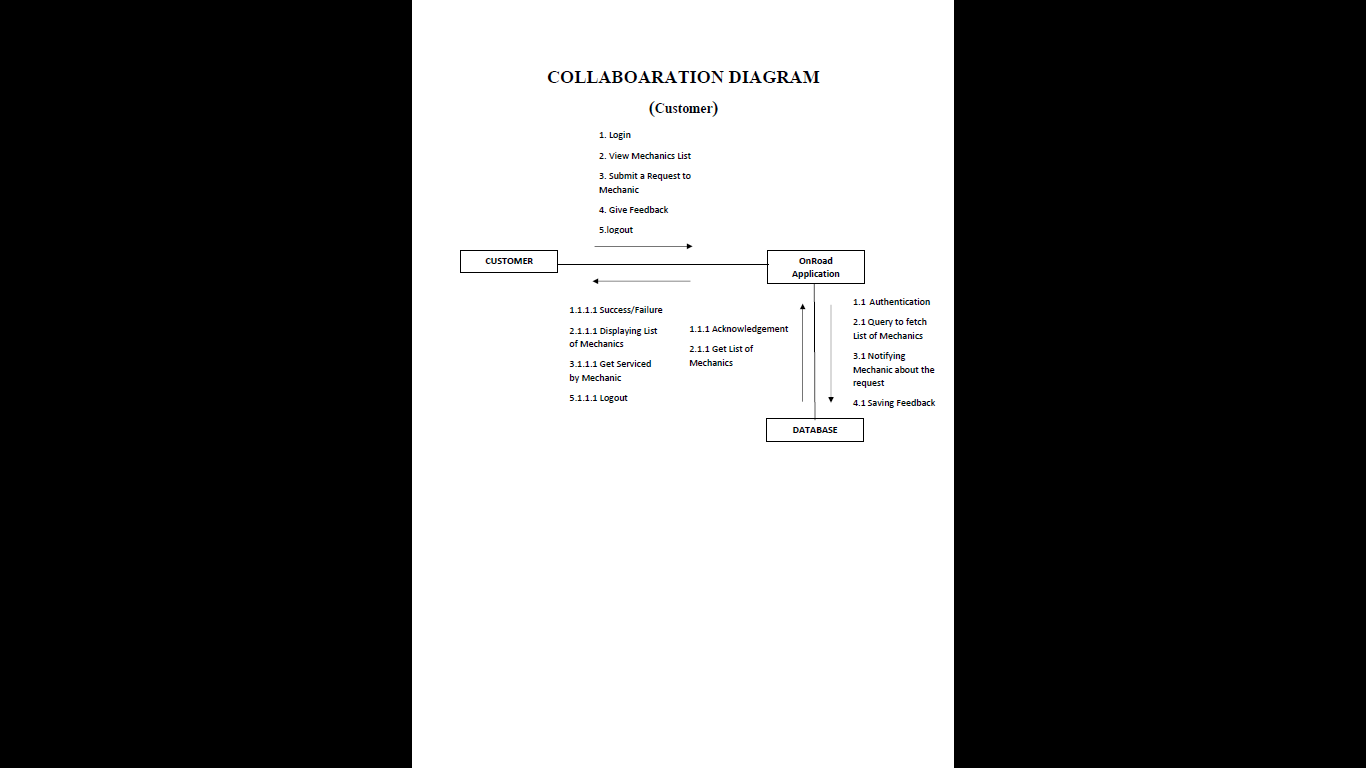
**3.3 Activity Diagram**

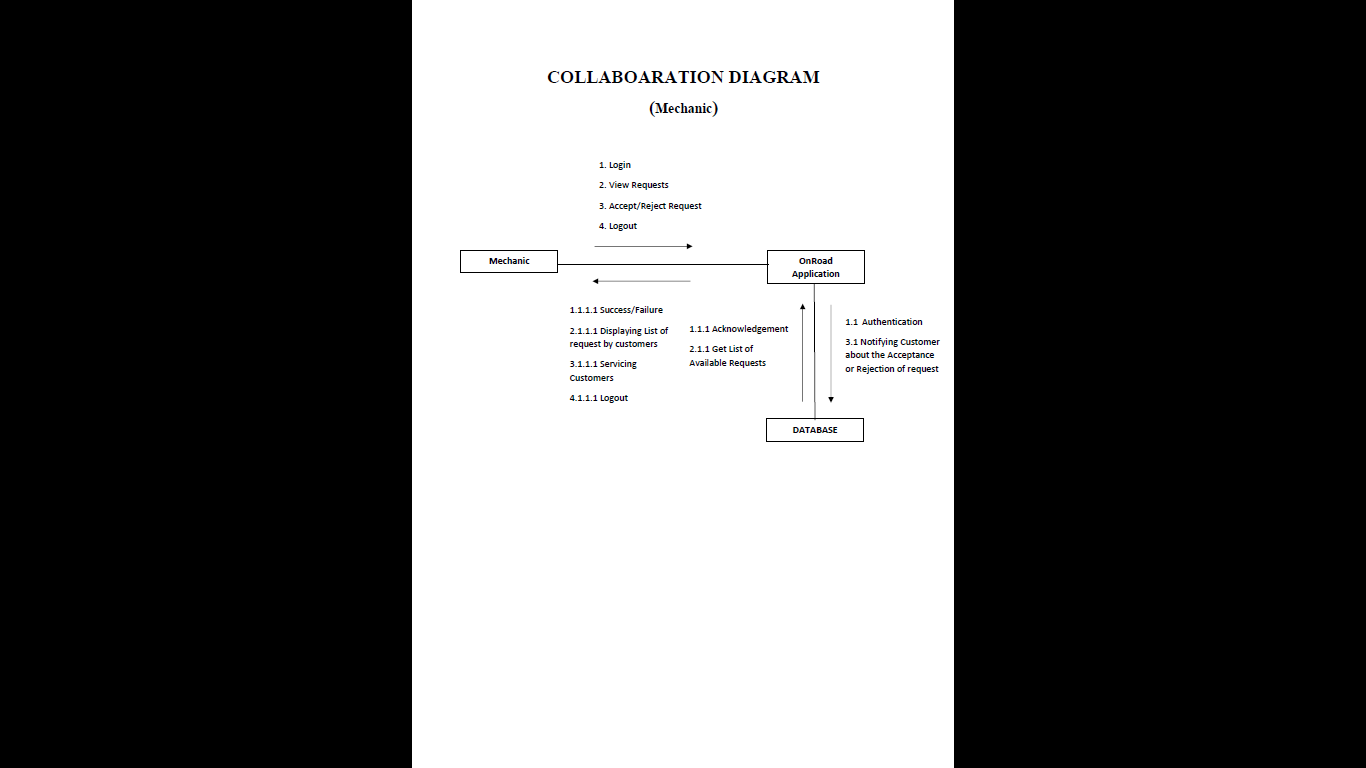
**Fig8: Activity Diagram(Customer)**



**Fig9: Activity Diagram(Mechanic)**

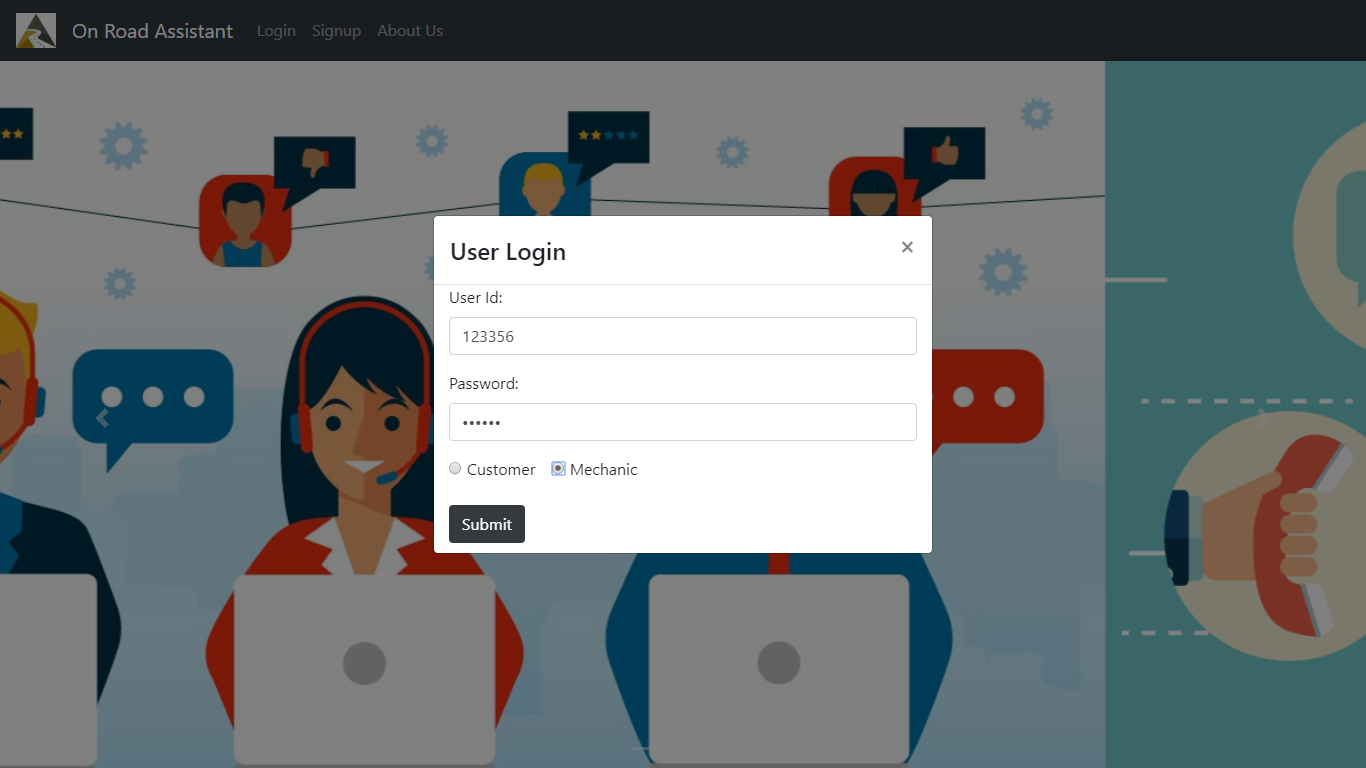
**3.4 Collaboration Diagram**

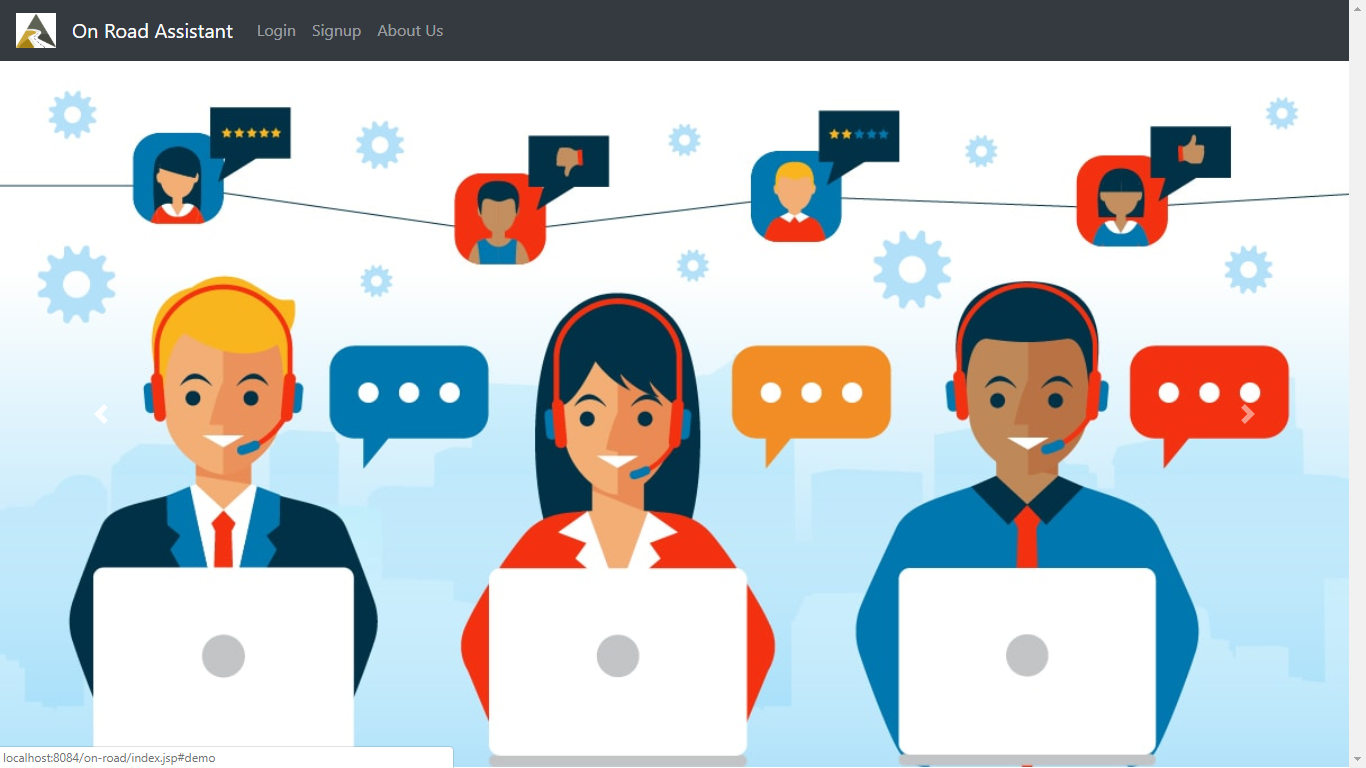
**Fig10: Collaboration Diagram(Customer)**



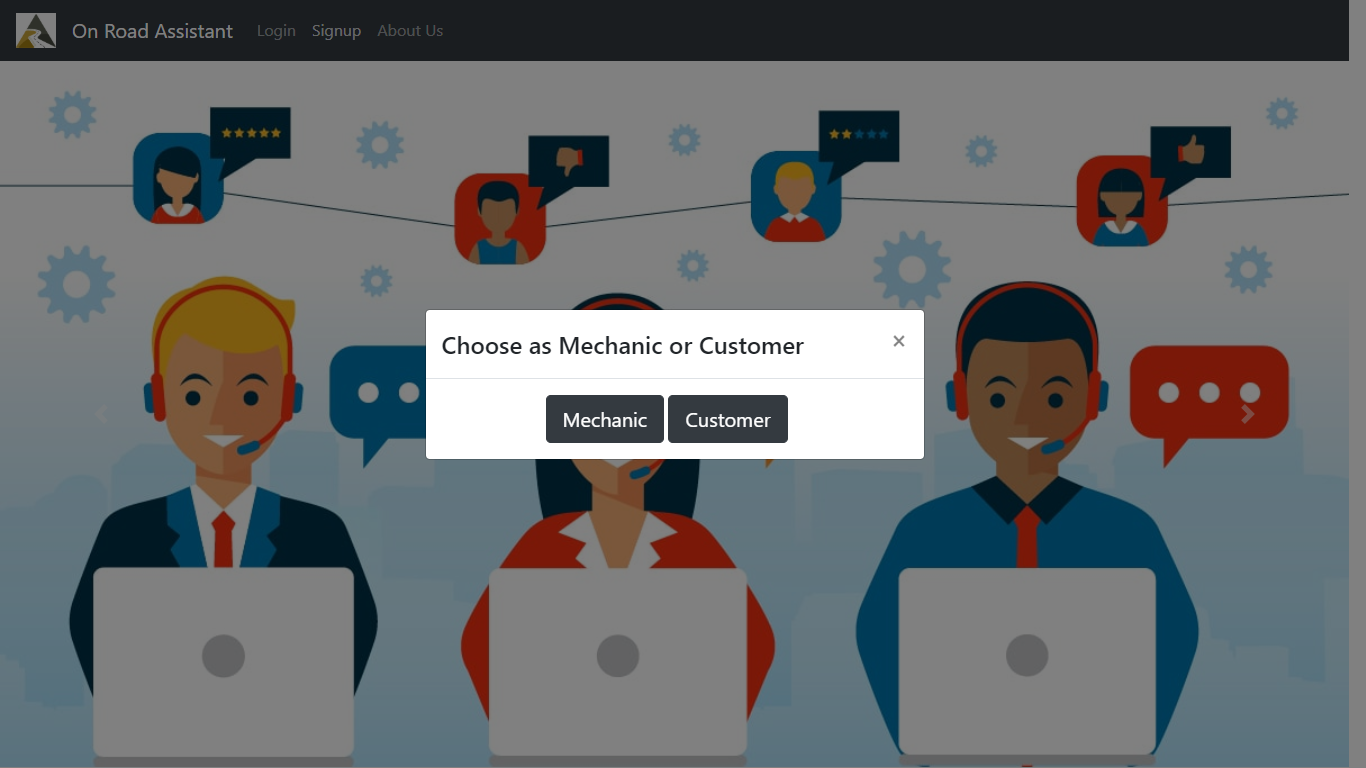
**Fig11: Collaboration Diagram(Mechanic)**

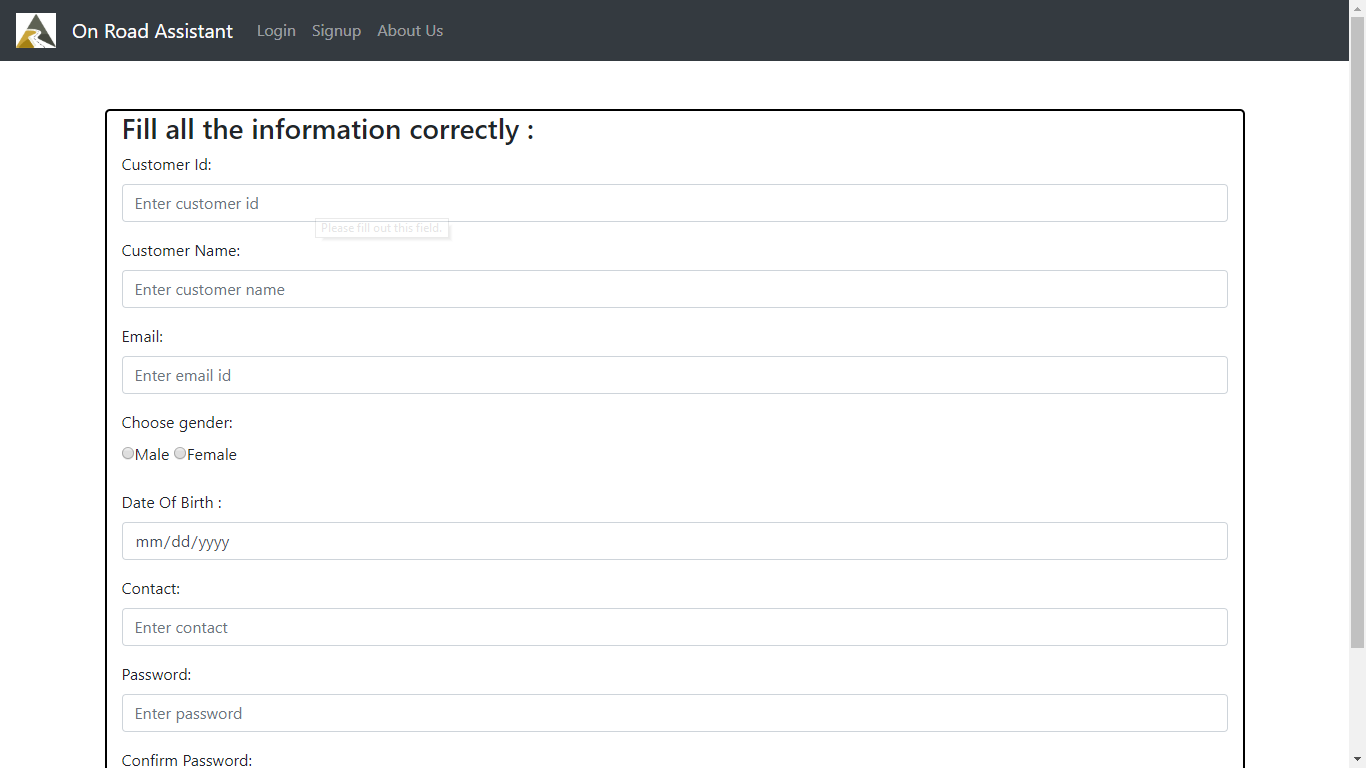
**4.UI Snapshots**

****

****

****

****

****

**5.Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC No** | **TC Name** | **Case Type** | **Expected Result** | **Actual Result** | **Result** |
| **1** | login | Wrong username/password | System throws error and prevent login | System displays error message “invalid username password” | Pass |
| **2** | login | Correct username/password | System redirects to next page | Displays user home page | Pass |
| **3** | Registration | Required field validations | Mandatory field is not blank | Registration done successfully | Pass |
| **4** | Registration | Required field validations | Mandatory field is not blank | Auto redirection to error page | Fail |
| **5** | Request Button(customer) | Request for service | Redirection to request page to view mechanic | Display list of available mechanic sorted by rating | Pass |
| **6** | Customer Requests | Select/Reject Request | Redirect to proper page | Displays action taken | Pass |
| **7** | Location | Track Mechanic | Customer can view mechanic location on map | Location can be viewed on map | Pass |
| **8** | Feedback | Customer rating to mechanic | Customer gives intentional feedback to mechanic after getting serviced | Saving feedback of customer as VG,G,Avg,P,VP | Pass |

**T4: Test cases**

**6.Bibliography & References**

1. Info@erdplus.com, “ERDPlus,” ERDPlus. [Online]. Available: <http://www.erdplus.com/>

2. “Where Developers Learn, Share, & Build Careers,” Stack Overflow. [Online]. Available: <https://stackoverflow.com/>

3. "Build software better, together", *GitHub*, 2019. [Online]. Available: <https://github.com>

4. Mark Otto, "Bootstrap", *Getbootstrap.com*, 2019. [Online]. Available: https://getbootstrap.com/.

5. "W3Schools Online Web Tutorials", *W3schools.com*, 2019. [Online]. Available: https://www.w3schools.com/.