| **Software Design Specifications**      **FREEGX**  **[Freelancing Platform with Project Management]**     | **Project Code** | 4805 | | --- | --- | | **Supervisor** | Mr. Ubaid Aftab Chawala | | **Co Supervisor** | Mr. Sayed Yousuf | | **Project Team** | 19k-0218 Saad Ur Rehman  19k-1373 Abdul Shakoor  19k-1499 Muhammad Anas | | **Submission Date** | 25th Nov 2022 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Document History

| **Version** | **Name of Person** | **Date** | **Description of change** |
| --- | --- | --- | --- |
| 1 | Saad Ur Rehman | 21th Nov 2022 | Document Created, Structured the document, Introduction which includes purpose,audience etc. |
| 2 | Abdul Shakoor | 22th Nov 2022 | Add few headings of overview of diagram , Add Use Case diagram |
| 3 | Muhammad Anas | 25th Nov 2022 | Add diagrams states , create function hierarchy |
| 4 | Abdul Shakoor | 26th Nov 2022 | Add interfaces, use case description. Finalizing SRS |

Distribution List

| **Name** | **Role** |
| --- | --- |
| Mr. Ubaid Aftab Chawala | Supervisor |
| Mr. Sayed Yousuf | Co Supervisor |

Document Sign-Off

| **Version** | **Sign-off Authority** | **Sign-off Date** |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**Table of Contents**

[1.](#_heading=h.30j0zll) Introduction 5

[1.1.](#_heading=h.3znysh7) Purpose of Document 5

[1.2.](#_heading=h.2et92p0) Intended Audience 5  
1.3 Abbreviations 5

[1.4.](#_heading=h.tyjcwt) Document Convention 5

[2.](#_heading=h.3dy6vkm) Overall System Description 6

[2.1.](#_heading=h.1t3h5sf) Project Background 6

[2.2.](#_heading=h.4d34og8) Project Scope 6

[2.3.](#_heading=h.2s8eyo1) Not In Scope 6

[2.4.](#_heading=h.17dp8vu) Project Objectives 6

[2.5.](#_heading=h.3rdcrjn) Stakeholders 6

[2.6.](#_heading=h.26in1rg) Operating Environment 6

[2.7.](#_heading=h.35nkun2) System Constraints 7

[2.8.](#_heading=h.1ksv4uv) Assumptions & Dependencies 7

[3.](#_heading=h.2jxsxqh) External Interface Requirements 7

[3.1.](#_heading=h.z337ya) Hardware Interfaces 7

[3.2.](#_heading=h.3j2qqm3) Software Interfaces 7

[3.3.](#_heading=h.1y810tw) Communications Interfaces 8

[4.](#_heading=h.4i7ojhp) Functional Requirements 8

[4.1.](#_heading=h.2xcytpi) Functional Hierarchy 8

[4.2.](#_heading=h.1ci93xb) Use Cases 9

[4.2.1.](#_heading=h.3whwml4) Description of use cases 10

[5.](#_heading=h.2bn6wsx) Non-functional Requirements 11

[5.1.](#_heading=h.qsh70q) Performance Requirements 11

[5.2.](#_heading=h.3as4poj) Safety Requirements 11

[5.3.](#_heading=h.1pxezwc) Security Requirements 12

[5.4.](#_heading=h.49x2ik5) User Documentation 12

[6.](#_heading=h.147n2zr) References 12

[7.](#_heading=h.3o7alnk) Appendices [12](#_heading=h.147n2zr)

1. **Introduction**
   1. **Purpose of Document**

The purpose of this document is to provide understanding of working of FreeGx by giving system designs and modeling. This document will provide knowledge of each and every feature which will available in the final product. This document will also help to transfer the knowledge of freegx to its reader. This document will the work which we will implement in our final product. The Type of designs would be Object oriented designs because all design will have different stakeholders and how they are interacting with real world entities

* 1. **Intended Audience**

The target audience is the whole group of peoples who wants to acquire services from providers. The second target group is peoples whose occupation is freelancing. The last clientry is peoples who wanted to join any service providers.

**1.3 Abbreviations**

| FYP | Final Year Project |
| --- | --- |
| WBS | Work breakdown structure |

* 1. **Document Convention**

The font size which will be enough for anyone to be able to read is **10px** with font family is **Aerial** . For Headings the size is **14px** with bold weight.The font size which will be enough for anyone to be able to read is **10px** with font family is **Aerial** . For Headings the size is **14px** with bold weight.

1. **Overall System Description**
   1. **Project Background**

After the pandemic COVID-19, many people lost their job and shifted to the online mode of working in search of authentic and reliable freelancing platforms, some of them succeeded while some failed to get projects as they were new in the market. Those who got the projects still faced many problems on the platform like there were no project management tools, progress reports, adding team members as co-worker to reduce workload, and many others.

* 1. **Project Scope**

Our Project “FreeGx” is a web application where people can do freelancing with ease and search for sellers with extraordinary gigs to provide them with the best quality experience. Through this platform, Seller can add co-workers who will work along with the team lead which increases interpersonal and group-working skills. The seller can share the work breakdown structure with the co-workers so that they will get a clear idea of relevant tasks to perform. The seller can also share progress reports and documents with clients. There will be other features to address previous problems in freelancing platforms.

* 1. **Not In Scope**

The project will not contains online meeting options and bidding system. The client will approach to seller instead of seller bids on clients post

* 1. **Project Objectives**

The main objectives of our project are as follows:

* To provide a platform to work in a collaborative environment.
* To work with fellow co-workers to increase the productivity of the task assigned.
* To help create a Work Breakdown Structure to well organize the projects.
* To maintain weekly or monthly progress reports between sellers and clients.
  1. **Stakeholders**

There are three stakeholders,

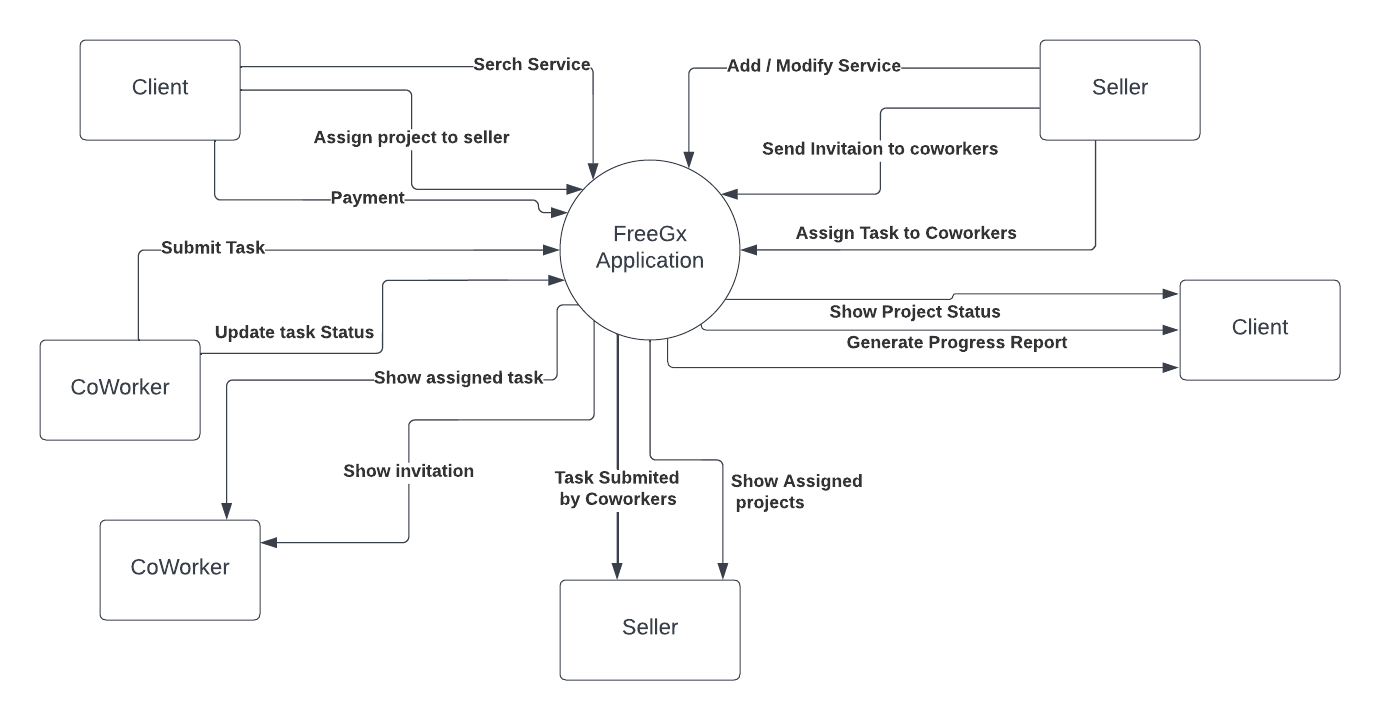
* Seller, who will be responsible to provide best quality services to the clients with respect to thier skill set
* Client, who are the backbones on any business platforms. Clients can see all listing of services by searching and category-wise.
* Co-worker, whose main aim is to complete all assigned tasks given by the seller and update the status after completion of the given task.
  1. **Operating Environment**

FreeGx can be used in any part of the world. You need few things to run our application. Firstly, The user must have personal computer with stable internet connection. In software perspectives. Any Operating system should installed in your computer. You must have browser that can help you to redirect to our application. Simpy type URL : [www.freegx.com](http://www.freegx.com) . You will be redirected to our application.

* 1. **System Constraints**
* **Software constraints:** No Wi-Fi, Service temporarily avaialble
* **Hardware constraints:** N/A
* **Cultural constraints** : N/A
* **Legal constraints:** N/A
* **Environmental constraints:** N/A
* **User constraints:** Project is developed for audience who are working as a freelancer or a client who wants to acquire any service.
  1. **Assumptions & Dependencies**

The system must be available with a stable internet connection. The system is dependent on cloud where the product will be deployed. The server should be up and running for infinite users. The security should also be there so that no any external such as malicious attack disrupt the website.

1. **External Interface Requirements**



* 1. **Hardware Interfaces**

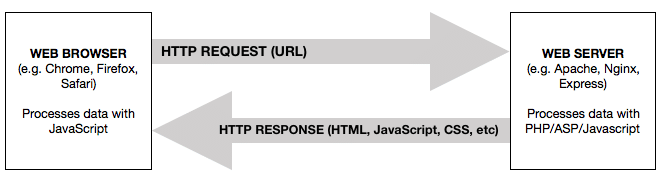
Not applicable

* 1. **Software Interfaces**

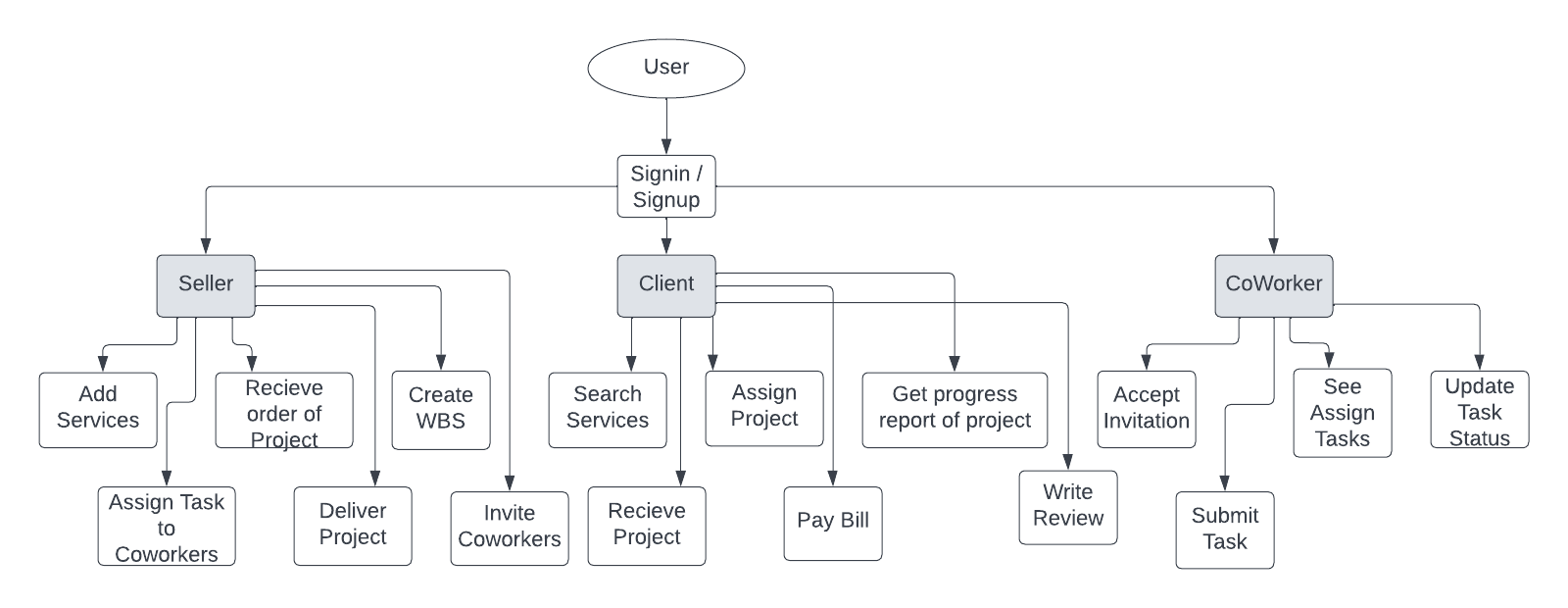
The front-end interface will be based on React and there will be a recused components present so that user does not need to wait for the whole page loading. Just a component will be reloaded. The onclick and onchange functions will be used for the state management The react components will be used for interchange information among screens. The software is dependent on the browser for generating response and packets through OSI model layer. The HTTP protocol will be used for data transfer.. The backend server should be running and scalable. The request will be get by the server where logical authentication, validation and security measures will be taken. The server will be interacting with the database using several queries. that retrieve the data from the database. The response generated will be sended to front end again.

* 1. **Communications Interfaces**

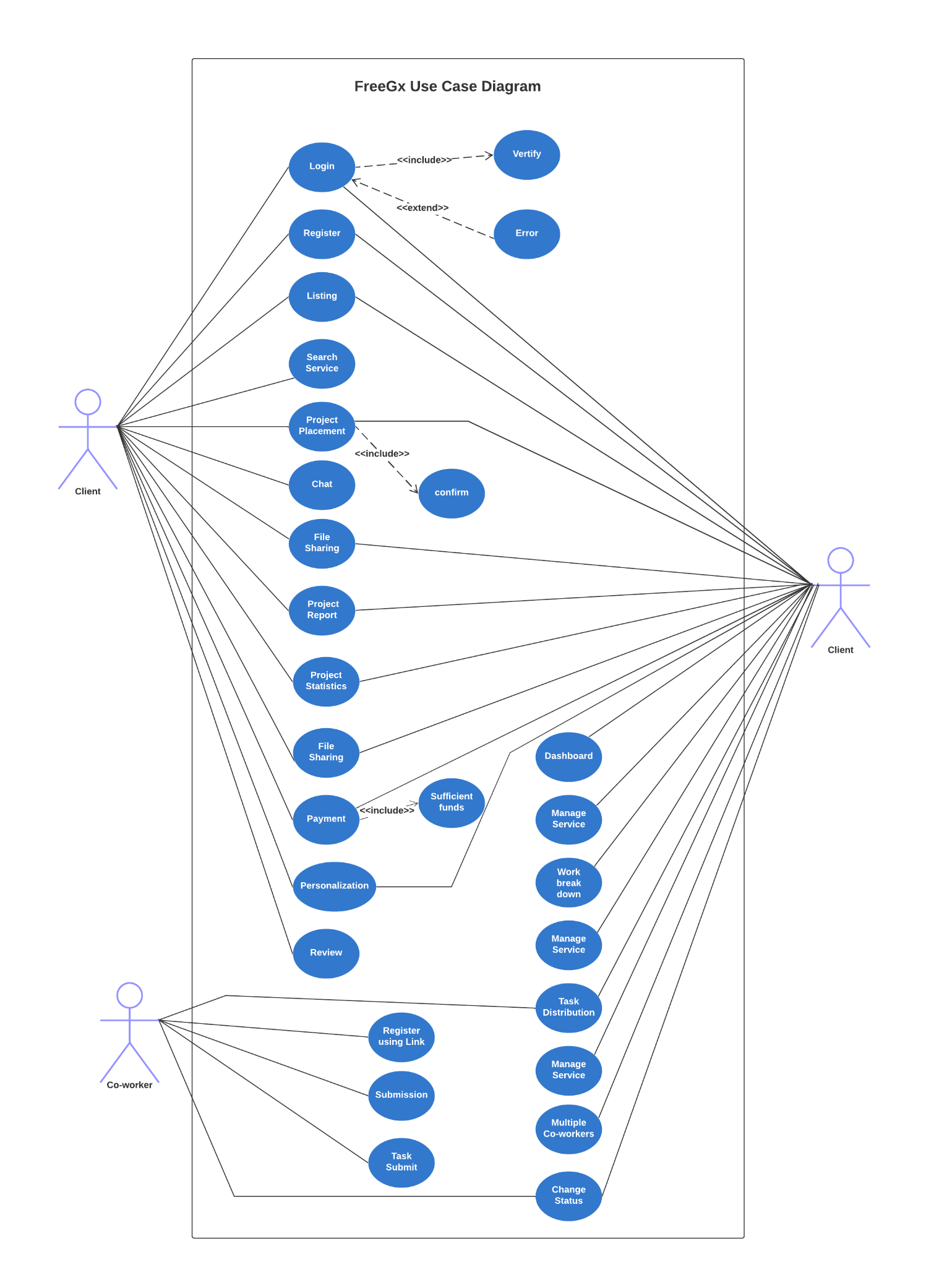
The web browser will be used to access the website. In FreeGx. The communication between client and server occurs through http protocol where the data is forwarded through the series of packets. The bulk data will be exchange through the asynchronous api calls to maintain the non functional requirements. The communication packets has a defined formats A request line with the header and the body contains data. several bits for checksum as well.



1. **Functional Requirements**
   1. **Functional Hierarchy**



* 1. **USE CASES**
     1. **FreeGx Use Case Diagram**

****

* + 1. **FreeGx Use Case Description**

| **LOGIN** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1 | | |
| **Actors:**  Seller , client | | | | |
| **Feature:** The use case will authenticate any user | | | | |
| **Pre-condition:** | | The user must access the introductory website or visit login portal | | |
| **Scenarios:**  When any user needs to acces his portal | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The user inputs correct credentials | | | Allow user to access portal |
| **2.** | The user inputs incorrect credentials | | | Denied and error generated |
| **Alternate Scenarios** | | | | |
| **1a:** The user will visit the introductory website.  **2a:** The user will enter username.  **3a.** The user will enter password.  **4a.** The user actions on this usecase. | | | | |
| **Post Conditions** The user must logged out when jobs completed. | | | | |
| **Step#** | **Description** | | | |
| **1** | The user will view the dashboard | | | |
| **Use Case Cross referenced** | | | Verify , error | |

| **PROJECT PLACEMENT** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2 | | |
| **Actors:**  Seller , client | | | | |
| **Feature:** The client can assign project to seller | | | | |
| **Pre-condition:** | | The users must be authenticated  The seller must have service  The client must access those service | | |
| **Scenarios:**  The client logins and search for the listings. When the service related to category. He will assign project to seller through that service | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The client place project order | | | The project will shown to user |
| **2.** | The seller accepts project | | | The project order placed and save to database |
| **Alternate Scenarios** | | | | |
| **1a:** The client will see all services.  **2a:** Find the service better of his choice related to his query.  **3a.** The client contacts service owner i.e. seller  **4a.** The client after confirmation places project. | | | | |
| **Post Conditions** The project will completed by the client. | | | | |
| **Step#** | **Description** | | | |
| **1** | The project billing use case starts. | | | |
| **Use Case Cross referenced** | | | N/A | |

| **BILLING** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 3 | | |
| **Actors:**  Seller , client | | | | |
| **Feature:** The client pay bill to seller | | | | |
| **Pre-condition:** | | The project placement need to occur and completed by seller | | |
| **Scenarios:**  The afterward of project placement usecase. When the seller gets the project. He try to complete in specified time. The Bill generates contain amount will be payed by the client | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The project completes | | | Bill generates by system. |
| **2.** | The client pay bills | | | The seller account increase and client account amount decrease. |
| **Alternate Scenarios:** The project cancels. No need to pay bills. | | | | |
| **1a:** The seller completes project.  **2a:** The bill generates..  **3a.** The client pays bills to client..  **4a.** The accounts updated.. | | | | |
| **Post Conditions** The review needed to submit | | | | |
| **Step#** | **Description** | | | |
| **1** | The client contains prompt screen for given ratings and review. | | | |
| **Use Case Cross referenced** | | | sufficient funds | |

1. **Non-functional Requirements**
   1. **Performance Requirements**

The main goal of our project is to provide flexibility and smooth operations and performance for that we components so instead of whole page reload. Only component will be reload on stage changes. Our project should be measurable, understandable, verifiable, equitable, and achievable. For that we made several conditions. For illustrate, The system will not take more than 2 seconds during login and sign up. There should be asynchronous API calls so that user will work reliable manner instead of seeing spinner. The system build to catch several errors on runtime also we made exceptions for taking steps when fault tolerance or any unusual events occurs.

* 1. **Safety Requirements**

The huge amount of attacks occurs in websites. So our website can be unsafe. To maintain safety, We build a software that management procedures, checklists, and validation procedures for using it. To ensure safety, we made code that catches exceptions also used built in packages that can allow to protect information. We used different strategies like hash code and salt method to encode our users critical information in database. The client will be shown when any undesirable activity is performed. When any unknown installation done by client. The software restrict him and only allow to continue when the client takes responsibility.

* 1. **Security Requirements**

The major security requirements include confidentiality. To ensure that the password and other information will be seen by another person. The system should available and it is requires that no any malicious activity occurs that loss the integrity of the system. We make sure to completes these requirements through efficient authentication and password management. It is required the authorization will be correct and access provided to user based upon his role in the system. Other security requirements include network security, data security, testing and validation etc.

* 1. **User Documentation**

1. The user will get video tutorial how to use the system.
2. The user will get introductory website which contains all the information related to software
3. Online support and FAQs will be provided.
4. The user will get proposals , designs , architectures , diagrams to understand all mechanism
5. Software requirements specification will be given to the user.
6. Software Design specification will be given to the user.

1. **References**

Not Applicable

1. **Appendices**

Not Applicable