Final Year Project

Software Requirement Specification

For

Go Green

Bachelor of Computer Science (BSCS)

By

Sr.	Name	Roll NO	Mobile No	Email
1	Kamran Siddique	Fa-2019-BSCS-355-H	0331-6640896	kamransiddique09@gmail.com
2	Maaz Mushtaq	Fa-2019-BSCS-354-H	0313-6804904	chmaaz21@gmail.com

Supervised By:

Sir Ahsan Ali Haroon **___________________ (Signature)**

Department of Computer Science

Lahore Garrison University

Lahore

Table of Contents

1. Introduction

1.1 Purpose

A gardening marketplace currently named Go Green (v: 1.0) is a project that is directed at helping users to create and manage jobs related to landscaping, gardening, and lawn care. Users will be able to post jobs, receive bids from talents, and hire professionals to complete the work. The app will also include a marketplace where users can buy fertilizers and other gardening products, and where talents can sell their products and services. The scope of the project includes the development of the mobile app, as well as all associated features, such as user registration, job posting, bidding, payment processing, and communication tools. The app will be available on Android devices. The project does not include the development of a web-based platform or integration with third-party tools and services. Additionally, the project will not include the development of advanced gardening tools or features such as plant identification or watering reminders. These features may be considered for future phases of development.

1.2 Document Conventions

This document uses the following conventions.

Convention	**Short Description**
User	A user, whose primary goal is to provide jobs and hire talents for executing those.
Talent	A nursery owner, whose primary goal is to apply for jobs and execute those that have been entrusted to them.
MONGODB	"a powerful, flexible, and scalable general-purpose database."
MONGOOSE	Node.js module simplifying connection to MongoDB
NODE.JS	"a JavaScript platform—a way to run JavaScript"
NOSQL	A common term for non-relational databases
JSON	JavaScript Object Notation, "A Simple Representation of Data"

1.3 Intended Audience and Reading Suggestions

Anyone with some basic knowledge of programming can understand this document. The document is intended for Developers, Software architects, Testers, Project managers and Documentation Writers. But anyone with programming background and some experience with UML can understand this document.

It is divided into 5 phases with sections 3, 4, 5 being intended for developers and software managers but other sections can be understood by anyone having little knowledge about software.

This Software Requirement Specification also includes:

- Overall description of the product
- External interface requirements
- System Features
- Other non-functional requirements

1.4 Product Scope

The gardening app will provide users with a platform to create and manage jobs related to landscaping, gardening, and lawn care. The app will include the following features:

- **User registration:** Users will be able to create profiles and log in to the app using email and password, or through social media accounts.
- **Job posting:** Users will be able to post jobs, including a description of the work, location, and budget. Talents will be able to browse and bid on available jobs.
- **Bidding:** Talents will be able to submit bids for jobs, including an estimated cost and timeline for completion. Users will be able to review and compare bids, and hire a talent for the job.
- **Payment processing:** The app will include a secure payment system for users to pay for jobs and for talents to receive payment for completed work.
- **Communication tools:** The app will include a messaging feature for users and talents to communicate and collaborate on jobs.

2. Overall Description

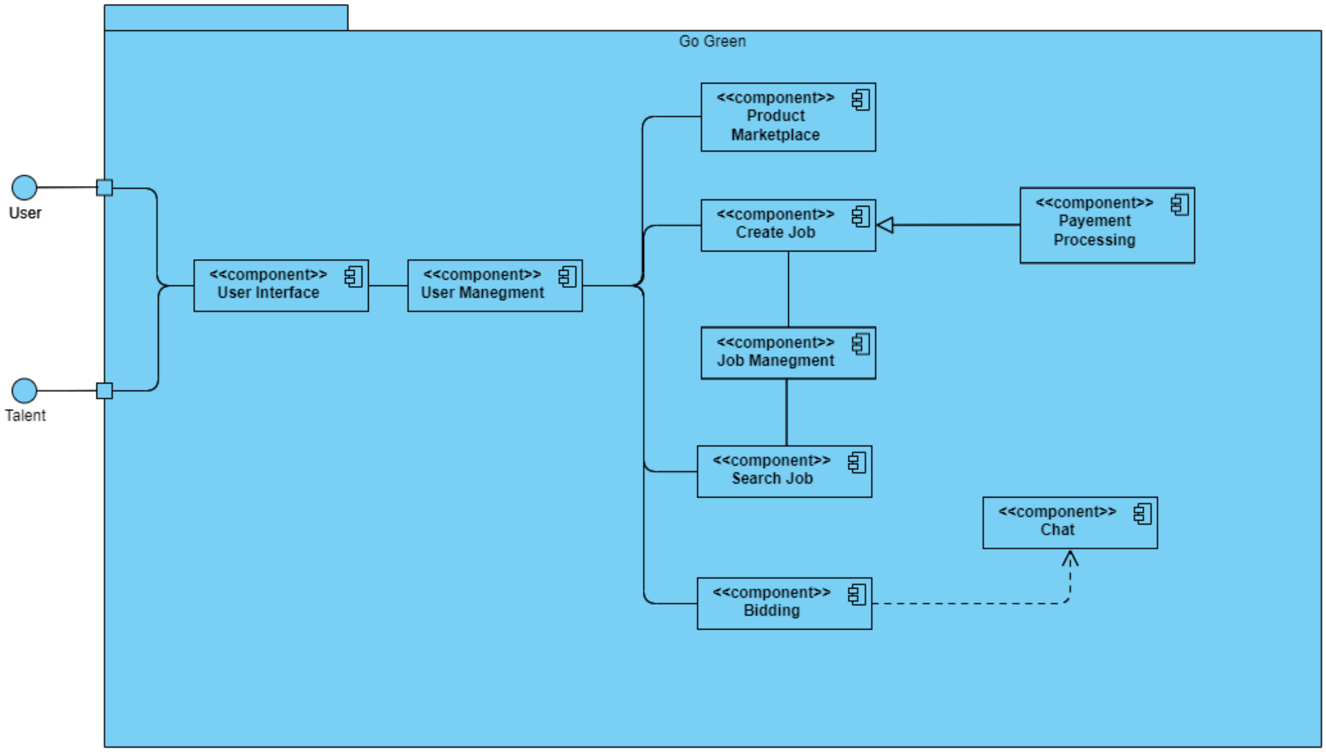
2.1 Product Perspective

This is a self-contained product. This app is designed to provide a platform for users to find and hire talent for landscaping, gardening, and lawn care services. The app will also include a marketplace for users to buy gardening products and for talent to sell their products and services. The app will offer a user-friendly interface, a wide range of services, and a secure payment system. The app will be targeted at homeowners, small business owners, and professionals in the landscaping and gardening industry. The app will be developed in phases, with future phases potentially including additional features and functionalities.

The app includes these components:

- **User interface:** The interface through which users interact with the app, including screens for creating and managing jobs, browsing and bidding on jobs, and communicating with other users.
- **Job management:** The component responsible for managing the creation, bidding, and completion of jobs. This might include functionality for posting jobs, reviewing bids, and marking jobs as complete.
- **Payment processing:** The component responsible for handling financial transactions within the app, including payment for jobs and the sale of products.
- **Product marketplace:** The component responsible for managing the sale of gardening products and services within the app. This might include a catalog of available products and functionality for users to browse and purchase products.
- **User management:** The component responsible for managing user accounts, including registration, authentication, and profile management.

Here is a component diagram:


Go Green
User
Talent
<<component>> User Interface
<<component>> User Manegment
<<component>> Product Marketplace
<<component>> Create Job
<<component>> Payement Processing
<<component>> Job Manegment
<<component>> Search Job
<<component>> Chat
<<component>> Bidding

2.3 User Classes and Characteristics

Users of the app should be able to post jobs that need to be completed, or browse and bid on jobs that have been posted by other users. For example, homeowners may post a job for landscaping work, and talents may submit bids to complete the work. They should be able to withdraw money from their account balance, either as payment for completed work or as a refund for cancelled jobs. This functionality would be facilitated through a secure payment system integrated into the app. Users of the app should be able to sell products or services through a marketplace feature within the app. For example, talents may offer a range of gardening products or services, such as fertilizers, tools, or lawn care services, and users may browse and purchase these items through the app.

Here are some user classes and characteristics:

- **Homeowners:** These users are homeowners who are looking for help with landscaping, gardening, and lawn care tasks. They may have limited knowledge or experience in these areas, and may be interested in hiring professionals to complete the work.
    - These users may be interested in finding affordable, reliable, and high-quality services. They may be willing to pay more for professional services, but may also be price-sensitive.
- **Small business owners:** These users are business owners who are looking for help with landscaping and gardening tasks at their commercial properties. They may be interested in hiring talents to complete one-time or ongoing tasks.
    - These users may be interested in finding cost-effective solutions that help to maintain the appearance of their properties. They may be willing to pay more for high-quality services, but may also be price-sensitive.
- **Professionals:** These users are talents who offer landscaping, gardening, and lawn care services through the app. They may have a range of skills and experience, and may offer a variety of services.
    - These users may be interested in finding a steady stream of work through the app. They may be willing to bid on a wide range of jobs, but may also have preferences for certain types of work or locations.

2.4 Operating Environment

The app is only compatible with Android smartphone running version 7 or higher, and requires an internet connection to function properly. The app may require a certain amount of processing power, memory, and storage to function properly.

2.5 Design and Implementation Constraints

Following are the design and implementation constrains that needed to be considered:

Front-end:

- Should be build using xml, Kotlin
- Design patterns to be used:
    - MVVM

    - Repository Pattern
- Retrofit Library to perform REST API calls

Back-end:

- The server side of the app should be built in NodeJS and Express.js
- Microservices architecture should be used in order to ensure reliability and availability of the app.
- MongoDB for database management
- Mongoose to interact with database
- Should implement CORS
- express-nosql-sanitizer for NoSQL injection

Deployment:

- Digital ocean
- Docker container and Kubernetes

2.6 User Documentation

User documentation for the mobile application will be provided in the form video tutorials.

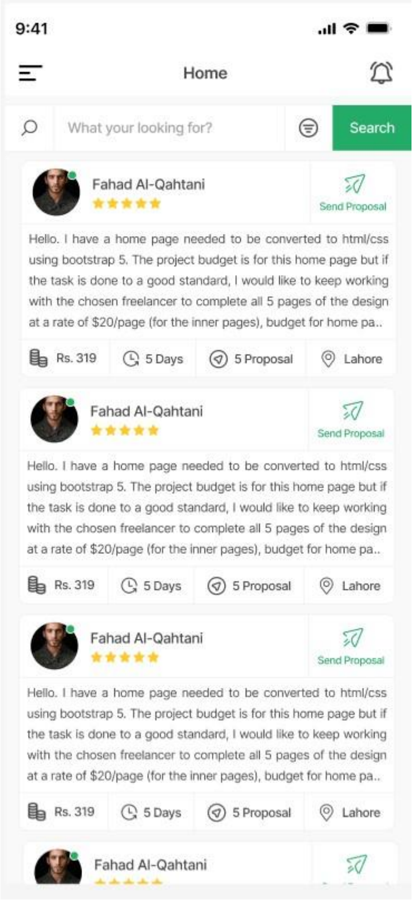
2.7 Assumptions and Dependances

No specific assumptions or dependencies are considered at this time.

3. External Interface Requirements

3.1 User Interfaces

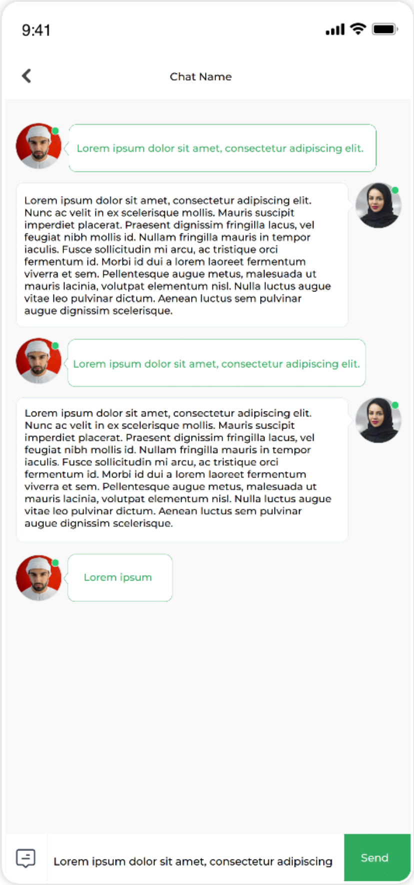
Here is some UI design for the app that needed to be follow.



Hello. I have a home page needed to be converted to html/css using bootstrap 5. The project budget is for this home page but if the task is done to a good standard, I would like to keep working with the chosen freelancer to complete all 5 pages of the design at a rate of $20/page (for the inner pages), budget for home pa..

Hello. I have a home page needed to be converted to html/css using bootstrap 5. The project budget is for this home page but if the task is done to a good standard, I would like to keep working with the chosen freelancer to complete all 5 pages of the design at a rate of $20/page (for the inner pages), budget for home pa..

Hello. I have a home page needed to be converted to html/css using bootstrap 5. The project budget is for this home page but if the task is done to a good standard, I would like to keep working with the chosen freelancer to complete all 5 pages of the design at a rate of $20/page (for the inner pages), budget for home pa..



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

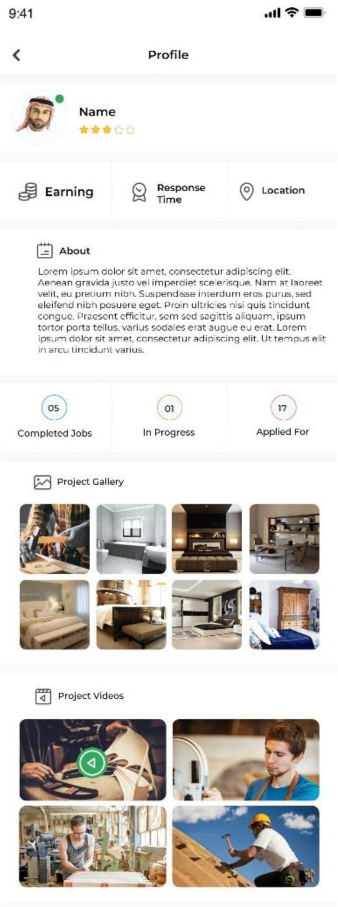
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac velit in ex scelerisque mollis. Mauris suscipit imperdiet placerat. Praesent dignissim fringilla lacus, vel feugiat nibh mollis id. Nullam fringilla mauris in tempor iaculis. Fusce sollicitudin mi arcu, ac tristique orci fermentum id. Morbi id dui a lorem laoreet fermentum viverra et sem. Pellentesque augue metus, malesuada ut mauris lacinia, volutpat elementum nisl. Nulla luctus augue vitae leo pulvinar dictum. Aenean luctus sem pulvinar augue dignissim scelerisque.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac velit in ex scelerisque mollis. Mauris suscipit imperdiet placerat. Praesent dignissim fringilla lacus, vel feugiat nibh mollis id. Nullam fringilla mauris in tempor iaculis. Fusce sollicitudin mi arcu, ac tristique orci fermentum id. Morbi id dui a lorem laoreet fermentum viverra et sem. Pellentesque augue metus, malesuada ut mauris lacinia, volutpat elementum nisl. Nulla luctus augue vitae leo pulvinar dictum. Aenean luctus sem pulvinar augue dignissim scelerisque.

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean gravida justo vel imperdiet scelerisque. Nam at laoreet velit, eu pretium nibh. Suspendisse interdum eros purus, sed eleifend nibh posuere eget. Proin ultricies nisi quis tincidunt congue. Praesent efficitur, sem sed sagittis aliquam, ipsum tortor porta tellus, varius sodales erat augue eu erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tempus elit in arcu tincidunt varius.

05	01	17
Completed Jobs	In Progress	Applied For

3.2 Hardware Interfaces

Any android OS supported smartphone having version greater than or equal to 7 with internet connection.

3.3 Software Interfaces

Operating system: The app should be designed to work with the latest version of the Android operating system, and should be tested on multiple versions and devices to ensure compatibility.

Database: The app will use MongoDB as its database management system and it will connect to the MongoDB server via an API to store and retrieve data. The data should be properly formatted and validated before being stored in the database to ensure data integrity.

Payment gateway: The app may use a third-party payment gateway i.e. Easypaisa, XPay to facilitate secure transactions between users and talents. This will ensure that all transactions are processed securely.

Push notification service: The app may use a third-party push notification service i.e. FCM to send notifications to users when events occur or messages are received. This could include things like new job postings, bids on jobs, or updates to account balances. The app should be designed to interface with this service to send and receive notifications.

Cloud storage: The app may use cloud storage service to store and retrieve user data, such as job listings, user profiles, and photos. This will ensure that data is accessible from any device, and will allow for easy sharing of data between users and talents. The app should interface with the cloud storage service through an API.

3.4 Communication Interfaces

In-app messaging: The messaging feature should allow users to send and receive text messages, images, and files. It should also allow users to create group chats and have the ability to mark messages as read or unread. Additionally, it should have the ability to block or report users if any suspicious or inappropriate behavior is detected.

Push notifications: The app may use push notifications to alert users of important events or updates. These notifications should be customizable, so users can choose which events or updates they want to be notified about. Additionally, the app should also have a mechanism to manage push notifications so the user can turn them on or off, as well as set custom notification sounds.

Email notifications: The app may send email notifications to users to alert them of important events or updates. These emails should include a clear subject line and a brief message, as well as a link back to the app for more information.

All the above communication interfaces should be designed with security and privacy in mind. For example, the messaging feature should be designed to encrypt messages.

All Network Communications will use HTTPS

4. System Features

4.1 Job listing and bidding

4.1.1 Description and Priority

The Job listing and bidding feature allows users to create and post jobs for landscaping and gardening tasks. Users can search for jobs by location, skill level, and price. They can also filter the list of jobs by location, skill level, and price. Users can view detailed information about a job, such as its description, location, and cost. They can also submit a bid for a job, specifying the amount they are willing to pay and a message that describes their qualifications and experience. Users can view the status of their bids, accept or reject bids and rate and review Talent after job completion. This feature is considered as a high priority feature as it is a core part of the app, allowing users to find and bid on jobs related to landscaping and gardening. Without this feature, the app would not provide a valuable service to its users.

4.1.2 Stimulus/Response Sequence

Stimulus: User opens the app and navigates to the job listing page.

Response: The app displays a list of available jobs, including their location, skill level, and price.

Stimulus: User filters the list of jobs by location, skill level, and price.

Response: The app updates the list of jobs to match the user's search criteria.

Stimulus: User selects a job from the list to view more details.

Response: The app displays more information about the selected job, such as its description, any attachments, location, and cost.

Stimulus: User submits a bid for the job by clicking on the "Apply" button.

Response: The app confirms that the bid has been submitted and displays the bid's status as "submitted" and also shows the bid amount.

Stimulus: The job owner receives a notification that there is a new bid and he can accept or reject it.

Response: If the job owner accepts the bid, the app updates the bid's status to "accepted" and sends a notification to the user. If the job owner rejects the bid, the app updates the bid's status to "rejected" and sends a notification to the user.

Stimulus: Once the job is completed, the talent is prompted to submit the final invoice.

Response: Once the invoice is submitted, the job owner can review the invoice and approve it, and the payment will be automatically processed. After the payment, the user is prompted to rate and review the talent.

4.1.3 Functional Requirements

REQ-1. Users must be able to search for jobs by location, skill level, and by price, keyword or category.

REQ-2. Users must be able to view detailed information about a job, such as its detailed description, location, cost, the skill required for the job.

REQ-3. Users must be able to submit a bid for a job, specifying the amount they are willing to pay and a message that describes their qualifications and experience.

REQ-4. Users must be able to view the status of their bids in real-time, including whether they have been accepted, rejected, or are still pending.

REQ-5. Users must be able to accept or reject bids.

REQ-6. Users must be able to rate and review talent after the job is completed on various parameters such as professionalism, punctuality, communication skills, and quality of work.

REQ-7. Job owner must be able to receive a notification for new bid and should be able to accept or reject it, with a reason of rejection.

REQ-8. Talent should be able to submit the final invoice, specifying the time and materials used.

REQ-9. Job owner should be able to review the invoice and approve it, and the payment should be processed automatically through a secure payment gateway.

REQ-10. Show a green message to talent if he already applied for the job.

4.2 Marketplace

4.2.1 Description and Priority

The marketplace feature allows users to view a list of products and services that are available for purchase. Users can search for products by category or keyword. Users can view detailed product information, including pricing, availability, and reviews. Users can purchase products and services through a secure payment gateway. Users can rate and review the products and services they purchase. Users can also post and sell their own products or services on the marketplace. This feature is considered as a medium priority feature as it is an important part of the app, but its functionality is not crucial for the core functionality of the app. Without this feature, users can still use the app to search, bid and complete jobs, but the marketplace feature provides more options for users to purchase or sell products and services related to gardening.

4.2.2 Stimulus/Response Sequence

Stimulus: User opens the app and navigates to the marketplace page.

Response: The app displays a list of available products and services related to gardening, including their images, description, price, and reviews.

Stimulus: User filters the list of products by category, keyword, and price.

Response: The app updates the list of products to match the user's search criteria.

Stimulus: User selects a product from the list to view more details.

Response: The app displays more information about the selected product, such as its detailed description, images, location of the seller, and reviews.

Stimulus: User click on “Buy” button to purchase the product

Response: The app displays the checkout page, where user can enter the shipping details and can select the payment method and payment gateway. Note: The talent is responsible for shipping the product.

Stimulus: User confirms the purchase.

Response: The app confirms the purchase and sends a notification to the seller.

Stimulus: User receives the product

Response: The app prompts the user to rate and review the product and the seller.

4.2.3 Functional Requirements

REQ-1. Users must be able to view a list of products and services available for purchase on the marketplace.

REQ-2. Users must be able to search for products by category, keyword, and price.

REQ-3. Users must be able to filter the list of products by multiple criteria such as location, category, keyword, and price.

REQ-4. Users must be able to view detailed information about a product, including pricing, images, availability, and reviews.

REQ-5. Users must be able to purchase products and services through a secure payment gateway.

REQ-6. Users must be able to rate and review products and services they have purchased.

REQ-7. The talent must be able to post and sell their own products or services on the marketplace.

REQ-8. Users must be able to view their order history.

REQ-9. Users must be able to cancel their orders before shipping.

REQ-10. Users must be able to return the products with a valid reason.

REQ-11. Users must be able to view the refund history.

REQ-12. Users must be able to view the review history.

4.3 User profiles and reviews

4.3.1 Description and Priority

The User profiles and reviews feature allows users to create a profile that includes information such as their name, location, skills, and experience. Users can also upload photos, videos, and documents related to their skills and experience. They can then use this profile to bid on and manage jobs. Users can also view the profiles of other users, including their ratings and reviews, and can also rate and review other users after a job completion. This feature is considered as a medium priority feature. It is an important part of the app as it allows users to showcase their skills, experience and helps in building trust among the users. Without this feature, users would not be able to find and hire talent they can trust. However, this feature is not as critical as the Job listing and bidding feature and the app can still function without it.

4.3.2 Stimulus/Response Sequence

Stimulus: User opens the app and navigates to the profile page.

Response: The app displays the user's own profile, including their name, location, skills, and experience.

Stimulus: User wants to edit their profile.

Response: The app displays an edit button which allows the user to edit their profile, including uploading photos, videos, and documents related to their skills and experience.

Stimulus: User wants to view another user's profile

Response: The app displays the selected user's profile, including their name, location, skills, job history and review, and experience.

4.3.3 Functional Requirements

REQ-1. Users must be able to create and edit their own profiles, which should include information such as their name, profile photo, location, skills, and experience.

REQ-2. Users must be able to upload photos, videos, and documents related to their skills and experience as part of their profile.

REQ-3. Users must be able to view the profiles of other users, including their ratings, job history and reviews.

REQ-4. Users must be able to rate and review other talent after a job completion.

REQ-5. Users must be able to view their own job and reviews history

REQ-6. Users must be able to reply to the reviews

4.4 Communications

4.4.1 Description and Priority

The Communications feature allows users to send and receive messages, notifications to and from other users within the app. This feature allows them to share information, and confirm appointments. Users can also receive notifications, such as job bidding notifications, job completion notifications, and payment notifications. Users can also receive automated email communications, such as order confirmation, shipping details and payment details. This feature is considered as a high priority feature as it allows users to communicate and share information with each other effectively and in a timely manner. Without this feature, the app would not be able to provide a seamless and efficient service to its users.

4.4.2 Stimulus/Response Sequence

Stimulus: User opens the app and navigates to the messages page

Response: The app displays the user's conversation history with other users.

Stimulus: User sends the message to already started conversion.

Response: The app sends the message to the recipient, who will receive a notification of the new message, then the recipient can respond to that message.

Stimulus: User receives a notification of a new message

Response: The app displays a notification on the screen and sends a push notification to the user, indicating that there is a new message.

Stimulus: User wants to view the message

Response: The app displays the message in the conversation history.

4.4.3 Functional Requirements

REQ-1. Users must be able to send and receive messages to and from other users within the app.

REQ-2. Users must be able to receive notifications in real-time, such as job bidding notifications, job completion notifications, and payment notifications.

REQ-3. Users must be able to receive automated email communications, such as order confirmation, shipping details and payment details.

REQ-4. Users must be able to read the messages offline

REQ-5. Users must be able to mark a message as read, unread, and delete

REQ-6. Users must be able to reply to the messages

REQ-7. Users must be able to attach the files, images and videos to the messages

4.5 Account management

4.5.1 Description and Priority

The Account management feature allows users to create and manage their accounts within the app. Users can view and update their personal information, change their password, and manage their billing information. This feature is considered as a high priority feature as it allows users to manage their account and perform essential tasks, such as managing personal information, managing billing information. Without this feature, the app would not be able to provide a seamless and efficient service to its users.

4.5.2 Stimulus/Response Sequence

Stimulus: User opens the app and navigates to the account management page.

Response: The app displays the user's account information, including name, email address, phone number, and billing information.

Stimulus: User wants to update their personal information.

Response: The app displays forms that allow the user to update their personal information, such as name, email address, phone number, and address.

Stimulus: User wants to change their password.

Response: The app prompts the user to enter their current password, and then enter a new password and confirm it.

4.5.3 Functional Requirements

REQ-1. User must be able to update the user's account information, including name, email address, phone number.

REQ-2. Users must be able to manage their billing information, such as adding, editing and deleting credit/debit card information

REQ-3. User must be able to change their password after they have provided their current password, and then entered a new password and confirmed it.

REQ-4. User must be able to configure notification settings.

4.6 Admin Panel

4.6.1 Description and Priority

The admin panel allows the administrator to manage the users and the content of the app. It includes tools to manage user accounts, review and approve job listings and bids, view transaction history and revenue, monitor app usage and performance, and access analytics and reports. The admin panel also allows the administrator to block or unblock a user, and also to add, update, or delete the products and services. This feature is considered as a high priority feature, as it allows the administrator to effectively manage the app and ensure that it is running smoothly. Without this feature, the administrator would not be able to effectively monitor the app and make necessary changes to improve its performance.

4.6.2 Stimulus/Response Sequence

Stimulus: Administrator logs in to the admin panel

Response: The app displays the dashboard, which includes an overview of the app's performance and key metrics

Stimulus: Administrator wants to view a list of users

Response: The app displays a list of users, including their name, email address, and account status.

Stimulus: Administrator wants to block a user.

Response: The app prompts the administrator to confirm the action, and then blocks the user's account.

Stimulus: Administrator wants to approve job listing

Response: The app displays a list of job listings, and allows the administrator to approve or reject each listing.

Stimulus: Administrator wants to view transaction history

Response: The app displays a list of transactions, including the date, amount, and status of each transaction.

Stimulus: Administrator wants to add/update/delete products

Response: The app allows the administrator to add/update/delete the products and services.

4.6.3 Functional Requirements

REQ-1. The administrator must be able to view a list of users and their account information, such as name, email address, and account status.

REQ-2. The administrator must be able to block or unblock a user's account.

REQ-3. The administrator must be able to approve or reject job listings.

REQ-4. The administrator must be able to view transaction history, including the date, amount, and status of each transaction.

REQ-5. The administrator must be able to manage the products and services, such as adding, updating, or deleting them.

REQ-6. The administrator must be able to view the revenue and transaction history.

REQ-7. The administrator must be able to monitor app usage and performance, such as the number of active users and the number of job listings.

REQ-8. The administrator must be able to manage the payment gateway

5. Other Non-Functional Requirements

5.1 Performance Requirements

The app should respond quickly to user inputs, with minimal latency and high throughput

5.2 Safety Requirements

Error handling: The app must handle errors and unexpected situations in a safe and predictable way.

Safety testing: The app must be rigorously tested to ensure that it meets safety requirements and standards.

Safety communication: The app must provide clear and precise communication to the user in case of an emergency or malfunction.

Safety validation: The app must be validated to ensure that it meets its safety requirements throughout the software development life cycle.

5.3 Security Requirements

- **User authentication:** Users must provide valid credentials in order to access the app.
- **Encryption:** Sensitive data, such as user passwords and transaction information, must be encrypted to protect it from unauthorized access.
- **Authorization:** Users can only access the resources and perform actions for which they have been granted permission.
- **Secure communication:** Data transmitted between the app and the server must be sent over a secure communication channel.
- **Input validation:** The app must validate all input to prevent injection attacks.
- **Session management:** The app must manage user sessions to prevent session hijacking.
- **Audit logging:** The app must log all security-relevant events for auditing and forensic purposes.
- **Password management:** The app must enforce strong password policies and handle password storage securely.
- **Data Leakage:** The app should prevent the accidental or intentional leakage of sensitive data

5.4 Software Quality Attributes

Functionality: The app should provide the required functionality and meet the needs of the users.

Reliability: The app should be dependable and perform its intended functions without failure.

Usability: The app should be easy to use and understand for the end-users.

Efficiency: The app should use resources (e.g., memory, network, CPU) in an optimal way to achieve its goals.

Maintainability: The app should be easy to maintain and modify to accommodate changing requirements.

Security: The app should protect sensitive data and prevent unauthorized access.

Testability: The app should be easy to test, including automated testing, to ensure that it meets its requirements.

5.5 Business Rules

- Users must be registered and logged in to use the app.
- Talent can bid on a maximum of 30 jobs in a day
- Only registered Talent can bid on jobs.
- Users can only withdraw money from their account after their job is completed
- The user can't bid on their own job listing.
- A user can only rate and review a freelancer after the job is completed
- User's account will be blocked if they violate any rules of the app
- Administrator can't bid on any job.
- A 10% service fee will be applied to each job completed by a freelancer and an additional withdrawal fee of Rs. 2 will be incurred for each withdrawal made by a user.

References

UI Idea: https://www.behance.net/gallery/93455247/Hareef-UI-UX-App-Design?tracking_source=search_projects_recommended%7Cfreelancer+app

Software Requirements Specification: https://github.com/ProjectRecommend/docs/blob/master/design-docs/SRS_final.md#introduction-terminology