Department of IT

University of Gujrat

STOPLIFY

Session: BSIT Spring 2020-2024

Project Advisor: Mr. Bilal Tariq

Submitted By

Hassam Sohail *20014156-040*

Ahmad Raza *20014156-028*

Rimsha Noor *20014156-052*

STATEMENT OF SUBMISSION

This is certify that **Hassam Sohail** Roll No. **20014156-040** and **Ahmad Raza** Roll No. **20014156-028** and **Rimsha Noor** Roll No. **20014156-052** has successfully completed the final year project named as **Stoplify** at the Department of Information Technology, University of Gujrat, to fulfill the requirement of the degree of __BS_____ **in Information Technology**.

_____ _____

Project Supervisor

Project Management Office,
Faculty of C& IT Evening-UOG

_____

Coordinator Evening/HOD

Acknowledgement

We truly acknowledge the cooperation and help make by _____, HOD/Coordinator, C&IT Evening Program, University of Gujrat. He has been a constant source of guidance throughout the course of this project. We would also like to thank Mr. _____ for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

1- Hassam Sohail

2- Ahmad Raza

3- Rimsha Noor

Date: July 23, 2024

Abstract

"Stoplify: **Empowering Education, Employment, and Skill Development through an Innovative Online Platform**"

Stoplify is an innovative online platform designed to revolutionize the way individuals engage with education, employment, and skill development. By offering a diverse range of courses in various subjects, Stoplify empowers users to expand their knowledge and expertise. Beyond education, our platform features a dynamic job marketplace, enabling seamless connections between skilled individuals and promising opportunities. Stoplify goes further by fostering real-time communication through chat modules, facilitating collaboration and networking. For companies, Stoplify simplifies the hiring process with the ability to create job listings and post projects effortlessly. On the flip side, users can showcase their skills by creating detailed profiles, complete with portfolios, expertise listings, and certification uploads. The platform's unique profile completion score encourages users to present a comprehensive picture of their capabilities. Stoplify stands out by providing a user-friendly interface where clients and freelancers can engage in meaningful interactions. With a commitment to transparency, users can access client reviews, user feedback, and ratings to make informed decisions. The platform's filtering options allow users to tailor their job searches based on hourly or fixed rates, ensuring a personalized and efficient experience. In essence, Stoplify is more than just a platform; it's a hub for learning, professional growth, and meaningful connections. Join Stoplify today to stop merely existing and start thriving in your career journey

TABLE OF CONTENTS

Chapter 1: Project Feasibility Report

1.1. Introduction

In the ever-evolving landscape of education, employment, and skill development, Stoplify emerges as a trailblazing online platform designed to revolutionize how individuals engage with these critical facts of personal and professional growth. This introduction serves as a gateway to understanding the core principles and functionalities that define Stoplify's innovative approach. By offering a diverse array of courses spanning various subjects, the platform empowers users to not only expand their knowledge but also to enhance their expertise. Beyond conventional educational offerings, Stoplify integrates a dynamic job marketplace, establishing seamless connections between skilled individuals and promising opportunities. The subsequent sections will delve into the platform's unique features, ranging from real-time communication through chat modules to simplified hiring processes for companies and the creation of comprehensive user profiles. Stoplify, with its commitment to transparency and user-friendly interface, stands out as more than just a platform; it stands as a transformative hub for learning, professional growth, and meaningful connections. Join Stoplify today to transition from a mere existence to thriving in your career journey.

1.2. Project/Product Feasibility Report

A Project/Product Feasibility Report assesses the viability and potential success of a project or product before its initiation. Based on the information provided about "Stoplify," we can discuss the various types of feasibility reports as follows:
There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

1.2.1. Technical Feasibility

Based on the tools and technologies used in our project, we can confidently state that "Stoplify" has strong technical feasibility. The tools are readily available, and our team possesses robust technical skills. We are highly motivated to complete this project successfully and efficiently. In the case of Stoplify, technical feasibility would involve assessing the platform's capability to offer a diverse range of courses, real-time communication features, and user-friendly interfaces.

1.2.2. Operational Feasibility

We are designing "Stoplify" to be user-friendly and fault-tolerant. Our goal is to create a product that users can operate seamlessly without encountering technical issues. We will provide initial guidelines to assist users in understanding the application's functionalities. Our entire team will be readily available to address any issues promptly.

1.2.3. Economic Feasibility

Evaluates the economic viability of the project, considering the costs and benefits.
Stoplify's economic feasibility would include analyzing the financial aspects of course offerings, job marketplace features, and the potential revenue streams.

Cost Estimate:

Total Project Cost: To be calculated.
Acquisition Cost: To be calculated.

Benefits:

Tangible Benefits: In the first year, "Stoplify" will offer cost-effective services and user-friendly features.
Intangible Benefits: Our focus is on providing excellent services and delivering accurate, helpful information to users.

1.2.4. Schedule Feasibility

Time management is crucial for meeting project requirements and to develop and launch the platform. We have allocated sufficient time to the project and divided tasks among team members. This ensures that "Stoplify" will be completed as per the planned schedule.

1.2.5. Specification Feasibility

Our project is built on well-defined requirements specifications, including features and functionalities, are achievable and align with the goals. For Stoplify, specification feasibility would involve assessing the practicality of features like chat modules, profile completion scores, and filtering options. These specifications are essential building

blocks that make “Stoplify” meaningful and aligned with user needs.

1.2.6. Information Feasibility

To ensure the reliability of “Stoplify”, we carefully identify and verify the sources of information required for the project. Our team is committed to ensuring that information is reliable, accurate, and relevant to the project's objectives.
Stoplify's information feasibility would involve ensuring that the platform has access to relevant course content, job listings, and user information.

1.2.7. Motivational Feasibility

Our motivation for “Stoplify” is to provide a platform for freelancers and clients to connect seamlessly. We aim to empower freelancers to showcase their skills globally while providing clients with easy access to a diverse pool of talent.

1.2.8. Legal & Ethical Feasibility

The tools used in this project are freely available, and we have the legal right to use them. We are committed to using these tools legally and only for the purposes outlined in our project.

1.3. Project/Product Scope

Stoplify aims to revolutionize the freelancing landscape by providing a comprehensive platform that bridges the gap between skilled freelancers and clients seeking professional services. The project's primary objectives encompass creating a global hub where freelancers can showcase their talents and clients can access a diverse array of services. With a commitment to user-friendliness, Stoplify will feature an intuitive interface, facilitating seamless interactions. The platform's efficiency will be evident in its swift and secure transaction. Prompt payment release, within 7 days of buyer request, further underscores the commitment to user satisfaction. Stoplify strives to be the go-to choice for those in search of cost-effective solutions and freelancers eager to expand their reach on a global scale. Key features include detailed user profiles, job portal, search filter, secure payment mechanisms. The scope explicitly excludes physical goods trading, event hosting, and the provision of legal or financial advice. Success hinges on factors like successful API integration, data security measures, and user adherence to platform guidelines. Periodic reviews will ensure the scope aligns with evolving project requirements.

1.4. Project/Product Costing

A metric is some measurement we can make of a product or process in the overall development process. Metrics are split into two broad categories:

- Knowledge oriented metrics: these are oriented to tracking the process to evaluate, predict or monitor some part of the process.
- Achievement oriented metrics: these are often oriented to measuring some product aspect, often related to some overall measure of quality of the product.

Most of the work in the cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulas linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by calibrating the model to your specific development environment, which basically involves adjusting the weightings of the metrics.

1.4.1. Project Cost Estimation By Function Point Analysis

P= UFP x CAF

Step1: Calculating UFP

Measurement Parameters	Counts	Weighting Factors			total
		Low	Medium	high	
External input (EI)	3	3*3	4	6	9
External out (EO)	2	4*2	5	7	8
External enquired (EQ)	2	3*2	4	6	6
Internal logical files (ILF)	1	7*1	10	15	7

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	3	◉	○	○
Number of User Outputs	2	◉	○	○
Number of User Inquiries	2	◉	○	○
Number of Files	1	◉	○	○
Number of External Interfaces	4	◉	○	○

Complexity Adjustment Table | FP Calculation

External Interface Files (EIF)	**4**	**5*4**	**7**	**10**	**20**

UFP = Total Count = 50

STEP 2: Calculating CAF (complexity adjust factor)

CAF = 0.65 + (0.01 x ∑Fi)

Where Fi = value adjusted factors based on response to the following 14 questions:

FP Calculation

NOTE: For any updates made on any of the entries, always click the 'Calculate Function Points' button to recalculate function points value.

Reset / Clear all form entries Calculate Function Points

RESULT	
PROJECT FUNCTION POINTS	51

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

So,

$\sum F_i = 0+5+2+2+2+0+5+5+2+2+5+2+0+5$

CAF = 0.65 + (0.01 x ∑ Fi)CAF = 0.65 + (0.01 * 37)

CAF = 1.02

Now,

F.P = UFP * CAF

= 50 * 1.02

F.P = 51

1.4.2. Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)

The Constructive Cost Model (COCOMO) is a procedural software cost estimation model developed by Barry W. Boehm. The model parameters are derived from fitting a regression formula using data from historical projects.
COCOMO 81 model estimate for the project cost by nominal effort involving a set of multipliers and development time for that project by selecting its mode.
For project mode selection, we have to look for tables that contain certain features, and all three modes are organic, semi-detached, and embedded. The mode will be selected according to the scaling done for these features. This estimation for mode selection is given below:

Optimistic LOC = 4400
Most Likely LOC= 6900

Pessimistic LOC= 10000
Estimated LOC = (Opt. + 4 Most Likely + Pessi.) / 6
= (4400 + 4 (6900) + 10000) / 6 = 7000 LOC
Typically our project size range in b/w **2 - 50 KLOC** and the mode is **Organic** so basic COCOMO is suitable for this range.

The Basic COCOMO equations take the form:

$E = a_b (KLOC)^{bb}$
$D = c_b (E)^{db}$
$SS = E/D$ persons
$P = KLOC/E$

E = effort
D = Deployment time
SS = staff size
P = productivity
a_b, b_b, c_b, d_b = Coefficients

Basic COCOMO Co- efficients

Project	a_b	b_b	c_b	d_b
Organic mode	2.4	1.05	2.5	0.38
Semidetached mode	3.0	1.12	2.5	0.35
Embedded mode	3.6	1.20	2.5	0.32

E = a (KLOC)^b
E = $2.4(7)^{1.05}$ = 18.52 PM
D = c(E)^d
D = $2.5(18.52)^{0.38}$ = 7.58 M
SS = E/D persons
SS = 18.52/7.58 = 2.44 persons
P = KLOC/E
P = 7/18.52 = 0.37

1.4.3. Activity Based Costing

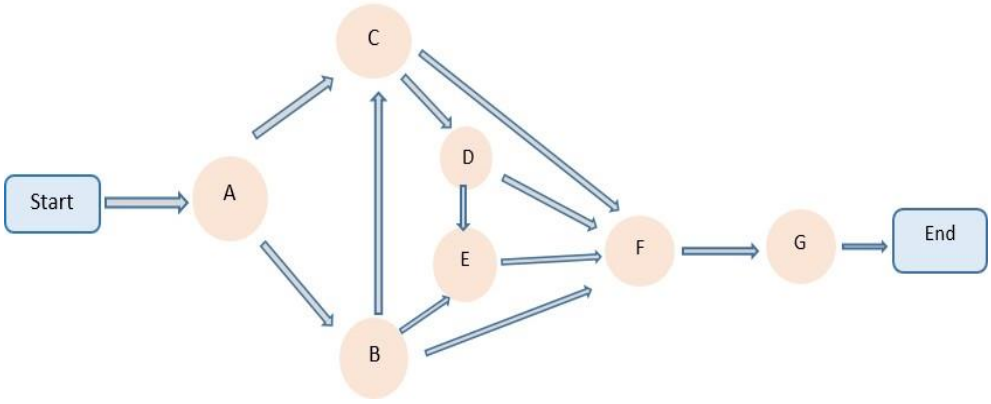
Sr.	Activities	Resources	Cost Rate	Duration
1	App Design, Layout, Structure	Figma	Free	2 weeks
2	Front End	React Native	Free	8 weeks

3	Development of App components	Node.js	Free	8 weeks
4	Back End	Node.js	Free	7 weeks
6	Integration	React Native	Free	2 weeks
7	Testing	Visual Studio Code	Free	2 week
8	Documentation	No Resources	Free	1 week

1.5. Task Dependency Table

Task	Task Name	Dependency	Duration (Weeks)
A	Feasibility & Requirement Gathering	none	3 Weeks
B	SRS & App Design, Layout, Structure	A	4 weeks
C	Development of Modules	A B	6 weeks
D	Implementation	C	5 Weeks
E	Backend	B, D	5 Weeks
F	Integration	B, C, D, E	6 Weeks
G	Testing	F	3 Weeks

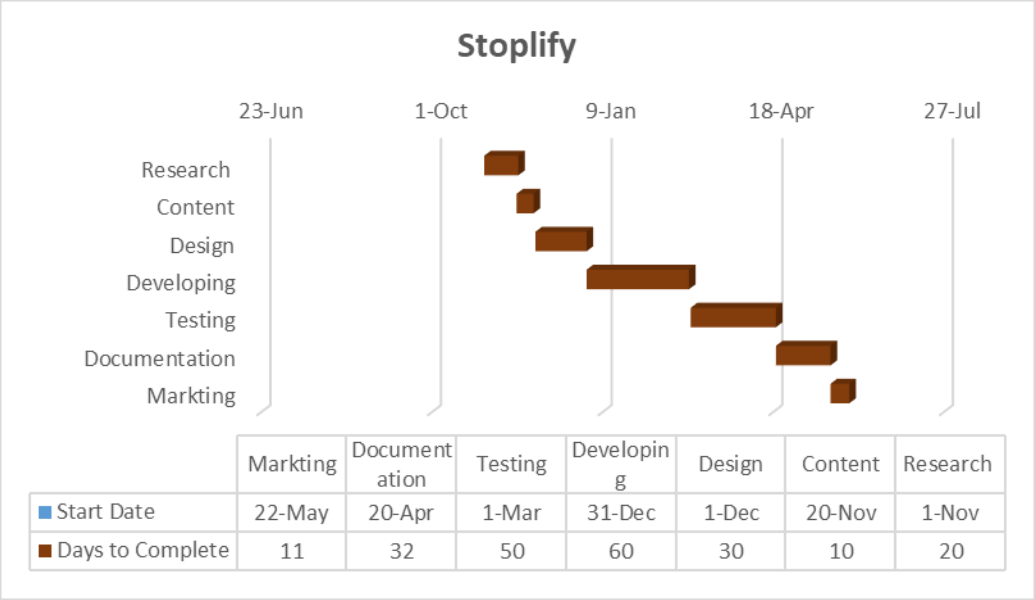
1.6. CPM - Critical Path Method



Network Diagram for the above-mentioned activities

Activity	Duration	ES	EF	LS	LF	Slack Time (ES-LS)	CriticalPath
A	3	0	3	0	3	0	Yes
B	4	3	7	3	7	0	Yes
C	8	7	15	7	15	0	Yes
D	6	15	21	15	21	0	Yes
E	5	21	26	21	26	0	Yes
F	6	26	32	26	32	0	Yes
G	3	32	35	32	35	0	Yes

1.7. Gantt chart



1.8. Introduction to Team member and their skill set

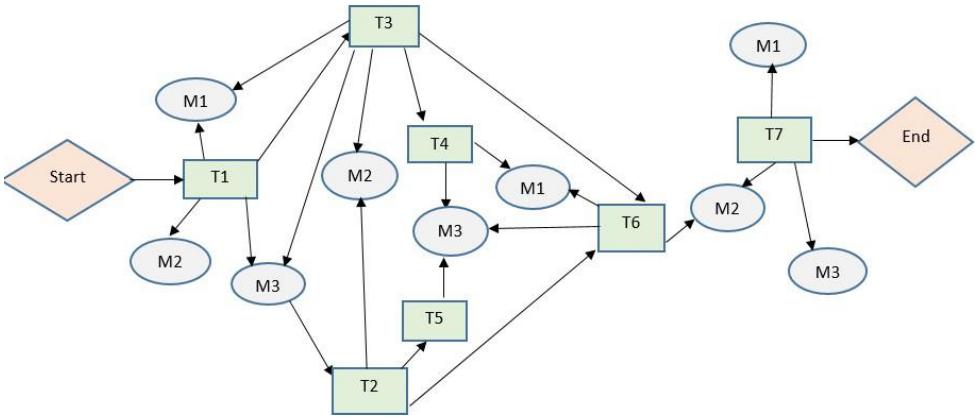
ID	Name	Roll#	Skill Set
M1	Hassam Sohail	20014156-040	UI/UX Designing, Frontend & Backend development
M2	Ahmad Raza	20014156-028	UI/UX Designing, Frontend development,
M3	Rimsha Noor	20014156-052	Documentation

1.9. Task and Member Assignment Table

Task	Duration (days)	Dependencies	Memebers
T1	4	N/A	M1, M2
T2	3	T1	M1, M2
T3	2	T2	M1, M3
T4	2	T3	M2
T5	1	T4	M3
T6	6	T5	M1, M3
T7	3	T6	M2, M1
T8	3	T7	M3
T9	5	T6	M2, M3
T10	5	T9	M3, M2

T11	5	T8,T10	M1, M2, M3
T12	2	T11	M1

Task durations and dependencies



1.10. Tools and Technology with reasoning

Tools and Techniques are as follows:

LANGUAGES:

React-Native:

- React Native is a popular open-source framework for building mobile applications using JavaScript and React. It allows developers to create cross-platform apps with a single codebase that can run on both iOS and Android devices.

Firebase:

- Firebase is a comprehensive mobile and web application development platform by Google. It provides a variety of services including real-time database, authentication, cloud functions, hosting, and more. Firebase simplifies backend development, enabling developers to focus on building the frontend and user experience.

MongoDB:

- MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It is designed to scale horizontally, providing high performance and flexibility. MongoDB is commonly used in modern web applications to handle large volumes of data with varying structures.

NoSQL:

- NoSQL, or "Not Only SQL," is a category of database systems that do not rely on the traditional relational database management system (RDBMS) structure. NoSQL databases are designed to handle unstructured or semi-structured data and can scale horizontally to accommodate large amounts of data.

Node.js:

- Node.js is a runtime environment that allows the execution of JavaScript code on the server side. It is built on the V8 JavaScript runtime and is known for its event-driven, non-blocking I/O model. Node.js is commonly used for building scalable and high-performance server-side applications.

Python (used for AI models and algorithms):

- Python is a versatile and high-level programming language widely used in various domains, including artificial intelligence and machine learning. Its simplicity and readability make it a popular choice for developing algorithms, machine learning models, and data analysis applications. Numerous libraries and frameworks, such as TensorFlow and PyTorch, support AI development in Python.

TOOLS:

VS Code:
A versatile and lightweight source code editor with robust features and extensive language support.

MS Word:
A widely-used word processing application for creating, editing, and formatting documents.

Lucidchart:
An online diagramming tool for creating flowcharts, diagrams, and visual representations.

MS Project:
Microsoft's project management software for planning, tracking, and managing projects.

MongoDB Compass:
A graphical user interface (GUI) for MongoDB, facilitating database visualization and manipulation.

1.11. Vision Document

To create an innovative online marketplace that empowers freelancers to showcase their skills and connects them with clients seeking high-quality services, fostering a global

community of collaboration and creativity.

Stakeholders:

1. **Freelancers:** Independent professionals offering a diverse range of skills and services.
2. **Clients:** Individuals or businesses seeking freelance services for their projects.
3. **Development Team:** The team responsible for creating and maintaining the platform.

Problem Statement:

The current absence of a comprehensive online platform hinders the efficient and transparent collaboration between freelancers and clients, leading to a fragmented freelance mobile app.

Solution:

The proposed solution is a dynamic and user-friendly online platform that facilitates seamless collaboration between freelancers and clients, allowing them to connect, collaborate, and achieve their project goals efficiently.

Benefits:

- **Empowering Freelancers**: Provides a platform for freelancers to showcase their skills and reach a global audience.
- **Efficient Collaboration**: Streamlines the process of finding, hiring, and working with freelancers.
- **Quality Assurance**: Enables clients to make informed decisions based on reviews and ratings.
- **Global Community**: Fosters a collaborative environment, connecting talents from around the world.

Future Enhancements:

- Skill Matching Algorithm: Implementing an algorithm to suggest freelancers based on client requirements.
- Integration of Virtual Collaboration Tools: Enhancing the platform with virtual collaboration tools for real-time collaboration.

1.12. Risk List

Following are the risks found while developing this system:

- Intense competition from existing freelance platforms.
- Time limit
- Failure to comply with local and international regulations.
- Lack of resources
- Disputes between freelancers and clients.
- DDos attacks

- Technical issues leading to platform downtime.

1.13. Product Features/ Product Decomposition

1. User Registration and Authentication.
2. User Profiles Creation.
3. Job Portal.
4. Proposals.
5. Search and Filter.
6. Skill Assessment.
7. Communication Tools.
8. Order Management.
9. Payment System.
10. Review and Rating System.
11. Mobile App.
12. Legal and Compliance

Chapter 2: Software Requirement Specification (For Object Oriented Approach)

2.1 Introduction:

Stoplify is a comprehensive online platform designed to revolutionize the freelance marketplace, seamlessly connecting clients with talented freelancers in a user-friendly environment. The platform facilitates a dynamic ecosystem encompassing proposal submission, job posting, and effective communication channels, ensuring a streamlined collaboration process. With a focus on enhancing the freelancing experience, Stoplify incorporates a sophisticated portfolio feature for freelancers to showcase their skills and achievements. Moreover, the platform integrates a robust learning module, empowering freelancers to continually develop their expertise. Embracing an Object-Oriented Approach, the software will be engineered to deliver scalability, maintainability, and flexibility, meeting the diverse needs of both clients and freelancers on the Stoplify platform

Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements
- Prioritize requirements
- Development of requirements traceability matrix

2.1.1 Systems Specifications

The following are the clauses that must be included while describing the system specifications.

Introduction

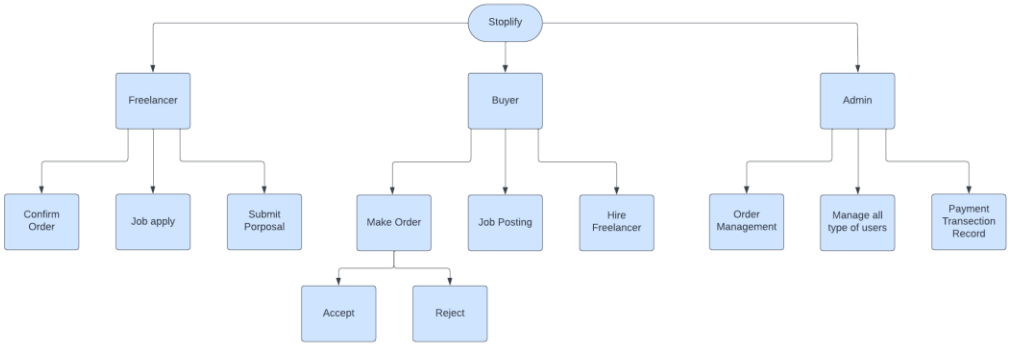
Stoplify is a cutting-edge online platform aimed at transforming the freelancing landscape, providing a robust and user-friendly ecosystem for clients and freelancers to collaborate effectively. This document outlines the detailed system specifications for the development of Stoplify, a platform inspired by the functionalities of Upwork but designed with additional features and enhancements for an improved user experience.

Existing System

Currently, Upwork serves as a prominent online freelance marketplace, connecting

clients with freelancers across various domains. Upwork allows users to post job opportunities, submit proposals, and engage in communication to facilitate successful collaborations. However, Stoplify aims to build upon this foundation, introducing novel features such as an advanced learning module, comprehensive portfolio management, and an optimized user interface for a more intuitive experience.

Organizational Chart



Scope of the System

The scope of Stoplify encompasses a wide range of functionalities, including but not limited to proposal submission, job posting, communication tools, freelancer portfolio management, and a dedicated learning module. The platform will be designed to support scalability, ensuring that it can accommodate a growing user base and evolving technological requirements. The system will prioritize security and user privacy, adhering to industry standards and regulations.

Summary of Requirements: (Initial Requirements)

User Authentication:

Implement secure user authentication mechanisms for both clients and freelancers.

Proposal Submission:

Allow freelancers to submit detailed project proposals, including relevant skills, experience, and project-specific details.

Job Posting:

Enable clients to post job opportunities with comprehensive descriptions, requirements, and budget details.

Communication Tools:

Provide messaging and collaboration tools to facilitate effective communication between clients and freelancers.

Portfolio Management: Develop a feature-rich portfolio system for freelancers to showcase their work, skills, and achievements.

Learning Module: Integrate a learning module to enable freelancers to enhance their skills through courses and resources.

2.1.2. Identifying External Entities

The identification of the external entities will be based on the information contained in your Abstract. This identification is done after two phases. We will map the "Green wood" case study to make things more comprehensible.
The Identification of External Entities is done in two phases.

a. Over Specify Entities from Abstract:

On the basis of the Abstract, one might identify the following entities from the Stoplify case study.

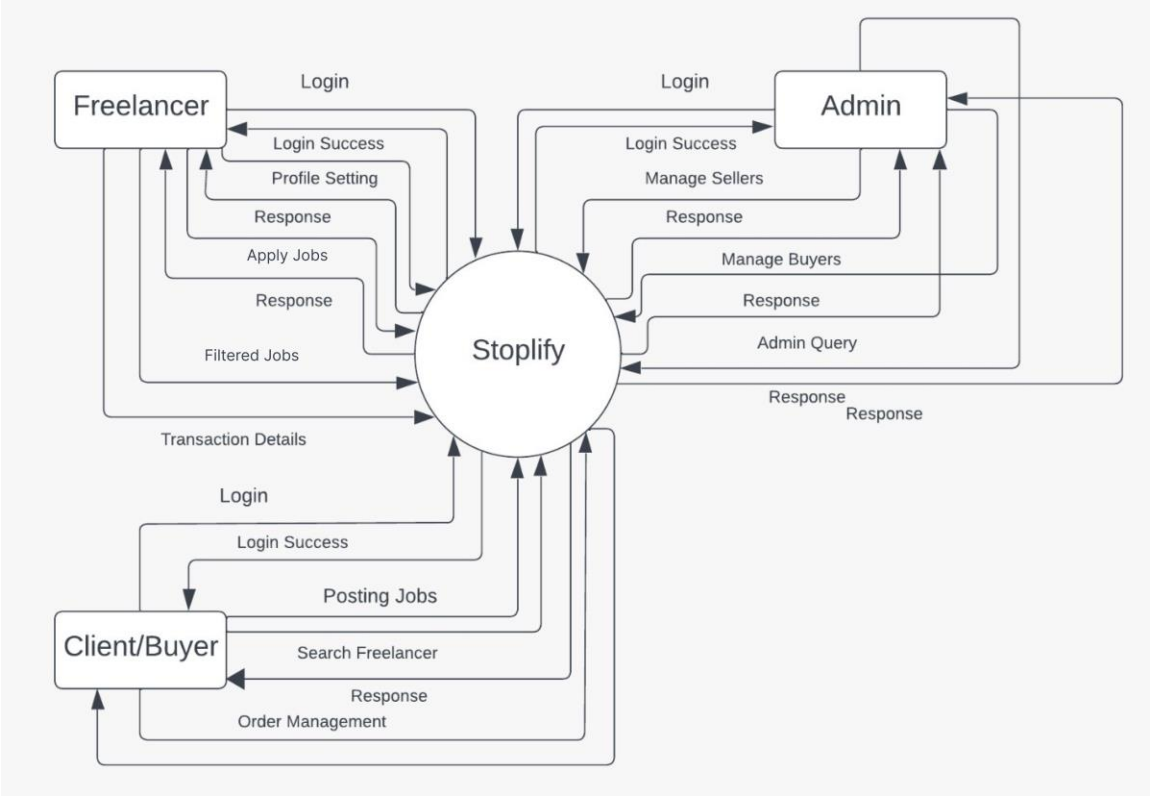
- Job Market Place
- Order
- Freelancer
- Job Filter
- Communication
- Learning
- Expertise Listing
- Profiles
- Client

b. Perform Refinement:

After over specifying the entities, you have to refine them on the basis of your business logic. The entities that are more related to our project are:

- Job Market Place
- Projects
- Freelancer
- Client

2.1.3. Context Level Data Flow Diagram:



2.1.4. Capture "shall" Statements:

Identify “shall” statements, as they would be all functional requirements.

Para #	Initial Requirements
1.0	Users “shall” visit the Stoplify platform.
1.0	Users “shall” register within the system.
1.0	The system “shall” allow users to register and verify through email.
1.0	The system “shall” provide sign-in for users.
1.0	Users “shall” log in to the system and “shall” be able to Forget their password.

1.0	User “shall” add the job post Title.
1.0	User “shall” add the main skills required for the work.
1.0	User "shall" select the estimated scope of their own work (Larger, Medium, Small) and also "shall" indicate the time required.
1.0	Users “shall” select the hourly rate or fixed price.
1.0	Users “shall” add the complete description of the project and review his uploaded details of project.
2.0	User “shall” view his Dashboard and different proposal.
2.0	User “shall” make the order to freelancer.
2.0	User “shall” give feedback to the freelancer on his job completion.
2.0	The system “shall” provide sign-in for freelancers.
2.0	The system “shall” provide sign-out for freelancers.
2.0	Freelancers “shall” log in to the system and “shall” be able to change their password.
2.0	Freelancer “shall” view the job page and view different jobs.
2.0	Freelancers “shall” submit the proposal according to his relevant job.
2.0	Freelancers “shall” able to manage their orders and communicate with client.
2.0	Freelancers “shall” be able to see the client-provided review.
3.0	The admin “shall” be able to log in to the Stoplify platform.
3.0	The admin “shall” be able to view all jobs.
3.0	The admin “shall” be able to view proposal.

3.0	The admin “shall” be able to view records of all orders.
3.0	The admin “shall” be able to view all users and freelancers.
3.0	The admin “shall” be able to create, delete, update the user and jobs.
3.0	The admin shall be able to take appropriate actions in case of policy violations.
3.0	The admin shall be able to send notifications to users and freelancers.

2.1.5. Allocate Requirements:

Allocate the requirements in the use cases.

Para #	Initial Requirements	UseCase Name
1.0	Users shall register within the system. The system shall allow users to register and verify through email.	UC_User_Registration_and_Verification
1.0	Users shall log in to the system and shall be able to forget their password.	UC_User_Log_in_and_Password_Recovery
1.0	User shall add the job post title, main skills required, select the estimated scope (Large, Medium, Small), and indicate the time required. Users shall select the hourly rate or fixed price.	UC_Job_Posting
1.0	Users shall add a complete description of the project and review their uploaded details.	UC_Project_Description
1.0	User shall view their dashboard and different proposals.	UC_User_Dashboard_and_Proposal_Viewing
1.0	User shall make an order to a freelancer.	UC_Freelancer_Order_Placement
2.0	User shall give feedback to the freelancer on job completion.	UC_Feedback_to_Freelancer

2.0	The system shall provide sign-in for freelancers. Freelancers shall log in and be able to change their password.	UC_Freelancer_Sign_in_and_Password_Change
2.0	Freelancers shall view the job page, view different jobs, and submit proposals according to their relevant job.	UC_Job_Viewing_and_Proposal_Submission
2.0	Freelancers shall be able to manage their orders and communicate with clients.	UC_Order_Management_and_Communication
2.0	Freelancers shall be able to see client-provided reviews.	UC_View_Client_Provided_Review
3.0	The admin shall be able to log in to the Stoplify platform.	UC_Admin_Login
3.0	The admin shall be able to view all jobs.	UC_Admin_View_Jobs
3.0	The admin shall be able to view proposals.	UC_Admin_View_Proposals
3.0	The admin shall be able to view records of all orders.	UC_Admin_View_Orders
3.0	The admin shall be able to view all users and freelancers.	UC_Admin_View_Users_and_Freelancers
3.0	The admin shall be able to create, delete, update users and jobs.	UC_Admin_User_and_Job_Management
3.0	The admin shall be able to take appropriate actions in case of policy violations.	UC_Admin_Policy_Enforcement
3.0	The admin shall be able to send notifications to users and freelancers.	UC_Admin_Notifications

2.1.6. Prioritize Requirements:

Requirements must be prioritized as this will help achieve tasks easily. Rank them as "highest, medium, and lowest".

Para #	Rank	Initial Requirements	Usecase ID	Usecase Name
1.0	Highest	Users shall register within the system. The system shall allow users to register and verify through email.	UC_1	UC_User_Registration_and_Verification
1.0	Highest	Users shall log in to the system and shall be able to forget their password.	UC_3	UC_User_Log_in_and_Password_Recovery
1.0	Highest	The system shall provide sign-in for freelancers. Freelancers shall log in and be able to change their password.	UC_4	UC_Freelancer_Sign_in_and_Password_Change

1.0	Highest	The admin shall be able to log in to the Stoplify platform.	UC_5	UC_Admin_Login
1.0	Highest	User shall add the job post title, main skills required, select the estimated scope (Large, Medium, Small), and indicate the time required. Users shall select the hourly rate or fixed price.	UC_6	UC_Job_Posting
2.0	Medium	User shall make an order to a freelancer.	UC_7	UC_Freelancer_Order_Placement
2.0	Medium	User shall give feedback to the freelancer on job completion.	UC_8	UC_Feedback_to_Freelancer
2.0	Medium	Freelancers shall view the job page, view different jobs, and submit proposals according to their relevant job.	UC_9	UC_Job_Viewing_and_Proposal_Submission
2.0	Medium	Freelancers shall be able to manage their orders and communicate with clients.	UC_10	UC_Order_Management_and_Communication
3.0	Medium	The admin shall be able to view all jobs.	UC_11	UC_Admin_View_Jobs
3.0	Medium	The admin shall be able to view proposals.	UC_12	UC_Admin_View_Proposals
3.0	Medium	The admin shall be able to view records of all orders.	UC_13	UC_Admin_View_Orders
3.0	Medium	The admin shall be able to view all users and freelancers.	UC_14	UC_Admin_View_Users_and_Freelancers
3.0	Medium	The admin shall be able to create, delete, update users and jobs.	UC_15	UC_Admin_User_and_Job_Management
3.0	Medium	The admin shall be able to take appropriate actions in case of policy violations.	UC_16	UC_Admin_Policy_Enforcement
3.0	Lowest	The admin shall be able to send notifications to users and freelancers.	UC_17	UC_Admin_Notifications
3.0	Lowest	User shall view their dashboard and different proposals.	UC_18	UC_User_Dashboard_and_Proposal_Viewing
3.0	Lowest	Freelancers shall be able to see client-provided reviews.	UC_19	UC_View_Client_Provided_Review

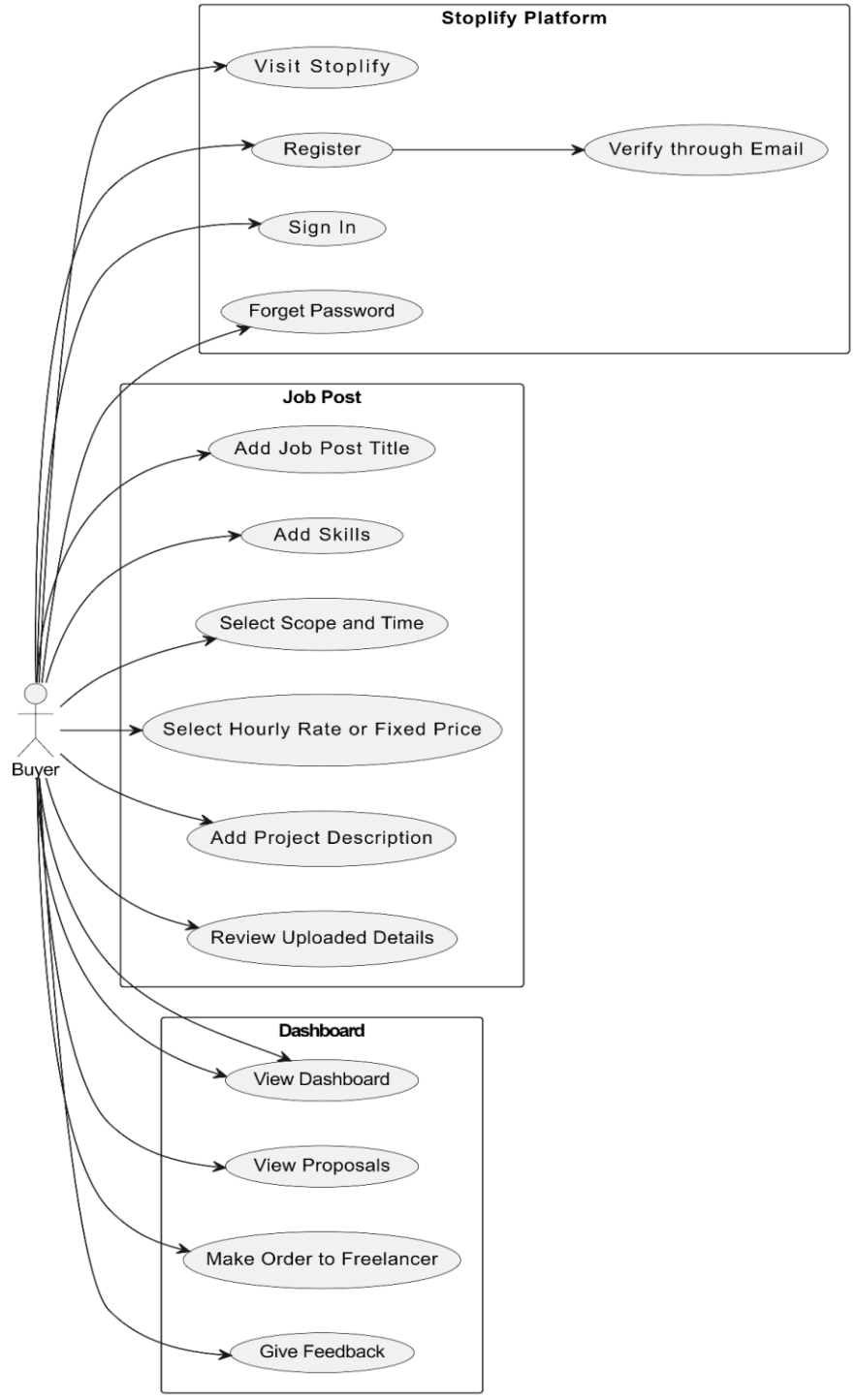
3.0	Lowest	Users shall add a complete description of the project and review their uploaded details.	UC_20	UC_Project_Description

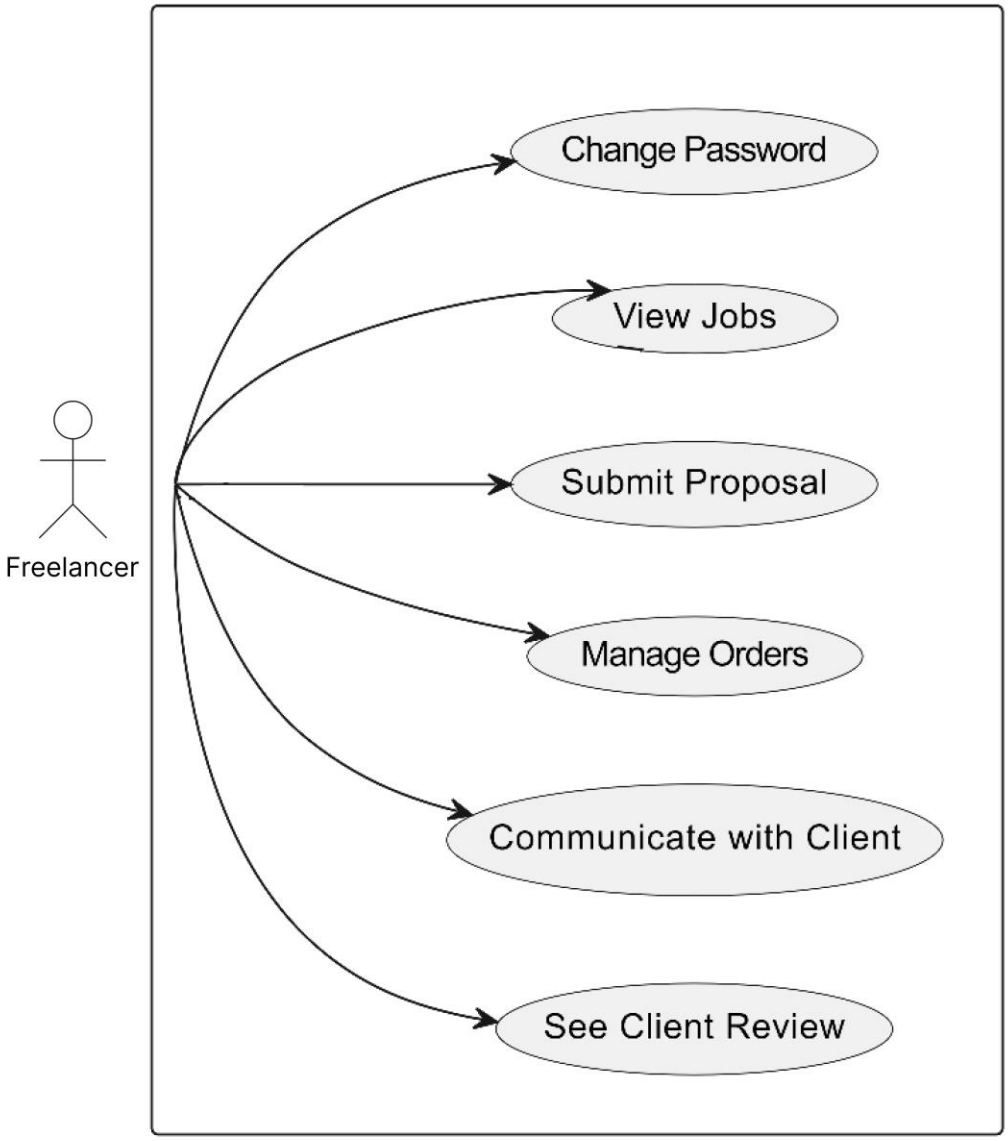
2.1.7. Requirements Trace-ability Matrix:

Sr #	Para #	System Specification Text	Build	Use Case Name	Category
1	1.0	Users shall register within the system. The system shall allow users to register and verify through email.	B1	UC_User_Registration_and_Verification	User
2	1.0	Users shall log in to the system and shall be able to forget their password.	B1	UC_User_Log_in_and_Password_Recovery	User
3	1.0	User shall add the job post title, main skills required, select the estimated scope (Large, Medium, Small), and indicate the time required. Users shall select the hourly rate or fixed price.	B1	UC_Job_Posting	User
4	1.0	Users shall add a complete description of the project and review their uploaded details.	B1	UC_Project_Description	User
5	2.0	User shall view their dashboard and different proposals.	B1	UC_User_Dashboard_and_Proposal_Viewing	User
6	2.0	User shall make an order to a freelancer.	B1	UC_Freelancer_Order_Placement	User
7	2.0	User shall give feedback to the freelancer on job completion.	B1	UC_Feedback_to_Freelancer	User
8	2.0	The system shall provide sign-in for freelancers. Freelancers shall log in and be able to change their password.	B1	UC_Freelancer_Sign_in_and_Password_Change	User
9	3.0	Freelancers shall view the job page, view different jobs, and submit proposals according to their relevant job.	B1	UC_Job_Viewing_and_Proposal_Submission	Freelancer

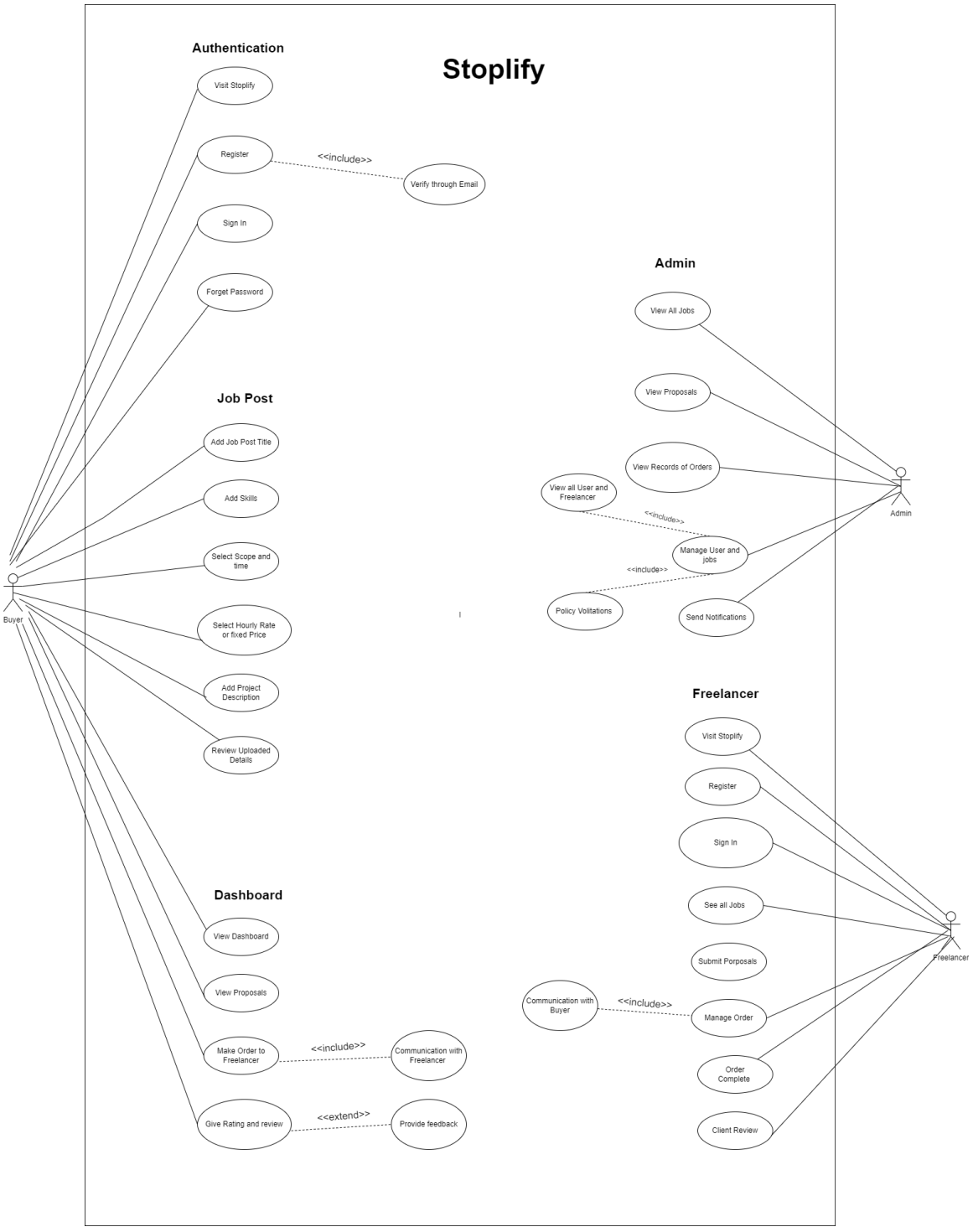
10	3.0	Freelancers shall be able to manage their orders and communicate with clients.	B1	UC_Order_Management_and_Communication	Freelancer
11	3.0 3.0	Freelancers shall be able to see client-provided reviews. The admin shall be able to log in to the Stoplify platform.	B1 B1	UC_View_Client_Provided_Review UC_Admin_Login	Freelancer Admin
12	3.0	The admin shall be able to view all jobs.	B1	UC_Admin_View_Jobs	Admin
13	3.0	The admin shall be able to view proposals.	B1	UC_Admin_View_Proposals	Admin
14	3.0	The admin shall be able to view records of all orders.	B1	UC_Admin_View_Orders	Admin
15	3.0	The admin shall be able to view all users and freelancers.	B1	UC_Admin_View_Users_and_Freelancers	Admin
16	3.0	The admin shall be able to create, delete, update users and jobs.	B1	UC_Admin_User_and_Job_Management	Admin
17	3.0	The admin shall be able to take appropriate actions in case of policy violations.	B1	UC_Admin_Policy_Enforcement	Admin
18	3.0	The admin shall be able to send notifications to users and freelancers.	B1	UC_Admin_Notifications	Admin

2.1.8. High Level Usecase Diagram:




Change Password
View Jobs
Submit Proposal
Manage Orders
Communicate with Client
See Client Review
Freelancer

2.1.9. Analysis Level Usecase Diagram:



2.1.10. Usecase Description

FreeLancer Usecases:

Use case No: 1	Change Password
Description	Allows the freelancer to change their account password for security reasons.
Precondition	Freelancer must be logged into their account.
Basic Flow	1. Freelancer navigates to the account settings. 2. Selects the option to change the password. 3. Enters the current password and the new desired password. 4. Confirms the new password. 5. Submits the changes.
Alternate Flow:	If the entered current password is incorrect, an error message is displayed
Post Condition	Freelancer's account password is updated.

Use case No: 2	View Jobs
Description	Enables the freelancer to browse available job opportunities.
Precondition	Freelancer must be logged into their account.
Basic Flow	1. Freelancer navigates to the job board or search functionality. 2. Views a list of available jobs. 3. Selects a job to view details.
Alternate Flow:	Filtering jobs based on specific criteria (e.g., category, payment range).
Post Condition	Freelancer has explored available job opportunities.

Use case No: 3	Submit Proposal
Description	Allows the freelancer to express interest in a specific job by submitting a proposal.
Precondition	Freelancer must be logged into their account and viewing a job.
Basic Flow	1. Freelancer clicks on "Submit Proposal" for a chosen job. 2. Enters details like bid amount, proposed timeline, and a message to the client. 3. Submits the proposal.
Alternate Flow:	Editing or withdrawing a proposal before the client responds.
Post Condition	Proposal is submitted to the client for review.

Use case No: 4	Manage Orders
Description	Allows the freelancer to manage their active orders, including ongoing and completed work.

Precondition	Freelancer must be logged into their account.
Basic Flow	1. Freelancer navigates to the order management section. 2. Views a list of current and past orders. 3. Accesses details of each order, including client information and project requirements.
Alternate Flow:	Updating the status of an order (e.g., marking it as complete).
Post Condition	Freelancer has a clear overview of their ongoing and completed orders.

Use case No: 5	Communication with Client
Description	Facilitates communication between the freelancer and the client.
Precondition	Freelancer must be logged into their account and have an active project.
Basic Flow	1. Accesses a messaging platform within the system. 2. Sends and receives messages with the client regarding project details.
Alternate Flow:	Initiating a communication channel when a new project is accepted.
Post Condition	Effective communication channel is established between the freelancer and the client.

Admin Usecases:

Use case No: 1	View all jobs
Description	Allows the admin to see a comprehensive list of all available jobs on the platform.
Precondition	The admin is logged in and has the necessary permissions to view job listings.
Basic Flow	• Admin navigates to the "View all jobs" section. • The system displays a list of all existing jobs.
Alternate Flow:	None
Post Condition	The admin has an up-to-date overview of all jobs.

Use case No: 2	View Proposals
Description	Permits the admin to review proposals submitted by freelancers for various jobs.
Precondition	The admin is logged in and authorized to access proposal data.
Basic Flow	• Admin selects the "View Proposals" option. • The system presents a list of proposals with relevant details.
Alternate Flow:	None
Post Condition	The admin is informed about freelancer interest in jobs.

Use case No: 3	View Records of Orders
Description	Enables the admin to see records of orders made on the platform.
Precondition	The admin is authenticated and has the required permissions for order information.
Basic Flow	• Admin navigates to the "View Records of Orders" section. • The system displays a log of all orders, including relevant details.
Alternate Flow:	None
Post Condition	The admin gains insights into the order history.

Use case No: 4	View all users and freelancers
Description	Allows the admin to access a comprehensive list of all users and freelancers registered on the platform.
Precondition	The admin is logged in and has the necessary privileges for user data
Basic Flow	• Admin selects the "View all users and freelancers" option. • The system presents a list with user and freelancer information.
Alternate Flow:	None
Post Condition	The admin has an updated view of all users and freelancers.

Use case No: 5	Manage User and jobs
Description	Permits the admin to perform actions related to user accounts and job management.
Precondition	The admin is logged in and authorized to make changes to user and job data.
Basic Flow	• Admin selects the "Manage User and jobs" option. • The system provides options for user and job management.
Alternate Flow:	The admin may need to handle exceptional cases, such as account suspension or job removal.
Post Condition	The admin successfully manages user accounts and job-related activities.

Use case No: 6	Policy Violations
Description	Alerts the admin about policy violations on the platform.
Precondition	The system continuously monitors user activities for policy violations.
Basic Flow	• The system detects a policy violation. • Admin receives a notification or accesses the "Policy Violations" section.
Alternate Flow:	Admin may need to take corrective actions, such as issuing warnings or suspending accounts.

Post Condition	The admin is aware of and addresses policy violations.

Use case No: 7	Send Notifications
Description	Allows the admin to send notifications to users or freelancers.
Precondition	The admin is logged in and has the authority to send notifications.
Basic Flow	• Admin selects the "Send Notifications" option. • The system provides a platform to compose and send notifications.
Alternate Flow:	Admin may need to specify the recipients and the type of notification.
Post Condition	The targeted users or freelancers receive the notifications.

Admin Usecases:

Use case No: 1	Visit Stoplify
Description	Buyer entity visits the Stoplify platform.
Precondition	None
Basic Flow	1. Buyer opens a web browser. 2. Buyer navigates to the Stoplify website. 3. The Stoplify homepage is displayed.
Alternate Flow:	None
Post Condition	Buyer is on the Stoplify platform.

Use case No: 2	Register
Description	Buyer entity creates an account on Stoplify.
Precondition	Buyer is on the Stoplify platform
Basic Flow	1. Buyer clicks on the "Register" button. 2. Buyer fills out the registration form with required details. 3. Buyer submits the registration form.
Alternate Flow:	None
Post Condition	Buyer has a registered account on Stoplify

Use case No: 3	Verify through Email
Description	Buyer entity verifies their account through email.
Precondition	Buyer has just registered on Stoplify.

Basic Flow	1. Buyer receives a verification email. 2. Buyer clicks on the verification link in the email.
Alternate Flow:	None
Post Condition	Buyer's account is successfully verified.

Use case No: 4	Sign In
Description	Buyer entity logs into their Stoplify account.
Precondition	Buyer has a registered and verified account.
Basic Flow	1. Buyer clicks on the "Sign In" button. 2. Buyer enters their credentials. 3. Buyer clicks on the "Sign In" button.
Alternate Flow:	None
Post Condition	Buyer is logged into their Stoplify account.

Use case No: 5	Forget Password
Description	Buyer entity resets their password.
Precondition	Buyer is on the Sign-In page and has forgotten the password.
Basic Flow	1. Buyer clicks on the "Forgot Password" link. 2. Buyer enters their email address. 3. Buyer receives a password reset email. 4. Buyer follows the instructions in the email to reset the password.
Alternate Flow:	None
Post Condition	Buyer's password is successfully reset.

Use case No: 6	Add Job Post Title
Description	Buyer entity creates a new job post by adding a title.
Precondition	Buyer is logged into their Stoplify account.
Basic Flow	1. Buyer navigates to the job posting section. 2. Buyer clicks on the "Add Job Post" button. 3. Buyer enters a title for the job post.
Alternate Flow:	None
Post Condition	Job post is created with a title.

Use case No: 7	Add Skills
Description	Buyer entity adds required skills to a job post.
Precondition	Buyer is in the process of creating a job post.
Basic Flow	1. Buyer adds skills to the job post.
Alternate Flow:	None

Post Condition	Job post includes specified skills.

Use case No: 8	Select Scope and Time
Description	Buyer entity defines the scope and time requirements for a job post.
Precondition	Buyer is in the process of creating a job post.
Basic Flow	1. Buyer selects the scope of the project. 2. Buyer specifies the timeframe for project completion.
Alternate Flow:	None
Post Condition	Job post includes defined scope and time requirements.

Use case No: 9	Select Hourly Rate or Fixed Price
Description	Buyer entity chooses between hourly rate or fixed price for a job.
Precondition	Buyer is in the process of creating a job post.
Basic Flow	Buyer selects either an hourly rate or a fixed price for the job.
Alternate Flow:	None
Post Condition	Job post includes the selected pricing model.

Use case No: 10	Add Project Description
Description	Buyer entity provides a detailed description of the project in the job post.
Precondition	Buyer is in the process of creating a job post.
Basic Flow	Buyer enters a detailed project description.
Alternate Flow:	None
Post Condition	Job post includes a comprehensive project description.

Use case No: 11	Review Uploaded Details
Description	Buyer entity reviews all the details uploaded for a job post.
Precondition	Buyer has filled in all the necessary details for a job post.
Basic Flow	Buyer reviews all the details entered for accuracy.
Alternate Flow:	None
Post Condition	Buyer confirms the accuracy of the entered details.

Use case No: 12	View Dashboard
Description	Buyer entity views their Stoplify dashboard.
Precondition	Buyer is logged into their Stoplify account.
Basic Flow	Buyer clicks on the "Dashboard" link.

Alternate Flow:	None
Post Condition	Buyer is on their Stoplify dashboard..

Use case No: 13	View Proposals
Description	Buyer entity reviews proposals received for a job post.
Precondition	Buyer is logged into their Stoplify account and has posted a job.
Basic Flow	1. Buyer navigates to the job post. 2. Buyer views the proposals submitted by freelancers.
Alternate Flow:	None
Post Condition	Buyer has reviewed proposals for the job post.

Use case No: 14	Make Order to Freelancer
Description	Buyer entity places an order with a selected freelancer.
Precondition	Buyer has selected a freelancer and is satisfied with the proposal.
Basic Flow	1. Buyer clicks on the "Make Order" button. 2. Buyer confirms the details of the order. 3. Buyer completes the order process.
Alternate Flow:	None
Post Condition	Order is placed with the selected freelancer.

Use case No: 15	Give Feedback
Description	Buyer entity provides feedback on a completed project.
Precondition	Buyer has received the completed project.
Basic Flow	1. Buyer navigates to the completed project. 2. Buyer provides feedback and ratings for the freelancer.
Alternate Flow:	None
Post Condition	Feedback is submitted for the completed project.

Chapter 3: Design Document (For Object Oriented Approach)

3.1. Introduction:

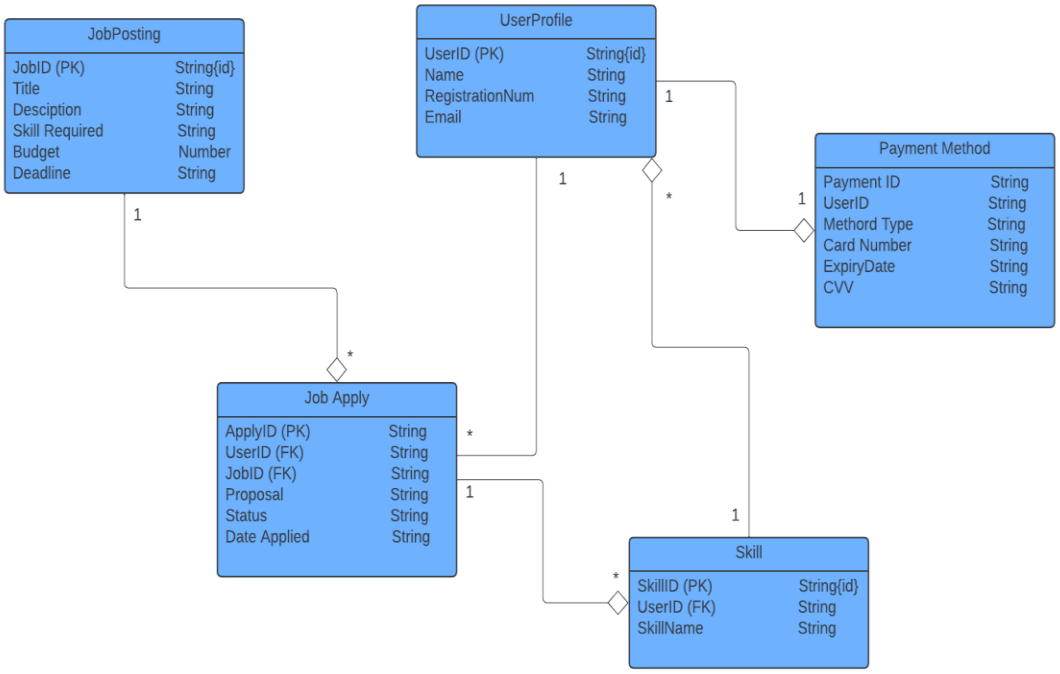
Stoplify is a comprehensive online platform designed to revolutionize the freelance marketplace, offering a seamless and user-friendly experience for both clients and freelancers. This project follows a robust Object-Oriented Approach in its design, emphasizing modularity, scalability, and maintainability. The system is structured around key object-oriented principles, with distinct classes representing entities such as Users, Proposals, Jobs, Portfolios, and Learning Modules. The design prioritizes encapsulation, inheritance, and polymorphism, fostering a flexible architecture that accommodates future enhancements and modifications. Through a meticulously crafted class hierarchy and well-defined relationships, Stoplify aims to provide a dynamic and intuitive environment for users to effortlessly navigate job postings, submit proposals, communicate efficiently, showcase portfolios, and engage in continuous learning. The Object-Oriented Approach ensures that the system is adaptable to evolving requirements, promotes code reusability, and facilitates a robust foundation for the development and evolution of the Stoplify platform.

1. Domain Model
2. System Sequence Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Operation Contracts
6. Design Class Diagram
7. State Transition Diagram
8. Data Model

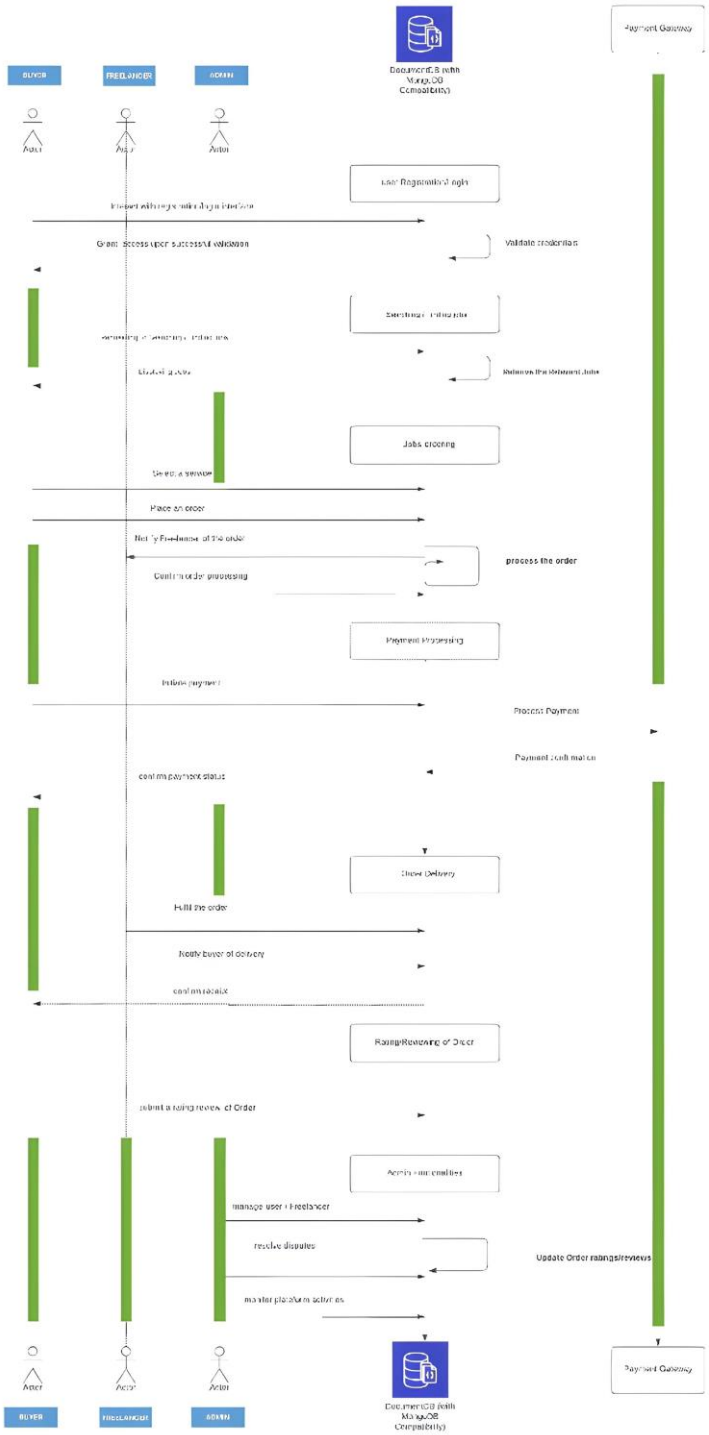
Now we discuss these artifacts one by one as follows:

3.2. Domain Model

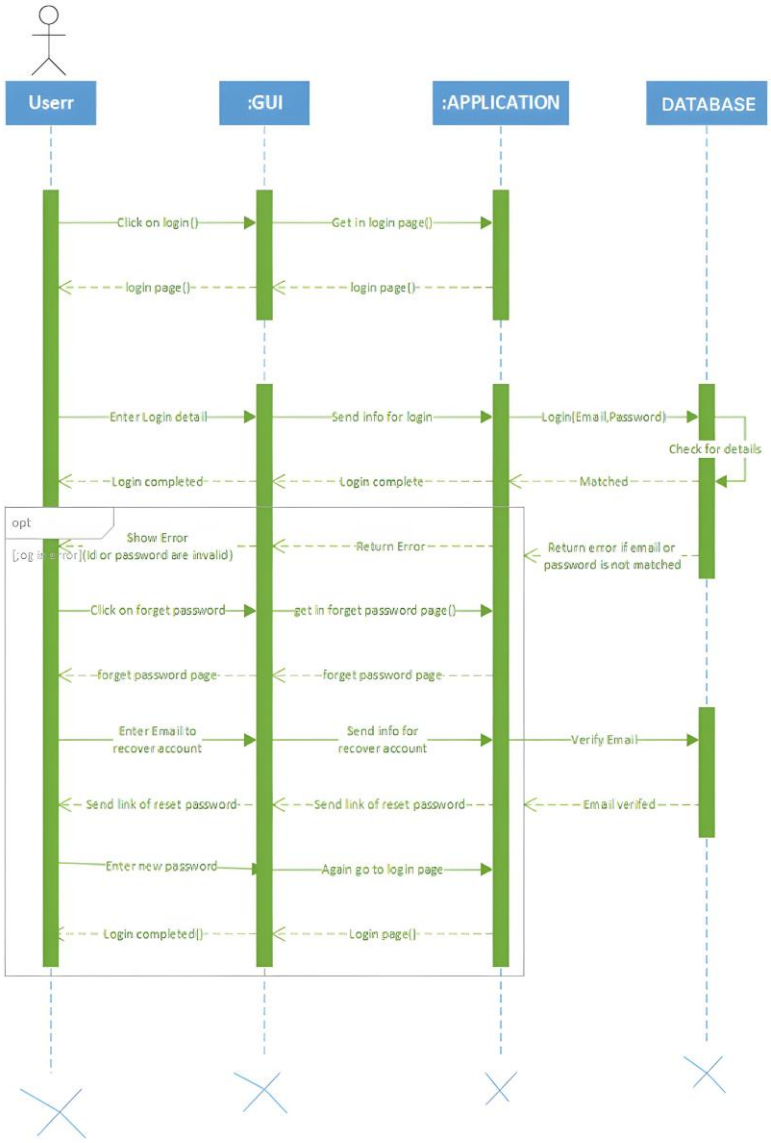
Domain model is a conceptual model that represents the entities and relationships within a specific problem domain. In this case, we have entities such as JobPosting, UserProfile, PaymentMethod, JobApply, and Skill. Let's create a diagram of the domain model along with descriptions and relationships.line definition will describe the domains necessary to build systems in the product line.

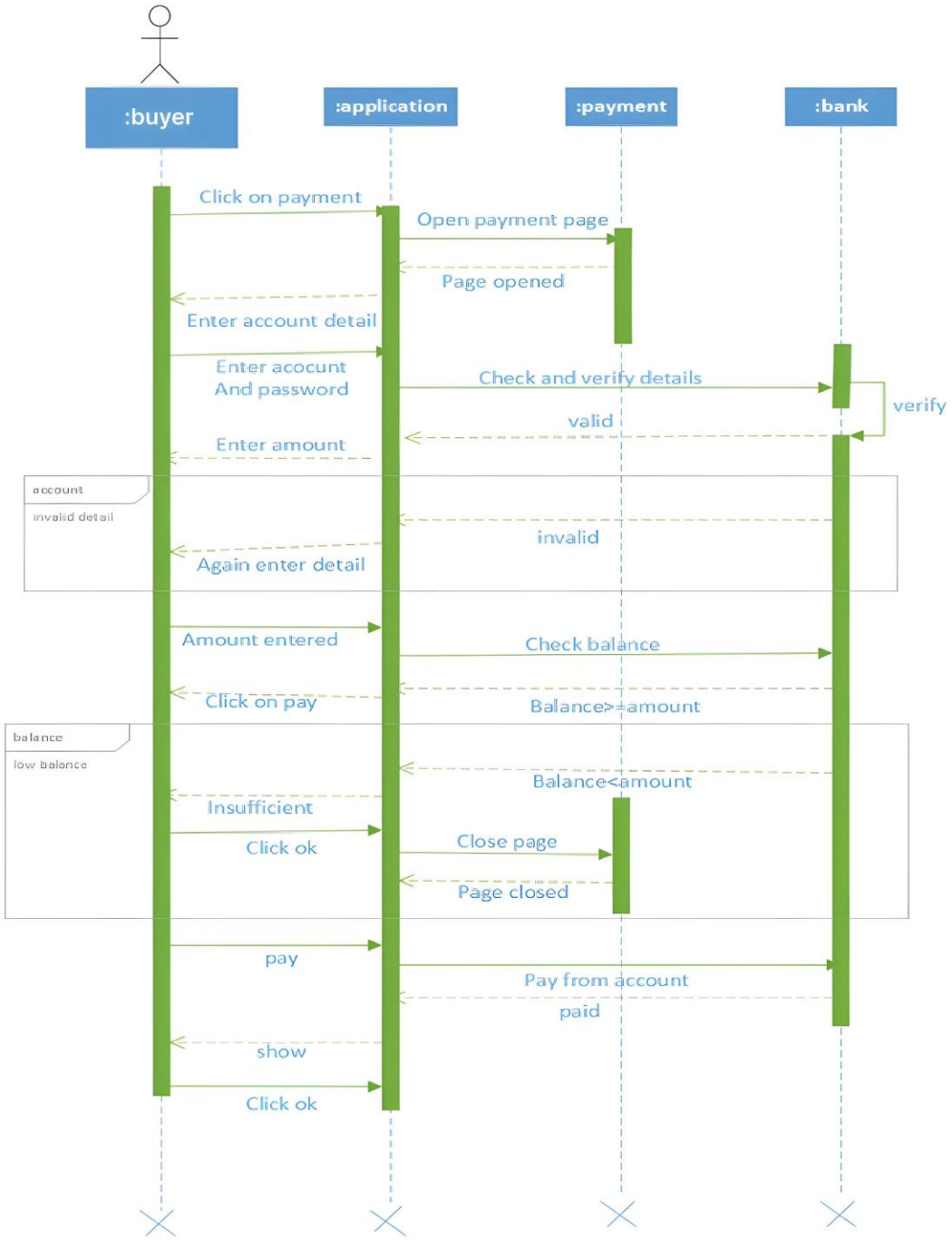


3.3. System Sequence Diagram

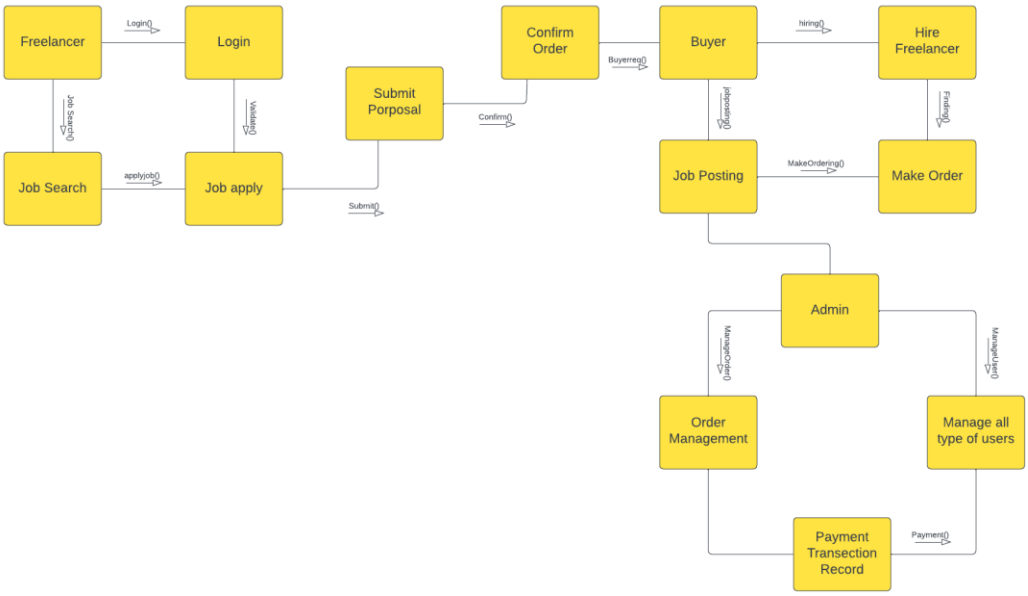


3.4. Sequence Diagram





3.5. Collaboration Diagram



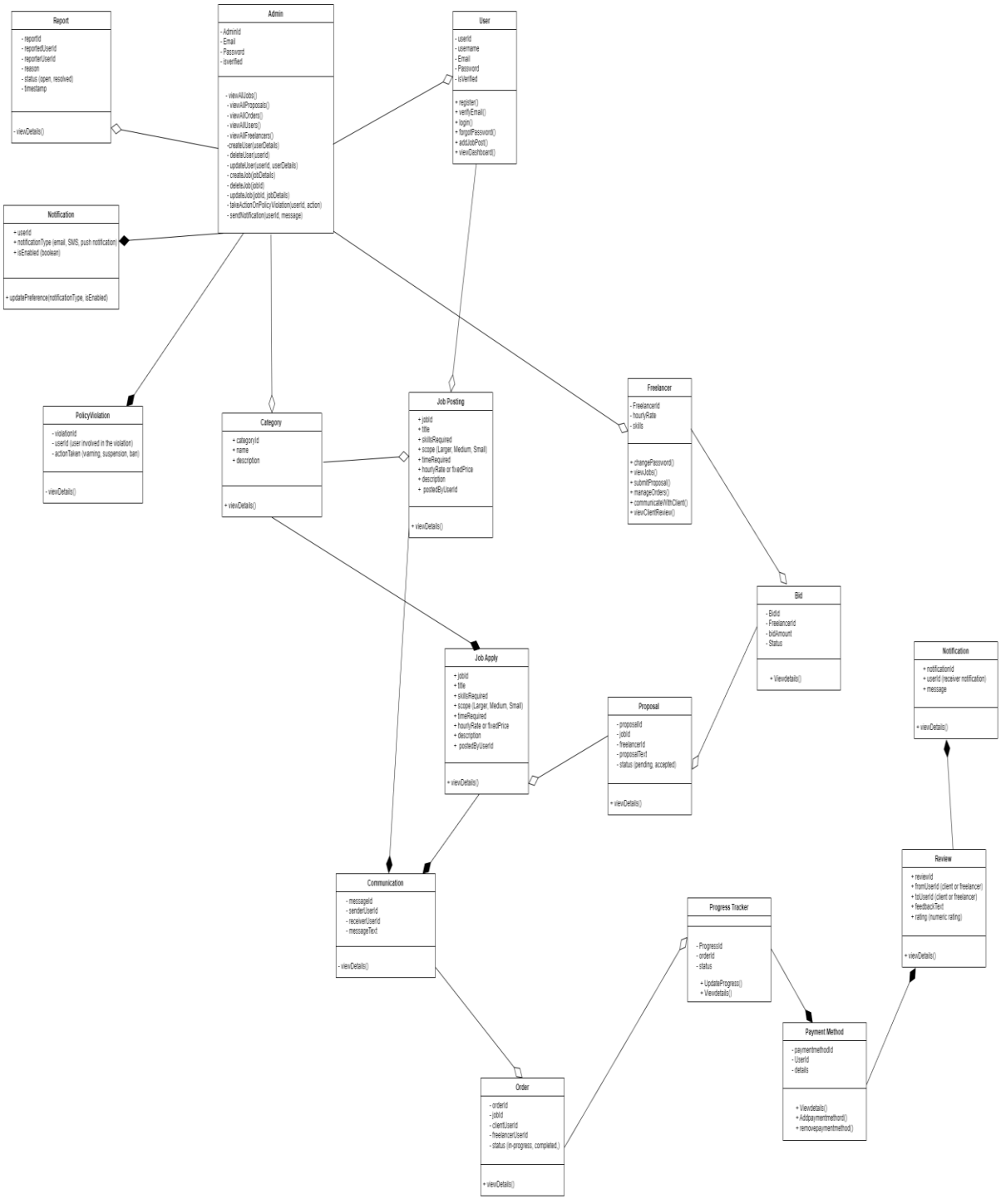
3.6. Operation Contracts

A UML operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does

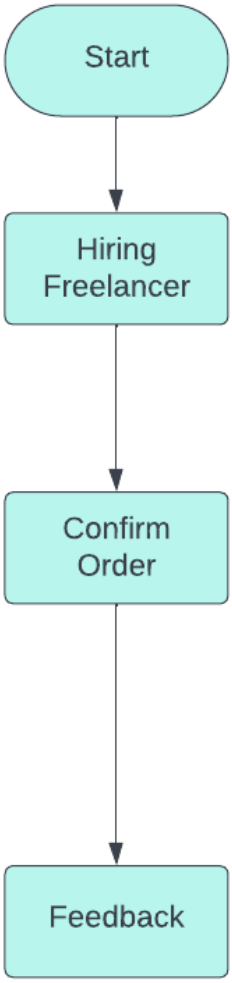
Operation Name	Responsibility	Cross-Reference	Pre-Condition	Post-Condition
User registration.	Register and verify via email.	UC_User_Registration_and_Verification	-	User registered and verified.
User log in and password recovery.	Enable log in; password recovery.	UC_User_Log_in_and_Password_Recovery	User account exists.	User logged in; password recovered if needed.
Job posting details.	Add job post title, skills, etc.	UC_Job_Posting	User is logged in.	Job post details added.
Verified Payment Methods	Encouraging users to link and verify payment methods for trust and security.	Uc_Payment_Methord	Users have valid accounts and are logged in.	Payment methods are successfully linked and verified.

Proposals	Allowing freelancers to submit proposals to clients for job opportunities.	Uc_Proposals	Job posting live, users logged in.	Proposals submitted and visible to clients.
Messaging and Communication	Facilitating client-freelancer communication via messaging.	Uc_Messaging	Users are logged in	Parties are now in real-time communication
Rating	Reviews	Uc_Reviews and Ratings	Must place an order	Rating must be stored in blockchain
View all users and freelancers.	Access and view user and freelancer details.	UC_Admin_Viw_Users_and_Freelancers	Admin is logged in.	All user and freelancer details displayed.

3.7. Design Class Diagram



3.8. State chart diagram



Chapter 3: Software Requirement Specification and Design Document (For structured Approach)

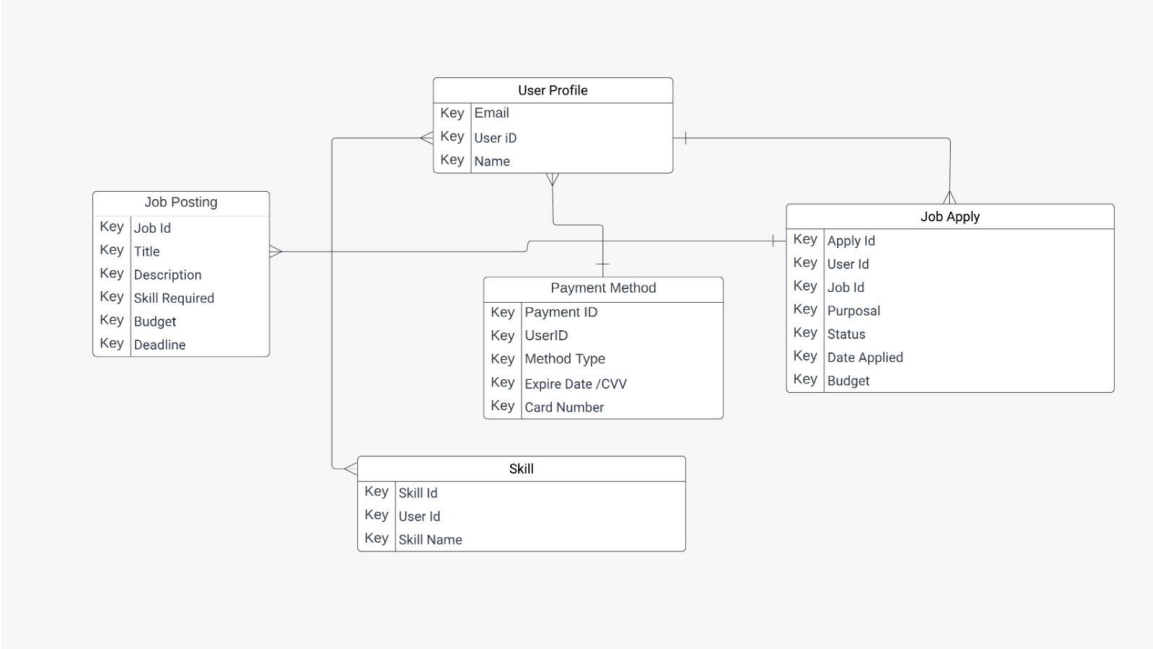
3.1. Introduction:

Analysis & Design Model for structured approach must contain following artifacts:

1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design
5. Component Level Design

3.2. Entity Relationship Diagram:

In the analysis model, Entity Relationship Diagram is used to understand the system under consideration with respect to entities involved and their relationships.



Each entity is documented by extracted its attributes, cardinality, and modality.

1. JobPosting Entity:

Attributes:

- JobID (Primary Key)
- Title
- Description
- SkillsRequired
- Budget
- Deadline
- CompanyName
- DatePosted

Cardinality and Modality:

- JobPosting **(1)** ------< JobApply **(Many) (One-to-Many)**
- JobPosting **(Many)** >------ JobApply **(1) (Many-to-One)**

2. UserProfile Entity:

Attributes:

- UserID (Primary Key)
- Name
- RegistrationNumber
- Email
- Signature

Cardinality and Modality:

- UserProfile **(1)** ------< JobApply **(Many) (One-to-Many)**
- UserProfile **(1)** >------ PaymentMethod **(Many) (One-to-Many)**
- UserProfile **(Many)** >------ Skill **(Many) (Many-to-Many)**

3. PaymentMethod Entity:

Attributes:

- PaymentID (Primary Key)
- UserID (Foreign Key)
- MethodType
- CardNumber
- ExpiryDate
- CVV

Cardinality and Modality:

- UserProfile **(Many)** ------< PaymentMethod **(1) (Many-to-One)**

4. JobApply Entity:

Attributes:

- ApplyID (Primary Key)
- UserID (Foreign Key)
- JobID (Foreign Key)

- Proposal
- Status
- DateApplied

Cardinality and Modality:

- UserProfile **(Many)** ------< JobApply **(Many) (One-to-Many)**
- JobPosting **(1)** ------< JobApply **(Many) (Many-to-One)**

5. Skill Entity:

Attributes:

- SkillID (Primary Key)
- UserID (Foreign Key)
- SkillName

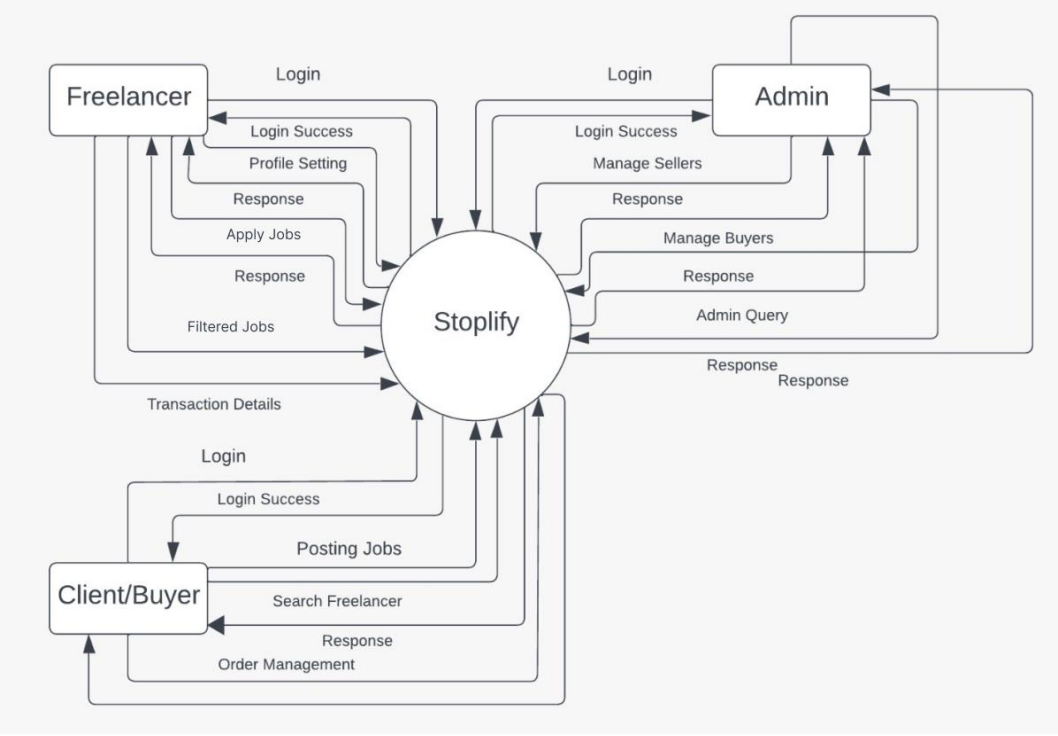
Cardinality and Modality:

- UserProfile **(Many)** >------ Skill **(Many) (Many-to-Many)**

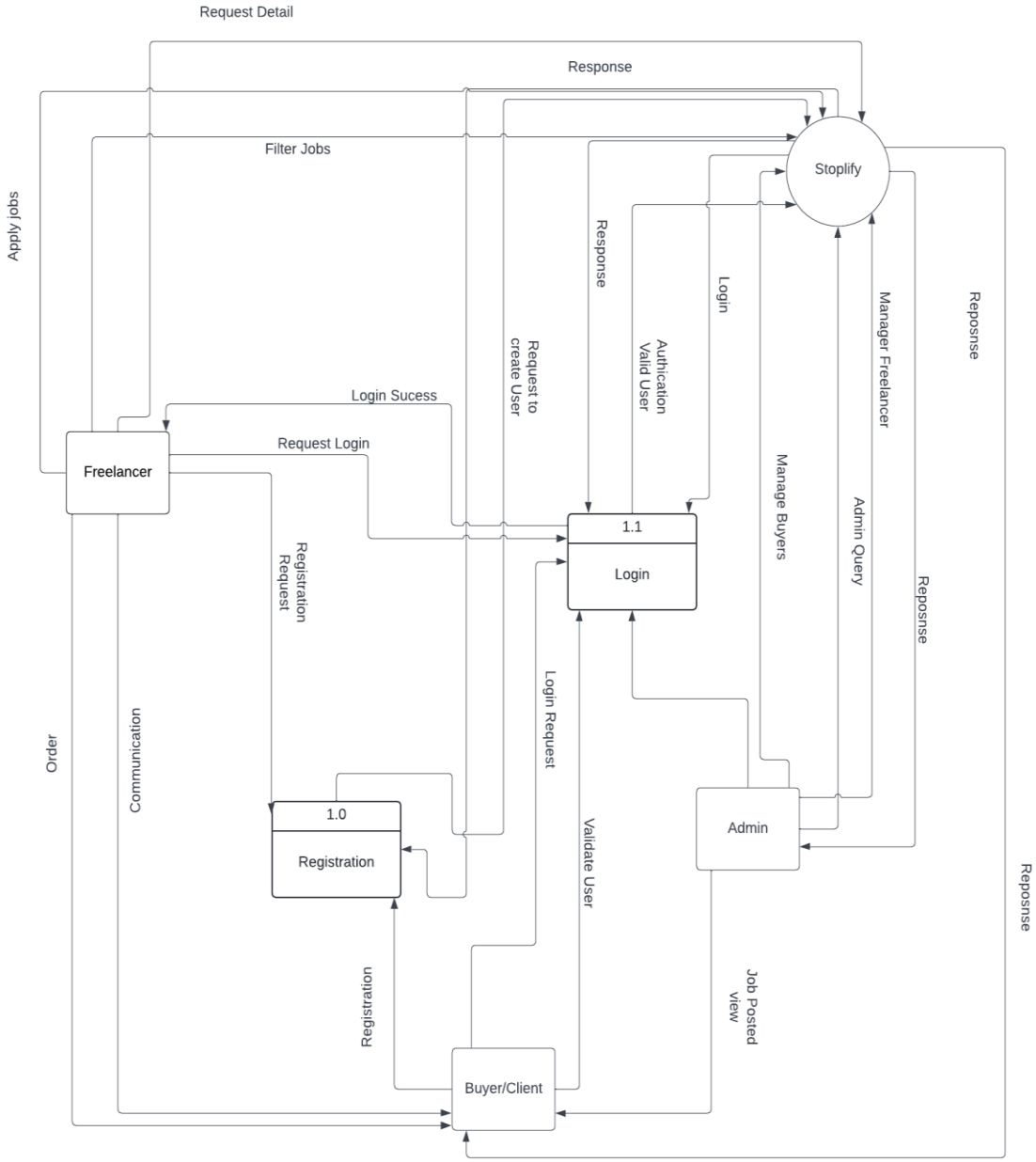
3.3. Data flow diagram (Functional Model)

DFD is all about to identify the major processes in your system and develop Data Flow Diagram up to required level.

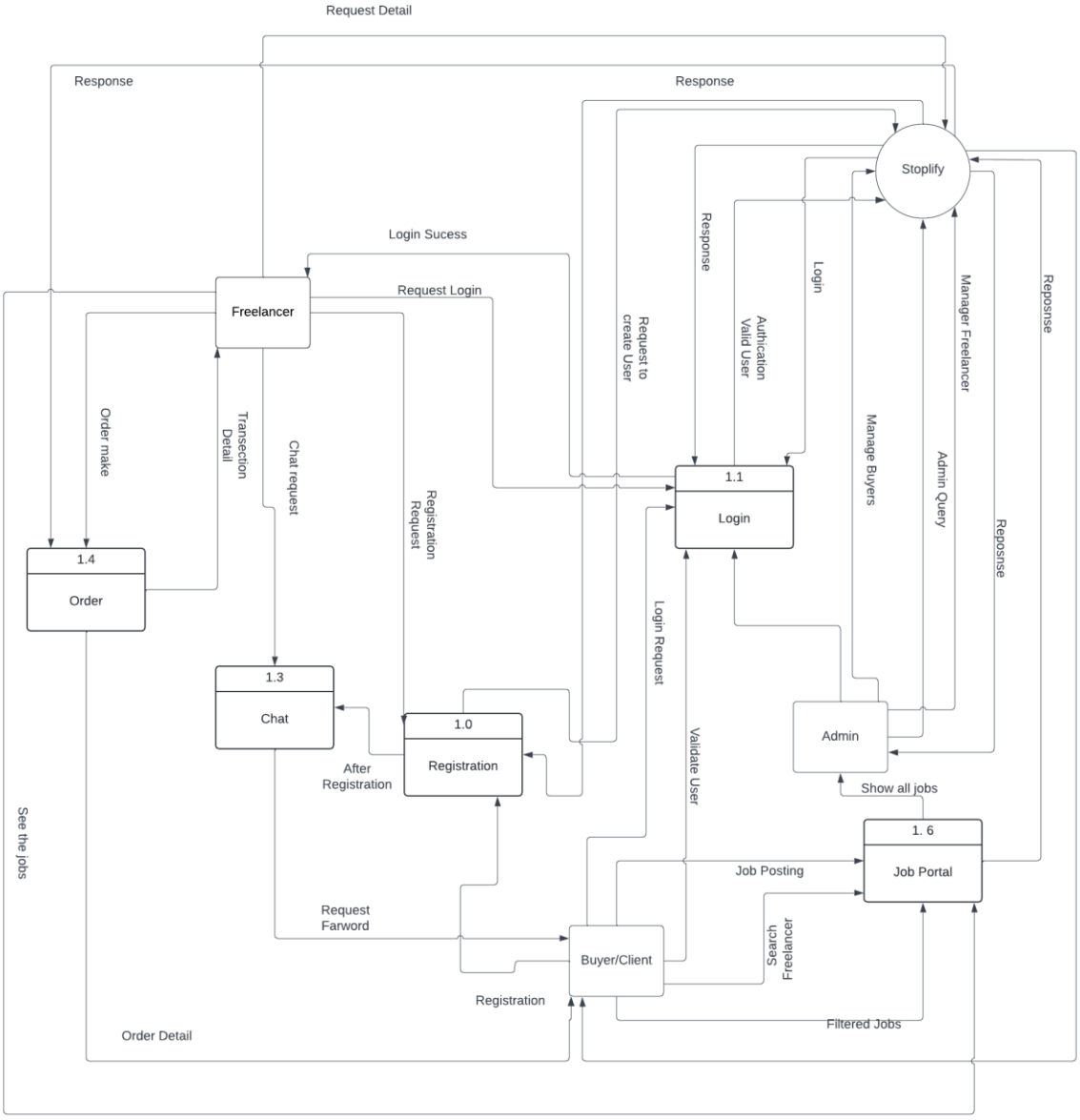
Context Level DFD:



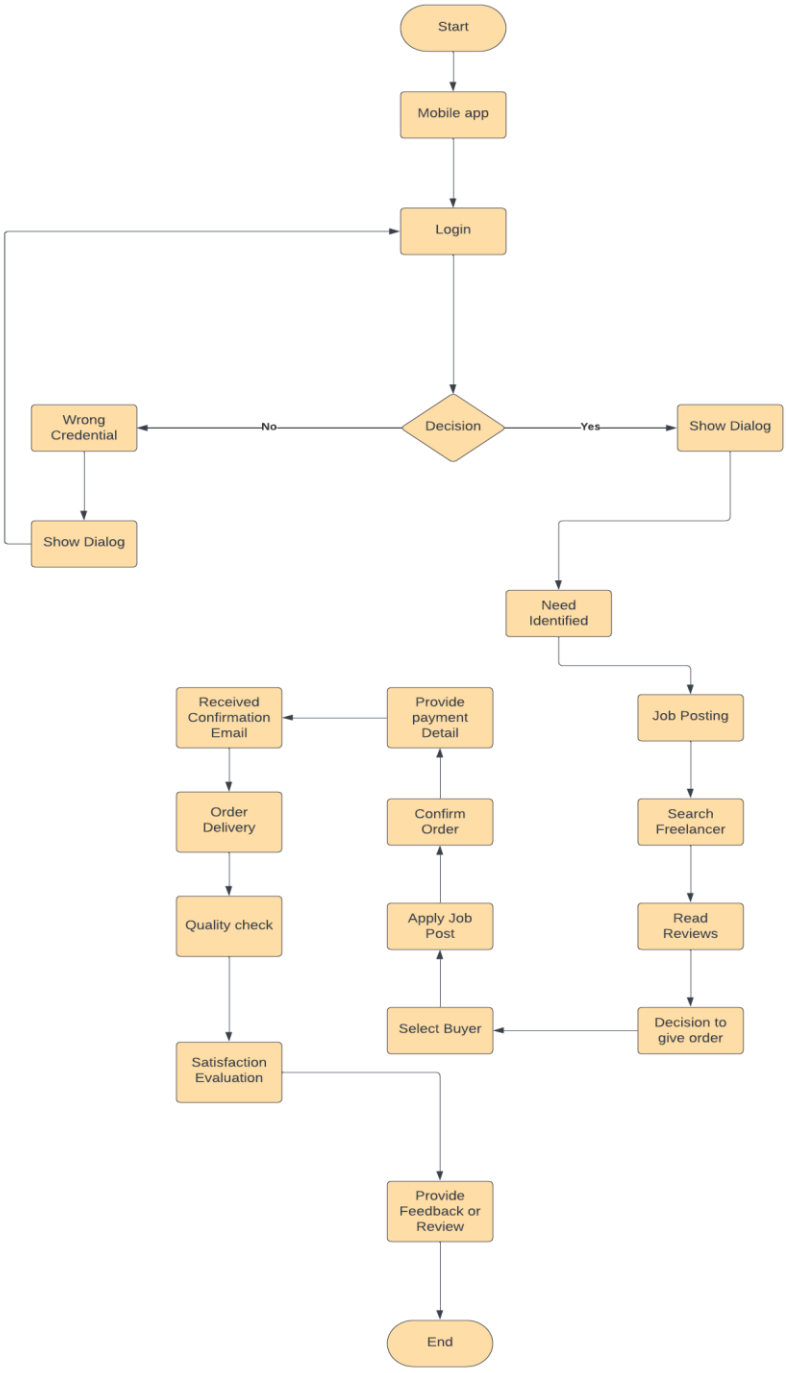
Level 1 Diagram



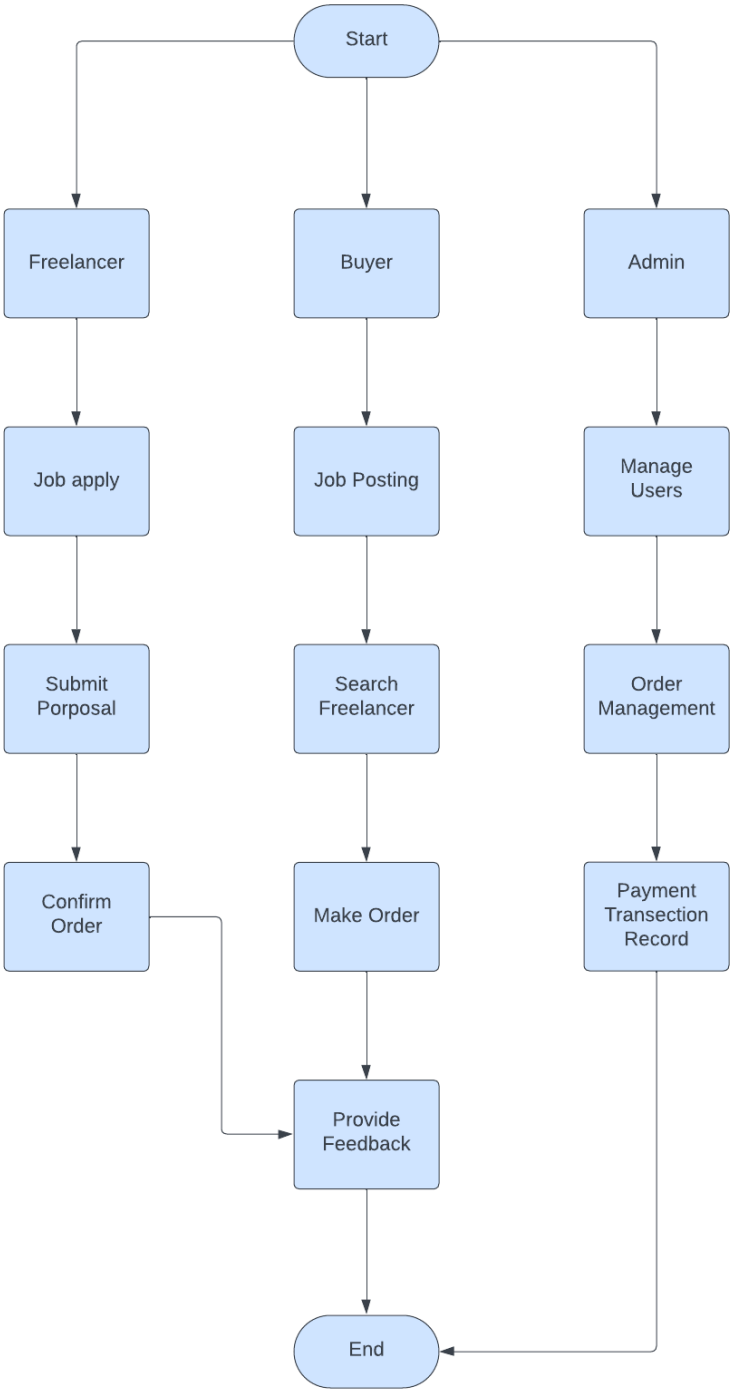
Level 2 Diagram



3.4. State Transition Diagram



3.5. Architectural design



Chapter 4: User Interface Design

4.1. Introduction

A user interface design consists of three main parts:
Page elements should be visualized on paper before building them in the computer. Just as you draw a site map to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

4.2. Site Maps

A site map's main benefit is to give users an overview of the site's areas in a single glance by dedicating an entire page to a visualization of the information architecture. If designed well, this overview can include several levels of hierarchy, and yet not be so big that users lose their ability to grasp the map as a whole. Some of the site maps we studied stretched over six screens on a standard 800x600 monitor. This is much too much. We recommend keeping the site map short; it should be no more than two-and-a-half times the window size most common among your users.
The greatest failures in our study came from site maps that attempted to lure the user into a dynamically twisting and expanding view, rather than presenting a simple, static representation of the information architecture. The site map's goal is to give users a single overview of the information space. If users have to work to reveal different parts of the map, that benefit is lost.
Dynamic site maps are basically an alternative way of navigating through the information space using a set of non-standard interaction techniques. For example, one site used a hyperbolic tree, where users had to click and drag clusters of links around the screen to expand areas of interest. Nobody could do this well.
A site map should not be a navigational challenge of its own. It should be a map.
As we have found again and again, users hated non-standard user interfaces that forced them to learn a special way of doing things for the sake of a single website. Site maps should be simple, compact layouts of links, and they should show everything in a single view.
If your site is large and complex, it is a good idea to include and index or site map that provides an outline of our site's content. This concept is something that we are all familiar with in the print world and it is very useful on the Web. An index is usually an alphabetical listing of key words that link to the appropriate pages in the site or it can be more like a table of contents. A site map is a graphical representation of the site's content. It doesn't usually have as much detailed information as the index has.

As web sites get more complicated, an index or site map is going to become more and more valuable and essential to the navigation of a good site.

- **Buyer panel**
- Splash Screen
- Login
- Register User
- Request email verify
- Select Join as a Client option
- Create a job
- Add Title
- Add Skills
- Add scope of project
- Add budget (Hourly or Fixed rate)
- Add Description
- Post a Job
- All Listed Jobs
- All Proposal
- View proposal (Specific jobs)
- Proposal Include Cover letter
- Hire Freelancer
- Start Order
- Billing Information
- Submit Order
- Submit Review and Rating
- Complete Order

- **Seller panel**
- Splash Screen
- Login
- Register User
- Request email verify
- Select Freelancer, Looking for work
- Complete Profile
- Upload picture, Professional Role
- Add Experience years
- Select Skills
- Add Country
- Add City
- Add State/province
- Add Zip Code
- All Jobs
- Job Details (Select specific job)
- Submit proposal for specific job

- See All Submitted Proposal
- Chat with buyer
- Order management and Completion
- Learning Skills of specific mentors
- Chat Bot implemented

4.3. Story boards

A storyboard is a sequence of single images, each of which represents a distinct event or narrative. It is also a visual representation of the script illustrating the interaction between the user and the machine. It can also be imagined as a film in visual-outline form.
A storyboard can be used in two ways. It describes the task, which are a series of images showing the user, environment and the machine. It also describes the interface, which represent series of screen images indicating the user's representation and the computer's response and work out interaction details when asking, "what happens next?" It also shows interaction sequence at a glance and helps develop usage scenarios to help develop tools & tasks.
All this can be done to construct a visual & verbal sequence that illustrates the interaction. Consider …

- Environment -- where system is used
- Visual cues -- what user can see
- Audible cues -- what user can hear
- Tactile cues -- what user can touch
- User input -- how the user communicates to the machine
- Machine output -- how the machine responds to the user
- User's emotions -- how user perceives and responds to the interaction
- Technology -- what technology is involved in performing the task
- Quality of experience -- what benefit is perceived

4.4. Navigational maps:

The next step is of navigational maps. In these maps, the storyboards are used as an input. The different display buttons or action buttons show the navigation from one screen to the other. In other words when one action button is pressed it would lead to other screens. This path and navigation would be shown.

Buyer Panel

Figure 4.1

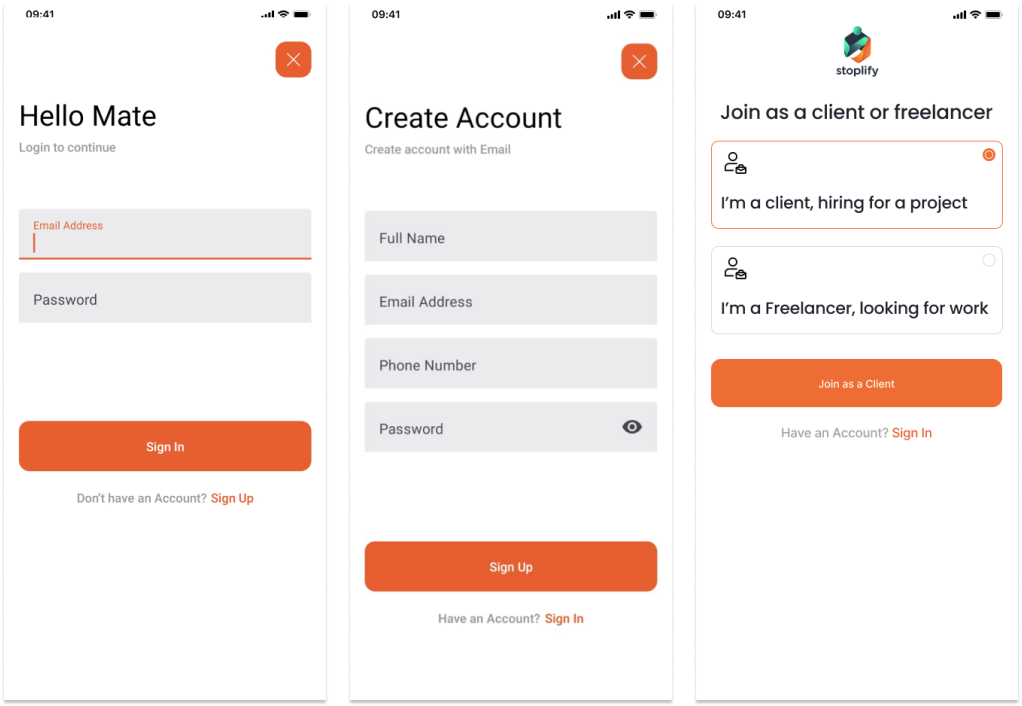


Figure 4.2

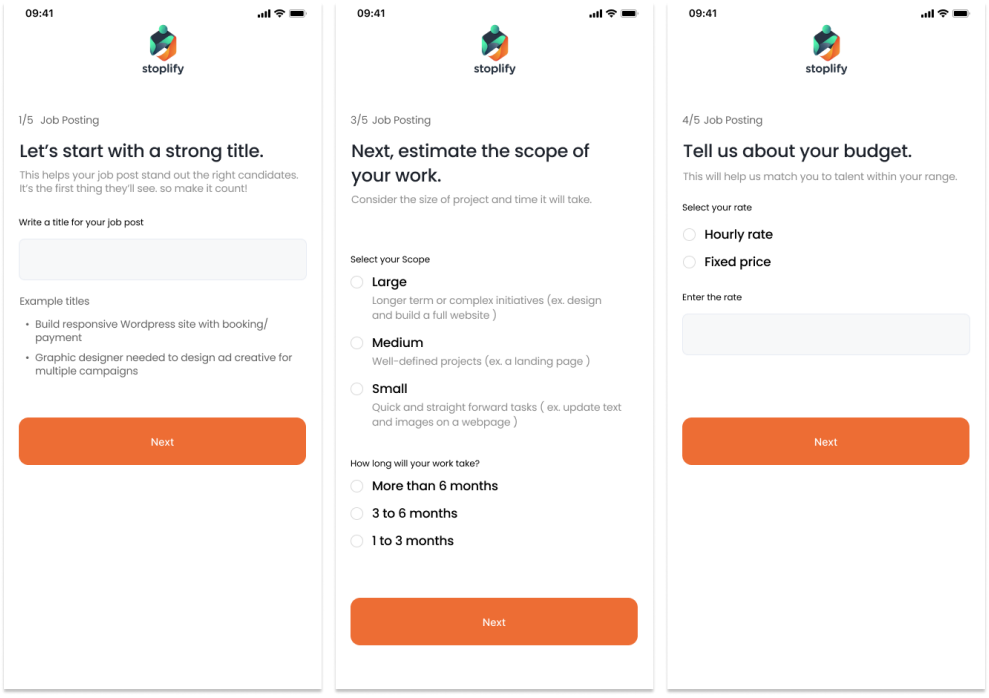


Figure 4.3

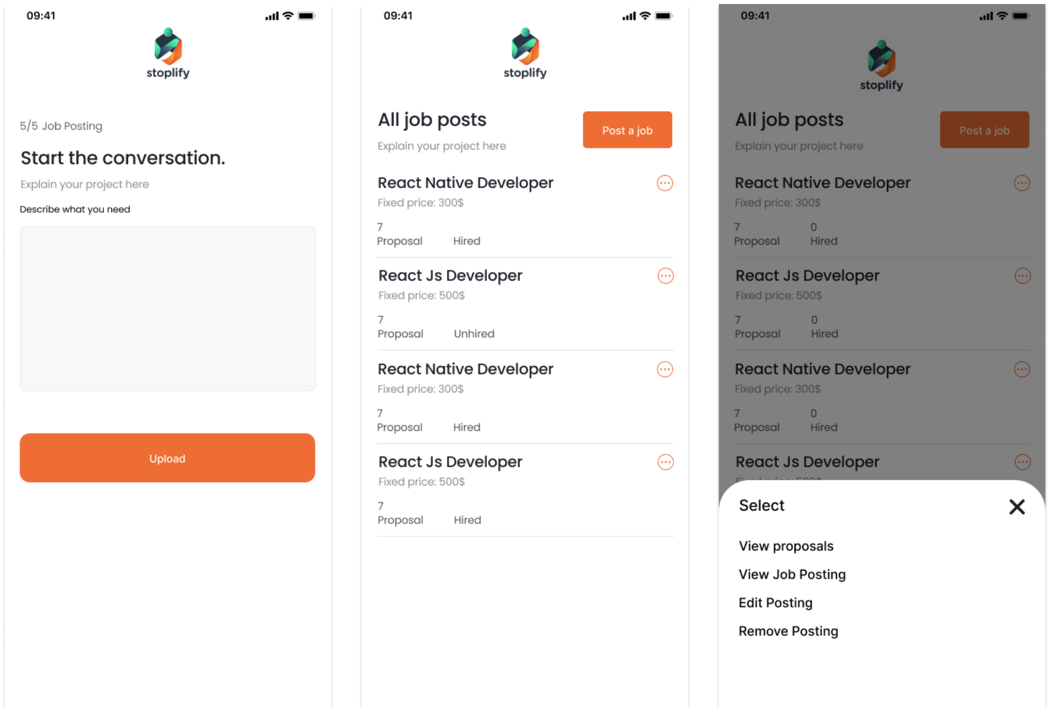


Figure 4.4

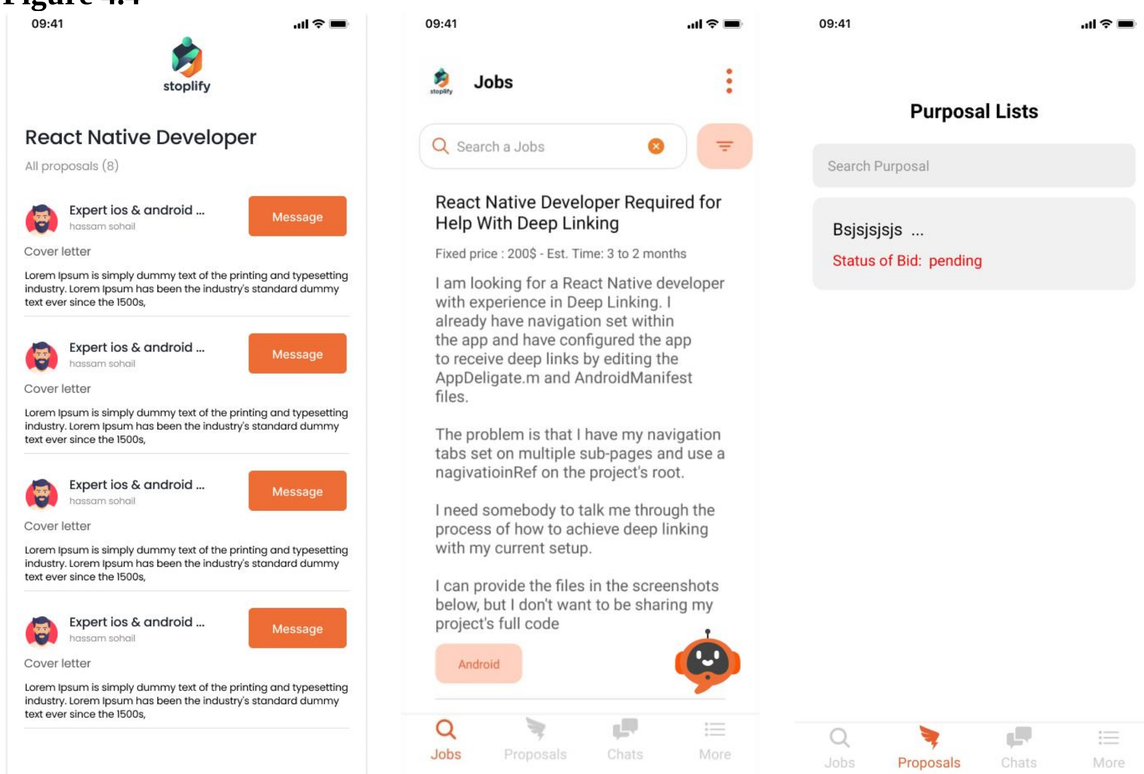
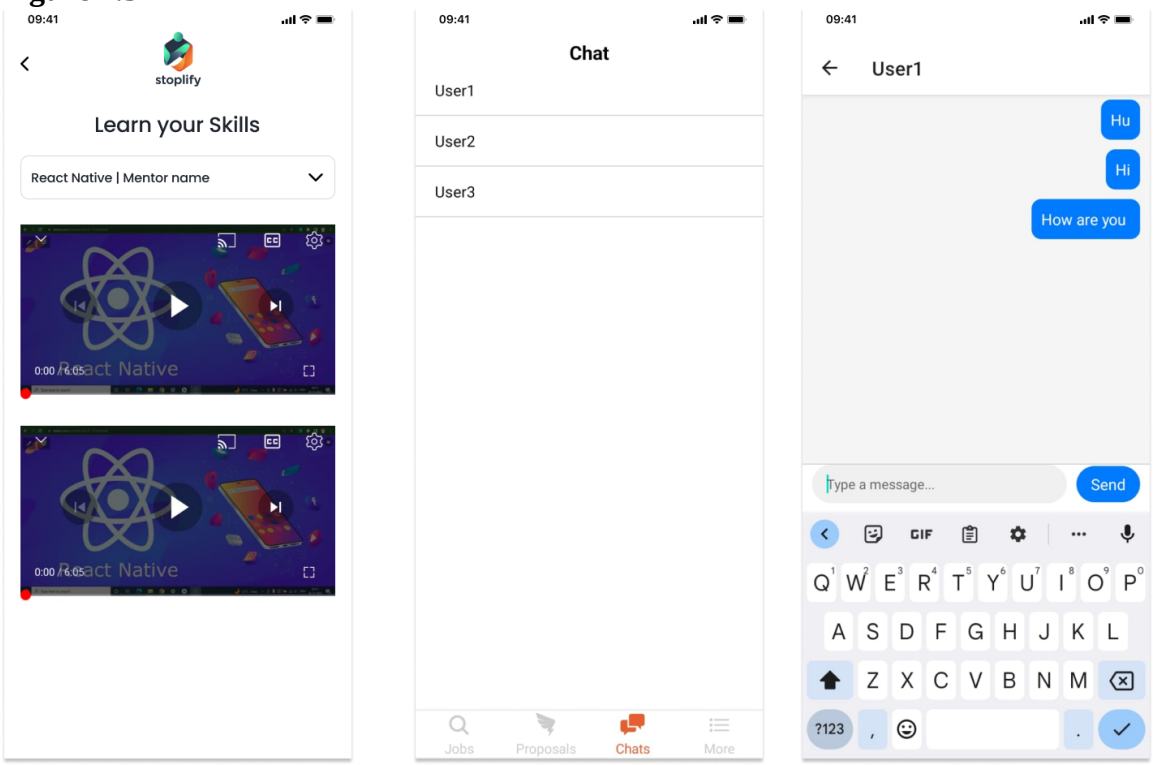


Figure 4.5



Chapter 5: Software Testing

5.1 Introduction:

SOFTWARE TESTING is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT). This chapter introduces testing software to the audience and justifies its importance.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

5.2. Test plan:

5.2.1. Purpose:

A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, and the limitations of the testing and the schedule of the testing activities. To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.2.2. Outline:

A test plan shall have the following structure.

5.2.2.1. Test plan identifier

Test plan identifier: ST-Test-01

5.2.2.2. Introduction

Summarize the software items and software features to be tested. The need for each item and its history may be included. References to the following documents, when they exist,

are required in the highest-level test plan:

a. Project authorization;
b. Project plan;
c. Quality assurance plan;
d. Configuration management plan;
e. Relevant policies;
f. Relevant standards.

In multilevel test plans, each lower-level plan must reference the next higher-level plan.

5.2.2.3. Test items

Identify the test items including their version/revision level. Also specify characteristics of their transmittal media that impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (e.g., programs must be transferred from tape to disk).
Supply references to the following test item documentation, if it exists:

a) Requirements specification
b) Design specification
c) Users guide
d) Operations guide
e) Installation guide
Reference any incident reports relating to the test items. Items that are to be specifically excluded from testing may be identified.

5.2.2.4. Features to be tested

- Registration
- Login
- Email Verification
- Change Password
- Forget Password
- Job Created
- Job Post
- Search Filter
- Proposal Submission
- Chatting
- Ordering
- Submit Review & Rating
- Chat Bot
- Learning Module
- View Profile
- Logout

5.2.2.5. Features not to be tested
Other than those mentioned above.

5.2.2.6. Approach
- Acceptance testing
- Unit Testing
- Integration Testing
- Usability Testing

5.2.2.7. Item pass/fail criteria
Works as inspected (desired output).

5.2.2.8. Suspension criteria and resumption requirements
Unexpected output (undesired output).

5.2.2.9. Test deliverables
Identify the deliverable documents. The following documents should be included:

a. Test plan;
b. Test design specifications;
c. Test case specifications;
d. Test procedure specifications;
e. Test item transmittal reports;
f. Test logs;
g. Test incident reports;
h. Test summary reports.

5.2.2.10. Testing tasks
● A sample data of stats will be needed for payment calculation.
● Mock data for signup
● Mock data for otp
● Mock data for verification

5.2.2.11. Environmental needs
No special environmental needs as the application can run on normal mobile applications with the internet

5.2.2.12. Responsibilities
- Hassam Sohail
- Ahmad Raza
- Rimsha Noor

5.2.2.13 Approvals

Name: Mr. Bilal Tariq Butt
Title: Project Supervisor
Date: Signature:_____

5.3. Test design specification

5.3.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.3.2. Outline

A test plan shall have the following structure:

a. Test plan identifier;
b. Introduction;
c. Test items;
d. Features to be tested;
e. Features not to be tested;
f. Approach;
g. Item pass/fail criteria;
h. Suspension criteria and resumption requirements;
i. Test deliverables;
j. Testing tasks;
k. Environmental needs;
l. Responsibilities;
m. Staffing and training needs;
n. Schedule;
o. Risks and contingencies;
p. Approvals.

5.3.2.1 Test plan identifier

Specify the unique identifier assigned to this test plan.

5.3.2.2. Introduction

In this section we will explain all the test design specification activities.

5.3.2.3. Test items

5.3.2.4. Features to be tested

- Registration
- Login
- Email Verification
- Change Password
- Forget Password
- Job Created
- Job Post
- Search Filter
- Proposal Submission
- Chatting
- Ordering
- Submit Review & Rating
- Chat Bot
- Learning Module
- View Profile
- Logout

5.3.2.5. Features not to be tested

Other than those mentioned above.

5.3.2.6. Approvals

Name: Mr. Bilal Tariq Butt
Title: Project Supervisor
Date: Signature:_____

5.3.2.7. Item pass/fail criteria

Works as inspected (desired output).

5.3.2.8. Suspension criteria and resumption requirements

Unexpected output (undesired output)

5.3.2.9. Test deliverables

Identify the deliverable documents. The following documents should be included:

a. Test plan
b. Test design specifications
c. Test case specifications
d. Test procedure specifications
e. Test item transmittal reports
f. Test logs

g. Test incident reports
h. Test summary reports

5.3.2.10. Testing tasks

- A sample data of stats will be needed for payment calculation.
- Mock data for signup
- Mock data for otp
- Mock data for verification

5.3.2.11. Environmental needs

No special environmental needs as the application can run on normal mobile applications with the internet.

5.3.2.12. Responsibilities

- Hassam Sohail
- Ahmad Raza
- Rimsha

5.3.2.13. Approvals

Name: Mr. Bilal Tariq Butt
Title: Project Supervisor
Date: Signature:_____

5.4. Test Case Specification

5.4.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.4.2. Outline

A test plan shall have the following structure:

5.4.2.1. Test case specification identifier

Specify the unique identifier assigned to this test case specification.

5.4.2.2 Test plan identifier

Test plan identifier: ST-Test-03

Table 5.1 "Test Case 1"

Test Case Title:	Signup with a valid email and password
Test Engineer:	Hassam Sohail
Test Case ID:	ST-ID-01
Test Scenario ID	ST-ID-01
Steps/Actions	• Click on Signup button on SignUp Page • Enter Fullname, email, password, phone number. • Click on Signup button
Expected Result	User record should be created and user should receive verification email
Actual Result	User record created and verification email received
Test Data/Preconditions	A working email and a random password
Status	Passed

Table 5.2 "Test Case 2"

Test Case Title:	Confirmation email after sign up
Test Engineer:	Ahmad
Test Case ID:	ST-ID-02
Test Scenario ID	ST-ID-01
Steps/Actions	• Click on Signup button on SignUp Page • Enter Fullname, email, password, confirm password. • Send Confirm email
Expected Result	Confirm email
Actual Result	Confirm email
Test Data/Preconditions	A working email and an invalid password
Status	Passed

Table 5.3 "Test Case 3"

Test Case Title:	Login with a valid email and password(existing user)
Test Engineer:	Hassam
Test Case ID:	ST-ID-04
Test Scenario ID	ST-ID-01
Steps/Actions	• Click on Login button on Landing Page • Enter email, password • Click on Login button
Expected Result	User redirected to its route based on its role
Actual Result	User redirected to its route based on its role
Test Data/Preconditions	A existing user's email and password
Status	Passed

Table 5.5 "Test Case 5"

Test Case Title:	Buyer Job Creation and posted
Test Engineer:	Ahmad
Test Case ID:	ST-ID-05
Test Scenario ID	ST-ID-01
Steps/Actions	• Choose Buyer and add title , description , and scope and budget of project
Expected Result	Job are created
Actual Result	Job are created
Test Data/Preconditions	User need to fill are input of job creation
Status	Passed

Table 5.6 "Test Case 6"

Test Case Title:	Seller Submit the proposal
Test Engineer:	Ahmad
Test Case ID:	ST-ID-06
Test Scenario ID	ST-ID-01
Steps/Actions	• Job are list on the main screen • Freelancer select the particular job and submit proposal • Click to apply button
Expected Result	Proposal are submitted , bid are uploaded
Actual Result	Proposal are submitted , bid are uploaded
Test Data/Preconditions	User need to fill the proposal input box and fill it
Status	Passed

Table 5.7 "Test Case 7"

Test Case Title:	Chatting
Test Engineer:	Hassam
Test Case ID:	ST-ID-06
Test Scenario ID	ST-ID-01
Steps/Actions	• After Submit proposal • Freelancer are chat with customer for understanding requirements
Expected Result	Chating
Actual Result	Chating
Test Data/Preconditions	User need to send message to customer
Status	Passed

Table 5.8 "Test Case 9"

Test Case Title:	Order management
Test Engineer:	Hassam
Test Case ID:	ST-ID-07
Test Scenario ID	ST-ID-01
Steps/Actions	• After chating , they can start the order • Order wil start with duration
Expected Result	Order will start and complete
Actual Result	Order will start and complete
Test Data/Preconditions	User need to Confirm the order
Status	Passed

Table 5.9 "Test Case 10"

Test Case Title:	Submit Review
Test Engineer:	Ahmad
Test Case ID:	ST-ID-08
Test Scenario ID	ST-ID-01
Steps/Actions	• Submit Review after complete the order • Customer are give the review
Expected Result	After order complete submit Review
Actual Result	After order complete submit Review
Test Data/Preconditions	User need to complete the order and give review
Status	Passed

5.5. Test procedure specification

5.5.1. Purpose
It will specify how the tester will physically run the test, the physical set-up required, and the procedure steps that need to be followed. It contains a sequence of actions required for the execution of the test.

5.5.2 Outline
A test procedure specification shall have the following structure:

5.5.2.1. Test procedure specification identifier
Test plan identifier: ST-Test-03.

5.5.2.2. Purpose
Explanation of the purpose of the test procedure specification Overview of the testing process and objectives.

5.5.2.3. Special requirements
- A sample data for registered users.
- Mock data for adding users
- Mock data for deployment
- Mock data for history

5.6. Test item transmittal report

5.6.1. Purpose
The goal is to identify the test items being transmitted for testing. It includes the person responsible for each item, its deployment requirements, and its status. Any variations from the current item requirements and designs are noted in this report.

5.6.2. Outline
A test item transmittal report will have the following structure:

a. Transmittal report identifier
b. Transmitted items
c. Location
d. Status
e. Approvals

Test plan identifier: ST-Test-04
.

5.6.2.1. Transmittal report identifier
None.

5.6.2.2. Transmitted items
None

5.6.2.3. Location
None

5.6.2.4. Status
None

5.6.2.5. Approvals

Name: Mr. Bilal Tariq Butt
Title: Project Supervisor
Date: Signature:_____

5.7. Test log

5.7.1. Purpose
To provide a chronological record of relevant details about the execution of tests.

5.7.2. Outline
A test log shall have the following structure:

a. Test log identifier;
b. Description;
c. Activity and event entries.

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test log or available to users of the log. Details on the content of each section are contained in the following sub clauses.

5.7.2.1. Test log identifier
Specify the unique identifier assigned to this test log.

5.7.2.2. Description
Information that applies to all entries in the log except as specifically noted in a log entry should be included here. The following information should be considered:

- Identify the items being tested including their version/revision levels. For each of these items, supply a reference to its transmittal report, if it exists.
- Identify the attributes of the environments in which the testing is conducted. Include facility identification, hardware being used (e.g., amount of memory being used, CPU model number, and number and model of tape drives, and/or mass storage devices), system software used, and resources available (e.g., the amount of memory available).

5.7.2.3. Activity and event entries
For each event, including the beginning and end of activities, record the occurrence date

and time along with the identity of the author. The information in 9.2.3.1 through 9.2.3.5 should be considered:

5.7.2.3.1. Execution description

Record the identifier of the test procedure being executed and supply a reference to its specification. Record all personnel present during the execution including testers, operators, and observers. Also indicate the function of each individual.

5.7.2.3.2. Procedure results

For each execution, record the visually observable results (e.g., error messages generated, aborts, and requests for operator action). Also record the location of any output (e.g., reel number). Record the successful or unsuccessful execution of the test.

5.7.2.3.3. Environmental information

Record any environmental conditions specific to this entry (e.g., hardware substitutions).

5.7.2.3.4. Anomalous events

Record what happened before and after an unexpected event occurred (e.g., A summary display was requested and the correct screen displayed, but response seemed unusually long. A repetition produced the same prolonged response). Record circumstances surrounding the inability to begin execution of a test procedure or failure to complete a test procedure (e.g., a power failure or system software problem).

5.7.2.3.5. Incident report identifiers

Record the identifier of each test incident report, whenever one is generated.

5.8. Test incident report

5.8.1. Purpose

To document any event that occurs during the testing process that requires investigation.

5.8.2. Outline

A test incident report shall have the following structure:

a. Test incident report identifier
b. Summary
c. Incident description
d. Impact

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test incident report or available to users of the incident report.

Details on the content of each section are contained in the following sub clauses.

5.8.2.1. Test incident report identifier
Specify the unique identifier assigned to this test incident report.

5.8.2.2. Summary
Summarize the incident. Identify the test items involved indicating their version/revision level. References to
the appropriate test procedure specification, test case specification, and test log should be supplied.

5.8.2.3. Incident description
Provide a description of the incident. This description should include the following items:

a. Inputs
b. Expected results
c. Actual results
d. Anomalies
e. Date and time;
f. Procedure step;
g. Environment;
h. Attempts to repeat;
i. Testers;
j. Observers.

Related activities and observations that may help to isolate and correct the cause of the incident should be included (e.g., describe any test case executions that might have a bearing on this particular incident and any variations from the published test procedure).

5.8.2.4.Impact
If known, indicate what impact this incident will have on test plans, test design specifications, test procedure specifications, or test case specifications.

5.9. Test summary report

5.9.1. Purpose
To summarize the results of the designated testing activities and to provide evaluations based on these results.

5.9.2. Outline
A test summary report shall have the following structure:

a. Test summary report identifier
b. Summary
c. Variances
d. Comprehensive assessment
e. Summary of results
f. Evaluation
g. Summary of activities
h. Approvals

5.9.2.1. Test summary report identifier
ST-Test-04

5.9.2.2. Summary
The following features were tested and their reference:

- Signup with a valid email and password ST-Test-1
- Signup with a valid email and invalid password(short) ST-Test-2
- Signup with an invalid email and invalid password(short) ST-Test-3
- Login with a valid email and password(existing user) ST-Test-4
- Buyer Job Creation and posted ST-Test-5
- Seller Submit the proposal ST-Test-6
- Chatting ST-Test-7
- Order management ST-Test-8
- Submit Review ST-Test-9

5.9.2.3. Variances
None.

5.9.2.4. Comprehensiveness assessment
We tested almost all necessary features. Only the repeated logics were not tested to save time and effort.

5.9.2.5. Summary of results
During the execution of the test, we did not face any anomalies. Our app performed the required functionality correctly

5.9.2.6. Evaluation
Overall the application performed as expected. However the similar components/features were not tested and are assumed to be optimal based on the assessment of these features.

5.9.2.7. Summary of activities

- Hassam Sohail
- Ahmad Raza
- Rimsha

5.9.2.8. Approvals

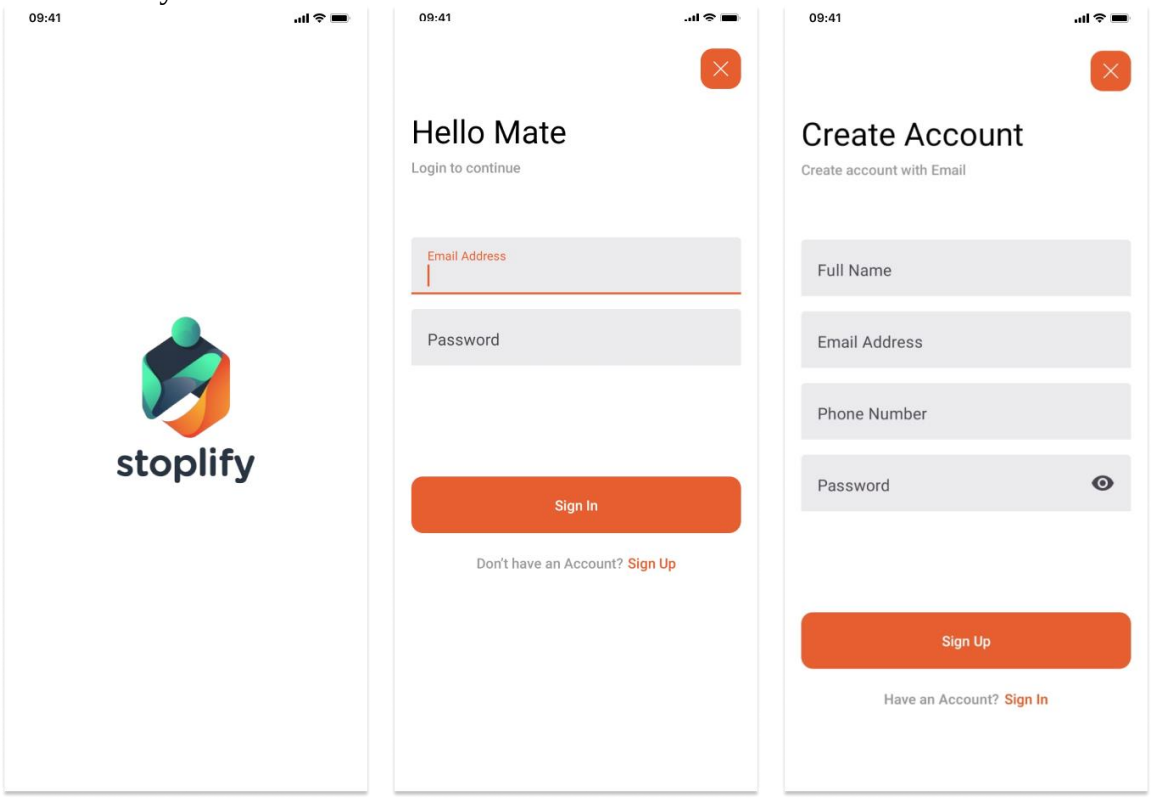
Name: Mr. Bilal Tariq Butt
Title: Project Supervisor
Date: Signature: _____

5.10. User Manual

The manual of this application is explained below.

Step 1

After clicking on app the logo will appear for seconds after that is will disappear automatically.



Step 2

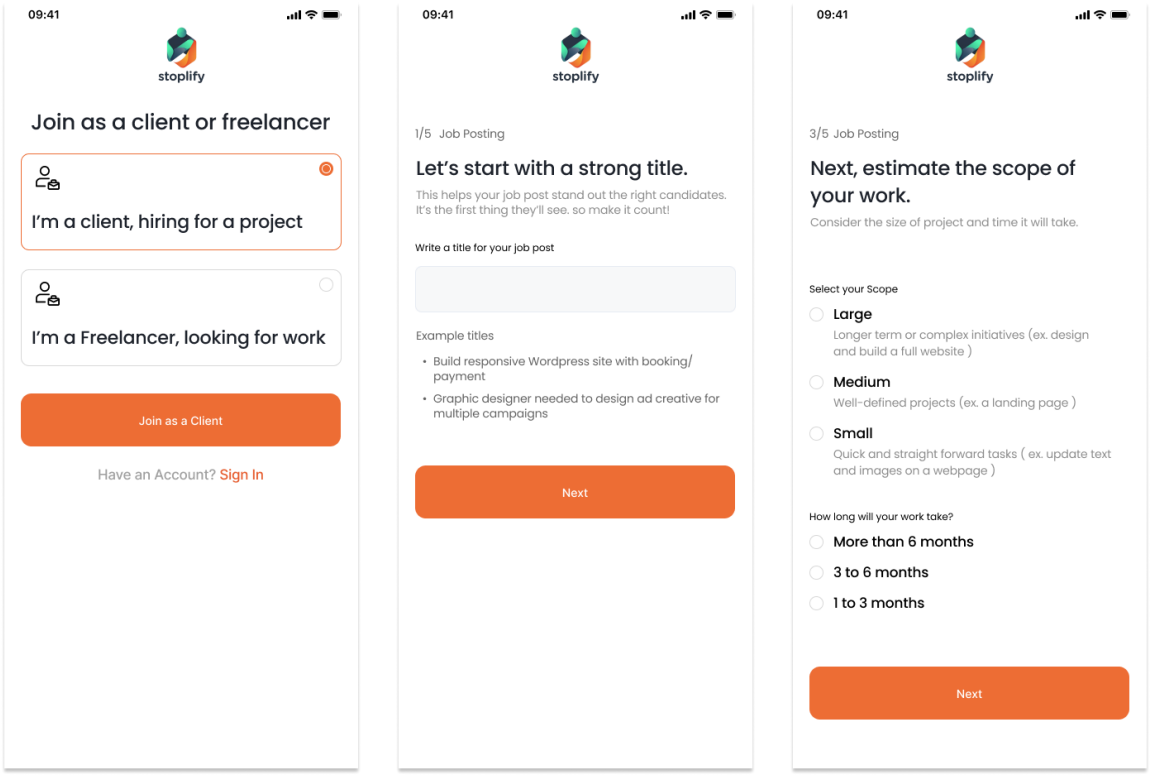
In order to use the application you will have to have an account you can setup new account.

On username field, you will have to enter the name, which will be seen by other customers.

On Email field, you will have to enter you email. On first password field, you will enter your secret password. On second password field, you will enter the same password to confirm. If you have already made an account then just press log in button. By clicking sign up button to create account.

Step 3

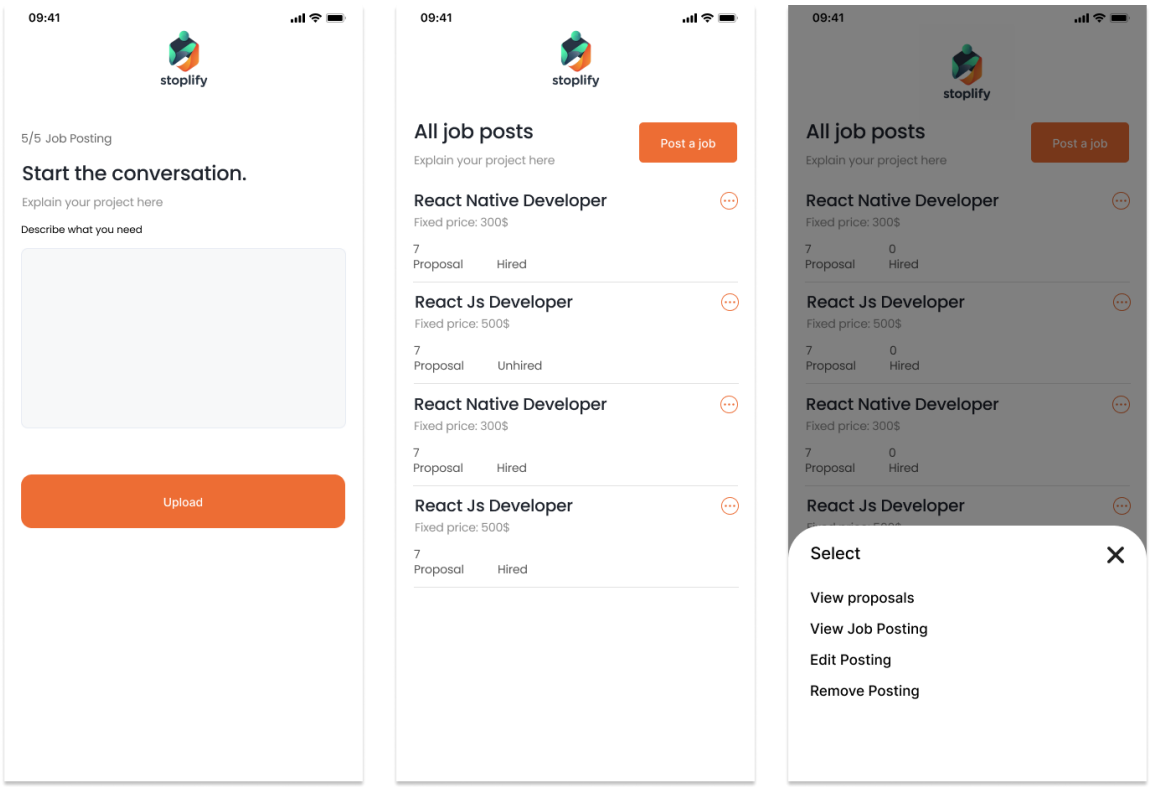
Now login by using your email and password.



Step 4

After you need to choose the one option from both,

1. Join as a Client
2. Join as a Freelancer



Step 5

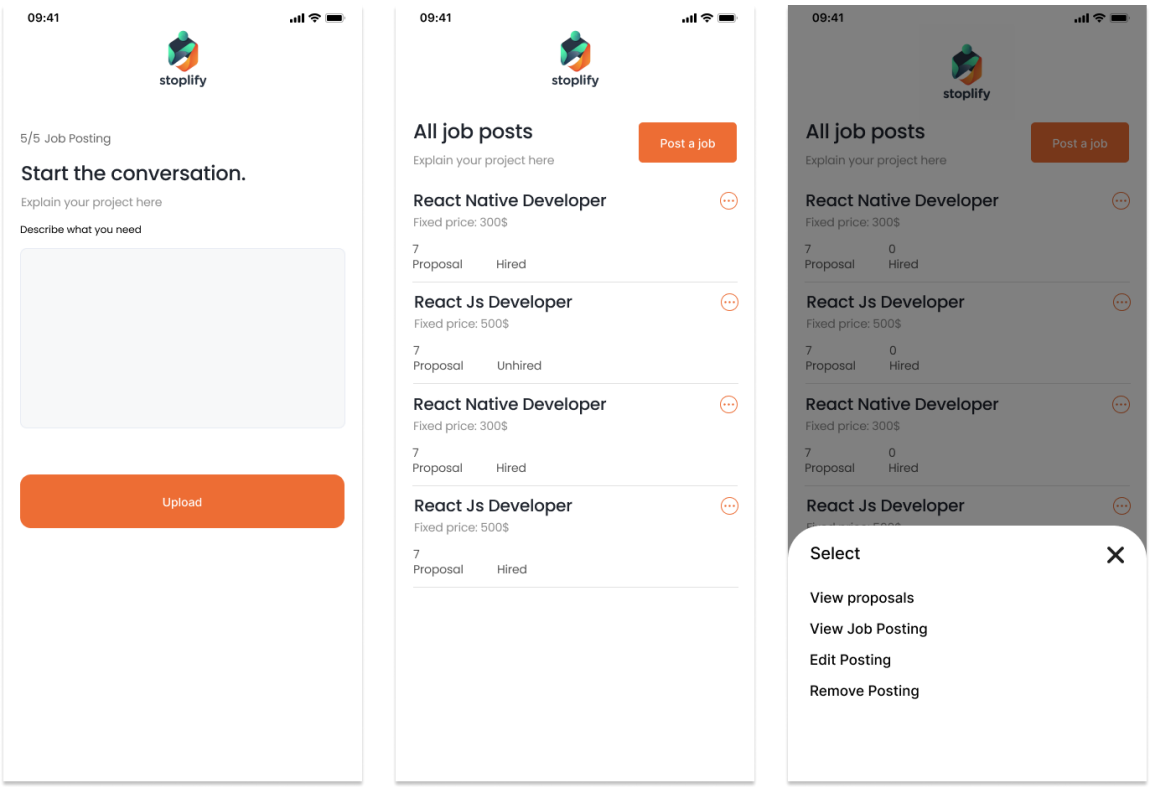
If User Can Select the step one , Join as a Client then follow this steps
Step 1: To create a job First Step to add a title

Step 2: To Select a skill and add the skill

Step 3: To Select the project Scope and Select the time line

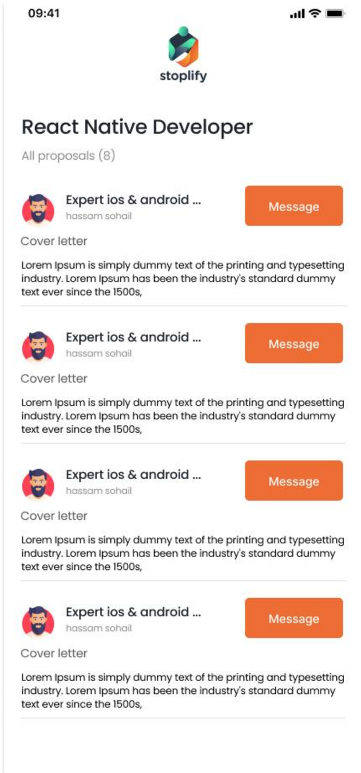
Step 4: To select the Project budget (Fixed or Hourly Rate)

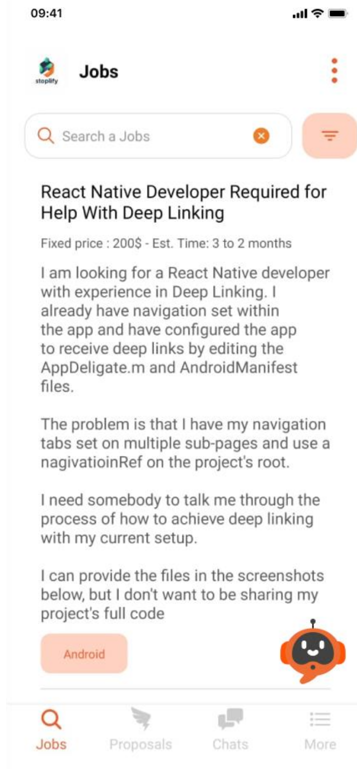
Step 5: To Add a Description of the project

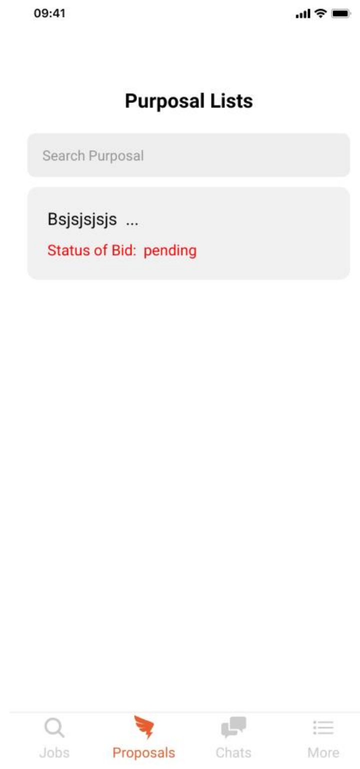


Step 6

After Upload the project the buyer see the list of Project on this screen.



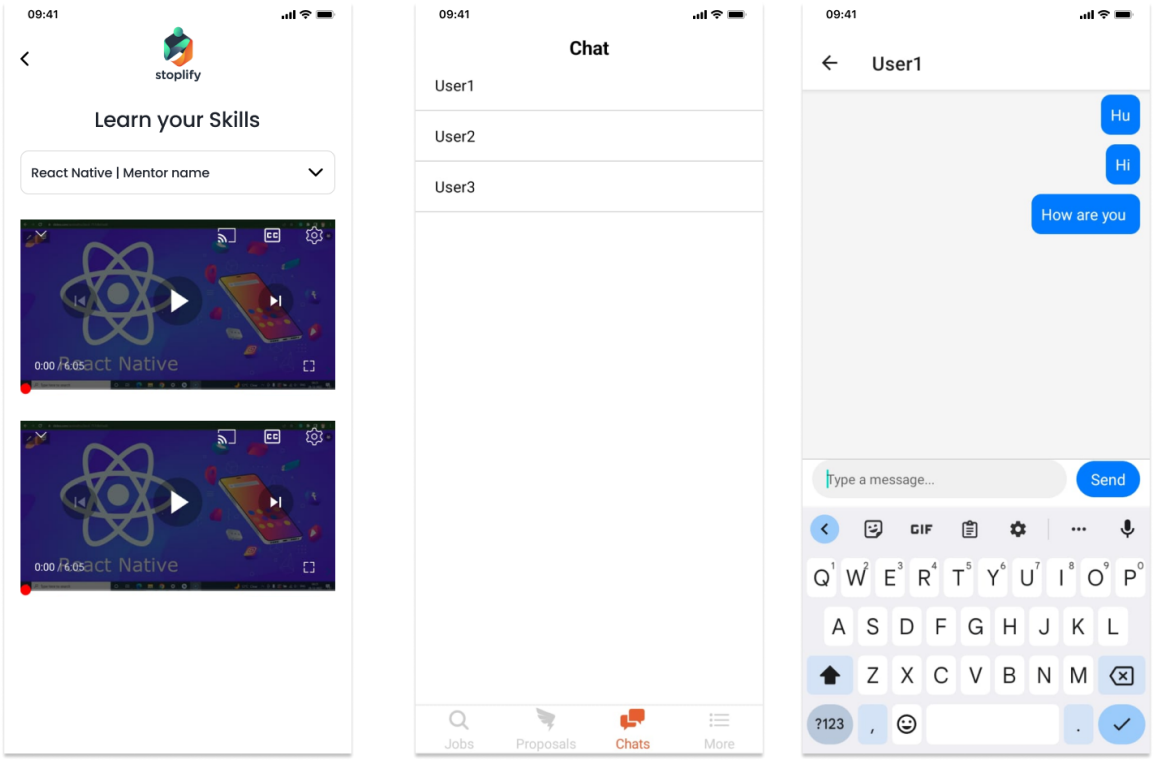




Step 7

After the click on the menu btn see the list of option shown

1. View Proposal
2. Edit Job
3. Delete Job
4. View Job posting



Step 8

See all Learning material and chat with client and freelancer