Go Green

Submitted By

Kamran Siddique (Fa-19-BSCS-355)

Maaz Mushtaq (Fa-19-BSCS-354**)**

Session 2019-2023

Supervised By

Sir Ahsan Ali Haroon

Department of Computer Science & Information Technology

Lahore Garrison University

Lahore

Go Green

A project submitted to the

Department of Computer Science

In

Partial Fulfilment of the Requirements for the

Bachelor's Degree in Computer Science

By

Kamran Siddique

Maaz Mushtaq

Internal Supervisor

Sir Ahsan Ali Haroon _____
Lecturer
Department of Computer Science

Chairperson

Dr. Arfan Ali Nagra _____
Head of Department
Department of Computer Science

COPYRIGHTS

This is to certify that the project titled "**Go-Green**" is the genuine work carried out by **Kamran Siddique, Maaz Mushtaq**, student of BSCS of Computer Science Department, Lahore Garrison University, Lahore during the academic year 2019-23, in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Kamran Siddique _____

Maaz Mushtaq _____

DECLARATION

I hereby declare that this project titled **"Go Green"** is not copied out from any other source. This authentic work done by signatory, in practical achievement for the degree "Bachelor of Science in Computer Science" at Computer Science Department, Lahore Garrison University, Lahore. Besides, it is declared that we have designed this WEB or project report completely on the basis of our own intention under the kind leadership of our internal supervisor Ms. Laraib Kanwal.

All the analysis, design & system development have been accomplished by the signatory. Moreover, this project has not been submitted to any other college or university.

Kamran Siddique _____

Maaz Mushtaq _____

ACKNOWLEDGEMENTS

First and for most uncounted thanks to ALMIGHTY ALLAH who created us worthy of all praise who mentor us in crises. Secondly all salutations are presented to our beloved PROPHET MUHAMMAD (PBUH) who empower us to recognize our creator and whose life is the best example for us. During this project, we received support, assistance or practical help from many people and it is bliss to acknowledge their intentions.

Our heartiest prayers go to our parents, siblings and to all family members for their assistance and moral support. Our success is possible only due to prayers or support of our mothers who really loves us and always pray for us in every moment of our life.

We are very thankful to all our family members, senior's fellows and we also feel vastly in debt to our supervisor Sir Ahsan Ali Haroon for providing us precise research-oriented opinions, advices, backing and assistance during the project and throughout the academic session.

DEDICATION

I dedicated this work to First of all, ALLAH Almighty who glorified us with the knowledge and bravery to complete this responsibility with elegance. Secondly, my affectionate supportive family whose prayers, advises and continuous support played a major role in achieving this goal. Finally, to the Lahore Garrison University and Especially, the department of Computer Science.

Table of Contents

List of Abbreviation

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"

Abstract

Gardening beginners and experts alike have quickly grown to love gardening apps, which give them digital tools to advance their horticultural activities. However, a look at the available gardening applications reveals some noteworthy drawbacks that restrict their usefulness and potential influence. This thorough literature analysis examines the state of gardening applications today and explains their functional limitations and flaws. Restricted plant offerings, communication difficulties, a narrow range of service offerings, few product possibilities, lacking of gardening tools, nursery owner's registration gaps, a lack of selling platforms, and a lack of job creation and binding mechanisms are a few of these. This review proposes a cutting-edge gardening app as a potential remedy to these noted constraints. By providing a wide variety of functionalities, this ground-breaking app seeks to fill the gaps in the gardening app ecosystem. These characteristics include a wide range of plants, improved communication tools, thorough services, a variety of goods and services, simple access to gardening equipment, user registration tools, a platform for the sale of services, and a system for creating and binding jobs. This assessment emphasizes the proposed app's disruptive potential while offering a thorough overview of the rapidly changing gardening app market. This creative app has the ability to completely change how gardening is done in the digital era by solving the present limitations and providing a comprehensive gardening experience. The abstract emphasizes how the gardening community is poised to undergo a fundamental transformation, with this new application paving the way for increased connectivity, information sharing, and productivity in the gardening industry.

Chapter 1
Introduction

1.1 Background

The idea behind the Go Green gardening marketplace stems from the increasing demand for landscaping, gardening, and lawn care services, as well as the growing interest in eco-friendly practices. Many homeowners and businesses are looking for reliable and skilled professionals to take care of their outdoor spaces, while others seek to engage in gardening and landscaping as a hobby or profession. However, the process of finding suitable talent and managing gardening projects can be challenging and time-consuming.

To address these issues, the Go Green app (v: 1.0) is being developed to provide a convenient and efficient platform where users can easily connect with talented professionals for their gardening needs. The app will not only cater to those seeking gardening services but also offer a marketplace for buying fertilizers and other gardening products. Furthermore, it will provide an opportunity for talented individuals to showcase and sell their products and services, fostering a thriving community of gardeners and gardening enthusiasts.

1.2 Overview

The Go Green app aims to revolutionize the gardening industry by creating a centralized platform that streamlines the process of finding, hiring, and managing gardening projects. By focusing on landscaping, gardening, and lawn care, the app caters specifically to the green space industry, providing a specialized and tailored solution for users.

Users of the Go Green app will have the following key features at their disposal:

User Registration: A simple and user-friendly registration process will enable individuals to create profiles and access the app's functionalities. They can register using their email and password or through their existing social media accounts.

Job Posting: Users can post details of gardening projects they need assistance with, specifying the type of work required, location, and budget. This feature empowers users to describe their needs accurately and attract suitable talents.

Bidding: Talented professionals can browse through the available jobs and submit bids based on the project's requirements, offering an estimated cost and timeframe for completion. This bidding system fosters healthy competition among talents, ensuring users receive competitive and reasonable offers.

Payment Processing: To facilitate seamless transactions, the app will incorporate a secure payment system that allows users to pay for completed jobs and talents to receive payment for their services.

Communication Tools: Effective communication is essential for successful project management. The app will provide messaging features, enabling users and talents to collaborate, clarify project details, and maintain clear communication throughout the process.

1.3 Objective

he main objectives of the Go Green app (v: 1.0) are:

1. **Connect Users and Talents:** To create a platform that connects users seeking gardening services with talented professionals, simplifying the process of finding suitable experts for their projects.

2. **Enhance User Experience:** To provide a user-friendly and intuitive app interface that ensures a seamless and enjoyable experience for all users, regardless of their level of technical expertise.

3. **Comprehensive Services:** Offer a wide range of gardening services, including pruning, plant care, landscaping, and more, to cater to diverse user needs.

4. **Streamline Project Management:** To offer robust features for job posting, bidding, and communication, facilitating efficient project management and fostering collaboration between users and talents.

5. **Expand Product Offerings:** Continually expand the selection of gardening products, including fertilizers, organic materials, and tools, to ensure users have access to high-quality supplies.

6. **Promote Green Practices:** To encourage eco-friendly gardening practices by providing a marketplace for green products and services, thereby contributing to a more sustainable environment.

7. **Scalability and Future Development:** To design the app with a modular and extensible architecture, enabling easy integration of additional features and functionalities in future updates, such as advanced gardening tools and plant identification capabilities.

Chapter 2
Literature Review

The advent of gardening apps has marked the beginning of new era of convenience and accessibility for gardening enthusiasts and professionals. However, it is essential to critically assess the existing landscape of these apps to identify areas where improvements can be made. This literature review delves into the constraints of current gardening apps and introduces a novel app designed to address these limitations.

The Current State of Gardening Apps

Limited Plant Variety: Many gardening apps in the market offer a restricted selection of plants, limiting users' choices for their gardens. This constraint can be frustrating for those looking to explore diverse flora.

Communication Challenges: Communication within existing gardening apps often falls short. The absence of direct channels for interaction hinders knowledge exchange among users, posing a barrier to the growth of a collaborative gardening community.

Inadequate Service Coverage: A significant portion of gardening apps offers only basic services, leaving users in search of specialized assistance for their unique gardening needs.

Scarce Product Options: The availability of fertilizers and organic products within these apps is often limited, impeding users' ability to access a wide range of gardening supplies.

Gardening Tools Deficiency: Gardening tools and equipment are frequently overlooked in current gardening apps, forcing users to seek these essential items through alternative means.

Registration Gaps: The absence of registration options for nursery owners, pruners, and caretakers restricts the growth of a connected gardening community and limits opportunities for professionals to offer their services.

Absence of Selling Platforms: Existing apps generally lack platforms through which nursery owners can directly showcase and sell their products and services to users.

Lack of Job Creation and Binding Systems: Many gardening apps fail to facilitate job creation or provide mechanisms for binding users and nursery owners together for ongoing gardening projects.

The Innovative Solution: Addressing Limitations

In response to the identified constraints of current gardening apps, an innovative gardening app emerges as a potential game-changer. This novel app seeks to bridge these gaps by offering a comprehensive suite of functionalities:

Expansive Plant Selection: The app boasts an extensive inventory of plants, granting users the freedom to choose from a diverse array of flora for their gardens.

Enhanced Communication: Prioritizing seamless communication, the innovative app facilitates direct interaction channels between users, nursery owners, and gardening professionals, fostering the exchange of knowledge and expertise.

Comprehensive Services: A wide range of services is available within the app, from basic pruning to specialized care, ensuring that all users' gardening needs are met within one platform.

Diverse Product Offerings: Users have access to a broad selection of fertilizers and organic products, enabling them to select the most suitable options for their plants.

Gardening Tools: Gardening tools and equipment are readily accessible within the app, simplifying the procurement of essential instruments for gardening activities.

User Registration: The app facilitates the registration of nursery owners, pruners, and caretakers, fostering a connected gardening community where professionals can offer their services and users can seek expert assistance.

Selling Platform: Nursery owners can showcase and sell their products and services directly to users through the app, expanding their businesses and reach.

Job Creation and Binding: In order to maintain the continuity of gardening projects, the app includes a job creation and binding mechanism that fosters long-term partnerships between user and nursery owners.

Chapter:2
Problem Definition

2.1 Problem Statement

The Go Green gardening marketplace seeks to revolutionize the landscaping, gardening, and lawn care services industry by creating a convenient and efficient platform for users to connect with skilled professionals and suppliers offering eco-friendly products. Despite its potential, several critical challenges need to be addressed to ensure the app's success and provide maximum value to its users.

The first challenge is the fragmented nature of the gardening services industry. With numerous individual professionals and small businesses operating independently, finding reliable and skilled gardeners becomes a daunting task for homeowners and businesses alike. The lack of a centralized platform leads to a time-consuming and inconsistent process of sourcing gardening services, often resulting in a mismatch of expectations and service quality.

Secondly, the limited availability of eco-friendly gardening products poses a significant obstacle to sustainable gardening practices. Homeowners and gardening enthusiasts keen on adopting environmentally friendly approaches face difficulties in accessing a diverse range of organic fertilizers, sustainable tools, and other green products. The Go Green app must address this issue by partnering with eco-conscious suppliers and creating a robust marketplace for environmentally friendly products.

Third, talented gardeners and horticulturists, especially those starting their businesses or working independently, struggle to gain visibility and attract clients. Without a dedicated platform to showcase their skills and services, these professionals may find it challenging to establish themselves in the competitive market. The Go Green app must offer a comprehensive profile system for gardeners to exhibit their expertise, customer reviews, and previous work, giving them a competitive edge and increasing their chances of gaining clients.

User trust and safety concerns are critical factors that can make or break the success of the Go Green gardening marketplace. Users need assurance of service quality, secure transactions, and protection of their personal information.

Chapter:3

Software Requirement Specification

3.1 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.

3.2 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

3.3 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.

3.4 Overall Description

3.4.1 Product Perspective

The Go Green gardening marketplace is a self-contained app that serves as a platform for users to find and hire talent for landscaping, gardening, and lawn care services. It also includes a marketplace for buying gardening products and for talented professionals to sell their products and services. The app features a user-friendly interface, offering a wide range of services, and ensures secure payment processing. It targets homeowners, small business owners, and professionals in the landscaping and gardening industry. The app will be developed in phases, with future possibilities for additional features. Key components include the user interface for interaction, job management for posting and bidding on jobs, payment processing for financial transactions, a product marketplace for sales, and user management for registration and authentication.

3.4.2 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.

3.4.3 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.

3.4.4 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

- Docker & Kubernetes
- Microservices architecture should be used in order to ensure reliability and availability of the app.
- NATS Streaming Server for syncing data between microservices
- Socket io for real time chat
- MongoDB for database management
- Mongoose to interact with database
- Should implement CORS
- Nodemailer to send emails to the user
- Hpp to protect against header pollution attacks
- Json web token for authentication
- express-NoSQL-sanitizer for NoSQL injection
- rate limiter to protect against DDoS attacks
- multer for uploading files
- AWS S3 Bucket for file storage purpose
- Ingress Nginx Load Balancer
- Ngrok for development

3.4.5 User Documentation

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

3.4.6 Assumptions and Dependencies

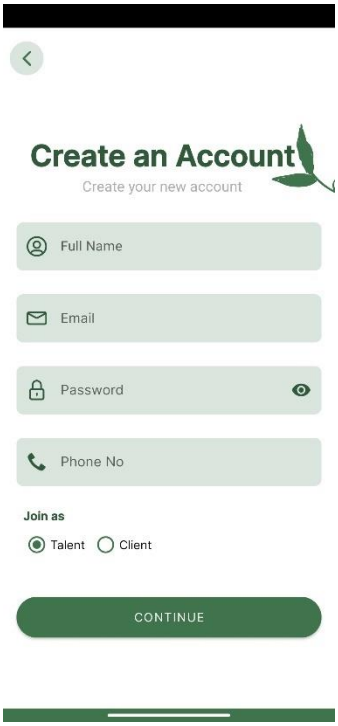
- System having windows 10 least.
- System having 100 mb of space.
- System having 8Gb ram.
- Android Studio.
- SDK for Android.

3.5 External Interface Requirements

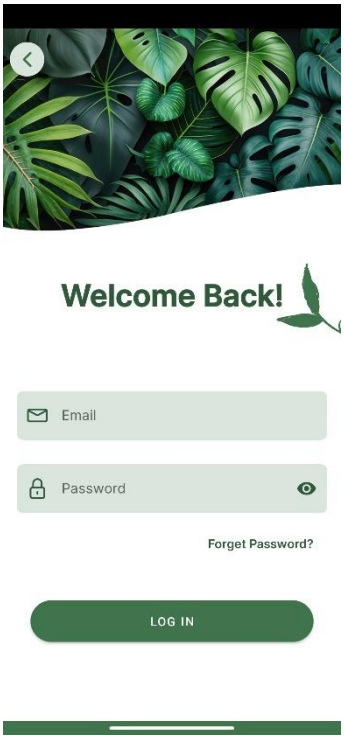
3.5.1 User Interfaces

Welcome Screen

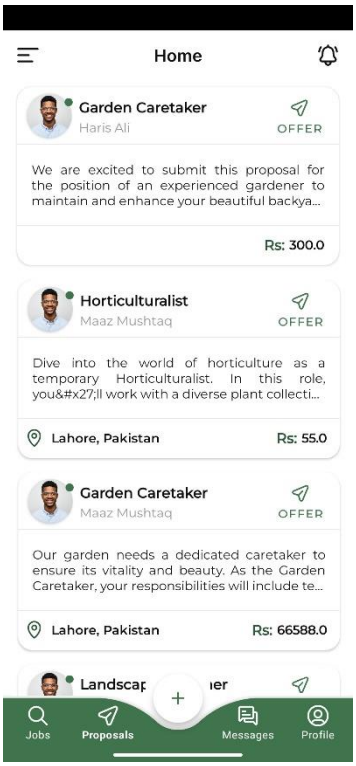
Sign-up Screen



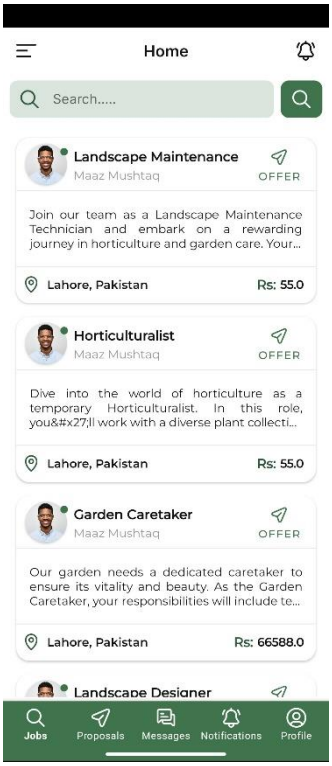
Sign-In Screen



Home Client



Home Talent



3.5.2 Hardware Interfaces

- Any android OS supported smartphone having version greater than or equal to 7 with internet connection.
- System having 4gb ram.

3.5.3 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., Easy-Paisa, X-Pay 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.

3.5.4 Communications 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

3.6 System Features

3.6.1 (Job listing and bidding) 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.

3.6.1.1 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.

3.6.1.2 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.

3.6.2 Marketplace 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.

3.6.2.1 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.

3.6.2.2 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.

3.6.3 User profiles and reviews

3.6.3.1 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.

3.6.3.2 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.

3.6.3.3 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

3.6.4 Communications

3.6.4.1 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.

3.6.4.2 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.

3.6.4.3 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

3.6.5 Account management

3.6.5.1 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.

3.6.5.2 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.

3.6.5.3 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.

3.6.6 Admin Panel

3.6.6.1 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.

3.6.6.2 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.

3.6.6.3 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

3.7 Other Nonfunctional Requirements

3.7.1 Performance Requirements

1. **Responsiveness:** The app must respond promptly to user interactions, such as tapping buttons, scrolling, and submitting forms. Users should experience minimal lag or delays to ensure a smooth and responsive user experience.

2. **Loading Time:** The app should load quickly and efficiently, particularly when accessing job listings, product catalogs, and user profiles. Faster loading times contribute to user satisfaction and retention.

3. **Scalability:** The app should be designed to handle increasing numbers of users, job postings, and product listings without significant performance degradation. Scalability is crucial to accommodate growth and ensure consistent performance during peak usage periods.

4. **Reliability:** The app should be highly reliable, with minimal downtime or system failures. Users should be able to access the app and perform essential tasks without disruptions or errors.

5. **Payment Processing Efficiency:** The payment processing system must be secure and efficient, ensuring quick and accurate financial transactions. Users expect a seamless payment experience without delays or technical glitches.

6. **Search and Filtering Speed:** The search and filtering functionality, such as finding specific gardening services or products, should return results swiftly and accurately.

7. **Communication and Messaging:** The messaging system should facilitate real-time communication between users, allowing them to interact without delays or message delivery issues.

3.7.2 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.

3.7.3 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

3.7.4 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.

3.7.5 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.

Chapter:4
Methodology

4.1 Methods and Approaches:

- **Agile Software Development**: The project follows an agile software development methodology, allowing for iterative and incremental development, frequent communication, and adaptability to changing requirements.
- **Use Case Driven Approach:** The app's functionalities and requirements are described in use cases, which could be used to drive the development process.

4.2 Tools and Techniques:

- **Android Studio:** The primary Integrated Development Environment (IDE) for Android app development using Kotlin and XML.
- **Node.js and Express.js:** Backend development using Node.js as the server-side runtime and Express.js as the web application framework.
- **MongoDB:** As the NoSQL database management system to store and retrieve data related to users, jobs, products, and reviews.
- **Retrofit Library:** For performing REST API calls between the mobile app and the backend server.
- **Docker and Kubernetes:** For containerization and efficient management of the app's deployment on the cloud platform, specifically Digital Ocean.

4.3 Algorithms:

- **Search Algorithm:** To facilitate users searching for jobs and products by location, skill level, category, or keyword, the app may use search algorithms to provide relevant results.

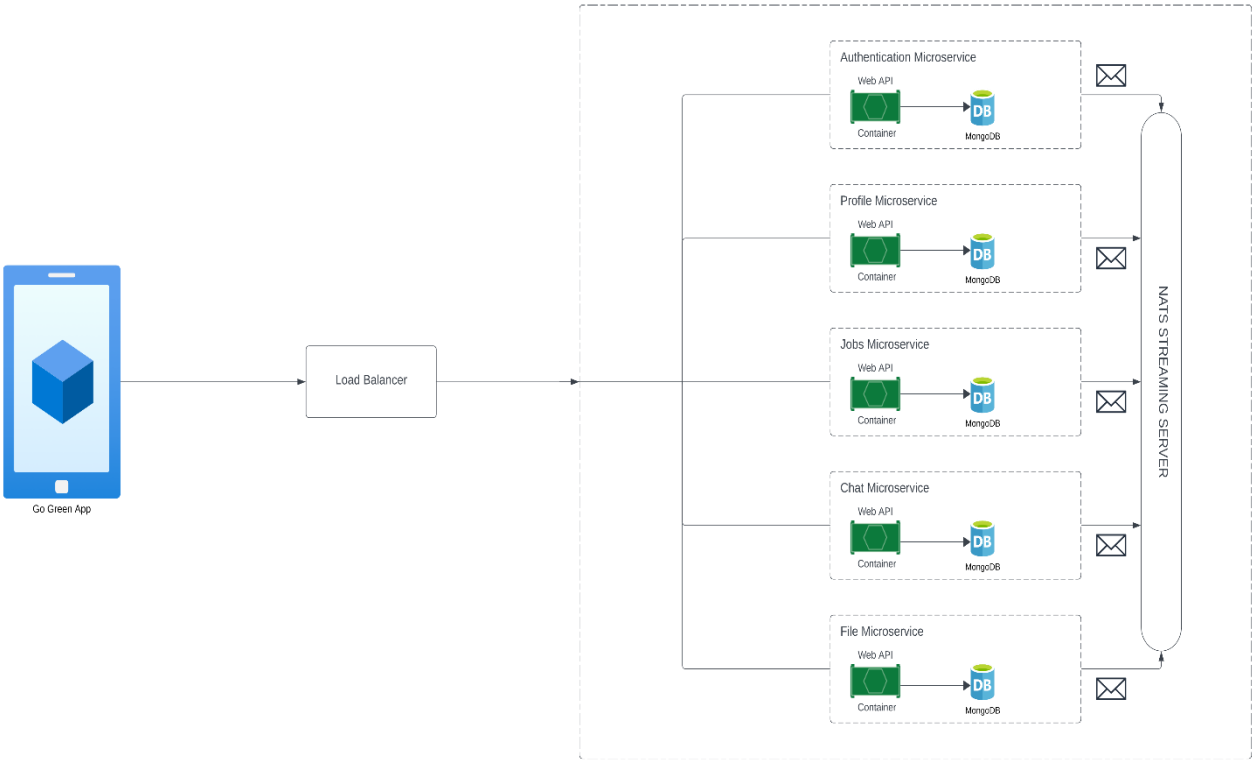
- **Payment Processing:** The app may utilize secure payment processing techniques and APIs provided by the chosen payment gateway (e.g., Easypaisa, XPay) to handle financial transactions.

4.4 Security Aspects:

- **User Authentication:** The app will likely use secure authentication methods (e.g., OAuth, JWT) to ensure only authorized users can access the app's features.
- **Encryption**: Sensitive data like passwords, payment information, and user data may be encrypted to protect against unauthorized access.
- **Input Validation**: To prevent injection attacks and ensure data integrity, input validation measures may be implemented to sanitize user inputs.
- **Secure Communication**: The app might use HTTPS (SSL/TLS) for secure communication between the mobile app and the backend server.
- **Access Control**: The app may implement access control mechanisms to restrict users' access to specific resources and operations based on their roles and permissions.
- **Error Handling:** The app should handle errors gracefully and avoid displaying sensitive information to users in error messages to prevent potential security risks.

Chapter:5

Detailed Design and Architecture



Authentication:

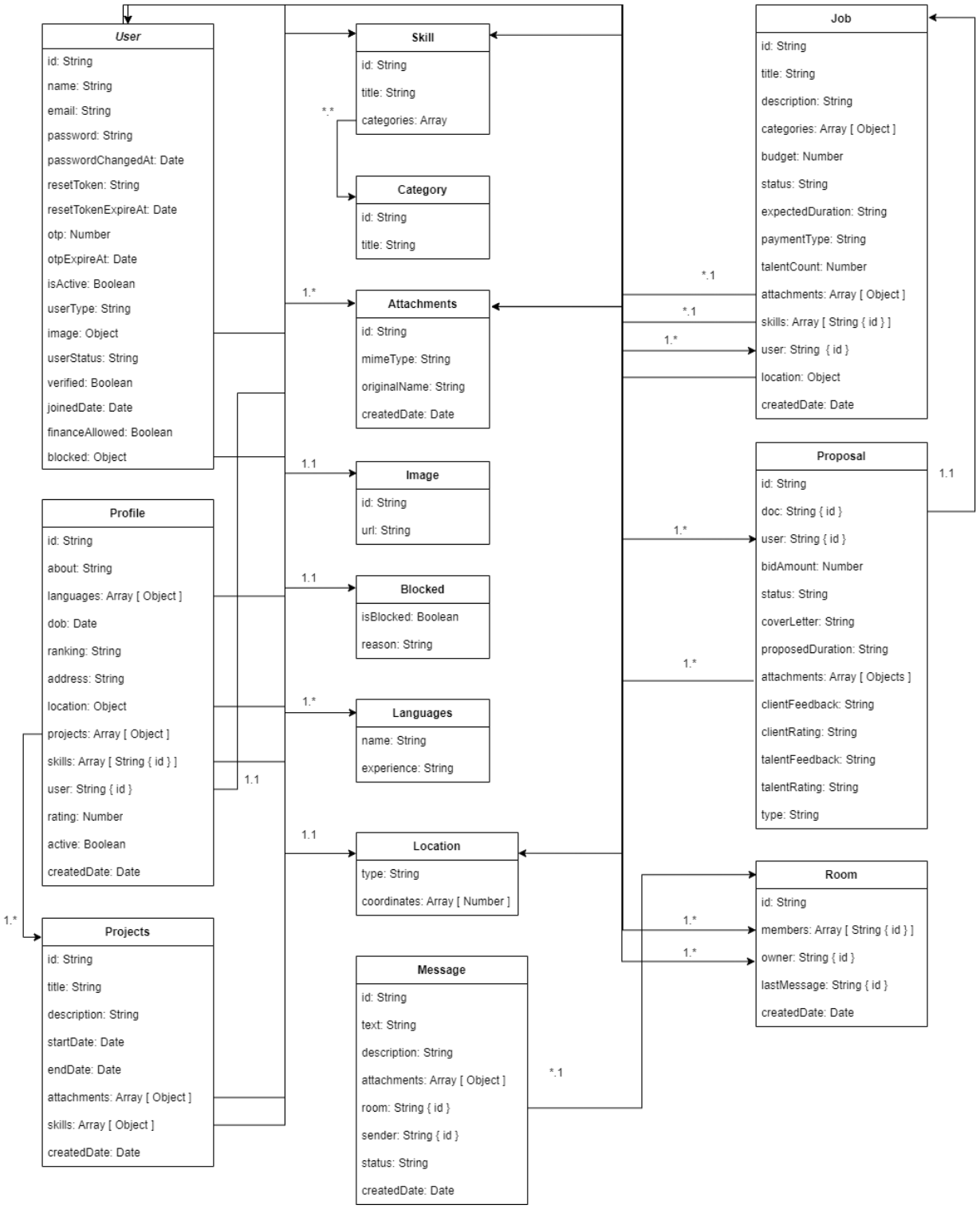
- Register User with Email
- Register User using third party like Google, Twitter, Facebook
- Login User
- Authenticate User

Profile:

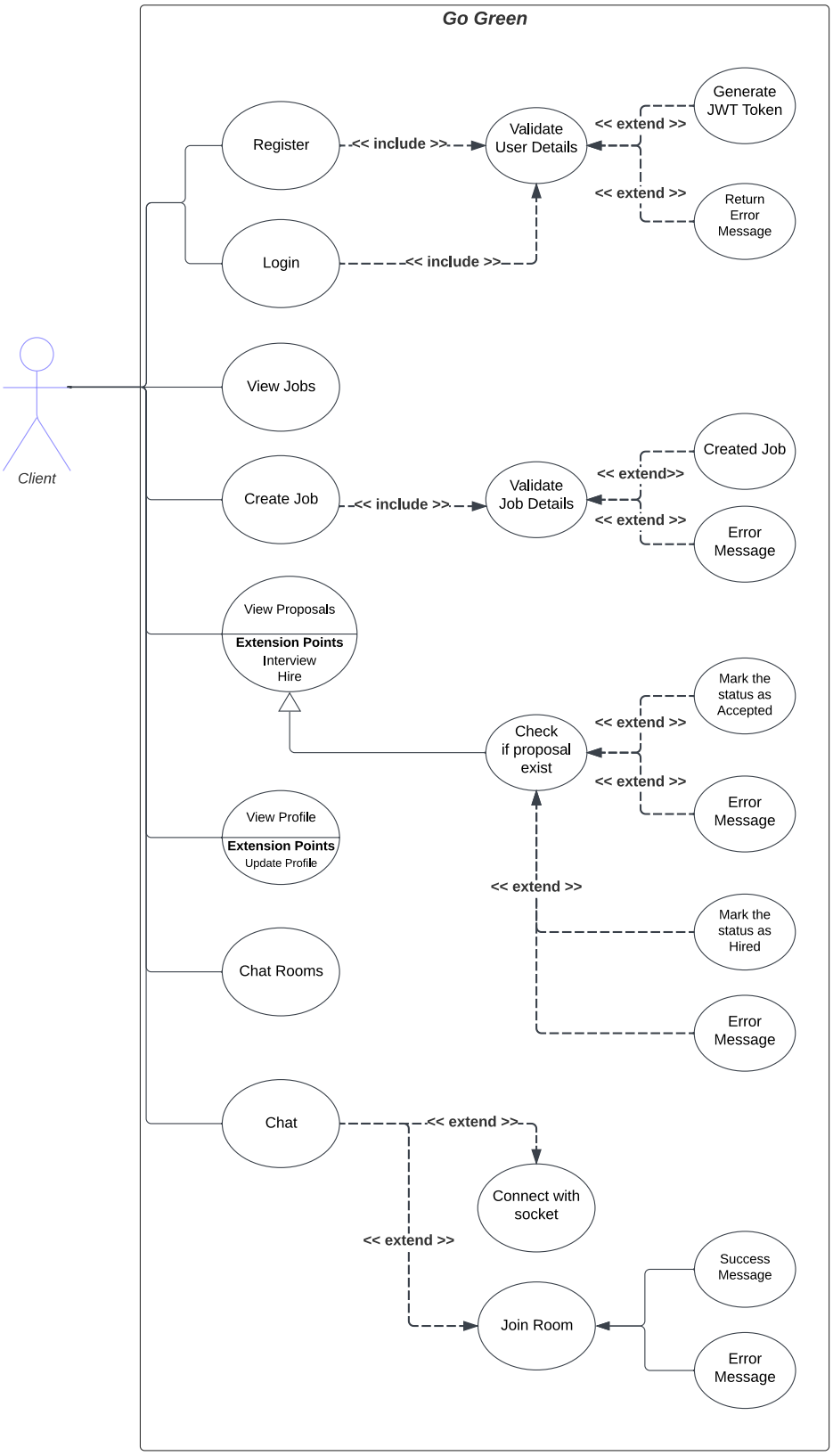
- Update User Info
  - Name
  - Email
  - Password
  - Phone No
  - Registration No

- Delete Account

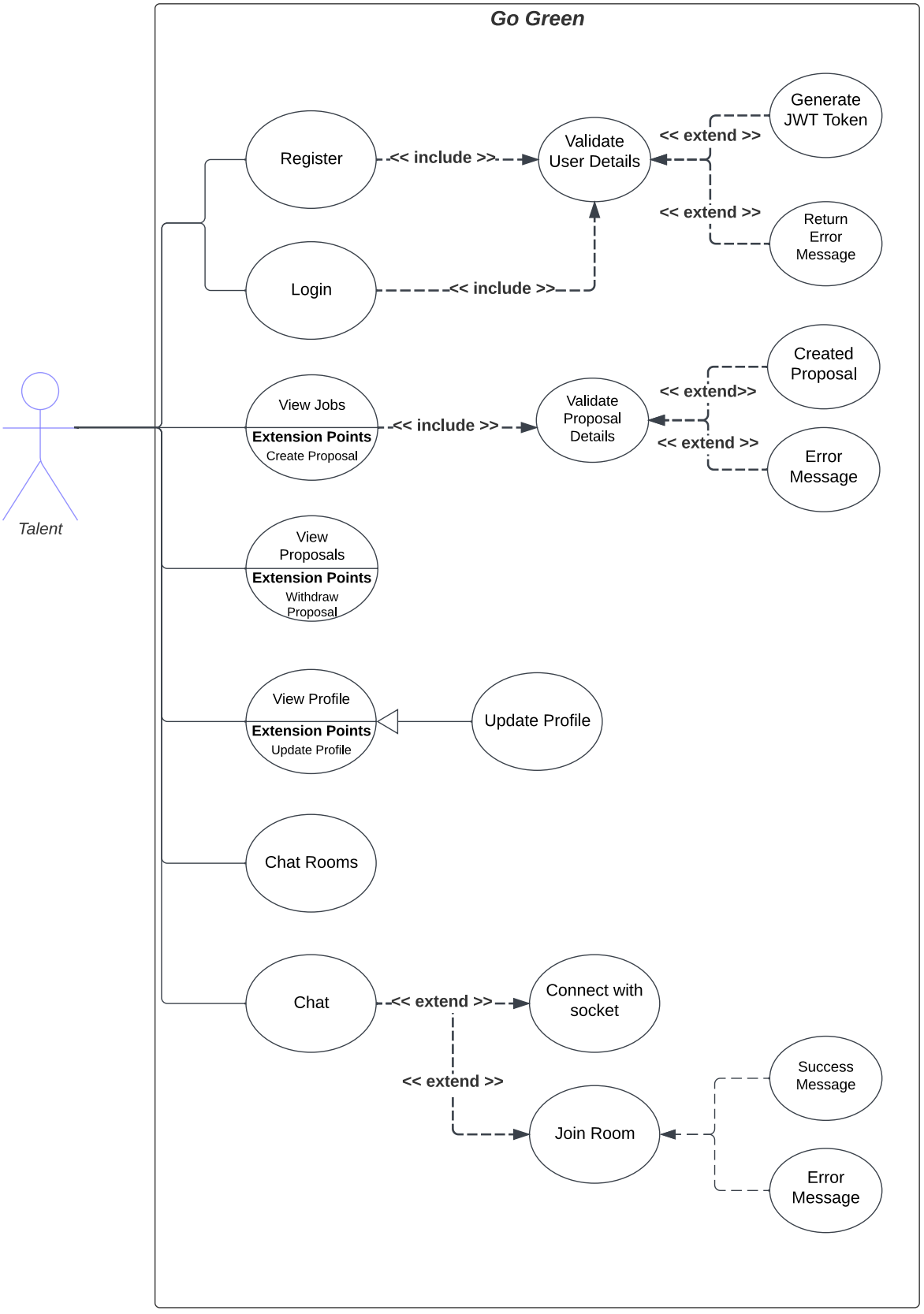
Class Diagram



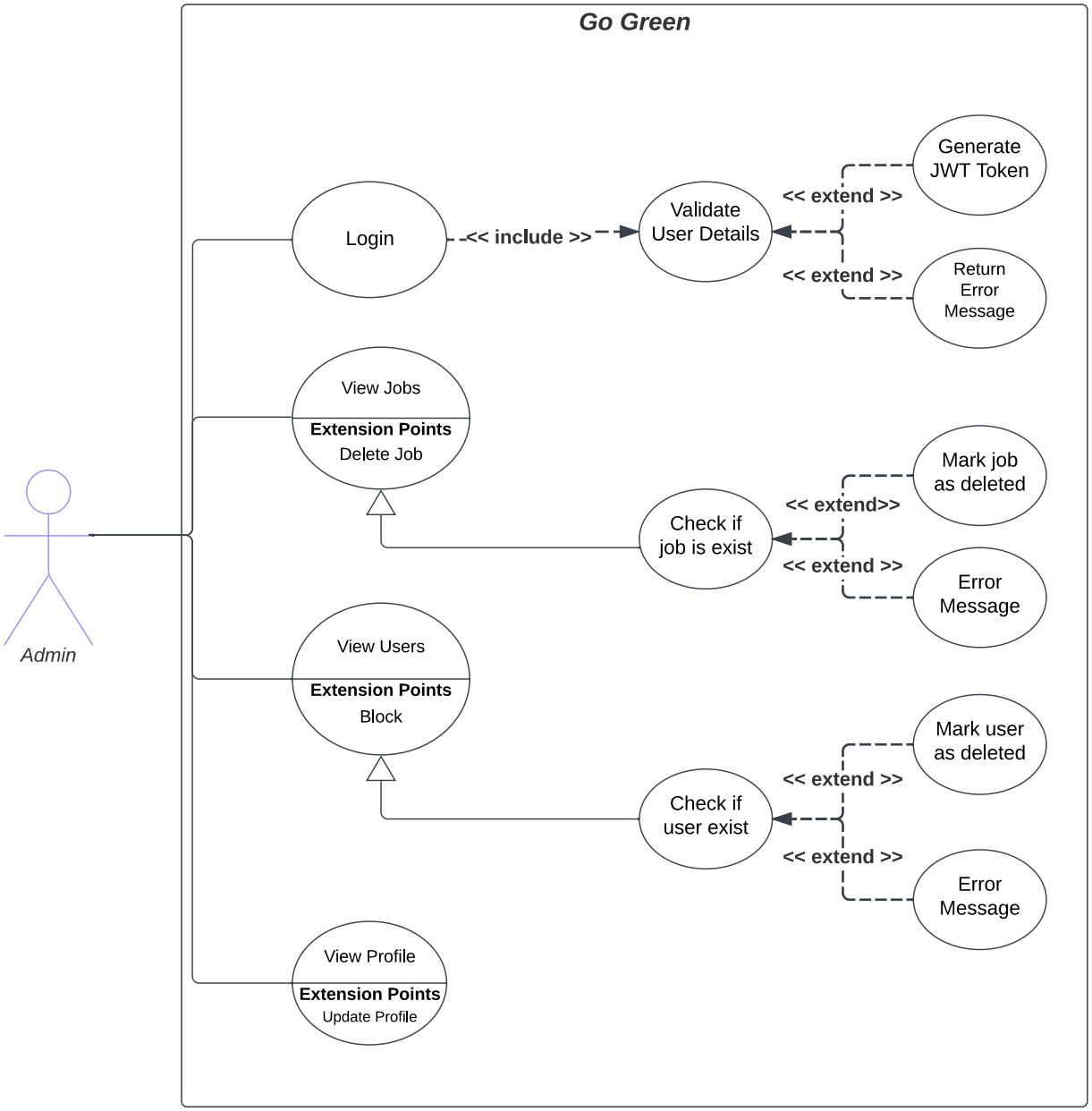
Client Use Case Diagram



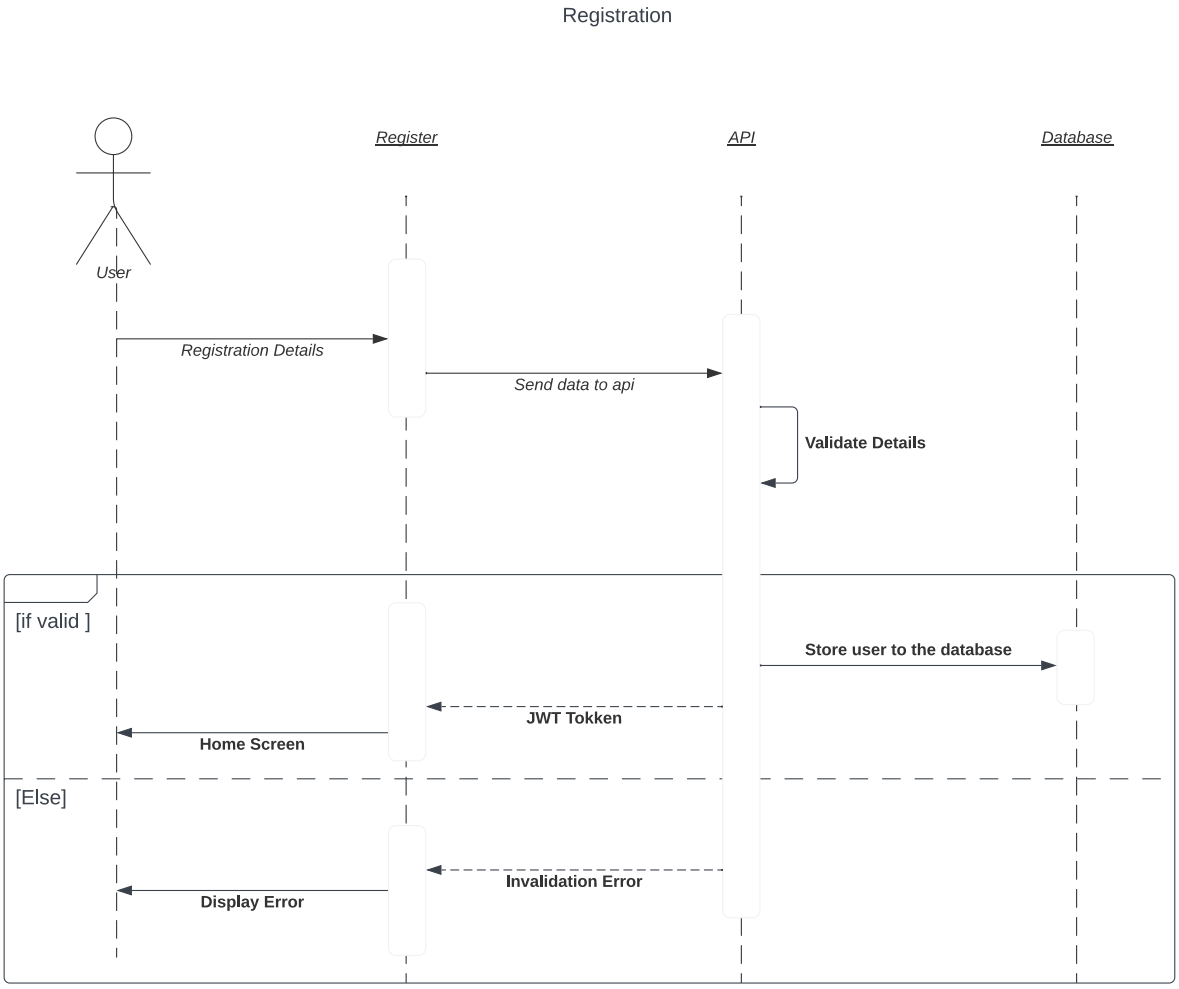
Talent Use Case Diagram



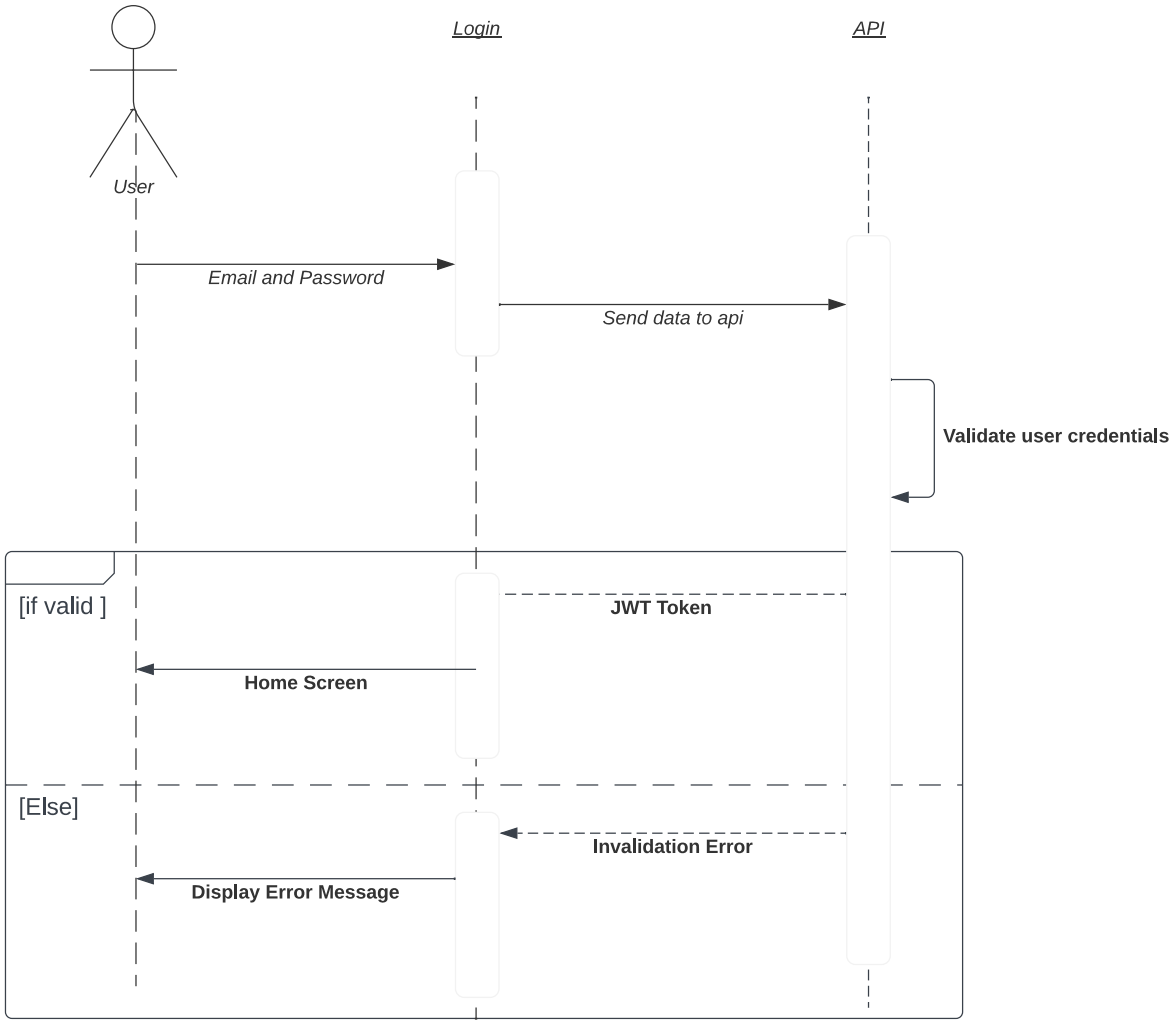
Admin Use Case Diagram



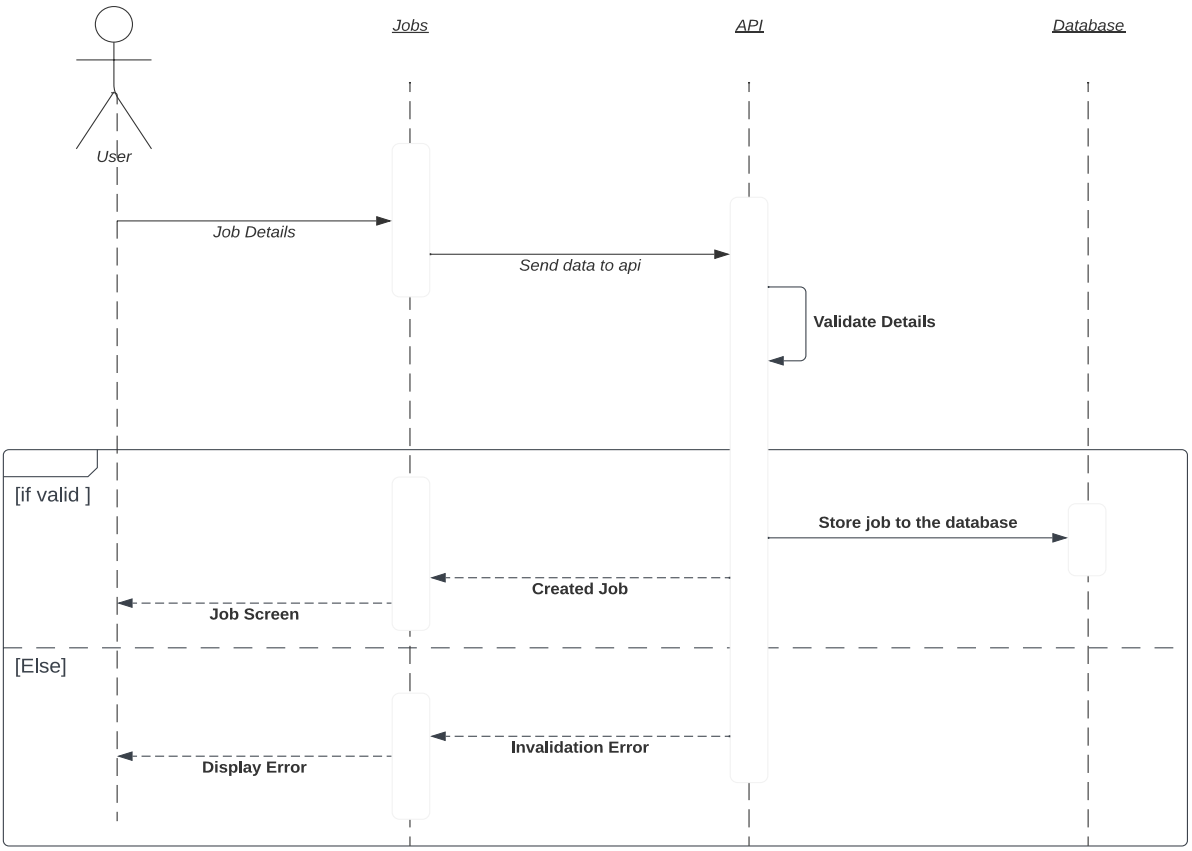
Sequence Diagram



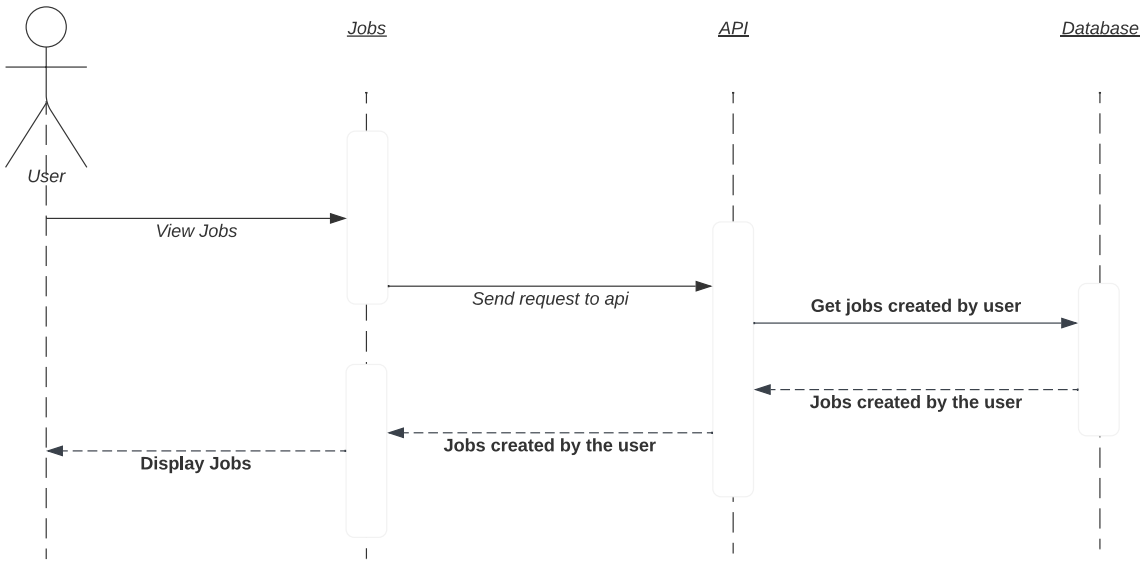
Login


Login
API
User
Email and Password
Send data to api
Validate user credentials
[if valid]
JWT Token
Home Screen
[Else]
Invalidation Error
Display Error Message

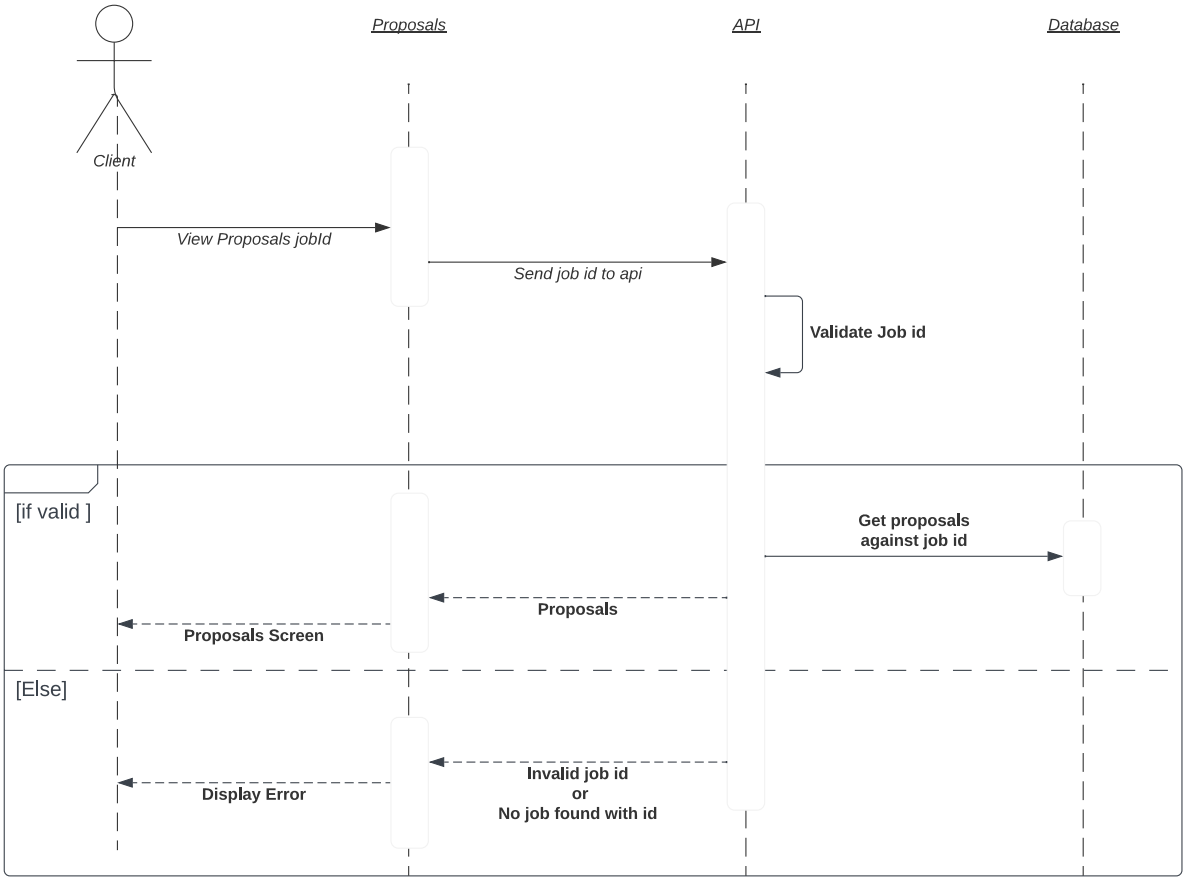
Create Job



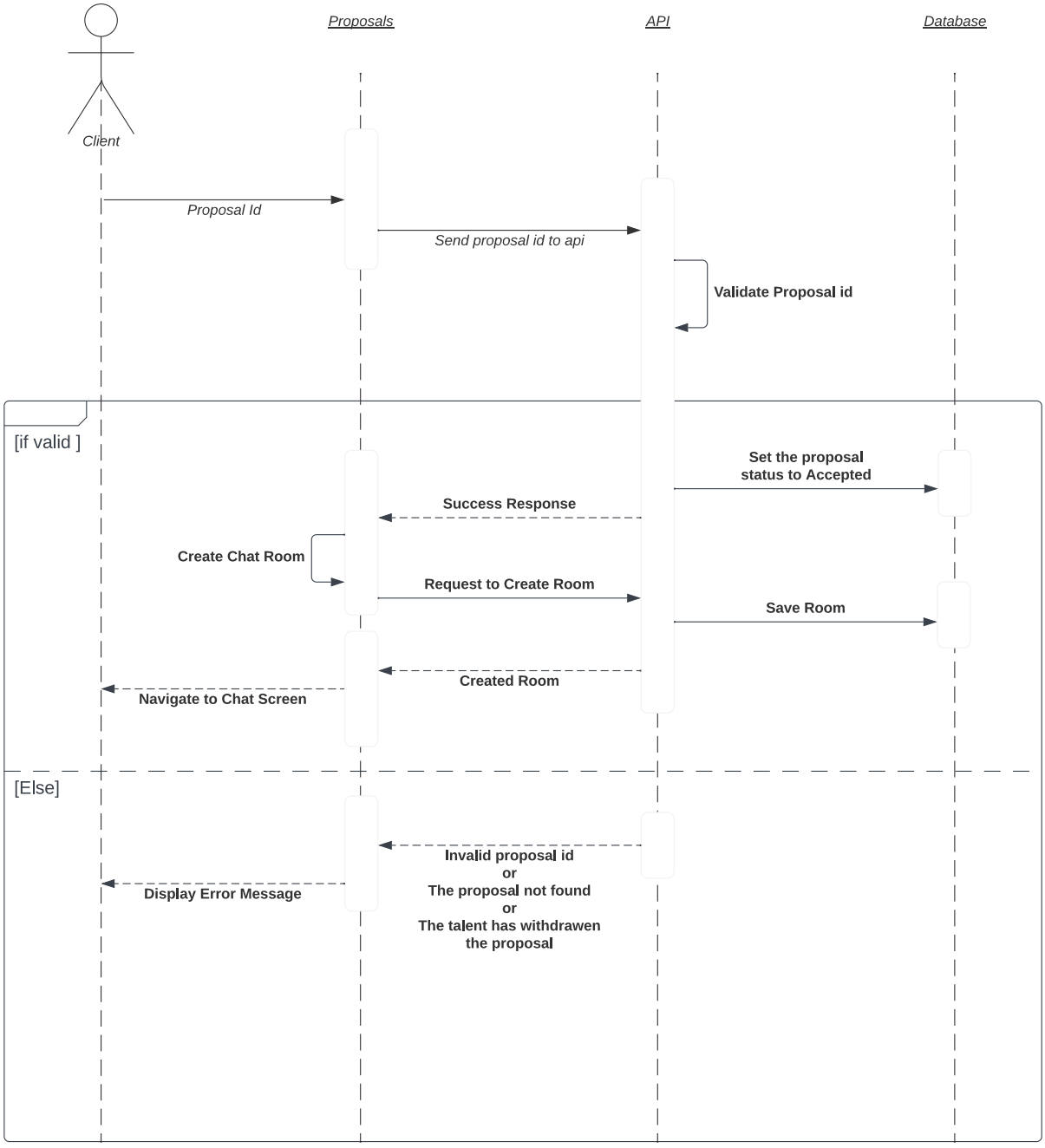
View Job



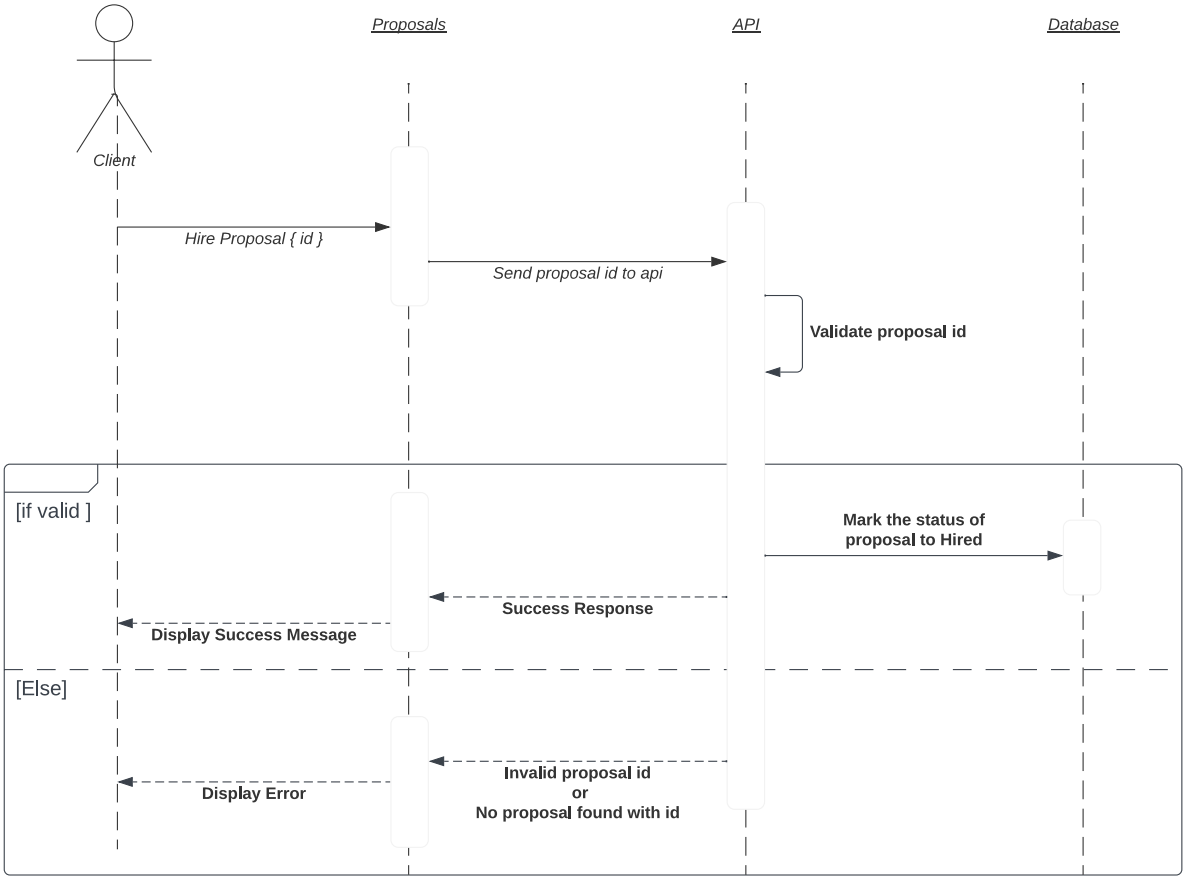
View Proposals



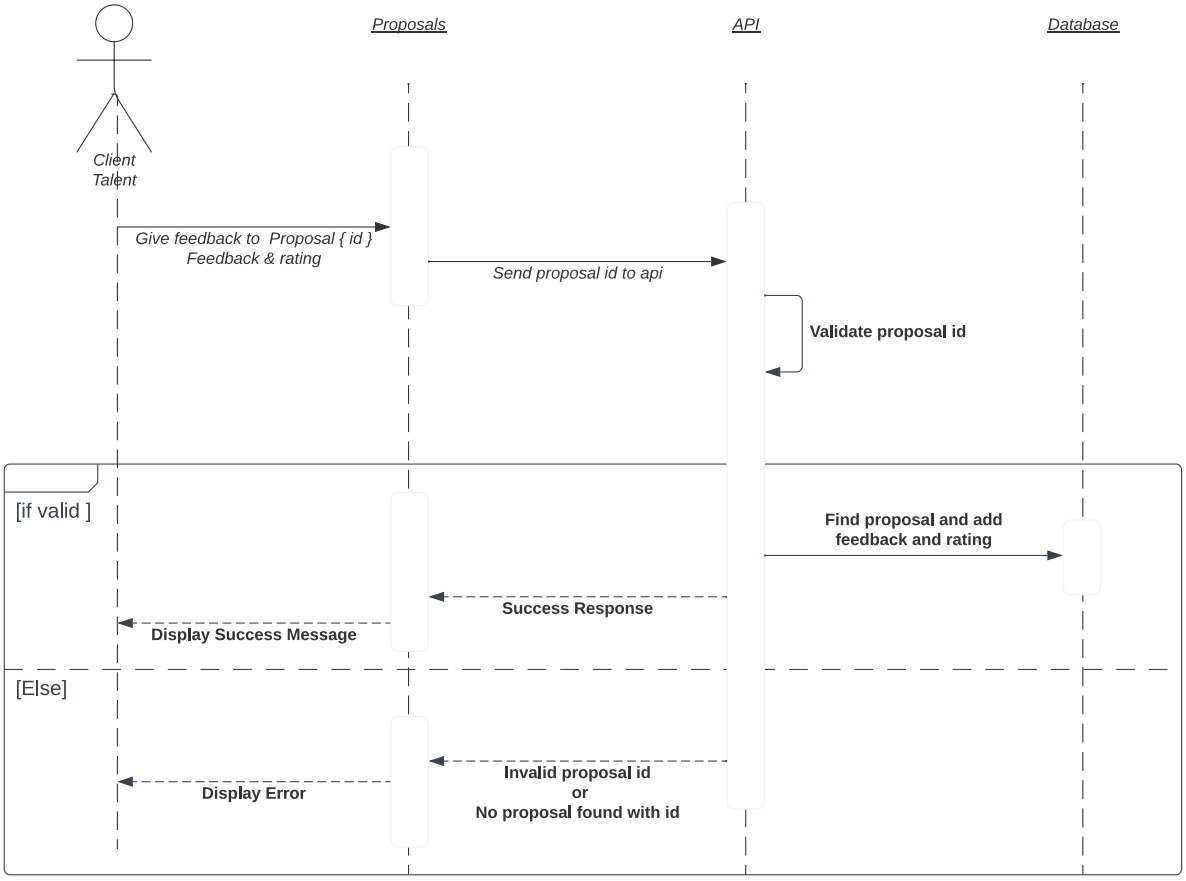
Interview Proposals



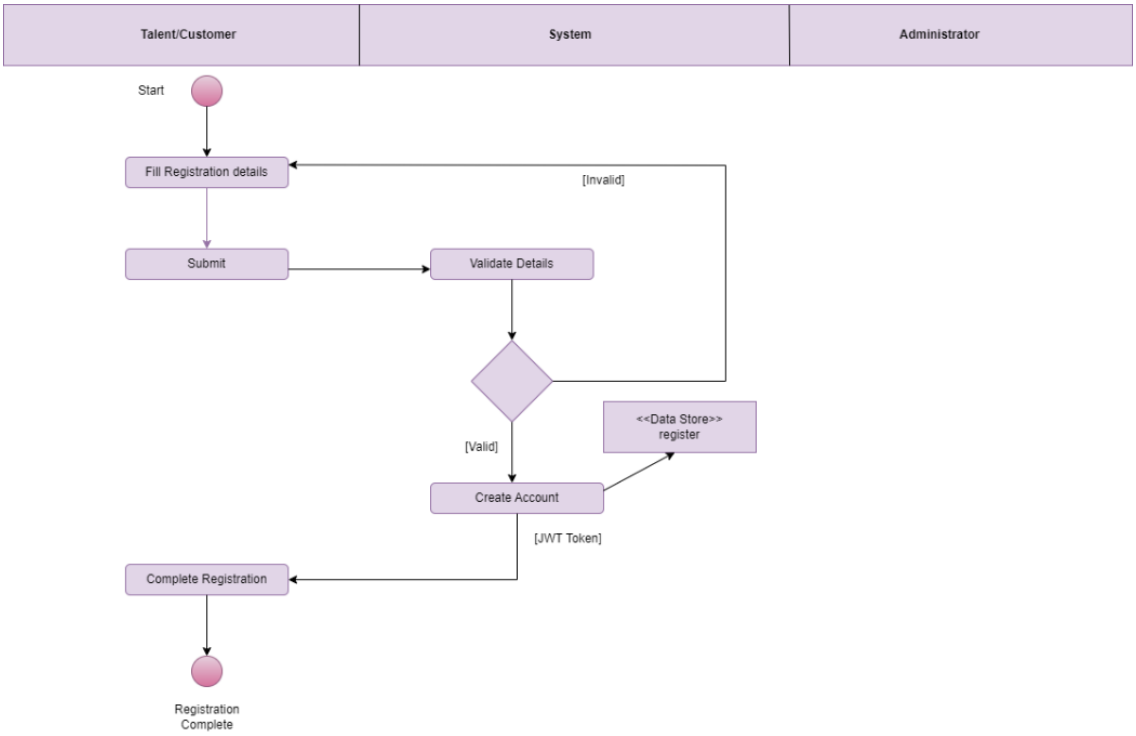
Hire Proposals



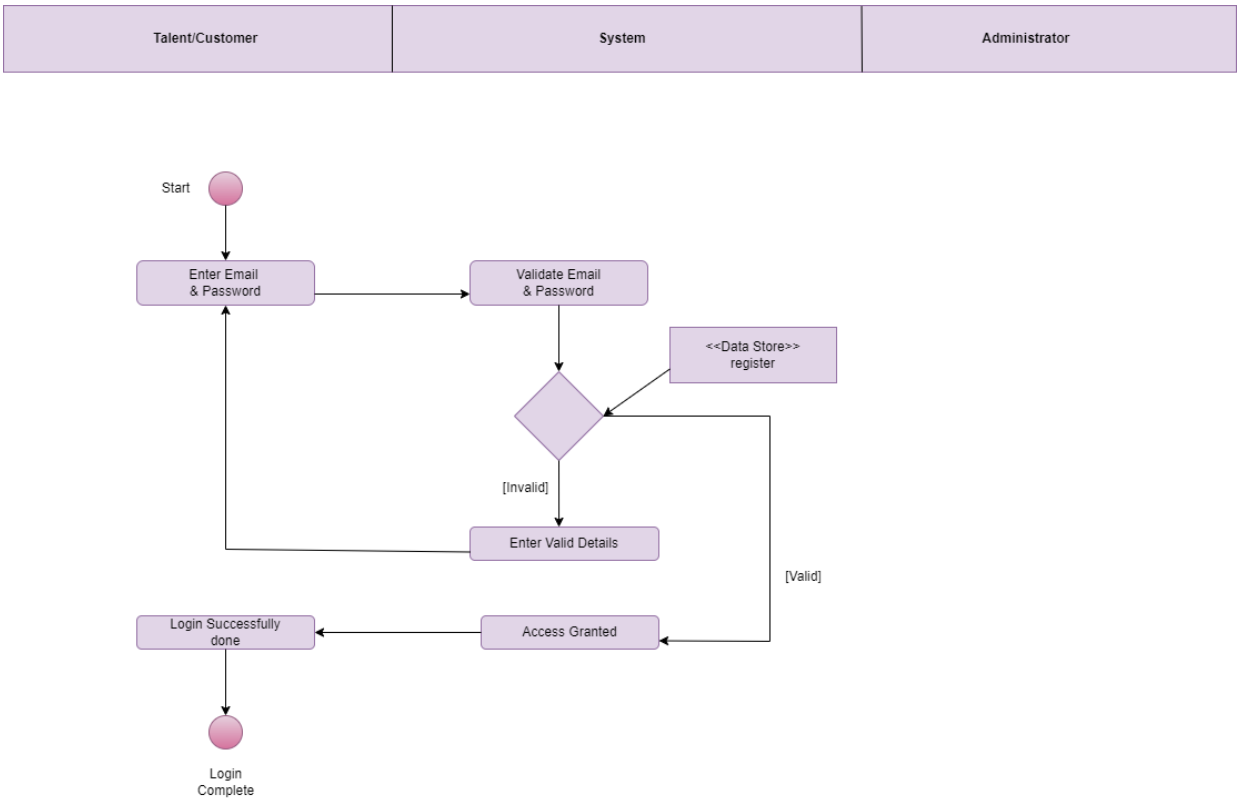
Give Feedback to
Proposal



Registration Activity Diagram



Login Activity Diagram



Chapter :6

Implementation and Testing

Development Methodology: The software development process for the Go Green app followed an Agile approach. Agile methodologies emphasize incremental and iterative development, ensuring continuous collaboration between developers, testers, and stakeholders. This iterative approach allowed for frequent feedback and allowed the team to adapt to changing requirements efficiently.

Version Control System: To manage source code changes and collaborate effectively, a Git version control system was utilized. Git enabled multiple developers to work on different features simultaneously while maintaining a history of code changes and facilitating easy code merging.

Integrated Development Environment (IDE): Android Studio was the primary IDE used for developing the Android app. It provided a feature-rich environment with tools for coding, debugging, and testing Android applications. For server-side development using Node.js, Visual Studio Code was employed, which offered a lightweight and extensible IDE for JavaScript-based projects.

Continuous Integration (CI): To ensure code quality and detect issues early, a CI/CD pipeline was set up using Jenkins. The pipeline automatically triggered builds, tests, and deployment whenever changes were pushed to the repository. This helped maintain code stability and identify issues promptly.

Testing Methodologies*:* The software development process incorporated various testing methodologies:

1. **Unit Testing:** JUnit was used for unit testing the Android app. It ensured that individual components and functions worked as expected in isolation.

2. **Integration Testing:** For testing interactions between different components, integration testing using Mockito was performed. This helped identify any issues arising from component interactions.

3. **End-to-End Testing:** Espresso was employed for end-to-end testing of the Android app, simulating user interactions and validating the app's behaviour across multiple screens.

4. **API Testing:** Postman was used for API testing, ensuring that the server-side APIs responded correctly to various requests and returned the expected data.

Controlled Libraries and Templates:

To streamline development, the app utilized various controlled libraries and templates. For the Android app, libraries such as Retrofit, GSon, and Room Database were used for networking, data parsing, and local data storage, respectively. For the server-side, Express.js middleware and templates were employed to handle routing and manage views.

Code Walkthroughs:

Regular code walkthroughs and peer reviews were conducted to ensure code quality and maintainability. Senior developers and team leads participated in these walkthroughs to identify and address any code-related issues.

Performance and Scalability:

The proposed software was thoroughly tested for performance and scalability. Load testing was performed to assess the app's performance under heavy user traffic, ensuring it could handle simultaneous job listings and bids without significant performance degradation.

Accuracy and Reliability:

Extensive testing was carried out to ensure the accuracy and reliability of the app's functionalities. The app's behavior was compared against the original specifications, and any deviations or inconsistencies were rectified during the development process.

User Acceptance Testing (UAT):

The app underwent UAT, where a select group of end-users tested the app in a real-world scenario. Feedback from UAT helped identify any usability issues or user experience improvements required.

Security:

The app's security measures were critically analyzed to ensure the protection of user data and transactions. Implementation of user authentication, encryption of sensitive data, and secure API communications were among the measures taken to enhance security.

Scalability Testing:

The app was tested for scalability by simulating a growing number of users and job listings. This ensured that the app could handle increased user traffic and data without compromising its performance.

Chapter:7

Results and Discussion

7.1 Application Output

Following are the implemented screenshots of our cross-platform application:

7.1.1 Welcome Screen

The figure shows that the user welcome screen of the system. It highlights app logo with tagline, and a meaningful graphical detail showcasing the purpose of the app.

Figure 7. 1: Welcome Screen

7.1.2 Sign-Up

The figure shows that the user sign-up page of the system. To register an account, the user willbe required to provide basic information such as their email address and password. This information is essential to create a unique account for the user and to keep their information secure.

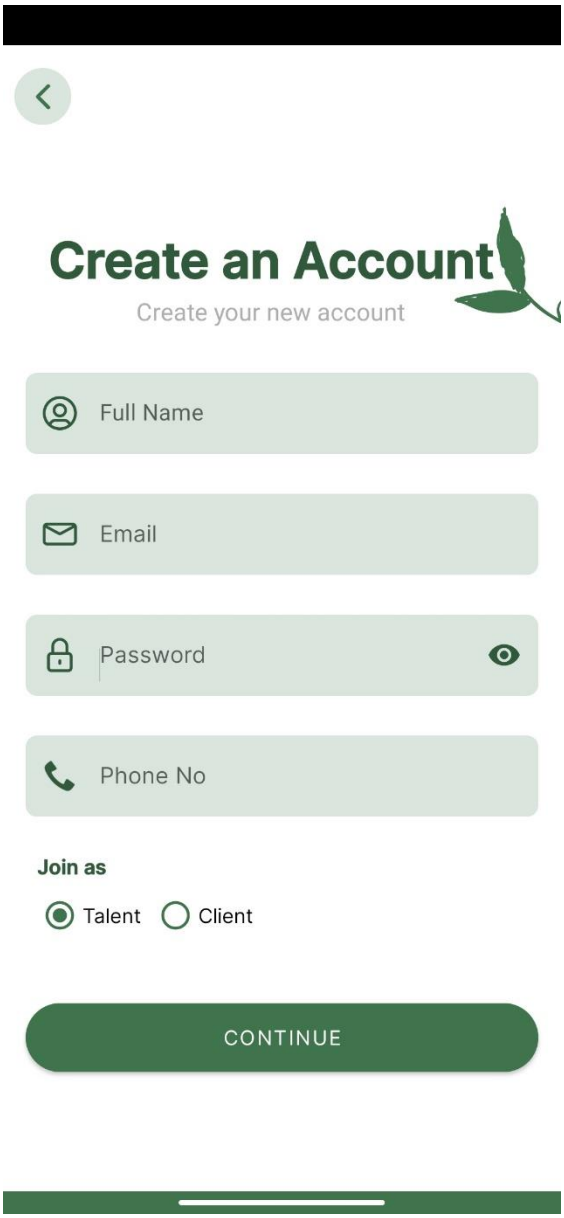


Figure 7. 2: Sign-Up Pages

7.1.3 Sign-In

The figure shows that the sign-in page of the application. The user is required to enter their email and password in the respective fields and click on the “Sign-in” button to access their account. The page also provides an option for the user to reset their password in case they haveforgotten it.

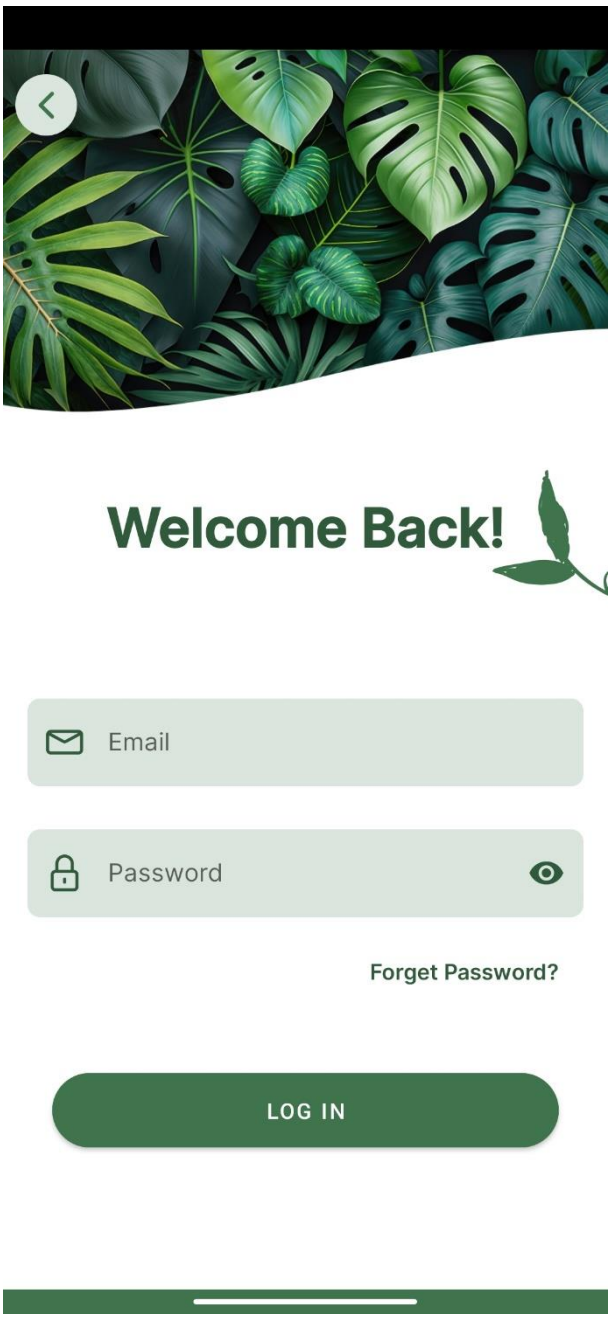


Figure 7. 3: Sign-In Page

7.1.4 Forgot Password

Figure shows the page for resetting the password in case a user forgets their password. To reset the password, the user needs to enter their registered the email addresses, and then further click the “Continue” button. An email will be sent to the user’s email address with instructionson how to reset their password.

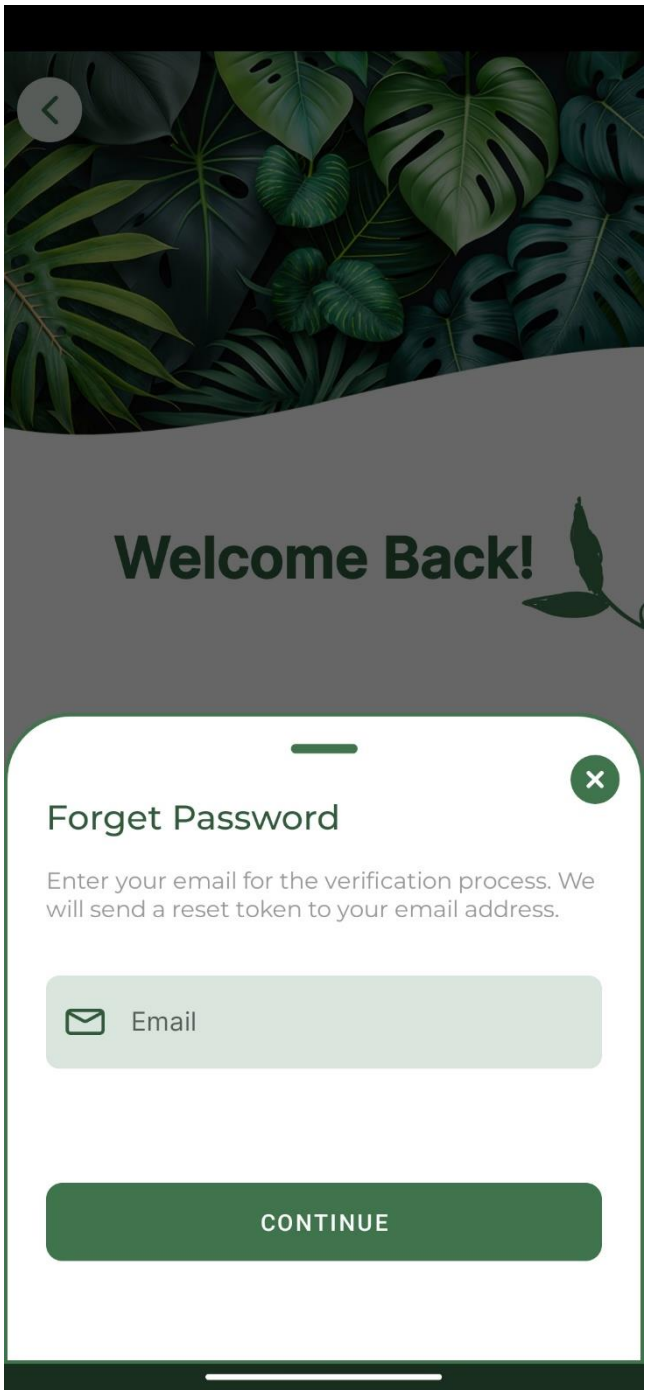


Figure 7. 4: Forgot Password Page

Chapter:8
Conclusion and Future Work

8.1 Future Work

While the current version of the "Go Green" app (v: 1.0) serves as a solid foundation for connecting users and talents in the gardening domain, there are several opportunities for future enhancements and expansions to further improve the app's capabilities and user engagement. Some potential areas for future work include:

1. **Web-Based Platform:** Consider developing a web-based platform in addition to the existing mobile app. This will expand the app's reach to users who prefer accessing services through a desktop or laptop.

2. **Integration with Third-Party Services**: Explore integration with third-party services for additional features, such as weather forecasts for gardening planning or an online payment gateway to provide users with more payment options.

3. **Advanced Gardening Tools:** Consider adding advanced gardening tools, such as plant identification and plant diseases detection through image recognition, personalized gardening recommendations, and watering reminders, to offer users a comprehensive gardening experience.

4. **Geographic Expansion:** Strategize for geographic expansion to cover a wider range of locations, attracting talents and users from various regions to create a more diverse and extensive gardening marketplace.

5. **Enhanced Security Measures:** Continuously improve the app's security measures to protect user data, transactions, and communications from potential threats or breaches.

6. **Social Media Integration:** Integrate social media platforms to allow users to share job listings, talents' profiles, and product purchases, increasing visibility and promoting the app's growth organically.

7. **Extended Marketplace:** Consider adding support for local industrialists related to gardening or agriculture such as tractor, thresher machine sellers or any other locals that want to connect to over platform to sell their products.

8. **Renting or selling of agricultural land:** Add functionality for renting or selling of agricultural land.

9. **Information Sharing:** Consider putting in an information sharing system on which the nursery owner, experience farmer or any expert can share information with the users about improving the production of their crop [Fasal] or nursery.

10. **Agricultural Automation System:** Try including an agricultural automation system in order to improve the productivity and quality of agro based products.

11. **Analytics and Reporting**: Implement analytical tools to gather insights into user behavior, job trends, marketplace performance and agricultural related data, allowing for data-driven decisions and targeted improvements.

8.2 Conclusion

The "Go Green" gardening marketplace mobile app (v: 1.0) has been successfully developed to address the needs of homeowners, small business owners, and talented professionals in the landscaping, gardening, and lawn care industry. The app provides a robust and user-friendly platform for users to post, bid on, and manage gardening-related jobs. It also features a marketplace for buying and selling gardening products and services. Throughout the development process, the Agile Software Development methodology was followed, enabling iterative progress and continuous feedback from stakeholders.

The "Go Green" app's key features, such as job listing and bidding, marketplace, user profiles and reviews, communications, and account management, have been implemented with careful consideration of user needs and expectations. The app offers an intuitive user interface, allowing users to seamlessly navigate through the various functionalities.

One of the app's strengths lies in the effective user profiling and review system, which fosters transparency and trust within the gardening community. Users can create detailed profiles, showcasing their skills, experience, and qualifications, while also sharing their feedback on

completed jobs. This system ensures that both users and talents can make informed decisions when bidding on jobs or hiring professionals, ultimately enhancing the overall user experience.

Furthermore, the app's integration with a secure payment gateway facilitates seamless and hassle-free transactions, ensuring that talents receive timely and fair compensation for their services. The communication tools within the app enable real-time interactions between users and talents, streamlining collaboration and ensuring that all parties stay informed throughout the job lifecycle. While version 1.0 of the app offers a strong foundation and meets the core requirements, there are several areas for future improvement and expansion. Considerations for future work include developing a web-based platform to reach a broader audience, integrating third-party services to offer additional features and options, and enhancing the app's security measures to safeguard user data and privacy.

References:

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

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

https://greengardens.pk/

https://www.upwork.com/

https://www.freelancer.com/

https://www.fiverr.com/

https://www.toptal.com/

https://www.guru.com/

https://www.interaction-design.org/literature/topics/ui-design

https://uxdesign.cc/10-tips-for-mobile-app-design-in-2022-23e70b57d927

https://www.mobileaction.co/blog/mobile-app-performance-optimization/

https://blog.testlodge.com/mobile-app-testing-best-practices/