
Software Specification

Requirements

for

Freelancing System

Version 1.1

Prepared by

Group Name: <place your group name here>

Nirav Pokiya	CE048	23ceuog125@ddu.ac.in
Rudra Prajapati	CE051	23ceubg128@ddu.ac.in

Instructor:	Ami Shah
Course:	Software Engineering
Lab Section:	<place your lab section here>
Teaching Assistant:	Ami Shah
Date:	<place the date of submission here>

CONTENTS.....	II
REVISIONS.....	II
1 INTRODUCTION.....	1
1.1 DOCUMENT PURPOSE.....	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	1
1.5 DOCUMENT CONVENTIONS.....	1
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
2 OVERALL DESCRIPTION.....	2
2.1 PRODUCT OVERVIEW.....	2
2.2 PRODUCT FUNCTIONALITY.....	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	3
2.4 ASSUMPTIONS AND DEPENDENCIES.....	3
3 SPECIFIC REQUIREMENTS.....	4
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	4
3.2 FUNCTIONAL REQUIREMENTS.....	4
3.3 USE CASE MODEL.....	5
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	6
4.1 PERFORMANCE REQUIREMENTS.....	6
4.2 SAFETY AND SECURITY REQUIREMENTS.....	6
4.3 SOFTWARE QUALITY ATTRIBUTES.....	6
APPENDIX A – DATA DICTIONARY.....	8
APPENDIX B - GROUP LOG.....	9

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

Our project is based on freelancing system where a client finds good freelancer to complete their projects and freelancer can have opportunity to apply their skills on real world projects. this document describing some details on our project and features.

1.1 Document Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the ***Freelancing system***, its requirements with respect to end-users. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered.

This document describes the requirements and scope of the project, that is who will be the users of our system and how our system will work for different users. moreover this document describes flow of our system.

1.2 Product Scope

The Freelancing System platform is designed to empower freelancers by providing opportunities for continuous skill development and professional growth. It allows freelancers to enhance their expertise and be compensated based on the projects they undertake, fostering a more competitive and skilled workforce. For clients, the platform offers a seamless experience in finding qualified individuals for various tasks, projects, and specialized jobs. This efficiency reduces the time spent searching for suitable talent, ensuring that both freelancers and clients can focus on delivering high-quality work and achieving their goals.

1.3 Intended Audience and Document Overview

Software Requirements Specification (SRS) document is intended for a variety of stakeholders involved in the development and deployment of the Freelancing System Project.

Developers : this is primary audience which will understand the SRS document and requirements

will work based on requirements.

Testers: after developers develops system testers will do some system testing for that.

Manager/client: will review work done by developers and checks that all requirements are fulfilled or not .

The Overview section provides a high-level understanding of the system's goals and scope for all readers. Functional Requirements detail the system's capabilities, guiding development and testing. Non-Functional Requirements outline key attributes like

performance and security, relevant for all stakeholders. System Design explains the technical architecture, mainly for developers and testers, with business alignment for managers/clients. Interfaces describe external system interactions, crucial for developers and useful for testers and managers. Use Cases offer specific system examples, helping all stakeholders, especially managers/clients and testers, visualize functionality

1.4 Definitions, Acronyms and Abbreviations

Definitions:

Freelancer : An individual who offers their services or skills on a temporary basis, typically through an online platform.

Client : An individual person or organization that hires freelancer to complete their project.

Project : A specific task or set of tasks assigned to freelancer by a client

Bid : An offer made by a freelancer to complete a project for a specified price.

Proposal :A detailed plan submitted by a freelancer outlining how they will complete projecting and also a request to the client that they are willing to accept their project.

Review : Feedback provided by a client or freelancer after the completion of a project.

Rating : A numerical score given by a client or freelancer to evaluate the quality of work or collaboration.

Acronyms and Abbreviations

API : Application Programming Interface

DB : Database

GUI : Graphical User Interface

HTTP : Hypertext Transfer Protocol

HTTPS : Hypertext Transfer Protocol Secure

JSON : JavaScript Object Notation

SRS : Software Requirements Specification

SQL : Structured Query Language

1.5 Document Conventions

This Document was created based on the IEEE template for System Requirement

Specification Documents.

1.6 References and Acknowledgments

Other Similar Freelancing Platforms : Upwork, Freelancer, Fiverr, Toptal.

An article of Everyday struggle of the freelancer on the available online platforms :
[The Everyday Struggles of Freelancers: A Compilation of problems and solutions - The Freelance Toolbox](#)

Basics knowledge and guideline in order to build Perfect Freelancing System :
[7-Step Guide to Building the Perfect System for Your Freelance Business | Bloom](#)

2 Overall Description

2.1 Product Overview

The Freelancing System was designed to streamline the process of matching freelancers with clients, ensuring fair compensation for services rendered. It offers an efficient job search and bidding mechanism, manages multiple tasks and projects, and provides integration with GitHub for seamless work transfer. This product is a new, self-contained system created to address the growing demand for flexible, remote work arrangements. It is not a follow-on member of a product family, nor is it a replacement for existing systems. Instead, it introduces a novel approach to managing freelance work, combining job searching, project management, and work transfer in one platform.

The need of this system arose from the evolving economy and increasing the number of individuals seeking freelancing opportunities. Traditional freelancing systems often fail to offer the flexibility of employment models and autonomy that a freelancer desires. Our system caters a satisfactory platform to fulfill such desires and fill the gap of many specific needs.

2.2 Product Functionality

Freelancer:

Freelancer Account Setup:

A freelancer can add their contact details and skill based details on site.

Skill-Based Filters and Recommendations:

- Implement a system that suggests projects based on the freelancer's skill set, experience, and completed projects.

Project Tracking

- Freelancers can track the progress of the projects they're working on and communicate updates to the client.

Freelancer Reviews

- Freelancers can view feedback and ratings from previous clients to help them evaluate their reputation and work quality.

Team formation

- Freelancer could be a member of the team as well as form a team (if he wants) and he/she is paid according to their part of work.

Client:

Client account setup:

- Clients can add their work details and about their company or organization.
- For sophistication , the client could add their licenses and qualifications.

Project Management:

- They can post projects with time duration and required skills.
- Handling multiple projects and works at a time.

Freelancer Tracking :

- Companies can contact freelancers directly with freelancers and hire them based on their performance.

Freelancer Full-Time Contract:

- A company can hire a freelancer for full-time projects or for longer periods of time.

Team formation of project

- Clients could hire entire teams for bigger tasks, which would appeal to companies with large-scale needs.

7Design and Implementation Constraints

Hardware Limitations:

- **Memory Requirements:** Ensure the platform operates efficiently within the memory constraints of users' devices, considering both RAM and storage.
- **Timing Requirements:** The system should have minimal latency for real-time updates and interactions, especially during peak usage hours.

Interfaces to Other Applications:

- **APIs:** The platform must integrate with third-party APIs for payment gateways (e.g., PayPal, Gpay), email services and cloud storage providers.
- **Compatibility:** Ensure compatibility with widely-used project management tools to allow for data import/export.

2.3 Assumptions and Dependencies

Assumptions:

Internet Connectivity and Basic Digital Literacy :

- It is assumed that all end users will have access to the inter-connection networks source in order to open the website from the different web browsers
- The user must be able to read and understand the interface also be to navigate different components of the website.

Device Capabilities:

- Users' devices will have the necessary capabilities (e.g., modern operating systems, updated browsers) to support the platform's functionalities.

Security Compliance:

- Users will comply with basic security practices, such as using strong passwords and protecting their personal information.

Dependencies:

User Feedback and Adoption:

- Dependence on user feedback for continuous improvement and feature enhancements.
- Assumption that users will adopt the platform and provide constructive feedback for ongoing development.

DataBase and Management System:

- Dependence of robust ,secure and widely available database server MySQL to store and manage the user data.

External Libraries and Frameworks:

- Utilization of external libraries and frameworks (e.g.,Spring, Node.js,Django) for building the front-end and back-end of the platform.
- Reliance on open-source components, which may have updates or changes that could impact the system.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

1. Dashboard Interface

- **Logical Characteristics:**
 - **Overview:** Displays an overview of ongoing projects, notifications, earnings, and recent activity.
 - **Widgets:** Interactive widgets for quick access to important features such as messages, notifications, and project status.
 - **Customization:** Users can customize the dashboard by rearranging widgets according to their preferences.

User Interaction:

- **Touch Screen:** Users can interact with widgets through touch gestures or mouse clicks.
- **Menus:** Drop-down menus for accessing different sections such as Projects, Bids, Proposals, and Settings.

2. Project Management Interface

- **Logical Characteristics:**
 - **Project Listing:** Displays a list of projects with filters for status (e.g., active, completed), deadline, and budget.
 - **Project Details:** Detailed view of individual projects including descriptions, deadlines and assigned freelancers.
 - **Actions:** Buttons for editing project details, uploading files, and marking tasks as complete.
- **User Interaction:**
 - **Touch Screen and Menus:** Users can navigate through projects using touch gestures or mouse clicks. Menus provide quick access to project actions.

3. Freelancer Profile Interface

- **Logical Characteristics:**
 - **Profile Overview:** Displays freelancer's profile picture, bio, skills, ratings, and reviews.
 - **Portfolio:** Section to showcase past projects and work samples.
 - **Contact Information:** Secure messaging system for clients to contact freelancers.
- **User Interaction:**
 - **Touch Screen and Menus:** Users can view profiles, send messages, and leave reviews using touch gestures or mouse clicks.

3.1.2 Hardware Interfaces

User Devices

- **Desktops/Laptops:**

- **Logical Characteristics:** The platform must be compatible with various operating systems (Windows, macOS, Linux) and support different web browsers (Chrome, Firefox, Safari, Edge).
- **Physical Characteristics:** Interaction through keyboard, mouse, and monitor. Requires internet connectivity for real-time updates and interactions.

- **Tablets/Smartphones:**

- **Logical Characteristics:** The platform must be optimized for mobile operating systems (iOS, Android) and support touch interfaces.
- **Physical Characteristics:** Interaction through touchscreen. Requires internet connectivity for real-time updates and interactions.

Servers

- **Web Server:**

- **Logical Characteristics:** Currently , it is run on the localhost server with the local web browser such as firefox,microSoft Edge ,Chrome etc.
- **Physical Characteristics:** It is hosted on mySQL servers,Django servers for robust interaction with a large number of people.

Database Server:

- **Logical Characteristics:** Manages data storage and retrieval, ensuring data integrity and security. Supports SQL queries and concurrent access by multiple users.
- **Physical Characteristics:** Host on the MySQL server or dedicated server with high availability and redundancy.

Nature of Data and Control Interactions

- **Data Transmission:** The platform will transmit data such as project details, bids, messages, and payments between user devices and servers. Data must be encrypted to ensure security.
- **Control Interactions:** User actions such as submitting bids, sending messages, and processing payments will trigger API calls to the server, which will handle the requested operations and return responses.

3.1.3 Software Interfaces

3.2 Functional Requirements

R.1: User Registration and Authentication

The End User must provide all the necessary information in order to create their account in this System, also for the authentic and sophisticated user to upload the certifications and licenses.

R.1.1: User Registration

- **Input:**
 - First name, last name, email address, password
 - Optional: Profile picture, skills, portfolio link, GitHub Link
- **Processing:**
 - Validate email format and password strength
 - Store user details securely in the database.
- **Output:**
 - Success message confirming registration
 - Error message for missing or invalid input (e.g., email already exists)

R1.2: User Login

- **Input:**
 - Email address and password
- **Processing:**
 - Validate user credentials against the database
 - Create a session for the user upon successful login
- **Output:**

- User dashboard and success message if login is successful
 - Error message for invalid information
-

R.2: Profile Management

Profile Management allows users to create, update, and manage their personal or professional profiles. Users can edit details such as name, contact information, and preferences to ensure accurate and up-to-date information. This feature ensures a personalized experience and seamless interactions.

R.2.1: Update Profile

- **Input:**
 - Updated name, bio, skills, portfolio link, contact information
 - **Processing:**
 - Validate the format of the inputs (e.g., URLs provided)
 - Update user data in the database
 - **Output:**
 - Success message confirming the profile update
 - Error message if any input is invalid
-

R.3: Project Management

Project Management enables users to create, organize, and track their projects efficiently. It provides tools for setting deadlines, assigning tasks, monitoring progress, and managing resources, ensuring streamlined workflows and successful project completion.

R.3.1: Create a Project

- **Input:**
 - Project title, description, budget, required skills, project duration
 - Optional: Freelancer level, job type (fixed-price/hourly)
- **Processing:**
 - Validate the project description and budget
 - Store project data in the database with a unique project ID
 - Mark the project as "Open" for freelancer applications

- **Output:**
 - Success message and the listing of the project
 - Error message for missing or invalid data

R.3.2: Apply for a Project

- **Input:**
 - Freelancer's proposal , estimated completion time
- **Processing:**
 - Validate proposal and compatibility with the project budget
 - Store the freelancer's proposal data with the project reference in the database
 - Notify the project owner of new applications
- **Output:**
 - Success message confirming the submission
 - Error message for invalid proposals

R.3.3: View Project Status

- **Input:**
 - Freelancer or project owner accessing a project
- **Processing:**
 - Retrieve project data and its current status from the database
 - Display status based on the user's role (freelancer or project owner)
- **Output:**
 - Display project status (open, in-progress, completed)
 - Show relevant project details, and updates

R.3.4 Update the Project Details

- **Description:**

If the client need to update or add some feature , functionals or any other details of the project , it would be easily possible.However repeatedly updation of the project sometimes put the freelancer at discomfort situations,so update could be done only two times.On updation the client should not add works which is difficult for the freelancer, if freelancer think the updated work is more for him/her,he could do the original project without any update ,also he/she are able to request more compensations for the updated project to the client.

- **Input :**

- Select the project which client needs to update.

- **Processing :**

- Check whether the project is not update when only few days are remaining for the submission or the submission date is extended and price.
 - Verify that the client and freelancer agree on the revised timeline and compensation.
 - Repeated updates (more than thrice) is not allowed , and gives warning on the last update.

- **Output:**

- If the current freelancer agreed , the work/project would be updated.
 - Otherwise, the original project scope will be followed.

R.4: Review and Rating

Reviewing and ratings of the clients helps the client in what ways they need to change their specifications of their projects. Also freelancers can judge the behaviours from reviews and rating of the client. While that of freelancers , they could know in what field they could enhance their skills.

- **Input :**

- Username of end-user.
 - Rating on the :Quality of work ,Timeliness,Reasonable Project Price .

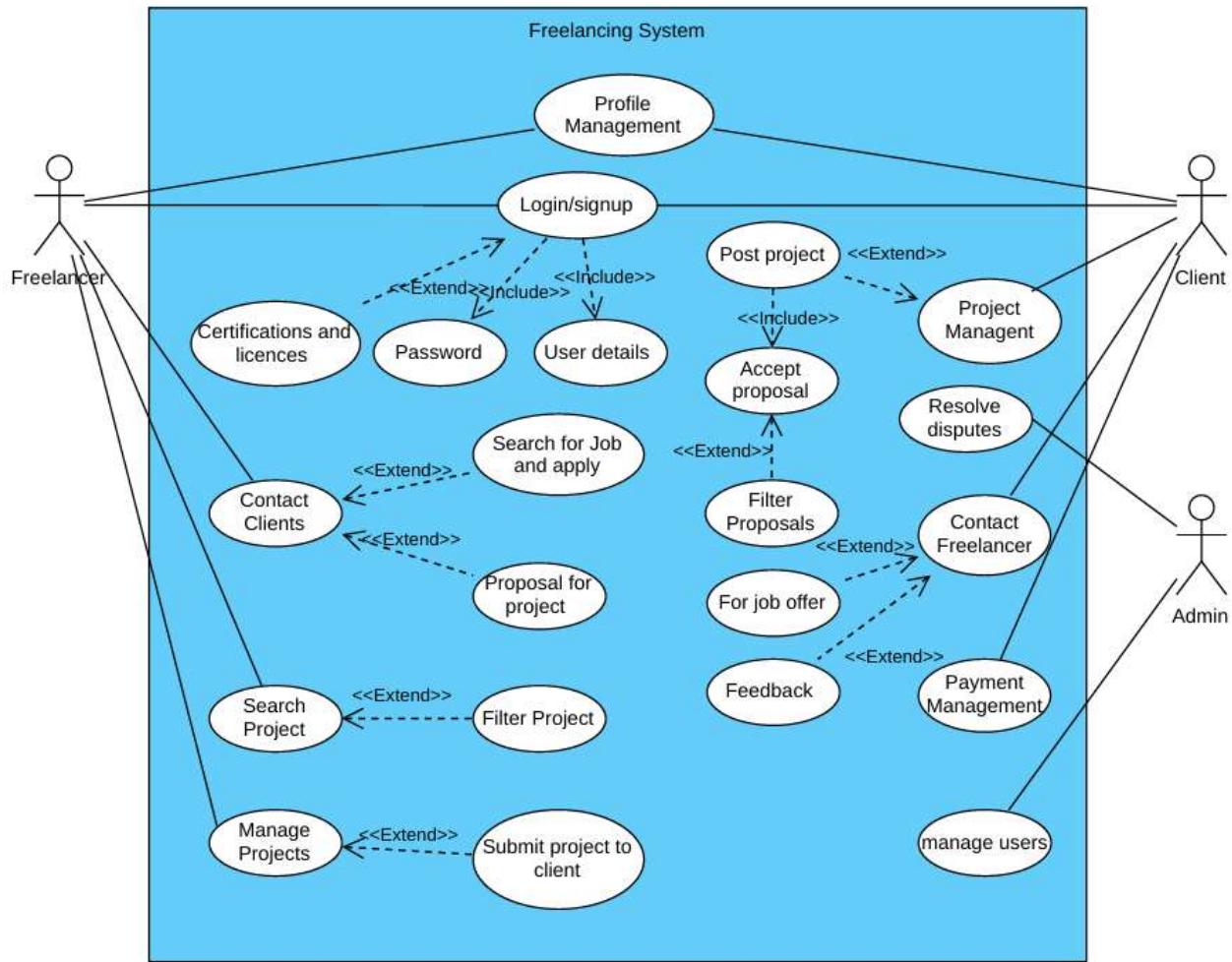
- **Processing :**

- Users can leave reviews for each other once a project is marked as completed.
 - Ratings and reviews are stored and displayed publicly on user profiles.

- **Output:**

- Review has been submitted publicly.

3.3 Use Case Model



3.3.1 Use Case 1 : Login and SignUp

Purpose -To allow registered users (freelancers and clients) to securely log in to the FreelanceConnect platform and access their accounts.

Priority - Highly prioritized as almost every interaction requires user name or ID with others and in order to carry out any operations.

Preconditions - User must have verified account on particular platform.And also they must have an access to the internet.

Actors – Freelancer and Clients

Extends – On registration, it provides an additional option to attach their official documentations such as Degree,Licenses and Certificates ,Work Experience etc.

Description

When any end-user accesses the website , they will be asked for their respective ID of email or any platform . Later they will be asked to fill the form based on whether they are client or freelancer and they have to fill in the details, for special authentication they could upload their certificates,licenses,degrees etc. After filling the details , they could browse and navigate the components of our website i.e better accessibility of the website.

Exception- If the given email ID is invalid , then the prompt will show an error message .Also if any field is empty , the end-user will be forced to fill in the details else they could not further access the website.

Post conditions - User registration is completed and redirected to their respective dashboard.

3.3.2 Use Case 2 :Contact Clients

Purpose -Contacting clients based on the requirement for the Freelancer. For example , if they want to contact(email or by any other means) client to discuss about project or for the job they wanted to apply.

Priority - The priority for contacting clients is depending on the Freelancer ,hence its Priority is medium.

Preconditions - User must have logged in and must have valid contact details of client

Actors – Freelancer

Extends – A freelancer can contact client based on his/her choice

Description

In order to contact a Client they must navigate to the client profile or job listings.Once they select the client which a freelancer wants to contact with they could select the contact client as well as write the purpose for contacting him/her.They could add some of description in contact form and after submitting ,it displays the confirmation message on freelancer side that the message is successfully sent to theClient.

Post conditions - Message is sent and has been notified to the Client.

3.3.3 Use Case 3 :Browse Project

Purpose -This allows a freelancer to browse and search for the projects that match their skills and interests

Priority - Medium ,because it's the Freelancer choice whether he/she wanted to carry out a project or a job.

Preconditions - Users must have logged in and there must be projects or jobs posted by a client.

Actors – Freelancer

Extends – A freelancer can filter the search operation based on what he/she required.

Description

A freelancer may want to browse through various projects to find suitable work opportunities. This involves searching for projects based on various criteria such as skills, budget, deadline, and location. Freelancers can view project details, save projects for later, or proceed to apply for them.

Postconditions-Freelancer finds and views detailed project listings.Freelancer may choose to apply or save the project for later.

3.3.4 Use Case 4 :Manage Project

Purpose -To allow freelancers to manage their projects effectively, including viewing the status of all projects they have worked on and submitting project deliverables.

Priority - High - It is essential for freelancers to keep track of their projects and submit deliverables on time.

Preconditions - Users must have logged in and there must be active projects listed on the freelancer's profile..

Actors – Freelancer

Extends – Viewing project status of all projects done till now and ubmitting project deliverables.

Description

Freelancers need to manage their ongoing and completed projects. This involves viewing the status of each project, tracking deadlines, and submitting deliverables. The system should facilitate easy project management and ensure freelancers can keep their clients updated.

Postconditions-Freelancer views the status of their projects.Freelancer submits project deliverables

3.3.5 Use Case 5:Upload Project

Purpose- To allow clients to upload project requirements and relevant documents for freelancers to review and respond to.

Priority - High - It is essential for clients to provide detailed project requirements to attract suitable freelancers and ensure the project's success.

Preconditions - User (client) must be logged in.

Actors – Client

Description

A client needs to upload detailed project requirements and relevant documents to provide clear instructions to freelancers. This process involves selecting the project, uploading the necessary files, and ensuring all information is complete and accurate.

Flow:

- **Client Navigates to Upload Section:**
 - i. Client logs in and selects the 'Manage Projects' option from the dashboard.
 - ii. Client selects the relevant project for which they want to upload requirements.
- **Upload Project Requirements:**
 - i. System displays an upload interface where the client can add files.
 - ii. Client selects files from their device and uploads them to the platform.
- **Add Descriptions (Optional):**
 - i. Client adds descriptions or notes for each file (optional).
 - ii. Client verifies that all necessary files are included such as deadlines .
- **Submit :**
 - i. Client submits the upload.
 - ii. System confirms the submission and updates the project listing with the new information.
 - iii. Freelancers are notified of the new upload.

Alternative Flow:

- **Client Cancels Upload:**

- Client decides not to proceed with the upload and cancels the action.
 - System retains any partially uploaded files for future completion.

Exceptions:

- **Upload Error:**

- If there is an issue with the file upload (e.g., file size too large, network issues), the system displays an error message and prompts the client to try again or contact support.
 -
 - System retains any partially uploaded files for future completion.

Postconditions- Enables a client to upload and will be enlisted to the job lists which will be seen by all the freelancers.

3.3.6 Use Case 6 :View Proposal

Purpose - To enable clients to view proposals submitted by freelancers for a project and accept the proposal they find suitable

Priority - High -It is essential for clients to review and there must be a request from the freelancer to initiate the project.

Preconditions - Users must have logged in and there must be active projects listed on the freelancer's profile..

Actors – Clients

Extends – The Client can also filter the proposal .Also accordance to his/her liking , can accept any one of the proposal.

Description

Client when hit the manage item ,it lead to page where all the projects are uploaded .And there by he/she select a particular project which is yet to be assigned to any client, and from view proposals sections ,displays all proposals from the freelancer.A clients filter the order of proposals list by changing filtering sections.And lastly if a client find a proposal which satisfies all his requirements and as well as the freelance is a noteworthy and thereby he/she can accept such proposal .

Postconditions-Freelancer views the status of their projects.Freelancer submits project deliverables

3.3.7 Use Case 7 :Payment

Purpose - To facilitate the secure and timely payment from clients to freelancers upon completion of a project.

Priority - High - Ensuring accurate and prompt payments is critical for the platform's trust and functionality.

Preconditions -User (freelancer and client) must be logged in.There must be a completed project with agreed-upon payment terms.

Actors – Clients

Extends –A client can have multiple ways to pay the agreed-payment to the freelancer.Such as UPI payment or a direct bank transfer.

Description

After a freelancer completes a project, the client needs to release the payment. This process involves verifying the project completion, processing the payment through a secure gateway, and confirming the transaction to both parties.If in case of any error while the payment ,if would corrected , and once again the Clients has to sent the payment to freelancer

Postconditions-Notification to the Client that the payment has been successfully completed and also freelancer is being notified that they have received the payment.

3.3.8 Use Case 8 :Profile Management

Purpose - To allow users (freelancers and clients) to manage their profiles, including updating personal information, skills, experience, and other relevant details.

Priority - High -It is always good practice to keep the profile details updating .

Preconditions -User (freelancer and client) must be logged in.

Actors – Clients and Freelancer

Description

Users need to manage their profiles to keep their information current and relevant. This involves updating personal details, adding new skills or experiences, and managing visibility settings. The system should provide a user-friendly interface for profile management. The user can

navigate to profile from the dashboard and by selecting update profile , he/she could update information as well as add the certificates, licenses and many more.

Postconditions-User profile is updated with the new information.System saves and reflects the changes across the platform.

4 Other Non-functional Requirements

4.1 Performance Requirements

P1. Response Time:

The system should respond to user actions (e.g., clicking a button, submitting a form) within 2 seconds during peak usage times to ensure a smooth user experience.

P2. Scalability:

The system should be able to handle significantly many concurrent users without a drop in performance, ensuring that it remains operational and efficient under high load.

P3. Availability:

The system should have an uptime of 99.9%, ensuring it is available to users almost all the time, with minimal downtime for maintenance.

P4. Data Processing:

The system can process and display search results within a few seconds to ensure users can quickly find and access relevant projects or freelancers.

4.2 Safety and Security Requirements

4.2.1. User Data Protection:

All user data must be encrypted in transit and at rest using industry-standard encryption protocols to protect against unauthorized access.

4.2.2. Data Privacy Compliance:

The system complies with relevant data privacy regulations to ensure the protection of user data and privacy rights.

4.2.3. Regular Security Audits:

The system undergoes regular security audits to identify and address potential vulnerabilities.

4.2.4. Mobile Connection Security:

All communications between the mobile app and the server must be secured using HTTPS to prevent interception and ensure data integrity.

4.3 Software Quality Attributes

4.3.1 Reliability:

The system must be reliable, with a target mean time between failure of at least 1 year. This will be achieved by rigorous testing, regular maintenance, and monitoring to identify and resolve issues promptly.

4.3.2 Adaptability:

The system should be adaptable to support different types of projects and client requirements. This will be achieved by designing the system with modular components and configurable settings that can be easily adjusted without extensive re-coding.

4.3.3 Usability:

The system must be user-friendly and intuitive, with a user satisfaction rating of at least 90%. This will be achieved by conducting user testing, gathering feedback, and iterating on the design to improve the user interface and experience.

4.3.4 Maintainability:

The system should be easy to maintain, with a goal of resolving issues within 24 hours. This will be achieved by maintaining comprehensive documentation, using clean and modular code, and implementing automated testing to quickly identify and fix bugs.

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>