



SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE AND TECHNOLOGY

Final Year Project Report

PakWork

Project Team:

Asaad Noman Abbasi (1912297)

Muhammad Ahsan (1912310)

Project Supervisor:

Muhammad Shahzad Haroon

Date: 9th November 2022

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science

in the

Faculty of Computing and Engineering Sciences

Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology

(SZABIST)

Karachi Campus

Declaration of Authorship

We, Asaad Noman (1912297) and Muhammad Ahsan (1912310), declare that this report titled, “PakWork” and the work presented in it are our own. We confirm that: This work was done wholly or mainly while in candidature for a bachelor’s degree at this University. Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated. Where we have consulted the published work of others, this is always clearly attributed. Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this report is entirely our own work. We have acknowledged all main sources of help. Where the report is based on work done by us jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed: Asaad Noman (1912297), Muhammad Ahsan (1912310)

Date: 9th November, 2022

Acknowledgment

In the name of ALLAH, the most beneficent and merciful who gave us the knowledge and courage to work on this research area. The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. We would first like to thank our supervisor “Sir Shahzad Haroon” of the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. The door to his office was always open whenever we ran into a trouble spot or had a question about our project or documentation. He consistently helped, cooperated and motivated us throughout the research. We would like to thank to our teachers who guided us in the light of their knowledge and experience. We would also like to express our gratitude to our loving parents and family members who helped and gave us encouragement. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of SZABIST, who gave the permission to use all required equipment and the necessary materials to complete the project. At the end, we would like to thank Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for providing us such an inspiring environment. The quality education, the cooperative faculty members and the challenging environment have always motivated and boosted the confidence level of each and every student who has been a part of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. And many thanks to all those who let us share their knowledge and resources.

Table of Contents

Table of Contents	4
Revision History	6
Iteration Plan.....	7
FYP Project Proposal	10
1. Introduction.....	11
2. Objective	11
3. Problem Description	11
4. Methodology	11
5. Project Scope	12
6. Feasibility Study.....	12
7. Solution Application Areas	12
8. Tools/Technology	12
9. Expertise of the Team Members.....	13
10. Milestones	13
11. Project Schedule.....	14
12. Work Breakdown Structure	14
13. References	19
Software Requirements Specification	20
1. Introduction.....	21
1.1 Purpose	21
1.2 Document Conventions	21
1.3 Intended Audience and Reading Suggestions.....	21
1.4 Product Scope	21
1.5 References	21
2. Overall Description	21
2.1 Product Perspective	21
2.2 Product Functions	22
2.3 User Classes and Characteristics	22
2.4 Operating Environment	23
2.5 Design and Implementation Constraints.....	23
2.6 User Documentation	23
2.7 Assumptions and Dependencies	23
3. External Interface Requirements	23
3.1 User Interfaces.....	23
3.2 Hardware Interfaces.....	27
3.3 Software Interfaces	27
3.4 Communications Interfaces	27
4. System Features	27
4.1 Freelancer Registration / Verification System.....	28
4.2 Gig / Job Creating & Posting.....	30
4.3 Profile Dashboard.....	32
4.4 Search System	34
4.5 Buyer Requests.....	36
4.6 Order Management.....	37
4.7 Custom Offers	39
4.8 Bidding System	40
4.9 Push Notification System	41
4.10 Chat System.....	43
4.11 Rating/Feedback System	44

4.12 Personalized Testing System.....	45
5. Other Nonfunctional Requirements	47
5.1 Performance Requirements.....	47
5.2 Safety Requirements.....	47
5.3 Security Requirements.....	47
5.4 Software Quality Attributes.....	47
5.5 Business Rules.....	47
6. Other Requirements	47
Appendix A: Glossary.....	48
Appendix B: Analysis Models	48
ERD:.....	48
System Use Case Diagram:	49
Domain Model:.....	50
Appendix C: To Be Determined List.....	51
Testing Documents.....	52
Diagrams.....	67
1. System Sequence Diagrams.....	68
1.1. Freelancer Registration / Verification	68
1.2. Gigs / Job Posting	68
1.3. Profile Dashboard	69
1.4. Search System	69
1.5. Buyer Requests	70
1.6. Order Management	70
1.7. Custom Offers	71
1.8. Bidding System	71
1.9. Push Notification System.....	72
1.10. Chat System	72
1.11. Rating/Feedback System.....	73
1.12. Personalized Testing System.....	73
Software Design Specification.....	74
1. Introduction.....	75
1.1 Purpose of this document	75
1.2 Scope of the development project	75
1.3 Definitions, acronyms and abbreviation	75
1.4 References	75
1.5 Overview of document	76
2. System architecture description.....	76
2.1 Section overview	76
2.2 General Constraints	76
2.3 Data Design	78
2.4 Program Structure.....	78
2.5 Alternatives Considered	79
3. Detailed description of components	79
3.1 Section overview	79
3.2 Component of Gigs & Job Feature	79
3.3 Component of Order Management and Rating/Feedback	80
4. User Interface Design	81
4.1 Section overview	82
4.2 Interface Design Rules.....	82
4.3 GUI Components.....	82
4.4 Detailed Description	82
5. Reuse and relationships to other products	82
6. Design decision and tradeoffs	82
7. Pseudo for components.....	82

8. Appendices.....	83
8.1 Class Diagram	83
8.2 Object Diagram	84
8.3 Statechart Diagram	85
8.4 Activity Diagram	86
8.5 Sequence Diagram.....	93
8.6 Collaboration Diagram.....	93
8.7 Use Case Diagrams.....	93
8.8 Component Diagram.....	93

Revision History

Name	Date	Reason For Changes	Version
Asaad	10 th Jan 2023	Revision for final	1.1

Iteration Plan

PakWork Iteration Plan

S.N o.	Features	FYP-I Iterations			
		Monthly Iteration-I	Monthly Iteration-II	Monthly Iteration-III	Monthly Iteration-IV
F1	Freelancer Registration & Verification System	Requirements			
		Design			
			Implementation		
			Testing		
F2	Client Registration System	Requirements			
		Design			
			Implementation		
			Testing		
F3	Profile Dashboard	Requirements			
		Design			
			Implementation	Implementation	
				Testing	
F4	Jobs / Gigs Posting	Requirements			
				Design	
				Implementation	
					Testing
F5	Order Management	Requirements			
				Design	
				Implementation	Testing
F6	Buyer Requests	Requirements			
				Design	
				Implementation	Testing
F7	Search System	Requirements			
				Design	
				Implementation	Testing
S.N o.	Features	FYP-II Iterations			
		Monthly Iteration-I	Monthly Iteration-II	Monthly Iteration-III	Monthly Iteration-IV
F8	Bidding System	Requirements			

		Design			
		Implementation	Implementation		
			Testing		
F9	Push Notifications	Requirements			
		Design			
			Implementation		
			Testing		
F10	Custom Offers	Requirements			
		Design			
			Implementation		
			Testing		
F10	Chat System	Requirements			
			Design	Design	
				Implementation	Implementation
					Testing
F11	Rating & Feedback System	Requirements			
				Design	
				Implementation	
					Testing
F11	Personalized Testing Systems	Requirements			
				Design	
				Implementation	
					Testing

FYP Project Proposal

1. Introduction

A Freelancing platform for freelancers. Where freelancers can easily sell their gigs and offer services to potential clients/businesses. We aim to provide a fully featured platform that provides everything necessary for a seamless process of a transaction between two entities.

2. Objective

“To design a platform for freelancers to provide their services to clients”.

3. Problem Description

Nowadays, Working remotely seems to be on a much larger scale as many companies whether it be local or offshore are shifting to either hybrid or completely remote-based office environments. This also relates to freelancers, who work remotely with these companies and have tended to rise their numbers. In 2022 it was estimated that 70.4 million freelancers were working around the globe remotely. If we talk about Pakistan, it saw a rise of 48% in the year 2021-2022, ranking it as the 4th most popular country in the freelancing industry globally. There are quite a few benefits of freelancing, such as international remittance coming into local banks via platforms, and the creation of micro-agencies that grow into proper software houses which provide employment opportunities and thus upbringing society. For buyers, it also provides quick and efficient solutions to their problems.

Pakistani freelancers face quite numerous challenges when it comes to freelancing on international platforms, the most common challenge they face is such platforms that provide these services charge quite a handful of service charges for this. Another issue that is often raised by experienced professional freelancers is that clients come with projects that were either unfinished or ruined by unskilled freelancers. Also, the problem of unverified users is on the rise, as on platforms such as Fiverr a single user can create multiple accounts and work, no such identity checks are put into place. Which can lead to fake personas conducting business on the platform. Another issue faced by freelancers is the order policy. Due to some policies regarding order acceptance freelancers face unjustified disputes.

4. Methodology

Now to address the ‘How’ part of the proposal. To tackle the challenges mentioned above, we aim to create a platform that targets freelancers with much easier policies and regional service charges. On the other hand, to tackle the issue of unprofessional and unskilled freelancers, we aim to provide a customized filtering process to ensure that only professional and skilled freelancers can be allowed to work on the platform. We also provide a customized test related to the job that is being bided by freelancers so that only matched freelancers can bid on outsourced projects by businesses, as this way we can tackle much of the filtering part we can offer to the clients. To tackle the issue of the order policy we will make sure that the freelancer has the choice to accept or decline customer job offers made by the client.

5. Project Scope

In the previous para, we talked about all the aspects of the scope of the project we are going to develop and work on. But the scope for this project is much more and some development/design aspects will not be implemented in this due course. Such as

- We are not going to implement a payment holding system. Which is a common feature used by these platforms to cater order claims/disputes.

6. Feasibility Study

Risks Involved:

- Unverified Freelancer using the platform

To tackle this problem, we will implement a system for the verification of CNIC / other documents for the freelancer

Resource Requirements: Cloud databases, Cloud service providers, Hosting servers

7. Solution Application Areas

In today's world, freelancing and remote work is on the rise. Pakistan is one of the top 4 service providers in the industry. Our platform aims to cater to the market of freelancers to connect them to potential international/local clients. International platforms charge hefty service charges. The money they make on the platform they withdraw to their local accounts. They are left with 75% of the money that they potentially make. Our platform aims to provide all the features that a competitive platform offers but with policies, service charges, and features to cater to the regional market.

8. Tools/Technology

Software Tools

- Visual Studio Code
- GitHub Desktop
- NPM
- MongoDB Compass
- Postman / Thunderclient
- Photoshop
- AdobeXD

Hardware Tools

- Laptop / PC
- Internet / WiFi Connection

Technologies

- JavaScript
- jQuery
- Node JS
- Express JS
- React JS
- MongoDB
- Firebase
- MySQL
- Socket.IO

9. Expertise of the Team Members

Asaad Noman Abbasi (1912297)

- **Courses Taken:** Fundamentals of Programming (4 GPA), Object Oriented Programming (4 GPA), Data Structures & Algorithms (3.25 GPA), Database Systems (4 GPA), Software Engineering (3.5 GPA), Web Technologies I (4 GPA), Design & Analysis of Algorithms (3.5 GPA)

Muhammad Ahsan (1912310)

- **Courses Taken:** Fundamentals of Programming (4 GPA), Object Oriented Programming (4 GPA), Data Structures & Algorithms (3.25 GPA), Database Systems (3.5 GPA), Software Engineering (3.25 GPA), Web Technologies I (4 GPA), Design & Analysis of Algorithms (3.25 GPA)

This project is of equal interest to the members as they both have experience in programming a web application and have done internships and taken relevant courses in university prior, in the domain of web development.

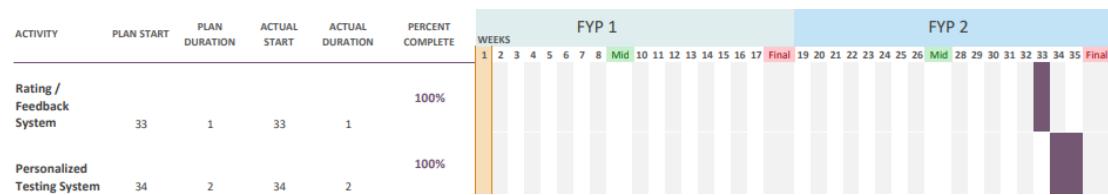
10. Milestones

1. Freelancer Registration / Verification System
2. Project Creation / Posting
3. Order Management
4. Buyer Requests
5. Search System
6. Profile Dashboard
7. Bidding System

8. Push Notification System
9. Custom Offers
10. Chat System
11. Rating/Feedback Systems
12. Personalized Testing System

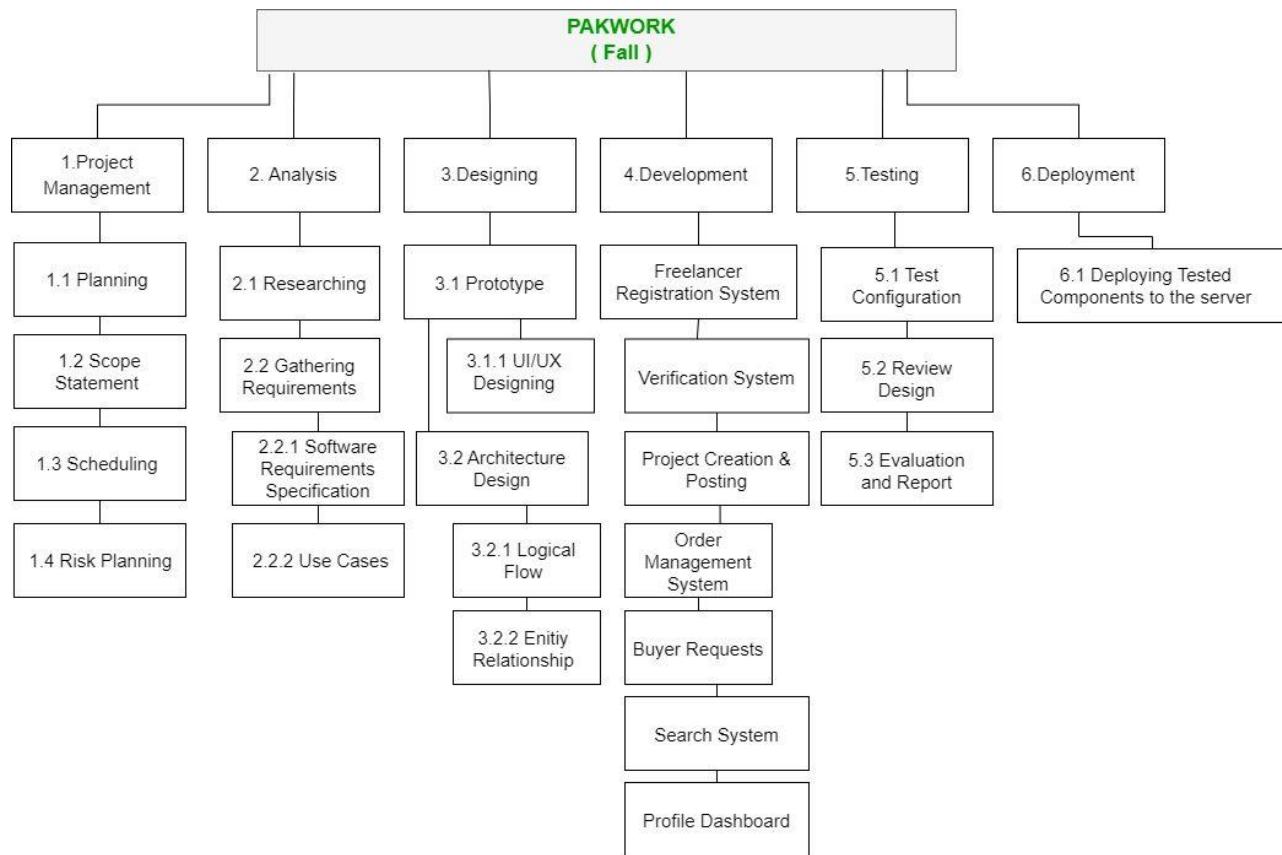
11. Project Schedule

PAKWORK (1 Year Plan)

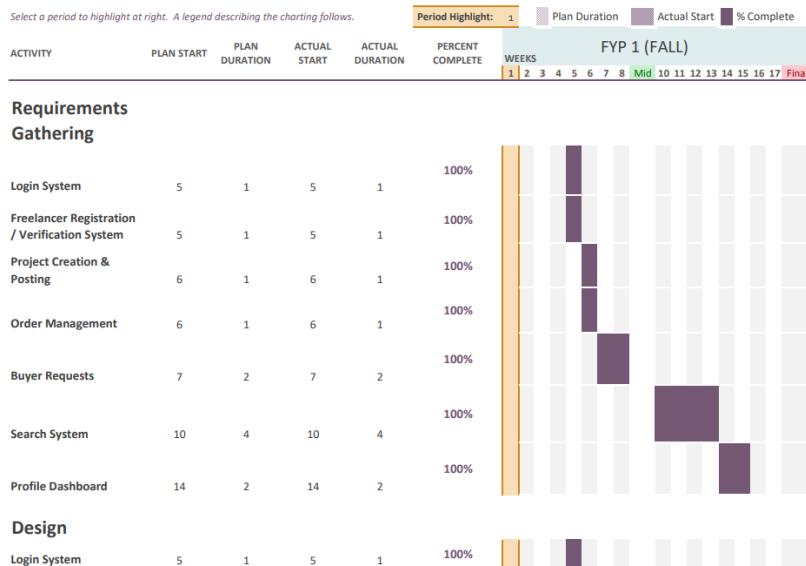


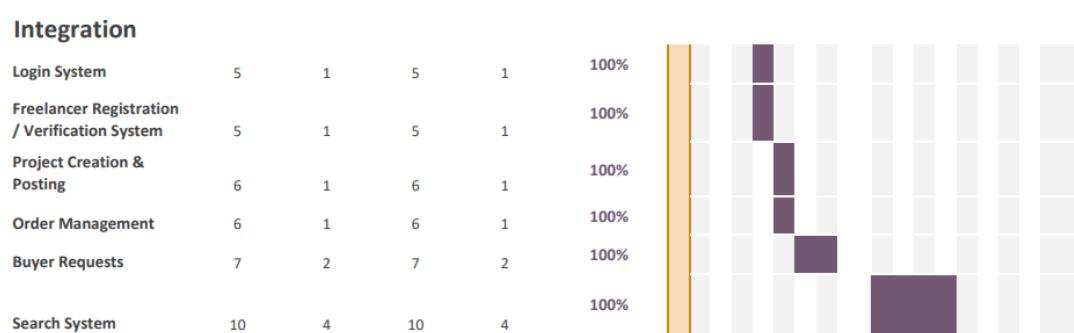
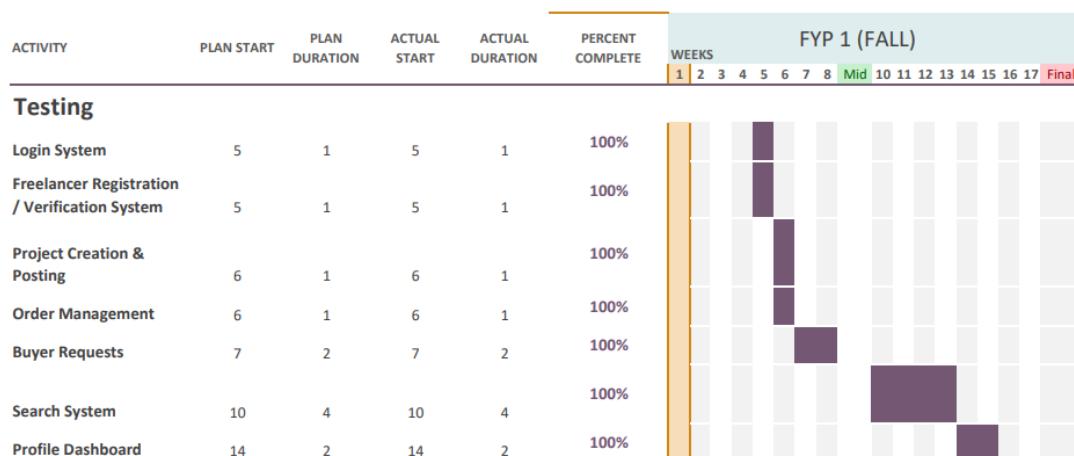
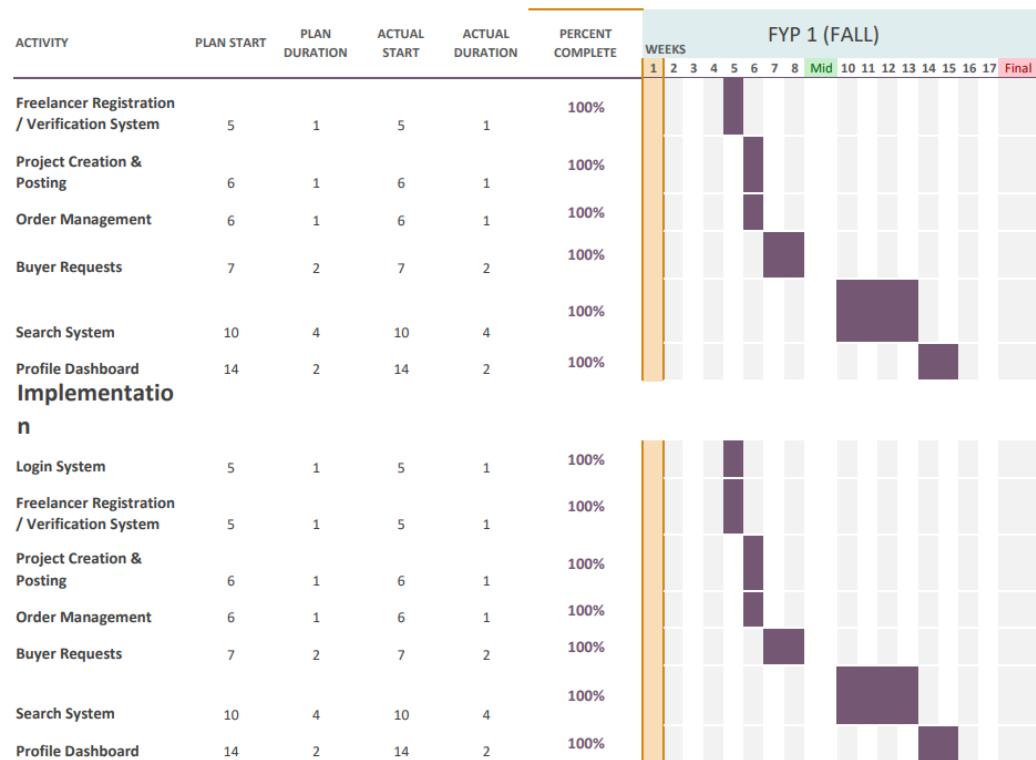
12. Work Breakdown Structure

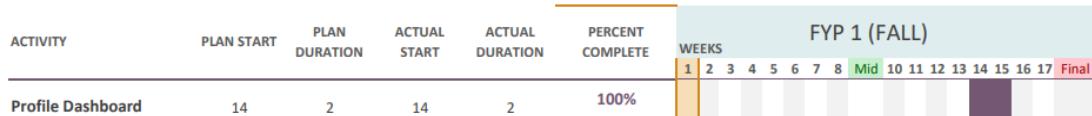
WBS & Gantt Chart (FYP 1) (Fall)



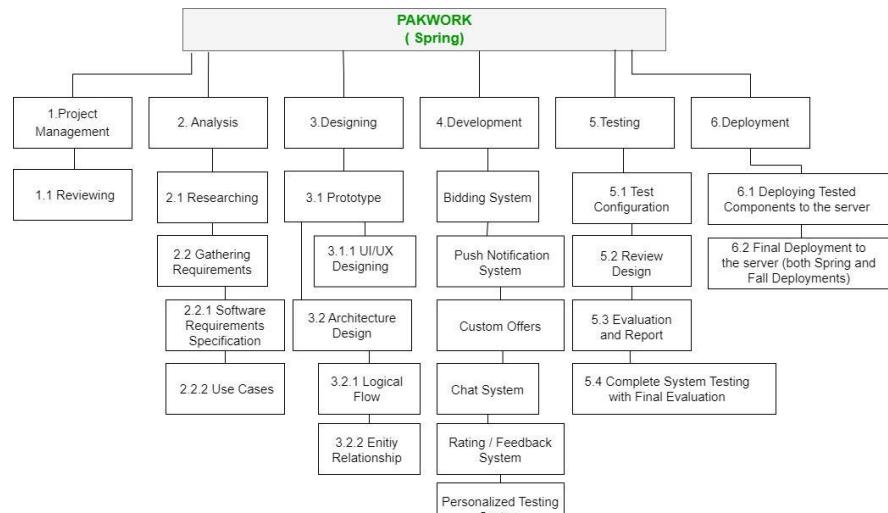
PAKWORK (FYP 1) (Fall) Timeline







WBS & Gantt Chart (FYP 2) (Spring)

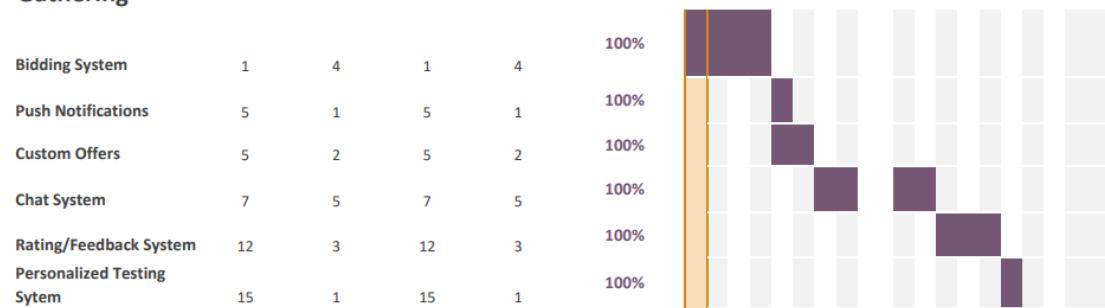


PAKWORK (FYP 2) (Spring) Timeline

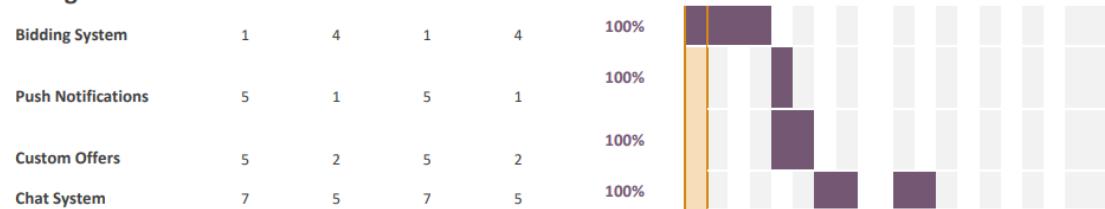
Select a period to highlight at right. A legend describing the charting follows.

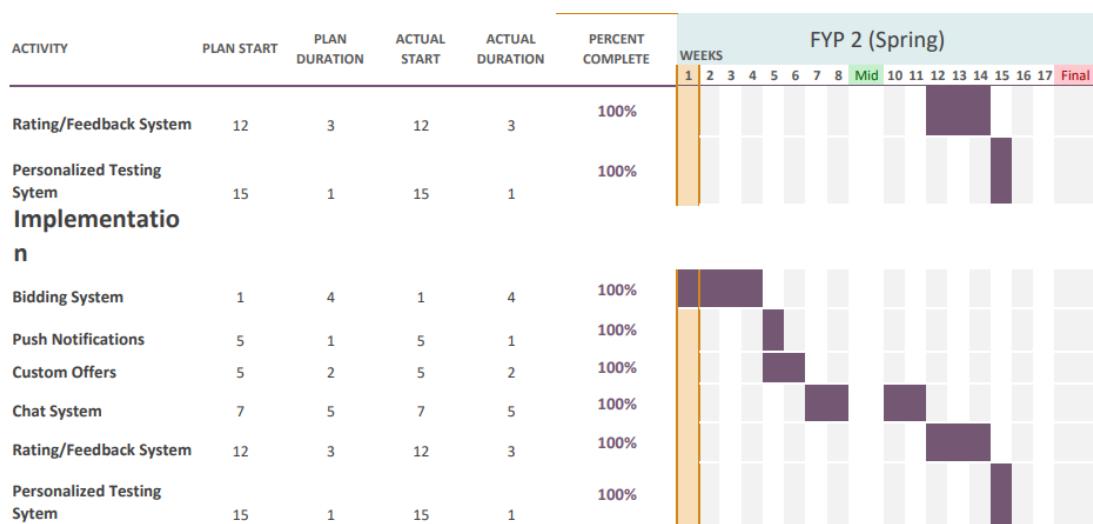
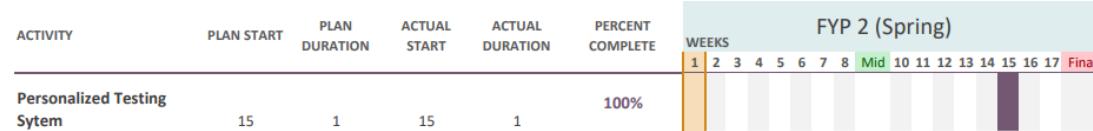
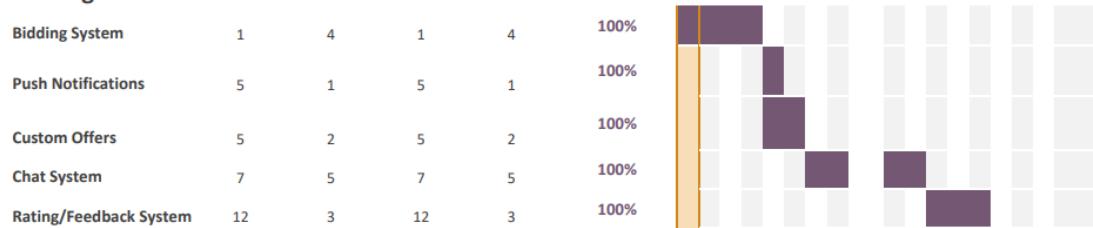
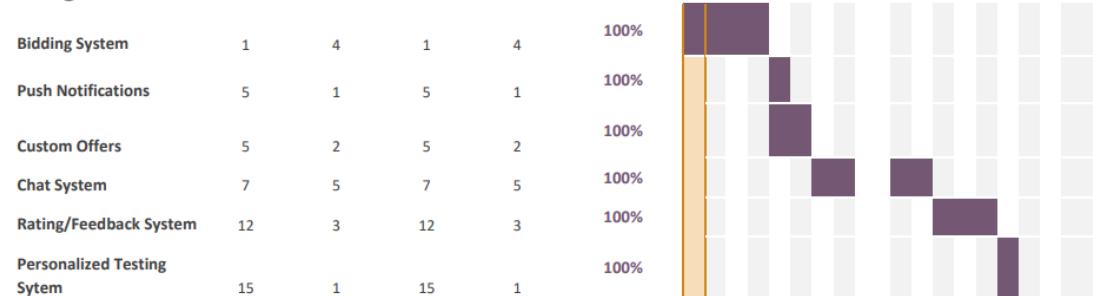


Requirements Gathering



Design



**Testing****Integration**

13. References

- Daniel Ruby “20+ Freelance Statistics Including All Facts & Data (2022)” DemandSage, August 13th 2022, <https://www.demandsage.com/freelance-statistics/>
- TechAbu “Top 10 Countries with The Most Freelancers 2022” TechAbu, June 12th 2022, <https://techabu.co/freelancing/top-10-countries-with-most-freelancers>

Software Requirements Specification

1. Introduction

1.1 Purpose

A Freelancing platform for freelancers. Where freelancers can easily sell their gigs and offer services to potential clients/businesses. We aim to provide a fully featured platform that provides everything necessary for a seamless process of a transaction between two entities.

1.2 Document Conventions

The format of this SRS is simple. Indentation is used on general topics and or specific points of interest with 'Times font' and are in a larger font size compared to the regular text. The headings have been Bold and the text size used is 14 and for the regular text the text size used is 12.

1.3 Intended Audience and Reading Suggestions

This document is to be read by the intended for project manager and teachers. This document is particularly intended for users of the website for understanding the product.

1.4 Product Scope

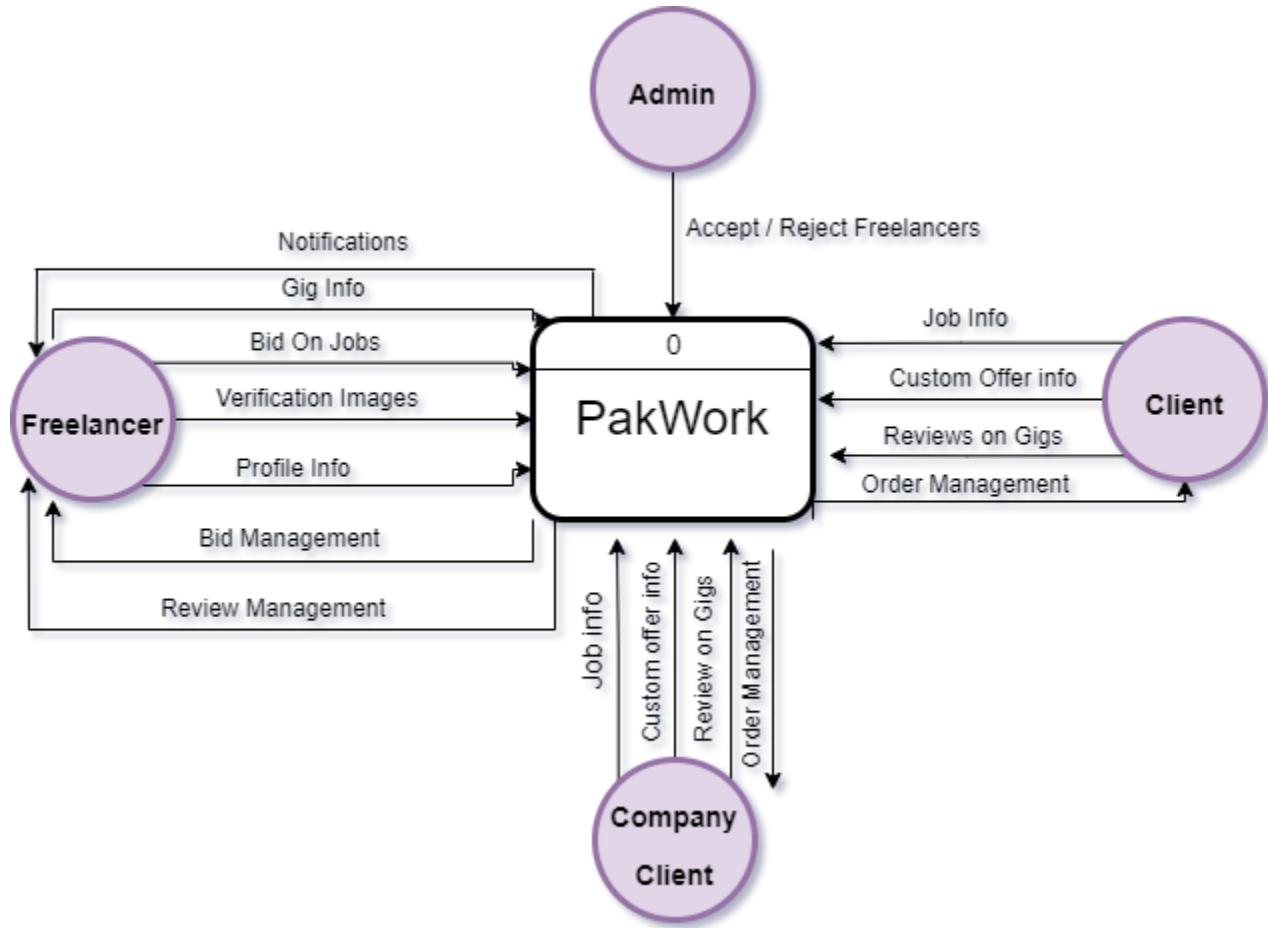
We aim to create a platform that targets freelancers with much easier policies and regional service charges. On the other hand, to tackle the issue of unprofessional and unskilled freelancers, we aim to provide a customized filtering process to ensure that only professional and skilled freelancers can be allowed to work on the platform. We also provide a customized test related to the job that is being bided by freelancers so that only matched freelancers can bid on outsourced projects by businesses, as this way we can tackle much of the filtering part we can offer to the clients. To tackle the issue of the order policy we will make sure that the freelancer has the choice to accept or decline customer job offers made by the client.

1.5 References

- IEEE Template for System Requirement Specification Documents: <https://goo.gl/nsUFwy>

2. Overall Description

2.1 Product Perspective



2.2 Product Functions

- Freelancer Registration & Verification System
- Gig / Job Creation & Posting
- Order Management
- Buyer Requests
- Search System
- Profile Dashboard
- Bidding System
- Push Notification System
- Custom Offers
- Chat System
- Rating/Feedback System
- Personalized Testing System

2.3 User Classes and Characteristics

There will be technically 4 types of users in the system

- **Freelancer:** The person that will be providing the services to the clients and the companies
- **Administrator:** The person responsible for verifying the freelancer's identity

- **Client:** The person that will publish jobs and custom offers to freelancers to do tasks.
- **Company Client:** Company that will publish jobs and custom offers to freelancers to do tasks.

2.4 Operating Environment

- **Software Preferences:** The website requires internet access to the internet and browsers such as Chrome, Firefox, MS Edge, Safari
- **Hardware Preferences:** Website can be accessed using PC/Laptop or Mobile/Tablet

2.5 Design and Implementation Constraints

Some system design limitation constraints come from a payment holding system. Which is a common feature which is used by these platforms to cater claims/disputes. Since we are not implementing a payment gateway system or wallet system, hence we are also not implementing this feature.

2.6 User Documentation

The website is self-explanatory for the user, that's why no user manual is provided.

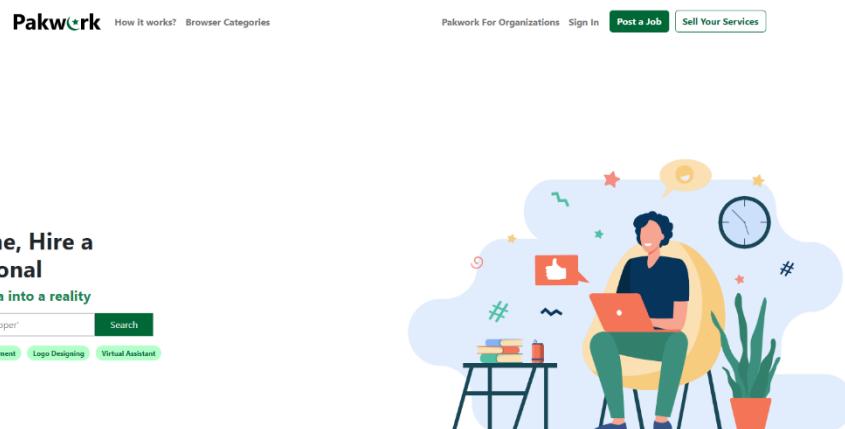
2.7 Assumptions and Dependencies

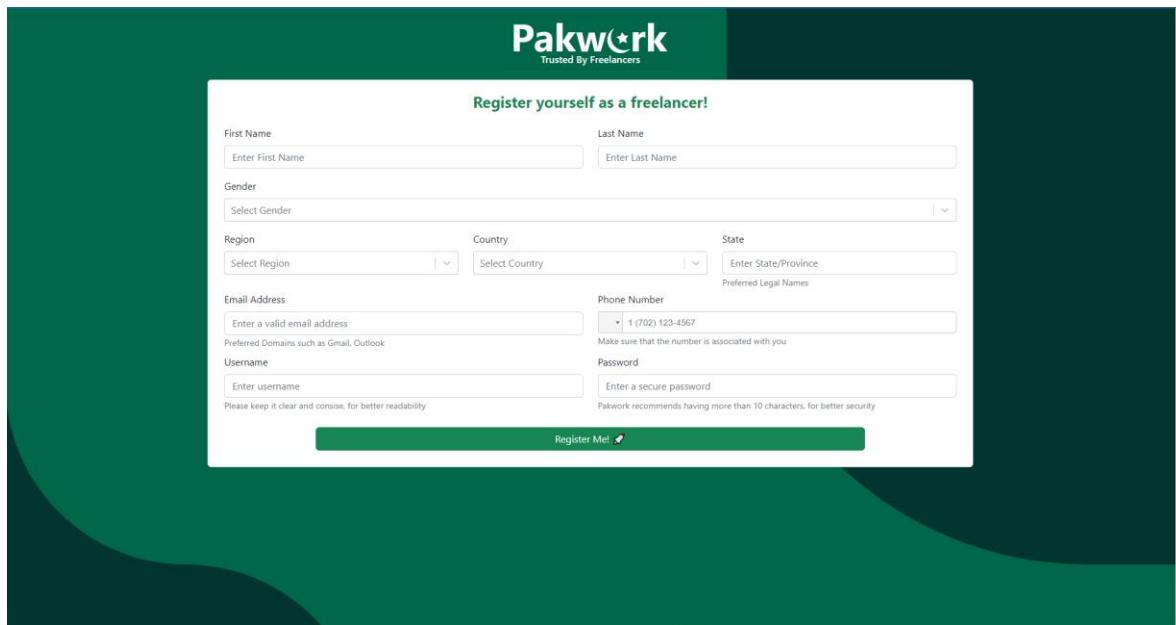
- User must have an active internet connection to access the website
- User must enter valid details into the website
- Proper identification documents are provided to be verified to use the platform

3. External Interface Requirements

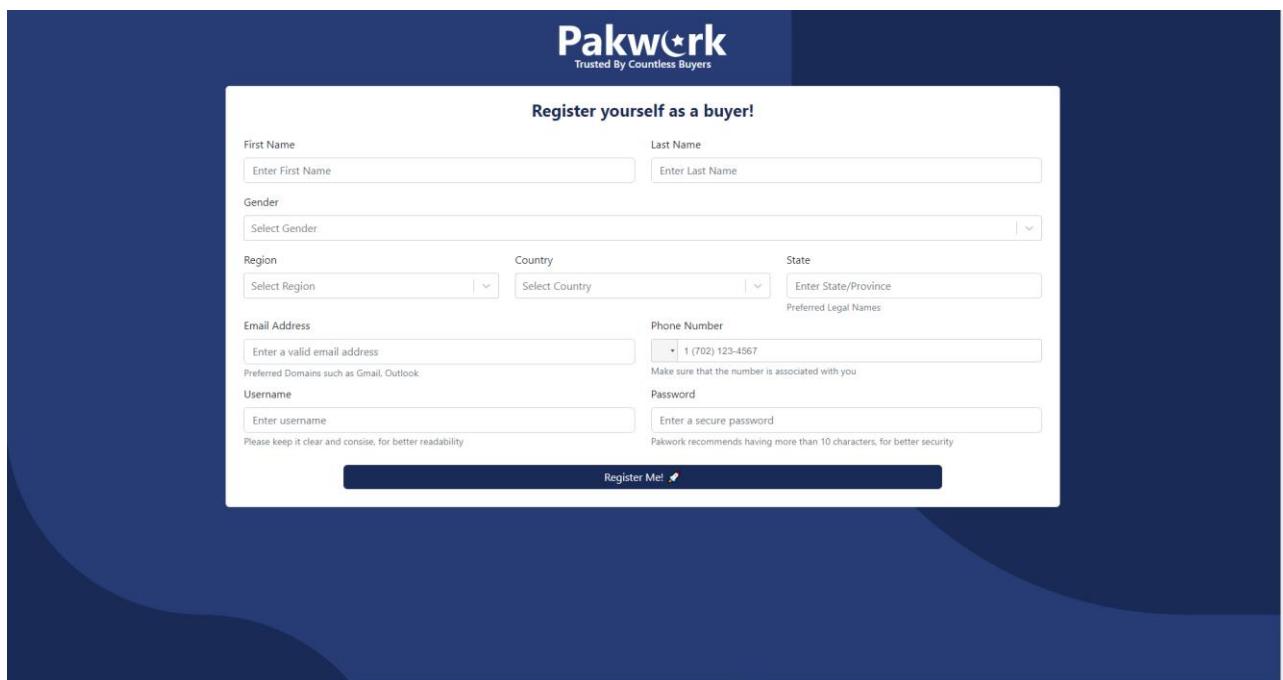
3.1 User Interfaces

Main Page

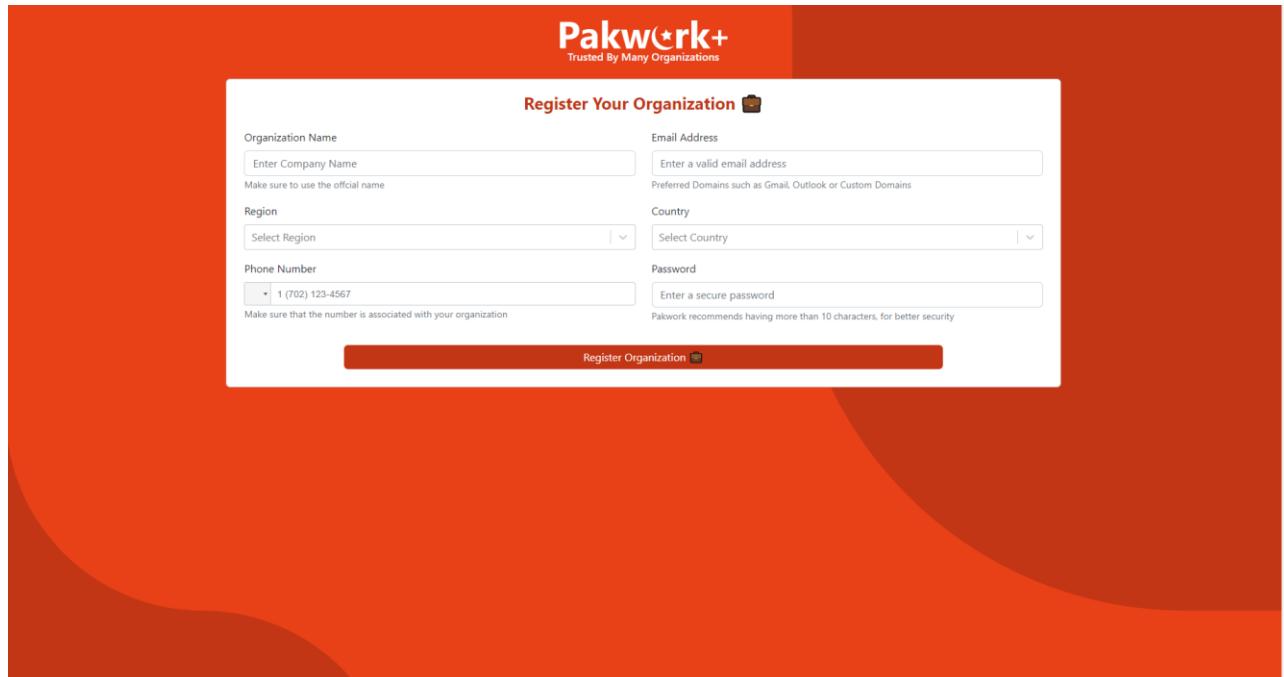




Client Registration



Company Client Registration

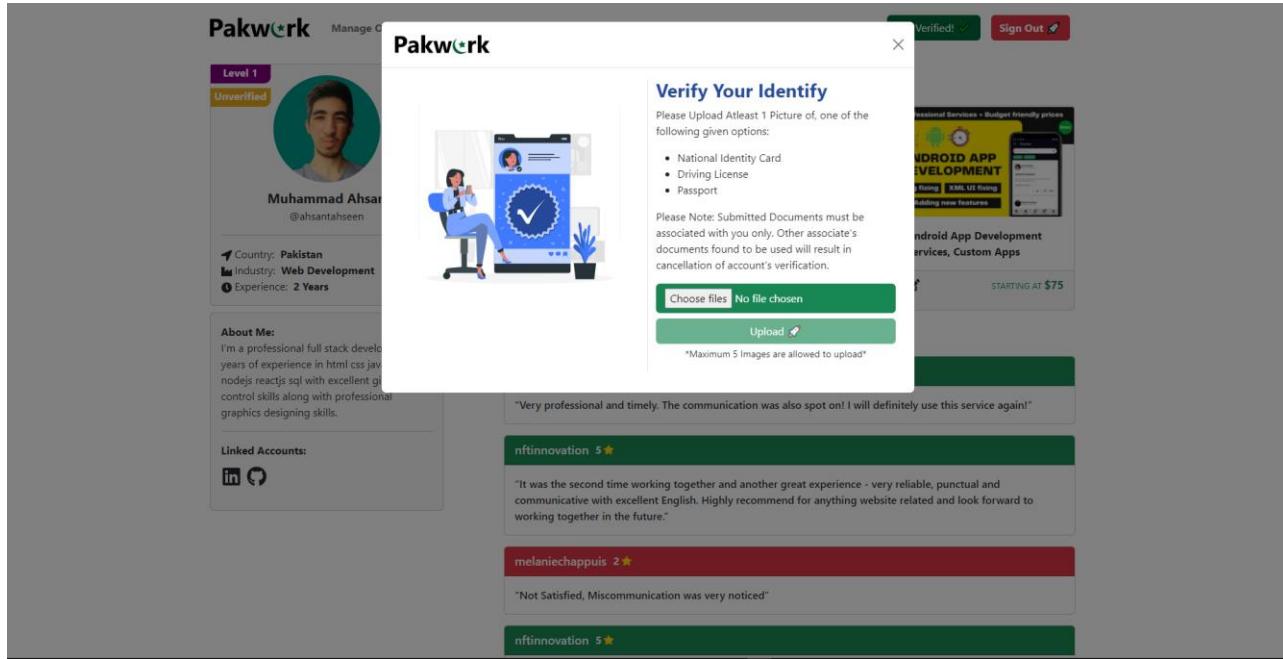


Profile Dashboard

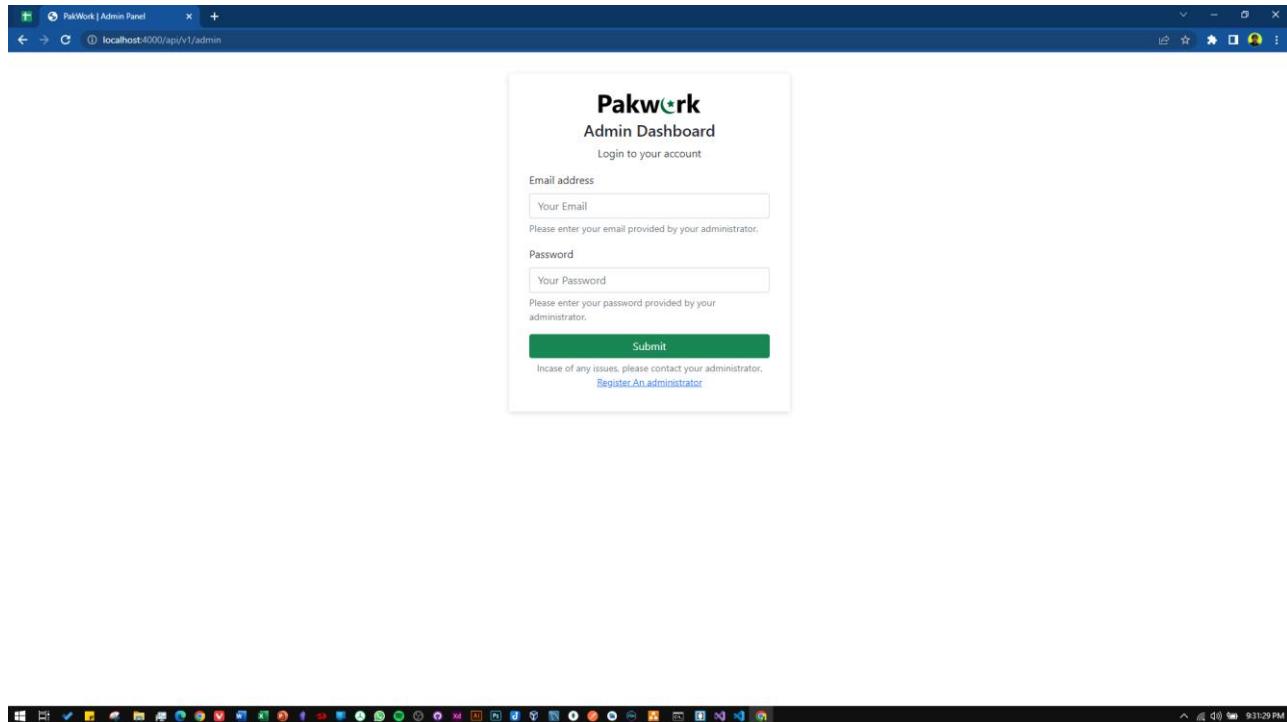
The dashboard for Muhammad Ahsan shows the following sections:

- Profile:** Level 1, Unverified. Includes a circular profile picture of Muhammad Ahsan, his name, and social links.
- About Me:** Describes him as a professional full stack developer with 2 years of experience in html css javascript nodejs reactjs sql with excellent git version control skills along with professional graphics designing skills.
- Linked Accounts:** LinkedIn and GitHub icons.
- Gig Options:** 'Create New Gig' button. Three service cards:
 - Custom Web Development Services:** Starting at \$15.
 - Convert Ms Excel, Google Sheets to HTML Calculator:** Starting at \$15.
 - Android App Development Services, Custom Apps:** Starting at \$75.
- Reviews:**
 - matratliff:** 5★, "Very professional and timely. The communication was also spot on! I will definitely use this service again!"
 - nftinnovation:** 5★, "It was the second time working together and another great experience - very reliable, punctual and communicative with excellent English. Highly recommend for anything website related and look forward to working together in the future."
 - melaniechappuis:** 2★, "Not Satisfied, Miscommunication was very noticed"
 - nftinnovation:** 5★

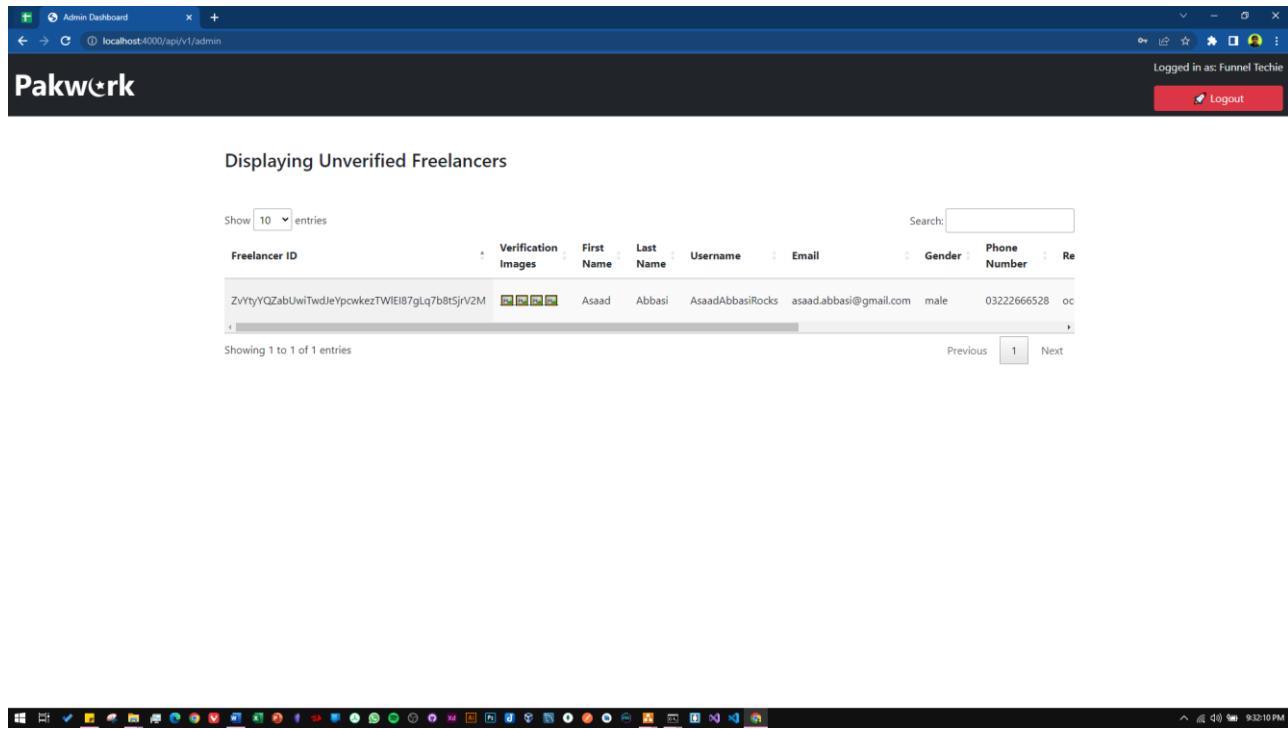
Upload Verification Images



Administrator Login



Displaying Unverified Freelancers



The screenshot shows a web browser window titled "Admin Dashboard" with the URL "localhost:4000/api/v1/admin". The page header "Pakwerk" is visible, along with a "Logout" button. The main content is titled "Displaying Unverified Freelancers". A table lists one freelancer entry:

Freelancer ID	Verification Images	First Name	Last Name	Username	Email	Gender	Phone Number	Re
ZvYtyYGZabUwiTwdJeYpcwkezTWIEI87gLq7b8t5jrV2M		Asaad	Abbasi	AsaadAbbasIRocks	asaad.abbas@gmail.com	male	03222666528	oc

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom right, there are "Previous" and "Next" buttons, with the number "1" in the center.

3.2 Hardware Interfaces

No hardware interfaces are required since the software has no direct relation or dependability on an external hardware device. But a good fast internet connection is recommended for usage of the website

3.3 Software Interfaces

- Visual Studio Code
- MySQL
- MongoDB
- Sockets IO

3.4 Communications Interfaces

Communication interface used in this system is HTTP (Hyper Text Transfer Protocol) and WebSocket protocol with polling as a fallback option for chat messages.

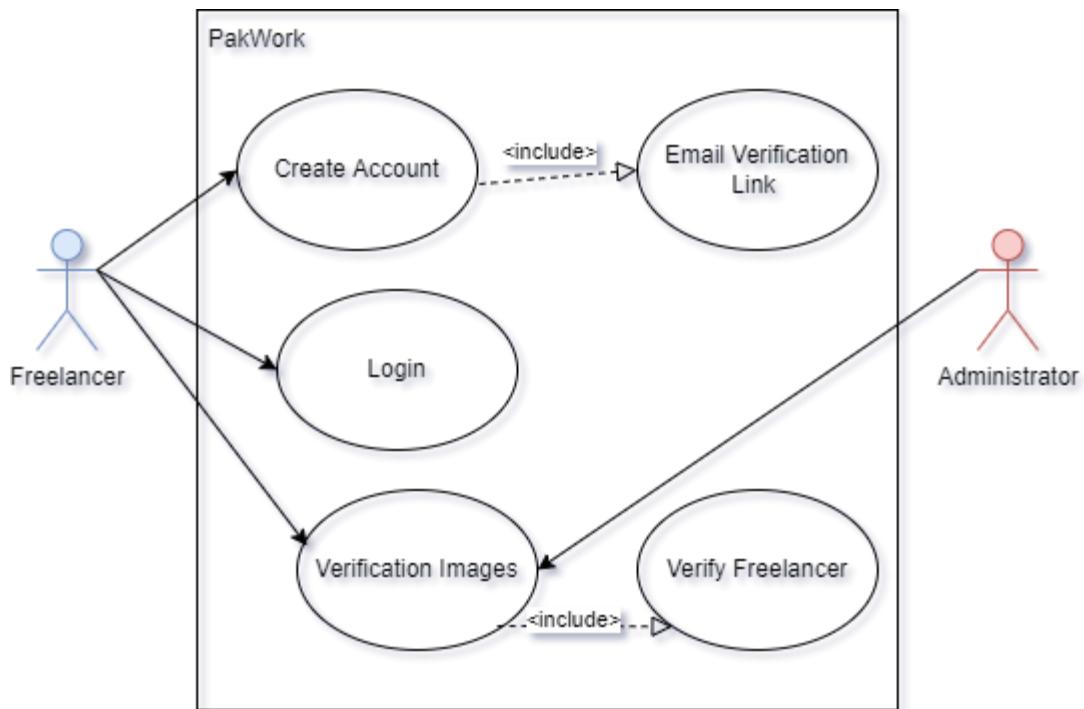
4. System Features

4.1 Freelancer Registration / Verification System

4.1.1 Description and Priority

This system is used to register and verify the freelancer before they can start posting gigs and bidding on jobs.

4.1.2 Stimulus/Response Sequences



Use Case Name:	Freelancer Registration / Verification System	
Summary:	Register and verify freelancer before they use the platform	
User/Actor:	Freelancer, Administrator	
Preconditions:	<ul style="list-style-type: none"> Internet connection necessary Valid information for registering 	
Basic Course of events/happy path		
Step	User Actions	System Actions

1	Freelancers inputs all valid information in registration field	Wait for the registration button to click
2	Registration button clicked	<ul style="list-style-type: none"> System sends an email verification link to user's email address Saves data to database
3	User Logs In & Uploads verification images	<ul style="list-style-type: none"> Save verification images in database Send's pending request to administrator to foresee

Alternate Flow

Step	User Actions	System Actions
1	Wrong Username or password entered when logging in	Displays an error message "Invalid username or password"

Post conditions

- Verification request is accepted/rejected by the administrator after which freelancer can use the platform

Author

Asaad Noman

4.1.3 Functional Requirements

Test Case ID	PW_001	Test Case Description	Testing Freelancer Registration / Verification System		
Created By	Asaad	Reviewed By	Ahsan	Version	1.9
QA Tester's Log	N/A				
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:				
1	Access to Chrome Browser				
2					
3					
4					
Test Scenario	Verify on entering the credentials the freelancer account is created and an email verification link is sent				
Step #	Step Details		Actual Results		

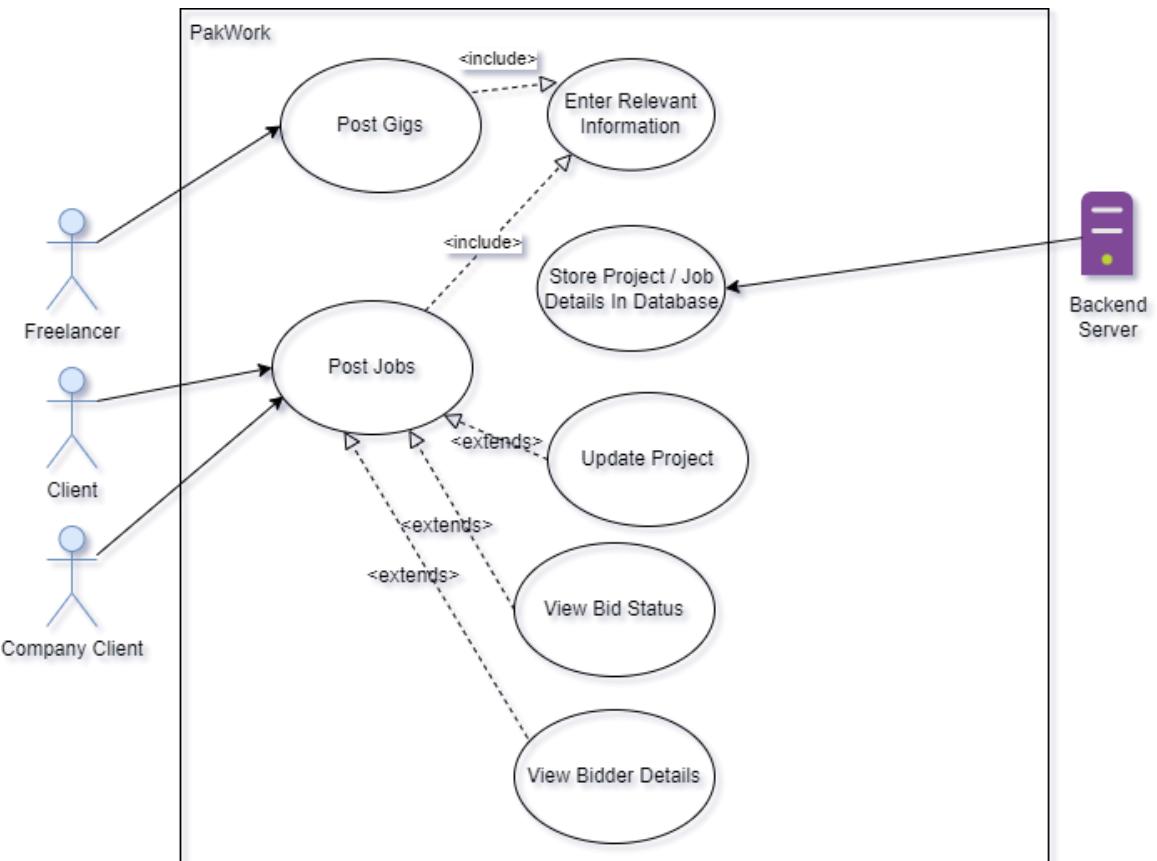
		Expected Results		Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/	Site should open	As Expected	Pass
2	Click “Sell your services”	Credential can be entered	As Expected	Pass
3	Click Register	Freelancer registered	As Expected	Pass
4	Email verification link sent	Link sent	As Expected	Pass

4.2 Gig / Job Creating & Posting

4.2.1 Description & Priority

This feature is used by the freelancer, client & the company client to post gigs, jobs, custom offers respectively

4.2.2 Stimulus/Response Sequences



Use Case Name:	Gig/Job Creation & Posting	
Summary:	Feature used by freelancers, client & company clients to post job, gigs & send custom offers	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	The users of the system must be logged in.	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The User clicks the create gig/job button	System displays a modal dialog for user to enter the necessary details
2	User Clicks Post Button	<ul style="list-style-type: none"> • Job/Gig gets saved in the database • Displayed on the profile dashboard
Alternate Flow		
Step	User Actions	System Actions
1	Some Details are left out and post button is clicked	Displays an error message to fill the empty fields
Post conditions		
The website is refreshed and updated information is displayed on the dashboard		
Author		
Asaad Noman		

4.2.3 Functional Requirements

Test Case ID	PW_002	Test Case Description	Testing Gig/Job Posting		
Created By	Asaad	Reviewed By	Ahsan	Version	1.0
QA Tester's Log	Testing this from the freelancer side				
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:				
1	Should be logged in				
2	Should be verified & activated				
3					
S #	Test Data				
1	title = Sell Scripts for automation				
2	startingPrice= 10				
3	detail = lorem ipsum				

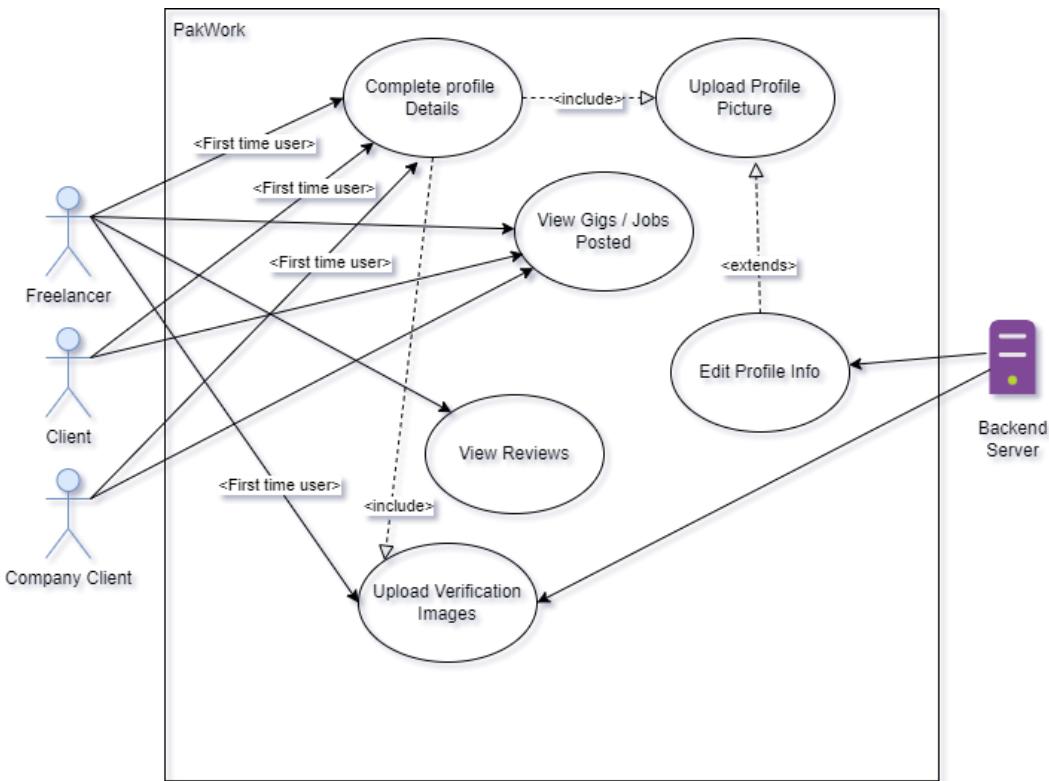
<u>Test Scenario</u>	Verify on posting that the gig is posted properly and is displaying			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not execute d / Suspend ed
1	Navigate to http://localhost:3000/profile/	Site should open	As Expected	Pass
2	Click create Gig button	Credential can be entered	As Expected	Pass
3	Click Post	Gig has been posted	As Expected	Pass

4.3 Profile Dashboard

4.3.1 Description & Priority

This feature is basically a dashboard for the user whether it be client, Company client or freelancer. They can view, create their jobs/gigs, change profile picture or update their bio or upload verification images.

4.3.2 Stimulus/Response Sequences



Use Case Name:	Profile Dashboard	
Summary:	A Feature that allows the users to see and update their profile bio, upload profile pictures, verification images and their gig and jobs posted	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	The users of the system must be logged in.	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The user clicks on profile dashboard	If profile not completed system tells to complete profile information
2	User completes profile info	System allows user to view all the necessary information on the profile dashboard
Alternate Flow		
Step	User Actions	System Actions
1	User tries to view profile without completing profile info	System continues to show error to complete profile information.
Post conditions		
Every information is available in the profile dashboard for the user to see.		
Author		
Asaad Noman		

4.3.3 Functional Requirements

Test Case ID	PW_003	Test Case Description	Testing the profile dashboard	
Created By	Asaad	Reviewed By	Ahsan	Version 1.0
QA Tester's Log	Reviewing the freelancer profile			
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed) Pass
S #	Prerequisites:		S #	Test Data
1	Should beloggedin		1	N/A
Test Scenario	Verify if all the data on profile dashboard is displaying correctly			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not

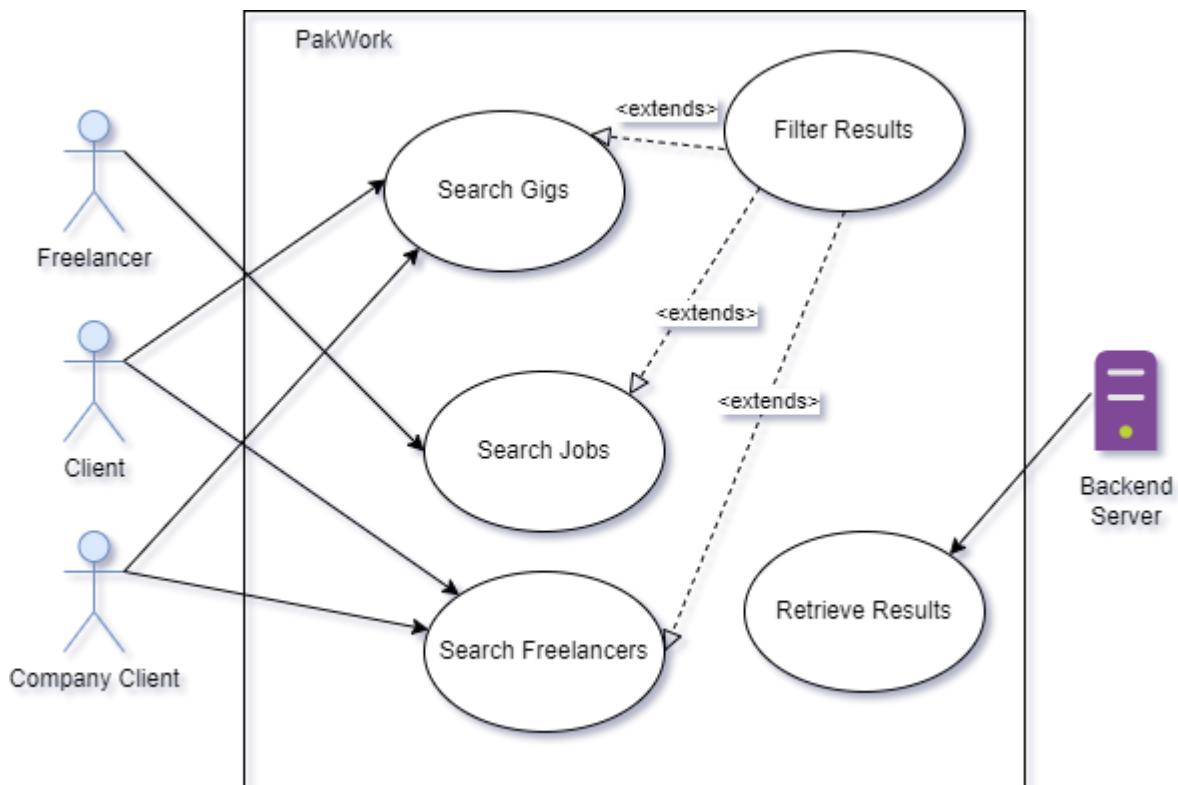
				execute d / Suspend ed
1	Navigate to http://localhost:3000/profile	Site should open	As Expected	Pass
2	Profile is displayed	All Information is being displayed	As Expected	Pass

4.4 Search System

4.4.1 Description & Priority

This feature will allow the clients and freelancer respectively to search for gigs / jobs on the platform

4.4.2 Stimulus/Response Sequences



Use Case Name:	Search System	
Summary:	This feature allows any type of users to search for desired data in the platform	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	Working internet connection	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The user clicks on search box	Does nothing
2	User selects type of information needed to search from combo box	Does nothing
3	User enters desired text	System starts giving suggestion based on the input given
4	User presses enter	A new page is loaded giving the matched result to the user query
Alternate Flow		
Step	User Actions	System Actions
1	User writes in search box	Result is not retrieved, since no matched query from the database. Data does not exist
Post conditions		
Search result is displayed		
Author		
Asaad Noman		

4.4.3 Functional Requirements

Test Case ID	PW_004	Test Case Description	Testing the search system		
Created By	Asaad	Reviewed By	Ahsan	Version	1.0
QA Tester's Log					N/A
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:			S #	Test Data

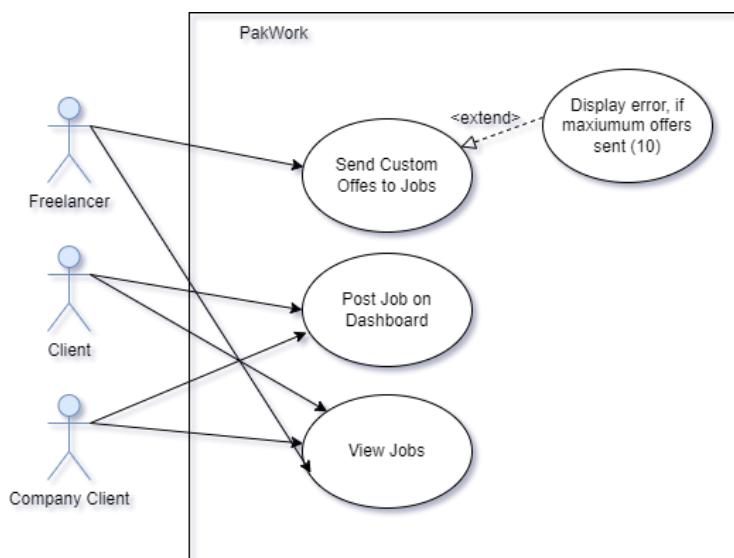
1	Internet Connectivity	1	Search box: Web Development	
Test Scenario	Verify if all the data on profile dashboard is displaying correctly			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/	Site should open	As Expected	Pass
2	Click on search box and select type of information needed and start typing	Information displayed dynamically	As Expected	Pass
3	Press enter	New page loaded and information displayed	As Expected	Pass

4.5 Buyer Requests

4.5.1 Description & Priority

This is basically a job board dashboard that allows miscellaneous jobs to be posted that do not fit into the categories offered in the platform.

4.5.2 Stimulus/Response Sequences



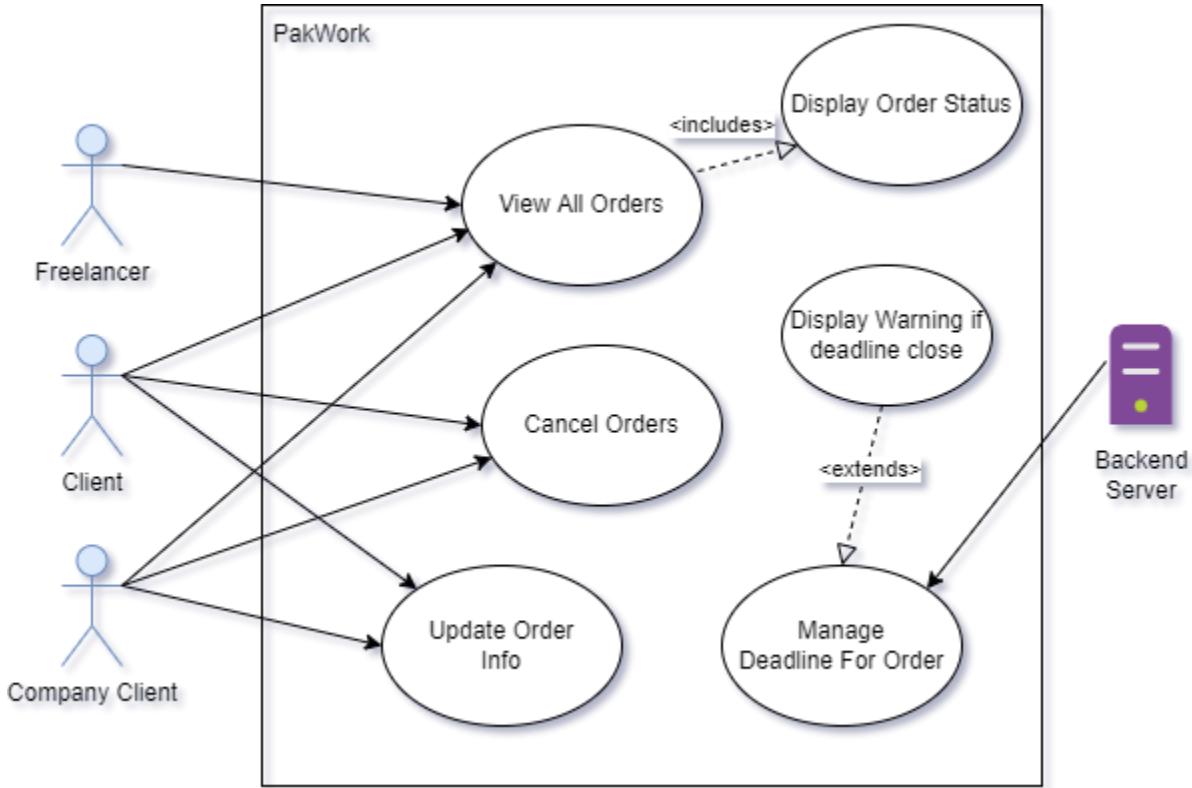
Use Case Name:	Buyer Requests	
Summary:	This is a job board dashboard for miscellaneous jobs.	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	Should be logged in	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The user clicks on buyer request dashboard	Displays all the current jobs posted and Create job button depending on the user
2	If user is a client/company client, they click create job button	Modal Dialog is showed
3	User enters desired information about the job	System waits for the post button to be clicked
4	User clicks the post button	New Job has been posted
Alternate Flow		
Step	User Actions	System Actions
1	If user is a freelancer, inputs the bid	Does nothing
2	User clicks the bid now button	New Bid has been placed and the bid is updated on the screen
Author		
Asaad Noman		

4.6 Order Management

4.6.1 Description & Priority

This is an order management feature where freelancers, clients and company clients can manage their orders respectively.

4.6.2 Stimulus/Response Sequences



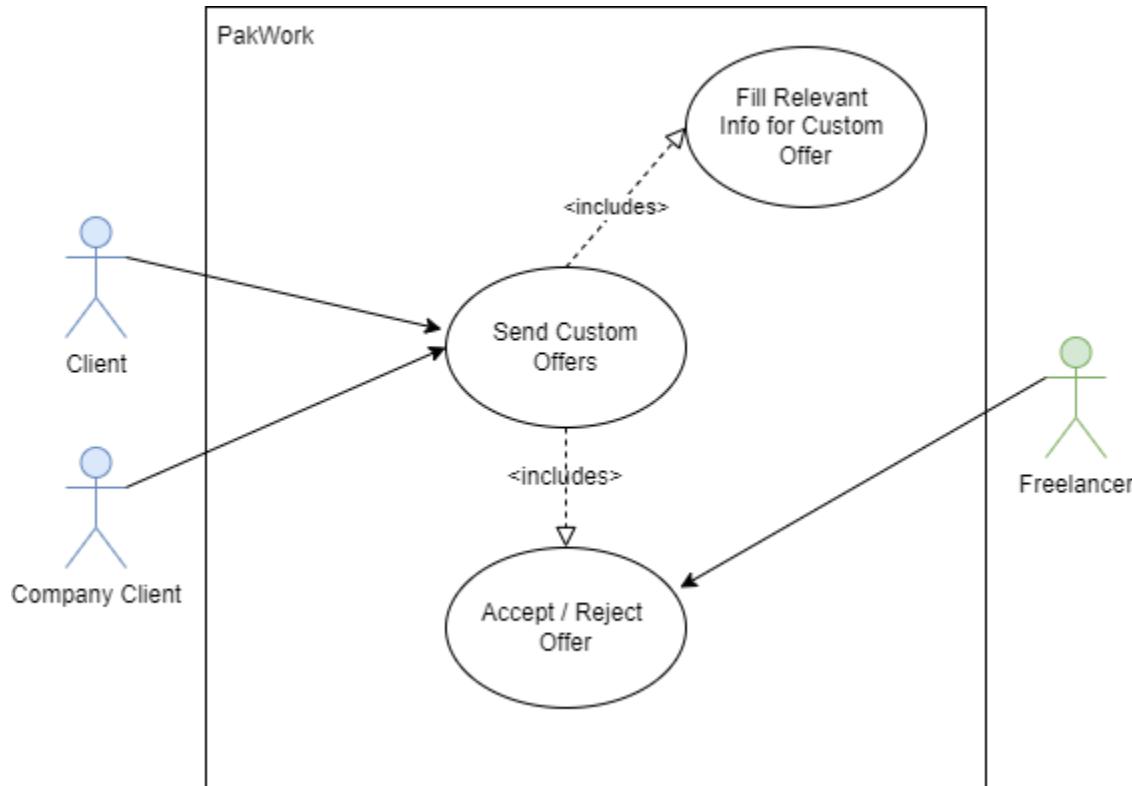
Use Case Name:	Order Management	
Summary:	This is an order management dashboard to manage the orders for freelancers, clients and company clients	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	<ul style="list-style-type: none"> • Should be logged in • Some order should be in progress 	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The user clicks on order management dashboard	System displays all the orders in a list
Author		
Asaad Noman		

4.7 Custom Offers

4.7.1 Description & Priority

This is a feature only for clients and company clients, to send custom offers to their desired/scouted freelancers.

4.7.2 Stimulus/Response Sequences



Use Case Name:	Custom Offers
Summary:	This feature allows clients and company clients to send custom offers to freelancers of their choice.
User/Actor:	Freelancer, Client, Company Client
Preconditions:	The users of the system must be logged in.
Basic Course of events/happy path	

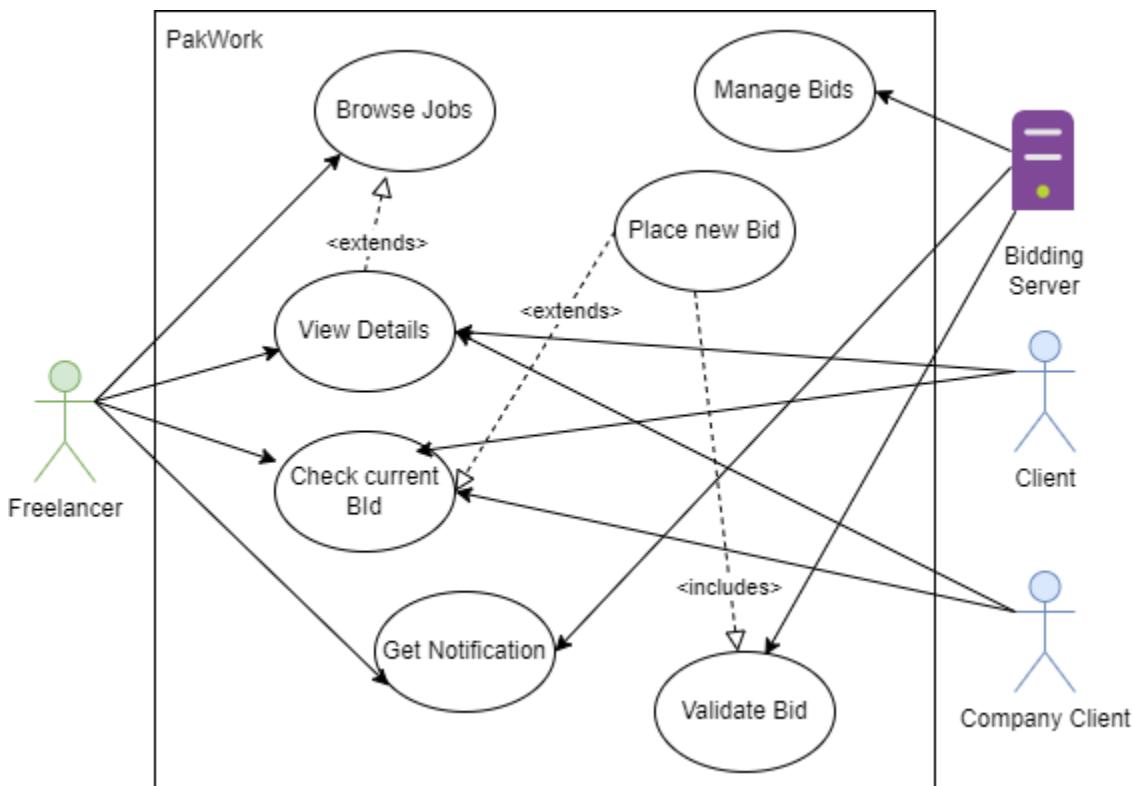
Step	User Actions	System Actions
1	The user clicks on profile of freelancer	System displays all information of the freelancers on a new page
2	User clicks custom offer buttons	System shows a modal dialog for inputting information
3	User fills all info and clicks the send offer button	Offer is sent to the freelancer and is awaiting response
Post conditions		
The offer status is pending until freelancer sees it and accepts or rejects the custom offer		
Author		
Asaad Noman		

4.8 Bidding System

4.8.1 Description & Priority

This is a bidding system for jobs, freelancers can bid on jobs and compete with other freelancers to quote the best price for a project.

4.8.2 Stimulus/Response Sequences



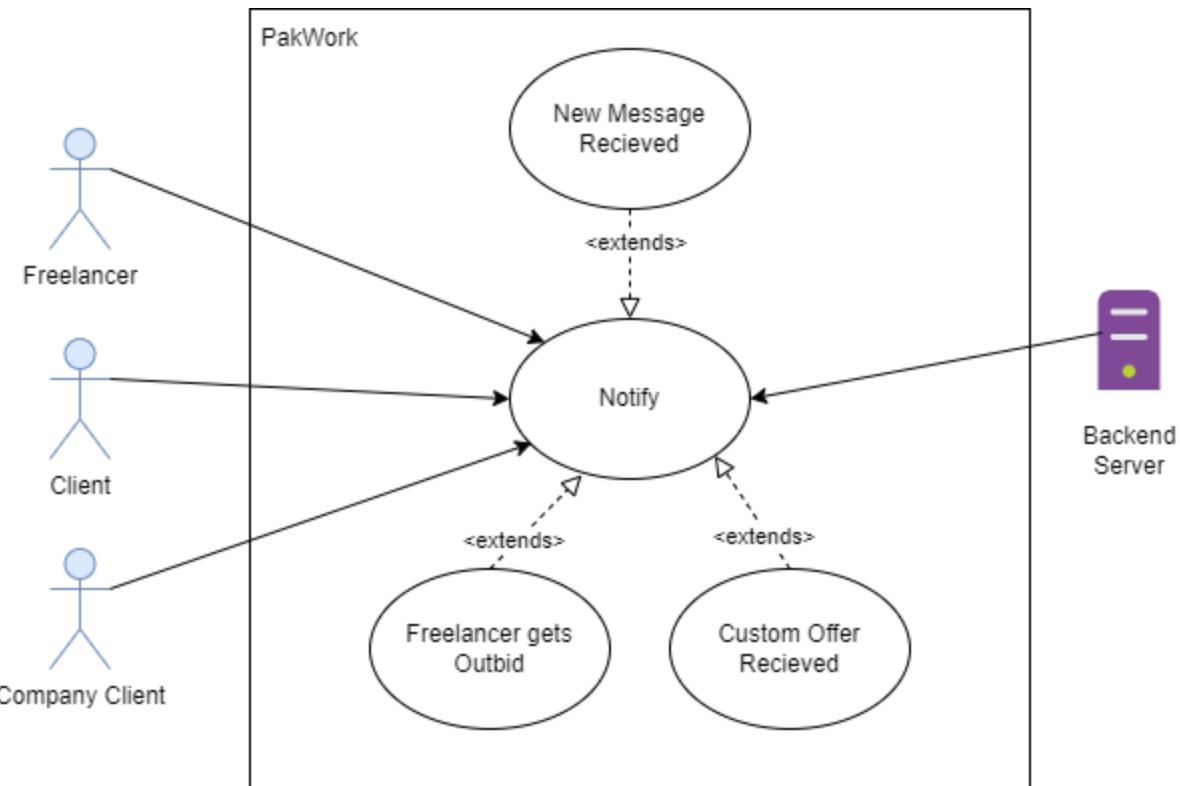
Use Case Name:	Bidding System	
Summary:	This is a bidding system for jobs, freelancers can bid on jobs and compete with other freelancers to quote the best price for a project.	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	Should be logged in	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	Freelancer views all the jobs listed	Does Nothing
2	Freelancer clicks detail button on the job posting	System shows all the relevant description and files related to the job posted
3	Freelancer enters bid and clicks the bid now button	System validates the bid, if all correct, bid is placed
Alternate Flow		
Step	User Actions	System Actions
1	If user is a freelancer, inputs an invalid bid	Displays an error message
Author		
Asaad Noman		

4.9 Push Notification System

4.9.1 Description & Priority

This notification system is used to notify user, if they have been outbidded or not, if they received a new chat message or custom offer.

4.9.2 Stimulus/Response Sequences



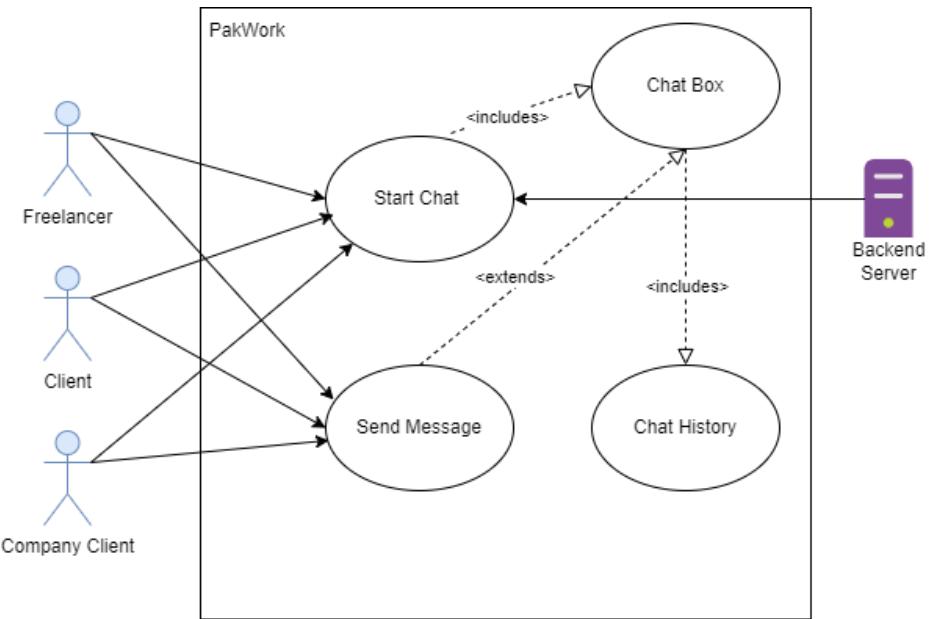
Use Case Name:	Push Notification System	
Summary:	This system allows the users to get notified when certain actions take place such as new message, outbid on jobs, custom offers	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	The users of the system must be logged in.	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	The user gets outbid on project	Sends a notification to the user, intimating of the outbid
Author		
Asaad Noman		

4.10 Chat System

4.10.1 Description & Priority

A Feature that can be used by the freelancer & clients respectively to communication with each other without leaving the platform.

4.10.2 Stimulus/Response Sequences



Use Case Name:	Chat System	
Summary:	A Feature that can be used by the freelancer & clients respectively to communication with each other without leaving the platform.	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	Should be logged in	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	Opens Chat Dashboard	Displays All User list with the user has chat with
2	Selects A User from list	Retrieves chat history from database and opens up a chat box with a socket connection

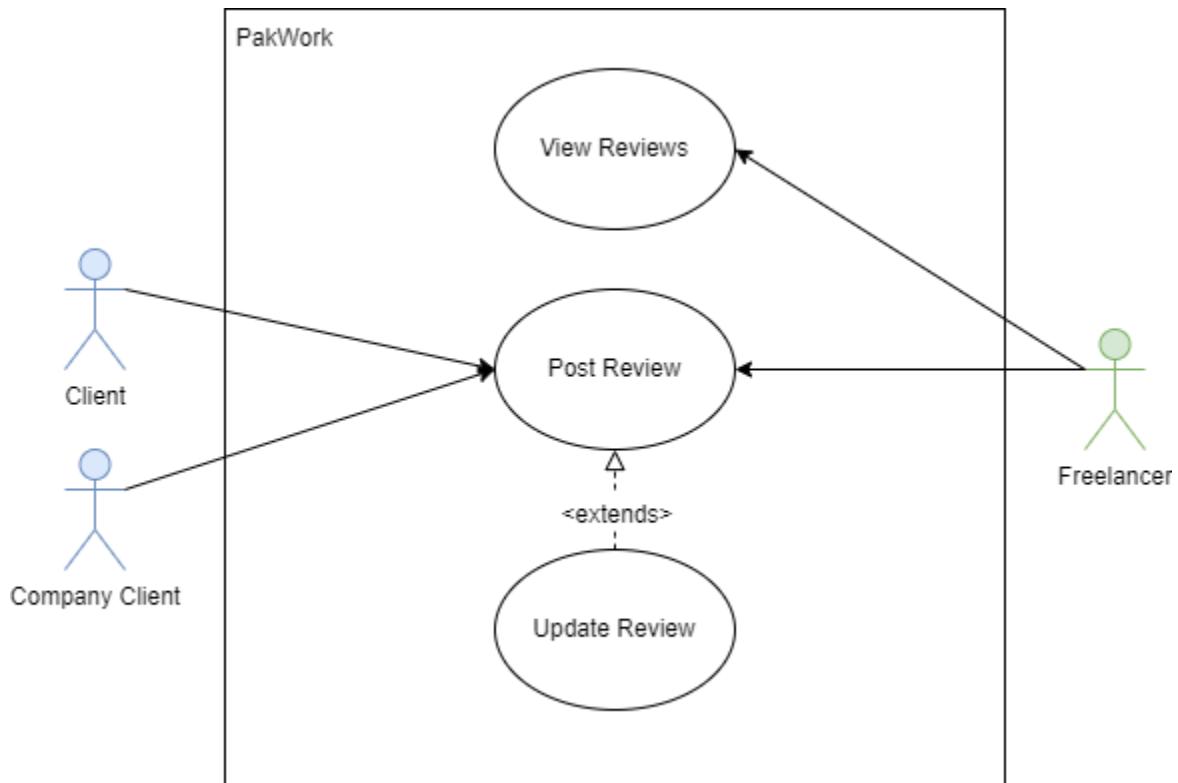
3	User sends message	Using sockets message is sent and message is stored in the database
Author		
Asaad Noman		

4.11 Rating/Feedback System

4.11.1 Description & Priority

This rating and feedback system is a feature for clients and company clients to leave a review of the work/performance of the freelancer on the job. This helps other user to differentiate between good and bad freelancers. Also helps the search system.

4.11.2 Stimulus/Response Sequences



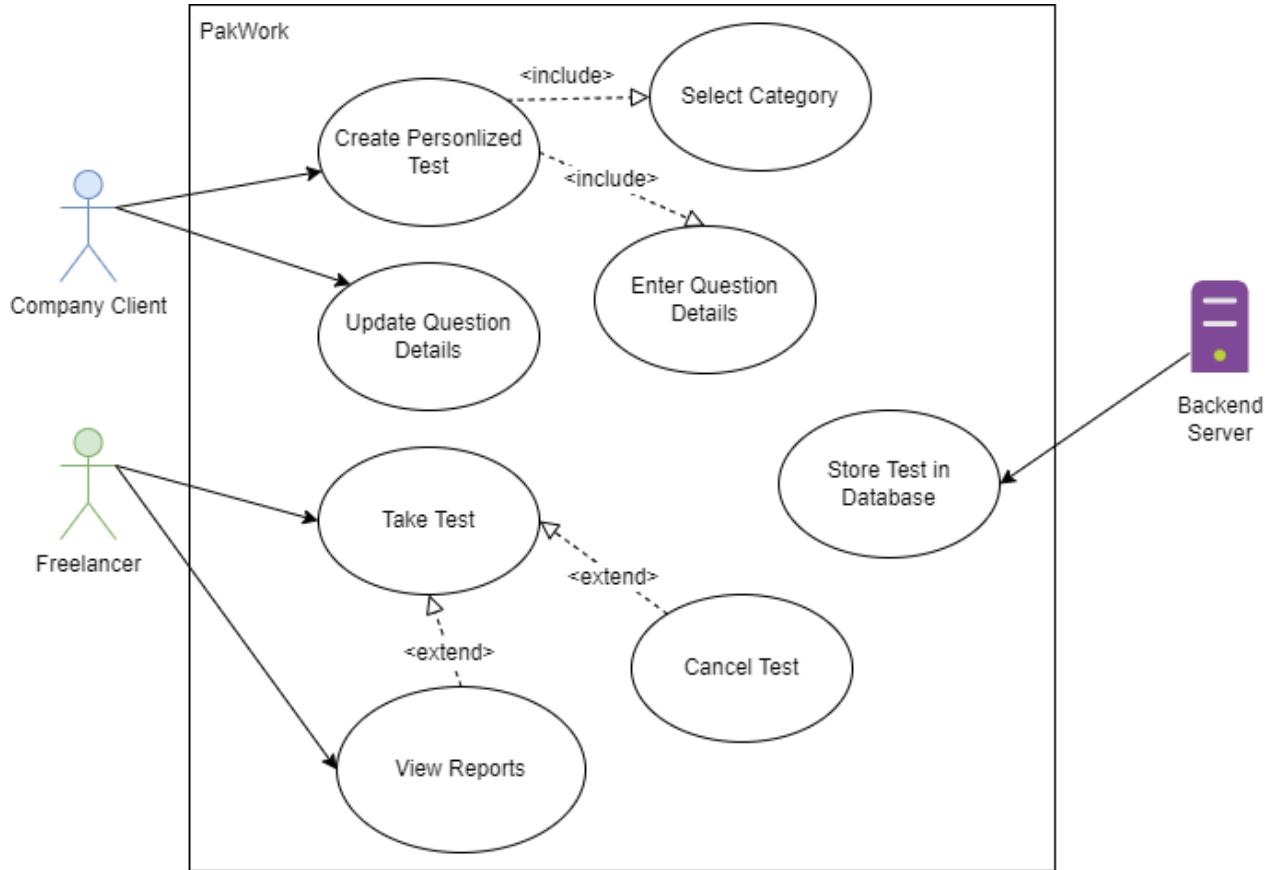
Use Case Name:	Rating/Feedback System	
Summary:	This rating and feedback system is a feature for clients and company clients to leave a review of the work/performance of the freelancer on the job. This helps other user to differentiate between good and bad freelancers. Also helps the search system.	
User/Actor:	Freelancer, Client, Company Client	
Preconditions:	<ul style="list-style-type: none"> • Should be logged in • Must be a verified buyer of the services of the freelancer 	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	Click the post a review button	Open up a modal dialog
2	User enters the details of the review, rating etc.	Does nothing
3	Clicks the submit button	System saves the review in the database and refreshes
Alternate Flow		
Step	User Actions	System Actions
1	User tries to submit without filling all information	Displays an error message
Author		
Asaad Noman		

4.12 Personalized Testing System

4.12.1 Description & Priority

This feature is available only to the Company client. This allows to set a test before project bidding for freelancer to filter the good ones from the bad ones. Also allows to create a customized test.

4.12.2 Stimulus/Response Sequences



Use Case Name:	Personalized Testing System	
Summary:	This feature is available only to the Company client. This allows to set a test before project bidding for freelancer to filter the good ones from the bad ones. Also allows to create a customized test.	
User/Actor:	Freelancer, Company Client	
Preconditions:	<ul style="list-style-type: none"> • Should be logged in • Must be logged in as a company client 	
Basic Course of events/happy path		
Step	User Actions	System Actions
1	Click the create test button	Open up a modal dialog
2	User enters the details of the test and adds questions	Does nothing
3	Clicks the submit button	Saves the question in the database
4	Sets the test when posting new job	System will ask freelancer to first take the test

Author
Asaad Noman

5. Other Nonfunctional Requirements

5.1 Performance Requirements

N/A

5.2 Safety Requirements

Do not deal with clients outside platform, for avoidance of being scammed.

5.3 Security Requirements

Freelancer verification should be done thoroughly to avoid the problem of impersonation and multiple profiles of the same person on the platform to maintain the integrity of the system. All information must be placed securely to avoid data loss and invasion of privacy.

5.4 Software Quality Attributes

- **Correctness:** The website should work as expected and contain minimal error and also should be bug free. It is achieved by systematically testing the system and its different components.
- **Maintainability:** System design of the website is done in such a way that it would be easy for the developer to maintain and fix the problems in the future.
- **Usability:** The user interface of the website is designed in such a way that it is easy to navigate and all features are self-explanatory.
- **Reliability:** The website provides secure and integrated communication system that connects freelancers to their clients with ease without making the need to leave the platform.
- **Testability:** This is a necessary attribute for SQA. This is thoroughly done as stated in the testing documents.

5.5 Business Rules

N/A

6. Other Requirements

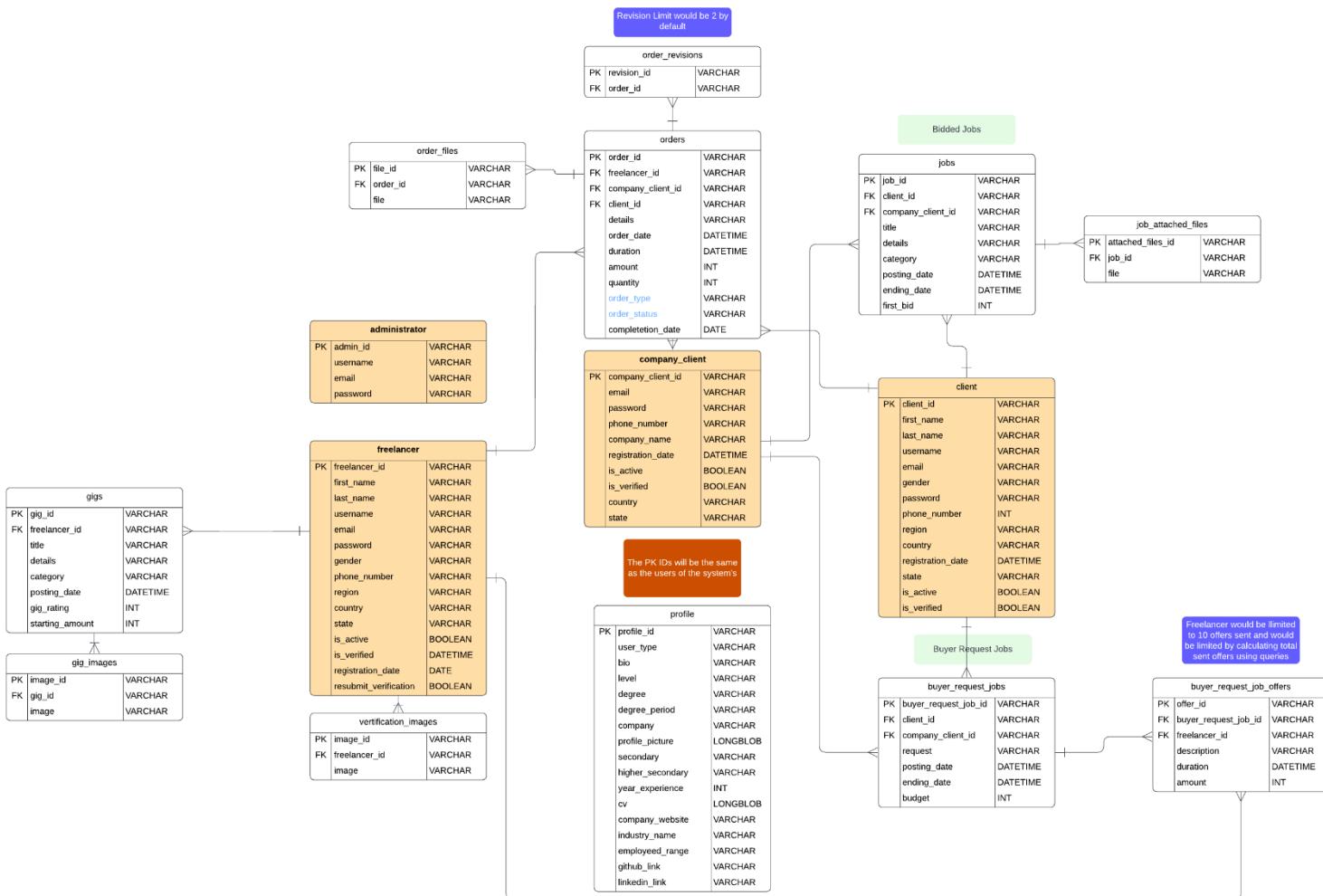
N/A

Appendix A: Glossary

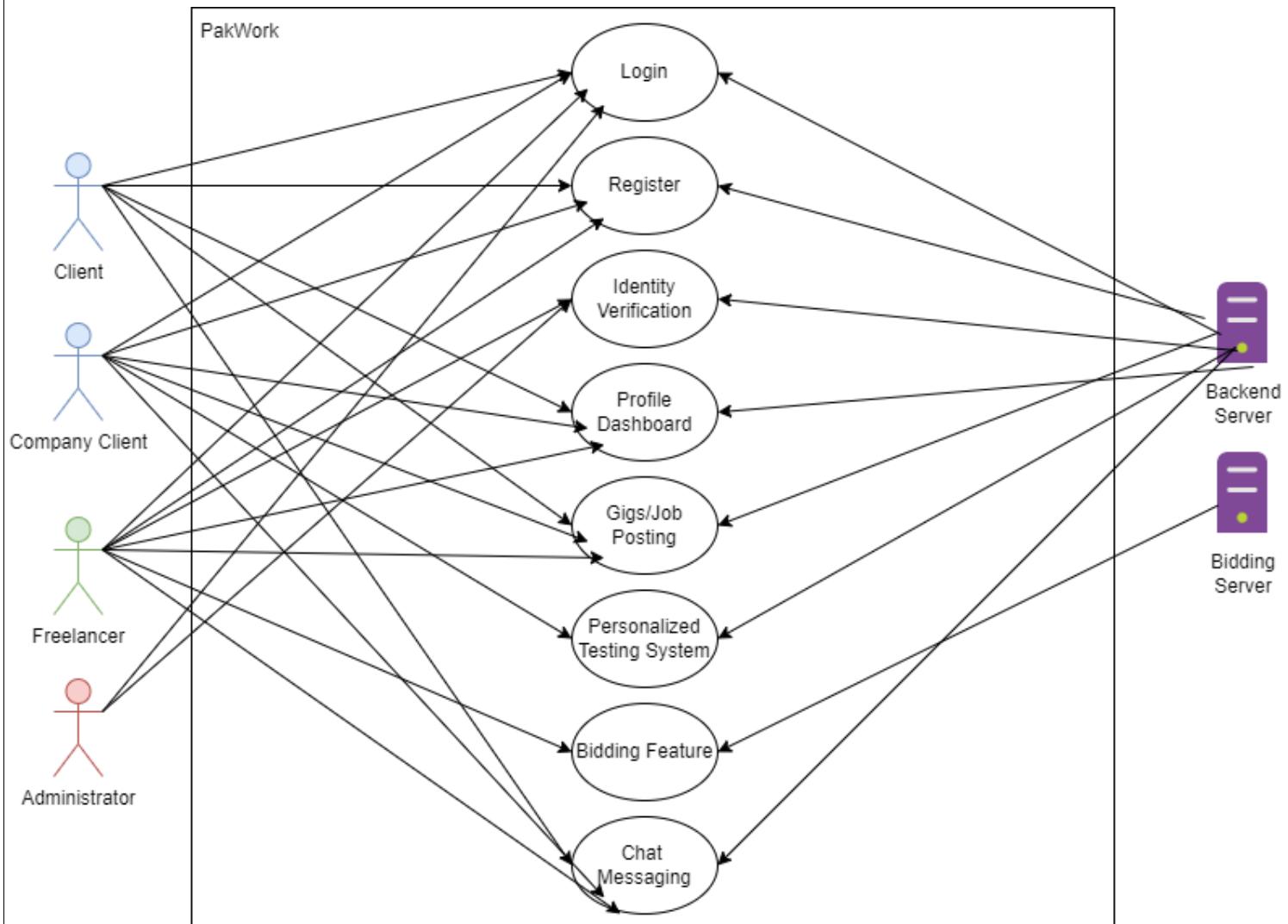
S.no	Terms	Description
1	User Interface	The means by which computer system interact with the user
2	Database	A structured set of data held in a computer
3	Sockets	A socket is a software architecture within a network node of a computer network that serves as an endpoint for receiving and sending data across a network
4	HTTP	Hypertext transfer protocol is a communication interface

Appendix B: Analysis Models

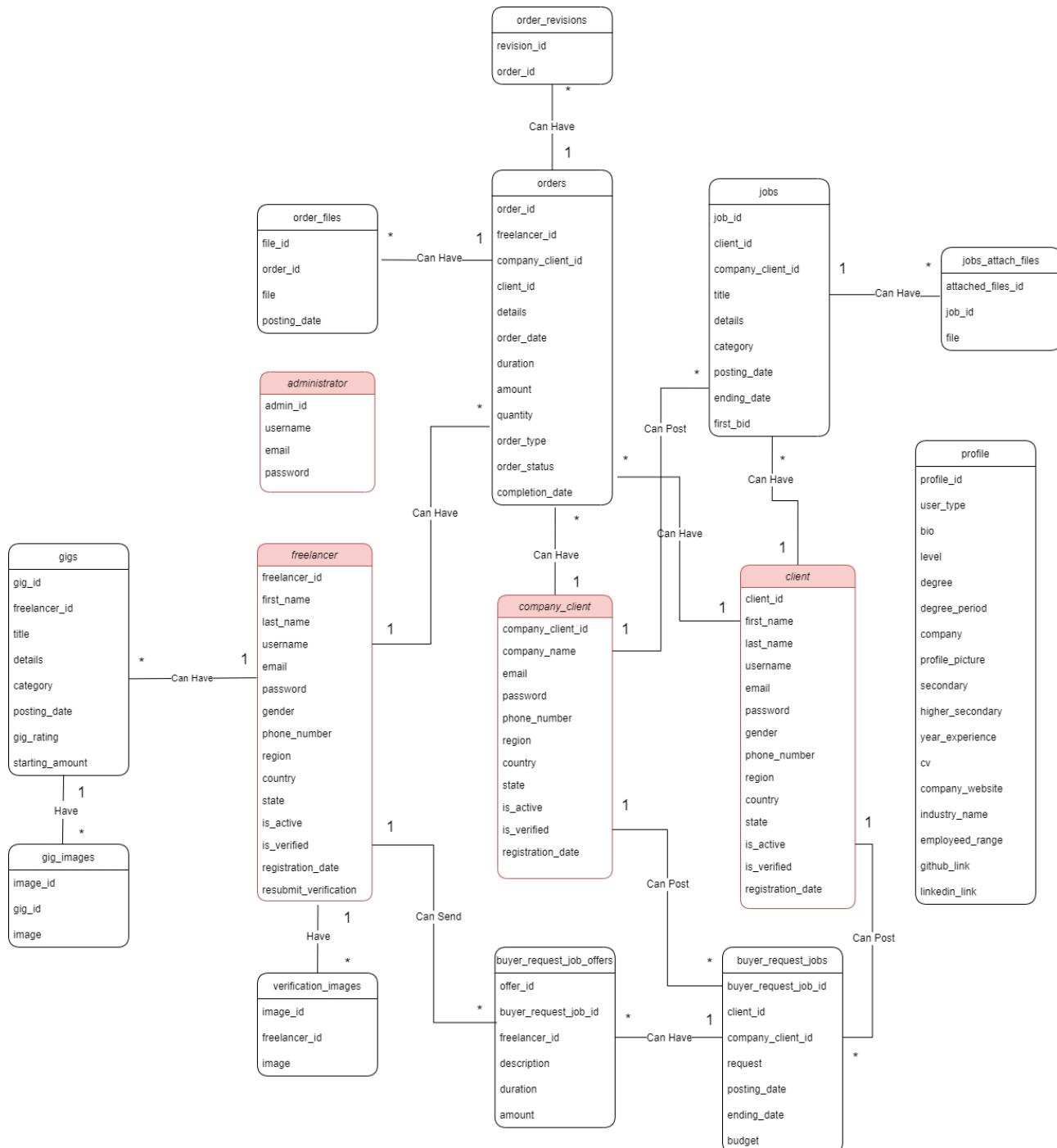
ERD:



System Use Case Diagram:



Domain Model:



Appendix C: To Be Determined List

N/A

Testing Documents

1. Freelancer Registration/Verification System

Test Case ID	PW_001	Test Case Description	Testing Freelancer Registration / Verification System		
Created By	Asaad	Reviewed By	Ahsan	Version	1.9
QA Tester's Log		N/A			
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #		Prerequisites:			
1		Access to Chrome Browser			
2					
3					
4					
Test Scenario		Verify on entering the credentials the freelancer account is created and an email verification link is sent			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to http://localhost:3000/	Site should open	As Expected	Pass	
2	Click “Sell your services”	Credential can be entered	As Expected	Pass	
3	Click Register	Freelancer registered	As Expected	Pass	
4	Email verification link sent	Link sent	As Expected	Pass	

2. Gig/Job Posting

Test Case ID	PW_002	Test Case Description	Testing Gig/Job Posting		
Created By	Asaad	Reviewed By	Ahsan	Version	1.0
QA Tester's Log	Testing this from the freelancer side				
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:				
1	Should be logged in				
2	Should be verified & activated				
3					
Test Scenario	Verify on posting that the gig is posted properly and is displaying				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to http://localhost:3000/profile/	Site should open	As Expected	Pass	
2	Click create Gig button	Credential can be entered	As Expected	Pass	
3	Click Post	Gig has been posted	As Expected	Pass	

3. Profile Dashboard

Test Case ID	PW_003	Test Case Description	Testing the profile dashboard		
Created By	Asaad	Reviewed By	Ahsan	Version	1.0
QA Tester's Log	Reviewing the freelancer profile				
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:				
1	Should beloggedin				
Test Scenario	Verify if all the data on profile dashboard is displaying correctly				
Step #	Step Details	Actual Results			

		Expected Results		Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/profile	Site should open	As Expected	Pass
2	Profile is displayed	All Information is being displayed	As Expected	Pass

4. Search System

Test Case ID	PW_004	Test Case Description	Testing the search system		
Created By	Asaad	Reviewed By	Ahsan	Version	1.0
QA Tester's Log	N/A				
Tester's Name	Asaad	Date Tested	9 th November 2022	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:			S #	Test Data
1	Internet Connectivity			1	Search box: Web Development
Test Scenario	Verify if all the data on profile dashboard is displaying correctly				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to http://localhost:3000/	Site should open	As Expected	Pass	
2	Click on search box and select type of information needed and start typing	Information displayed dynamically	As Expected	Pass	
3	Press enter	New page loaded and information displayed	As Expected	Pass	

5. Order Management

Test Case ID	PW_005	Test Case Description	Testing Order Management		
Created By	Ahsan	Reviewed By	Asaad	Version 1	
QA Tester's Log	Reviewing Ordering System				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed) Pass	
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sample Order: \$10 2 Days Delivery Random Files in Delivery	
2	Freelancer Account				
3	Placed Order				
Test Scenario	Delivering an order that was placed by a buyer				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard/orders	Site Should Open	As Exepcted	Pass
2	Click on the active order by looking at it's status	Order Page appear with appropriate information	As Exepcted	Pass
3	Click on the deliver order button	Delivery Panel with file upload options	As Exepcted	Pass
4	Upload nesscary files regarding to order and Click Submit	Order Files Delivered and Pending for buyer's approval	As Exepcted	Pass

Test Case ID	PW_006	Test Case Description	Testing Order Management	1
Created By	Ahsan	Reviewed By	Asaad	Version
QA Tester's Log	Reviewing Ordering System			
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)
S #	Prerequisites:		S #	Test Data
1	Internet Connectivity		1	Sample Order: \$10 2 Days Delivery
2	Buyer Account			
3	Placed Order			
Test Scenario	Accepting an order that has been delivered by a freelancer			

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspend ed
1	Navigate to http://localhost:3000/dashboard/orders	Site Should Open	As Exepcted	Pass
2	Click on the delivered order by looking at it's status	Order Page appear with appropriate information	As Exepcted	Pass
3	Click on the accept delivery button	Order accepted and marked complete	As Exepcted	Pass

6. Buyer Request System

Test Case ID	PW_007	Test Case Description	Testing Buyer Request System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing Buyer's request system				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sample Request: Website Creation	
2	Buyer Account		2	Timeline: 30 days	
			3	Budget: 350\$	
Test Scenario	Placing a buyer request for an particular sample website creation				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on Create A Request button	Buyer Request Panel should open with appropriate options	As Expected	Pass
3	Fill Appropriate Information and Click Submit	Buyer Request should be placed and displayed on the buyer's requests page	As Expected	Pass

Test Case ID	PW_008	Test Case Description	Testing Buyer Request System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing Buyer's request system				
Tester's Name		Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sample Offer: Offering to build website	
2	Freelancer Account			Asking Price: 380\$	
Test Scenario	Offering the buyer requests that are currently active				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on the Buyer Requests NavLink	Page with active buyer requests	As Expected	Pass
3	Click on the Offer Button	Offer Detail Panel should open with appropriate options	As Expected	Pass
4	Fill the appropriate information and click Send	Offer should be sent to the buyer	As Expected	Pass

7. Chat System

Test Case ID	PW_009	Test Case Description	Testing Chat System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing a chat system by sending a sample text				
Tester's Name		Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Text Message: "Hello"	
2	Logged In Any Account				
Test Scenario	Sending a sample text message to a particular user via chat				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on user profile	User Profile Page should open	As Expected	Pass
2	Click on send message button	Chat Page should appear with appropriate options	As Expected	Pass
3	Enter appropriate details in the option and Click send	Message will be sent to the user	As Expected	Pass

8. Rating/Feedback System

Test Case ID	PW_01 0	Test Case Description	Testing Feedback/Rating System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	This would be right after the order completion				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sample Feedback Text: "Good Service"	
2	Buyer Account		2	Sample Stars: 5	
3	Completed Order				
Test Scenario	Giving a feedback on the order that was completed				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on the leave a feedback button on the order page	Feedback Panel should be open with appropriate options	As Expected	Pass
2	Enter the appropriate details on the panel and Click submit button	Feedback will be sent on the order	As Expected	Pass

9. Push Notification System

Test Case ID	PW_01 1	Test Case Description	Testing Push Notification		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Testing if the push notifications are being sent				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sending Notification: Buyer Left Feedback	
2	Freelancer Account				
3	Completed Order				
Test Scenario	Buyer leaving feedback on the order notifying the freelancer				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Interact normally with your tasks and wait for the notification	Push notification from the website about the buyer leaving a feedback	As Expected	Pass

10. Bidding System

Test Case ID	PW_012	Test Case Description	Testing Bidding System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing Bidding System				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Sample Project: Build Website	
2	Buyer Account		2	Initial Bid: \$15	
Test Scenario	Creating a live bidding session for a sample project				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on the Live Bidding NavLink	Page with appropriate bidding information should appear	As Expected	Pass
3	Click on create a live bidding session	Bidding Session Panel should appear with appropriate options	As Expected	Pass
4	Enter appropriate details in the options and Click start	Bidding Session should start with the buyer placed in a live room	As Expected	Pass

Test Case ID	PW_013	Test Case Description	Testing Bidding System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing Bidding System				
Tester's Name		Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pas s
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Offering Bid: 35\$	
2	Freelance Account				
Test Scenario	Making a bid in the live bidding session				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on the Live Bidding NavLink	Page with appropriate bidding information should appear	As Expected	Pass
3	Select the appropriate bidding session and click join	Page with live bidding session should appear	As Expected	Pass
4	Select the appropriate bidding session and click join	Page with live bidding session should appear	As Expected	Pass
5	Enter appropriate bidding information and Click Place Bid	Offering Bid should be placed in the live session	As Expected	Pass

11. Personalized Testing System

Test Case ID	PW_014	Test Case Description	Testing the Personalized Testing System		
Created By	Ahsan	Reviewed By	Asaad	Version	1
QA Tester's Log	Reviewing the PTS				
Tester's Name	Ahsan	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	2 Sample Questions for a test: Q1. What is HTML Q2. What is Node.js	
2	Company Buyer Account				
Test Scenario	Creating a Personalized Test				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on the Personalized Tests NavLink	Page with appropriate Personalized Tests information should appear	As Expected	Pass
3	Click on the Create new Test	Test Panel will open with appropriate options to create a test	As Expected	Pass
4	Enter the appropriate infomration in the options and Click Submit	A Personalized Test should be created	As Expected	Pass

12. Custom Offers

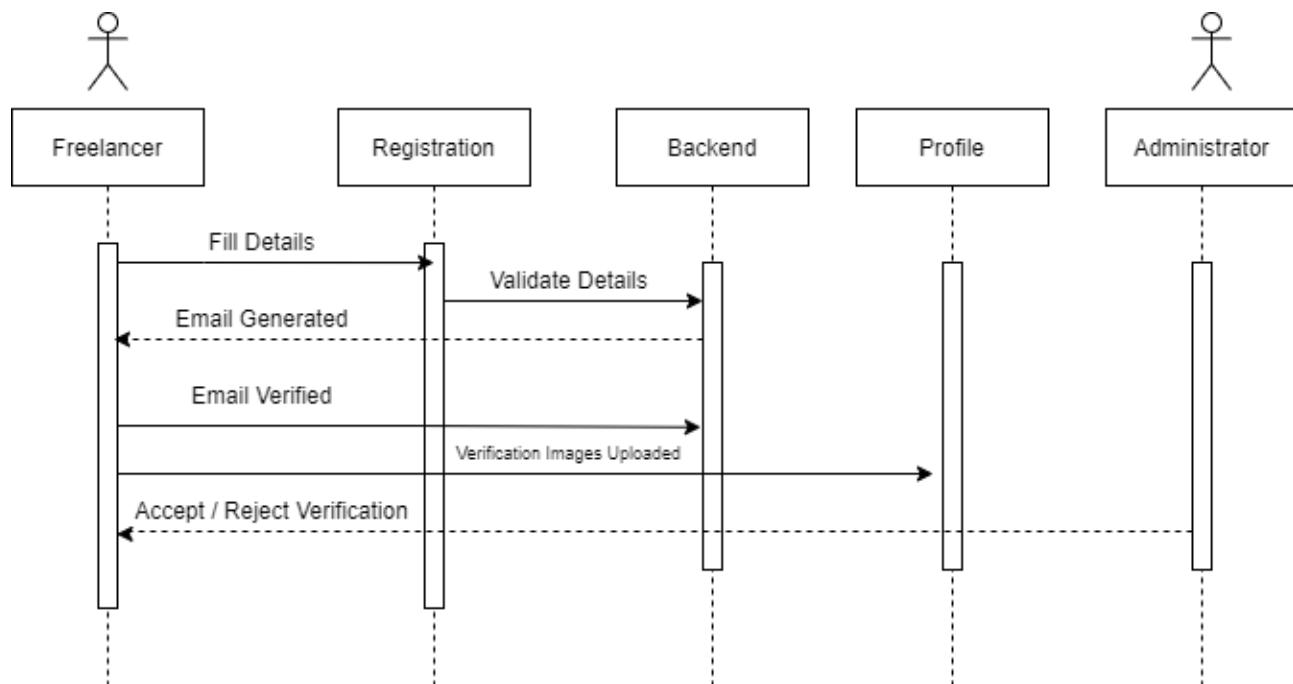
Test Case ID	PW_015	Test Case Description	Testing the custom offers		
Created By	Asaad	Reviewed By	Ahsan	Version	1
QA Tester's Log	Reviewing Custom Offers				
Tester's Name	Asaad	Date Tested	10th January 2023	Test Case (Pass/Fail/Not Executed)	Pas s
S #	Prerequisites:		S #	Test Data	
1	Internet Connectivity		1	Custom offers form data	
2	Company Buyer Account or client account				
Test Scenario	Creating a Custom Order				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:3000/dashboard	Site Should Open	As Expected	Pass
2	Click on the chat tab	Chat panel with users should display	As Expected	Pass
3	Click on any freelancer	Open chat window with freelancer	As Expected	Pass
4	Press Custom offer button, fill and send offer	A Form Should Appear and offer should be able to send	As Expected	Pass

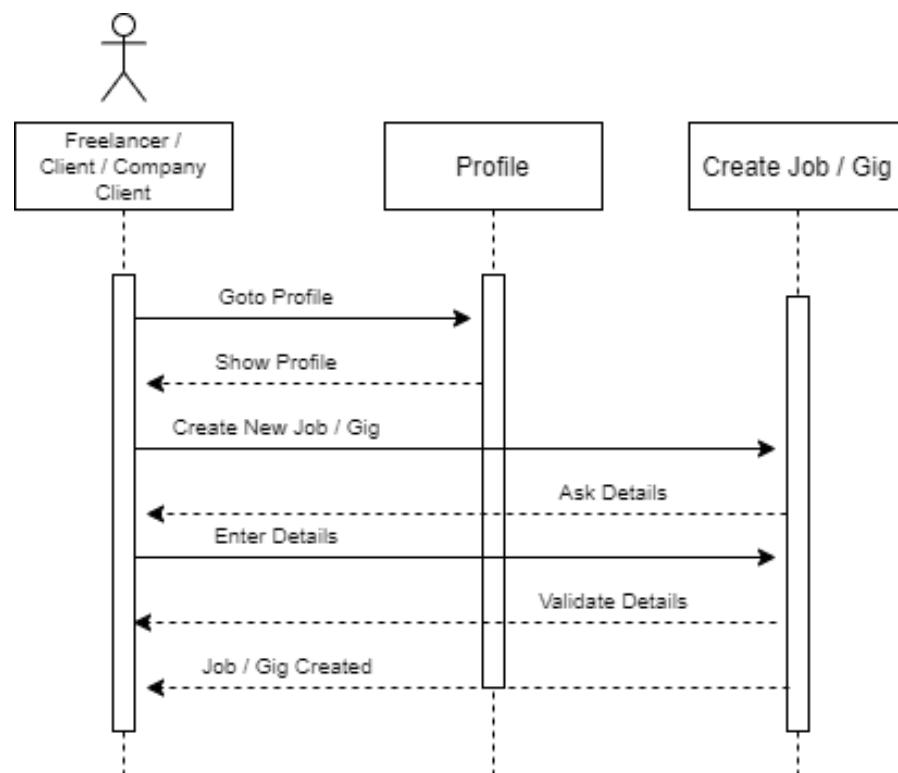
Diagrams

1. System Sequence Diagrams

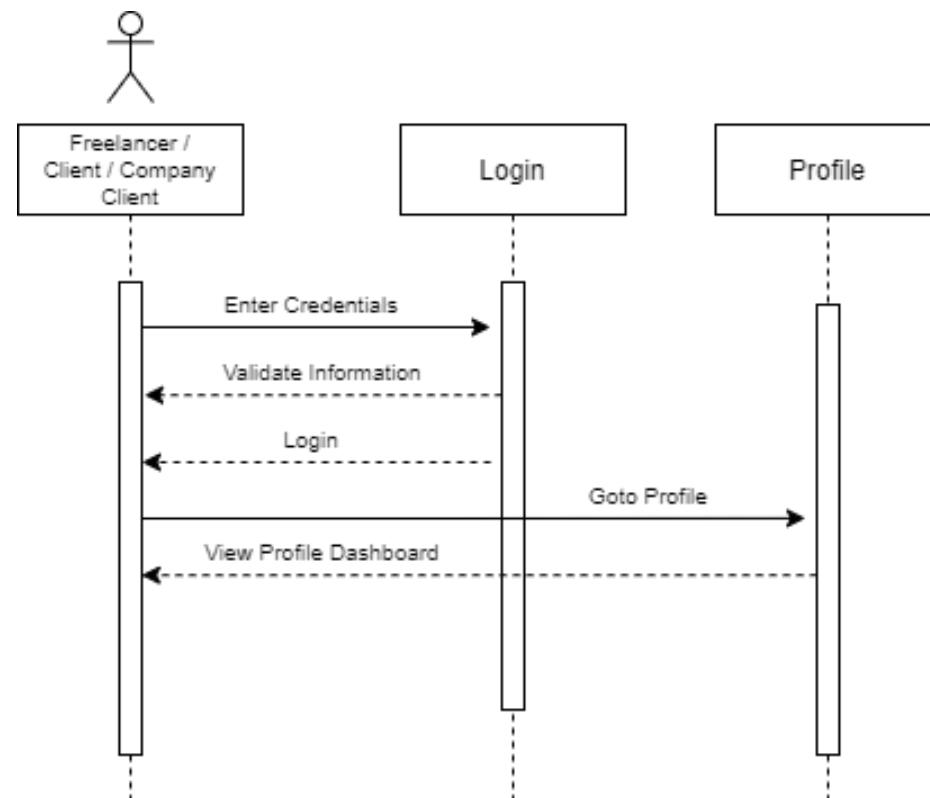
1.1 Freelancer Registration/Verification System



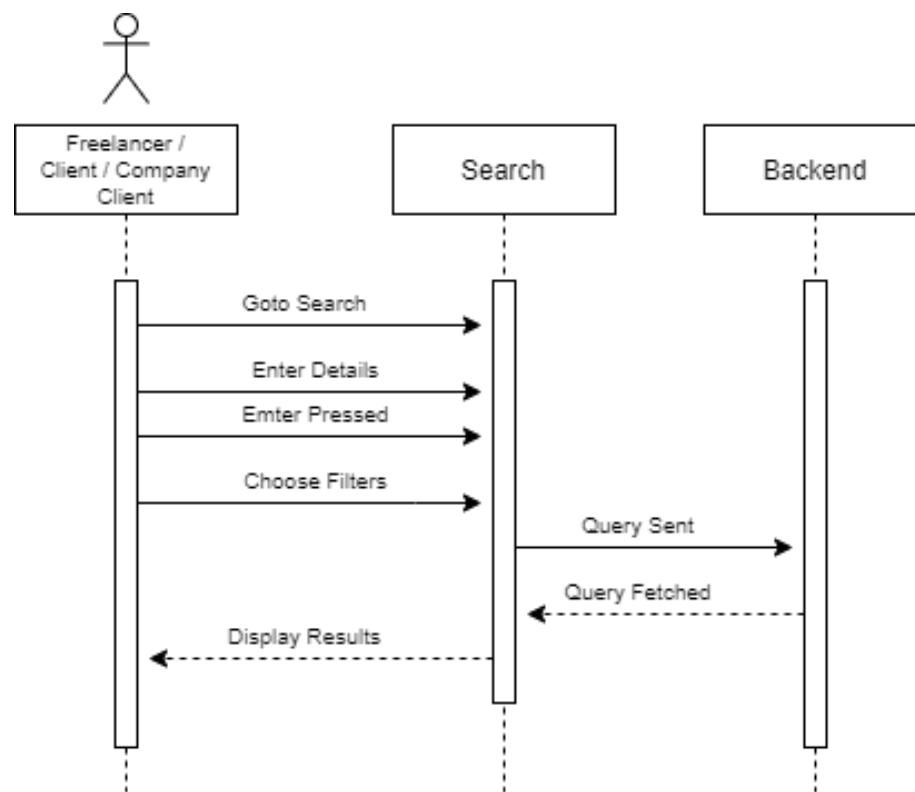
1.2 Gigs / Job Posting



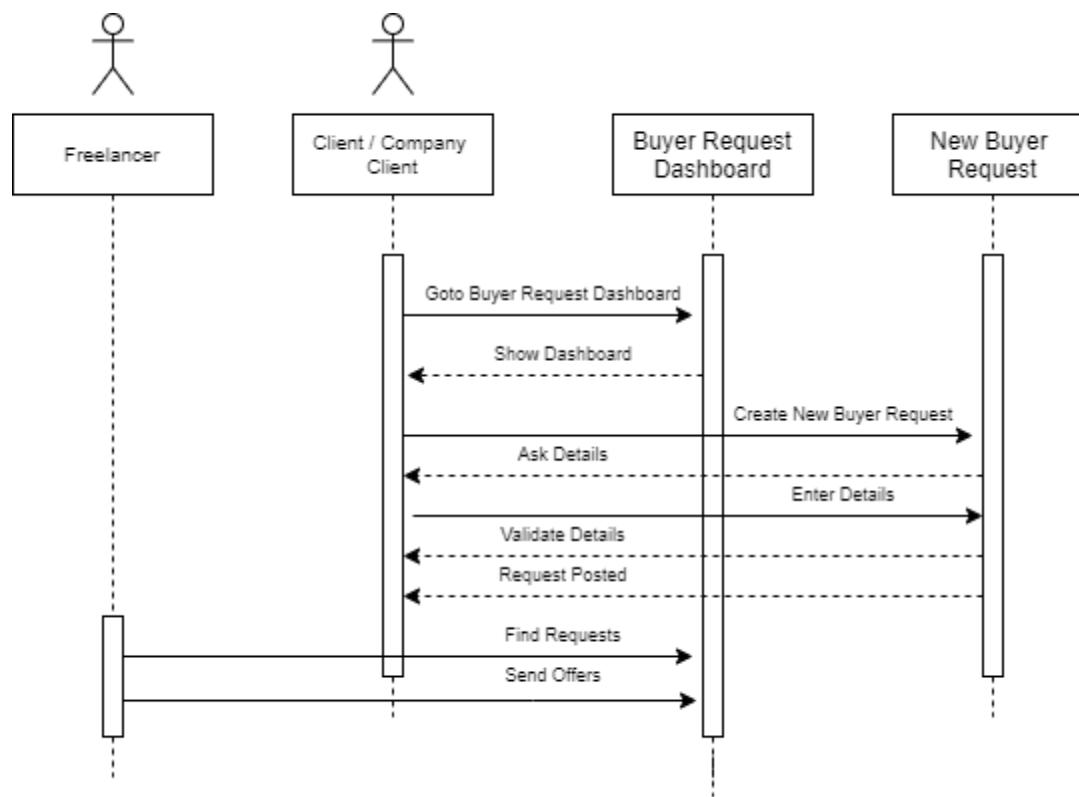
1.3 Profile Dashboard



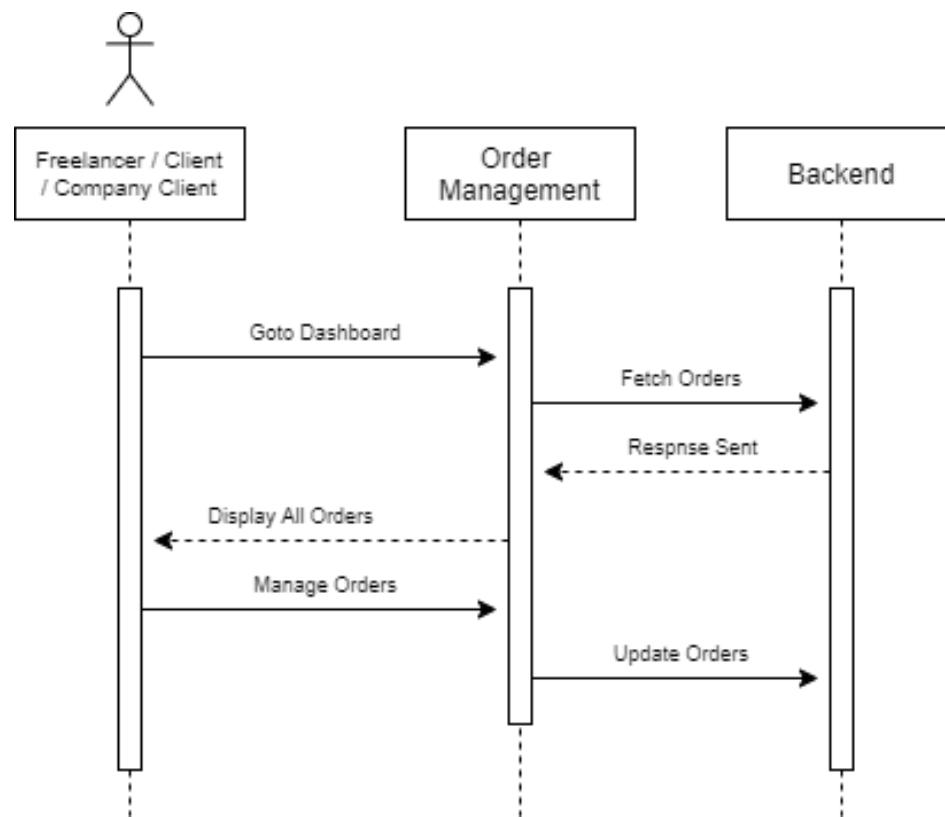
1.4 Search System



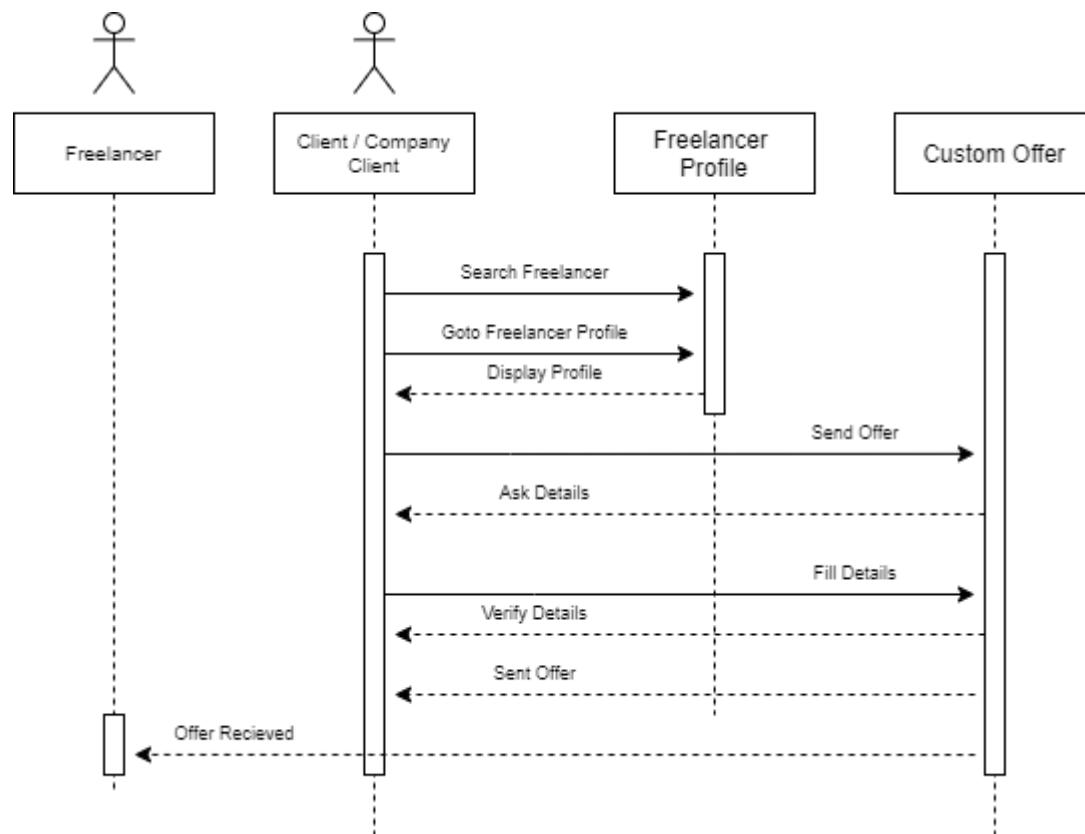
1.5 Buyer Requests



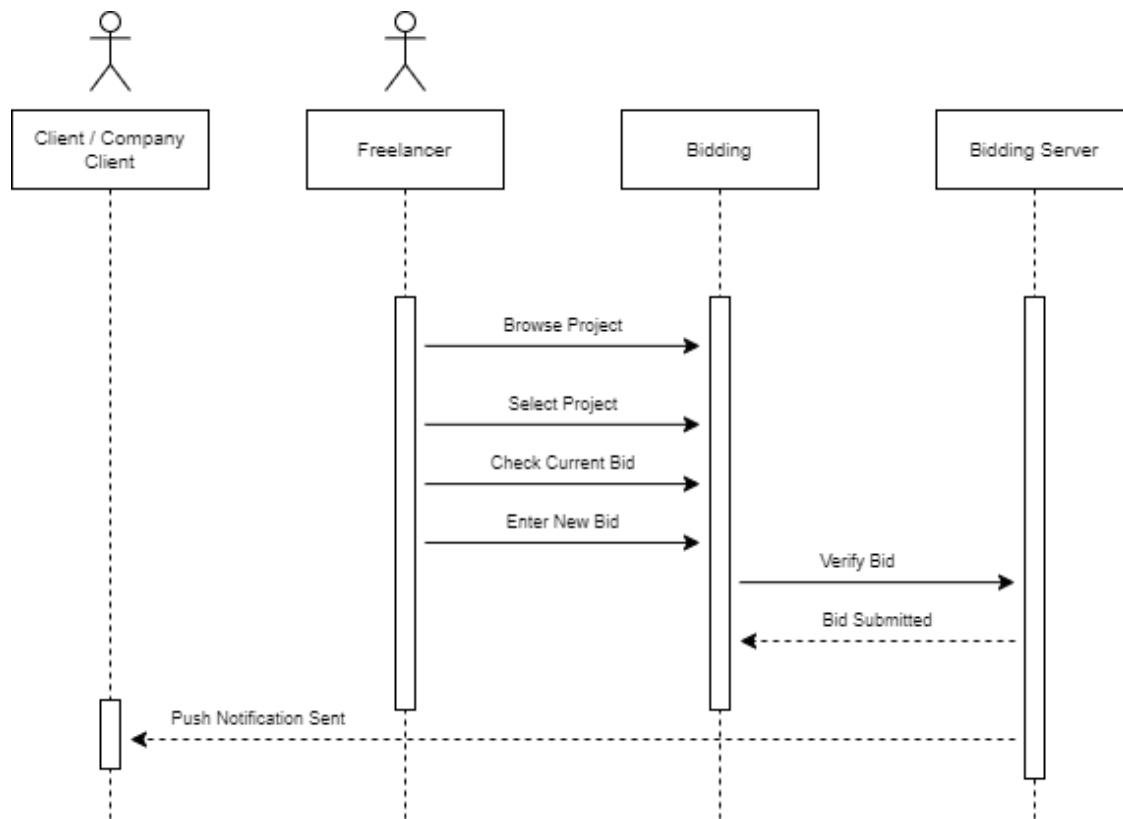
1.6 Order Management



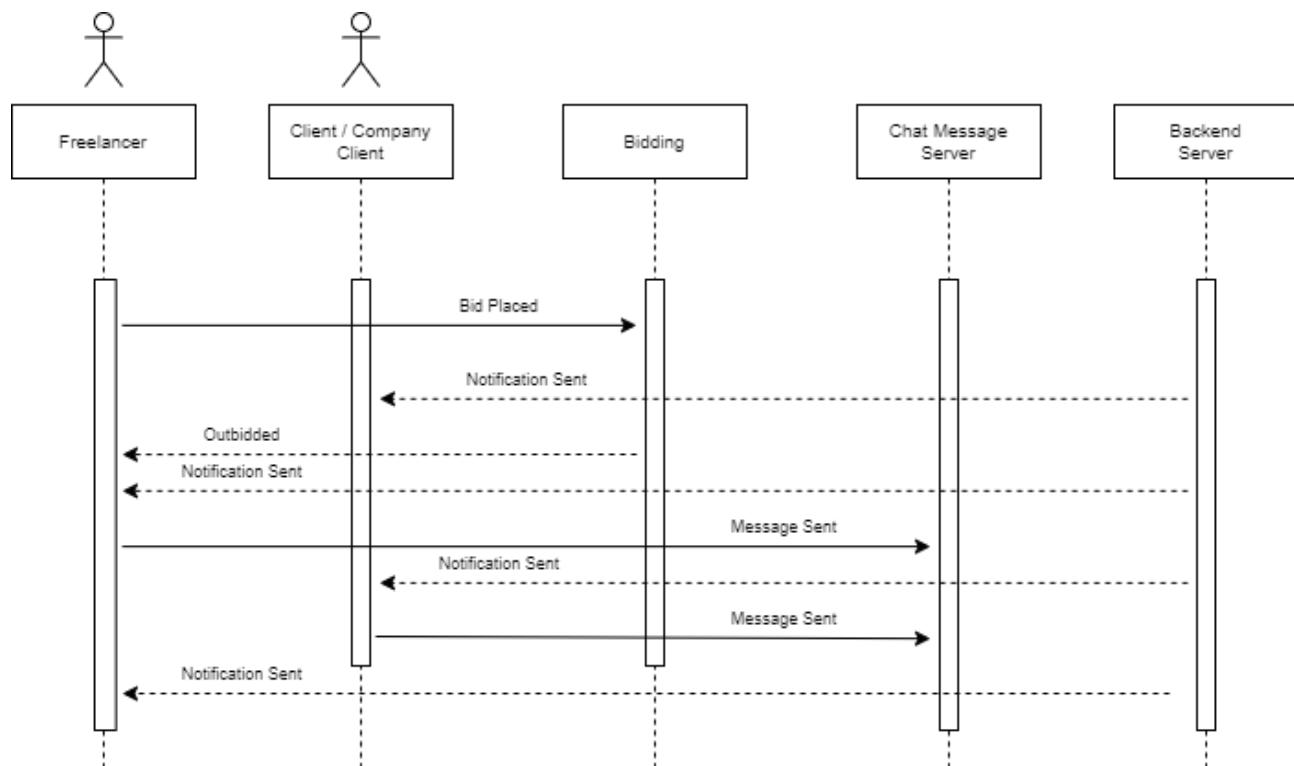
1.7 Custom Offers



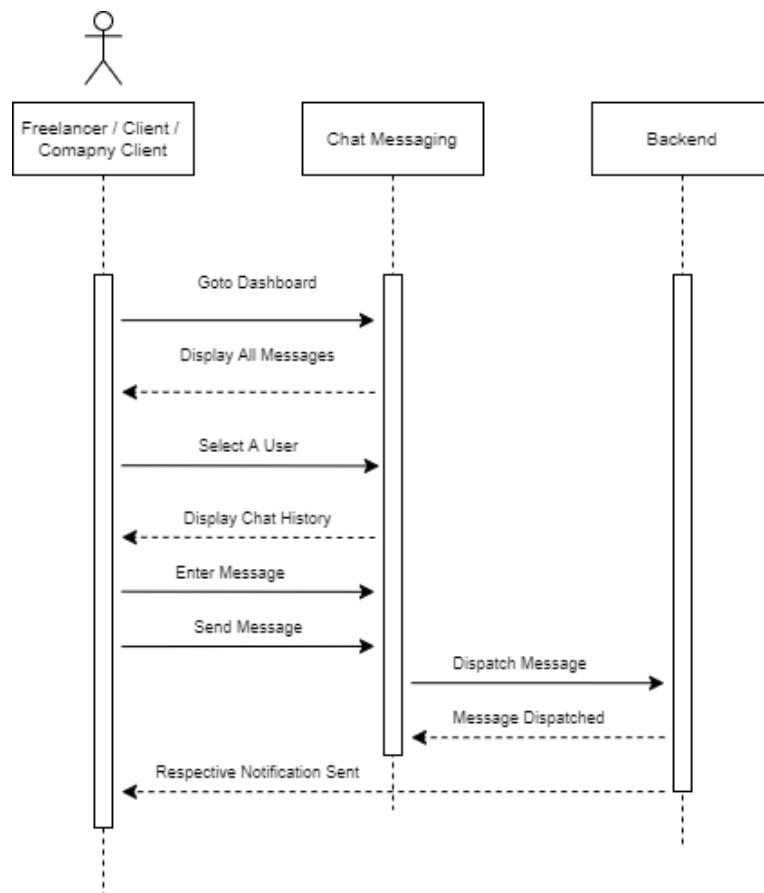
1.8 Bidding System



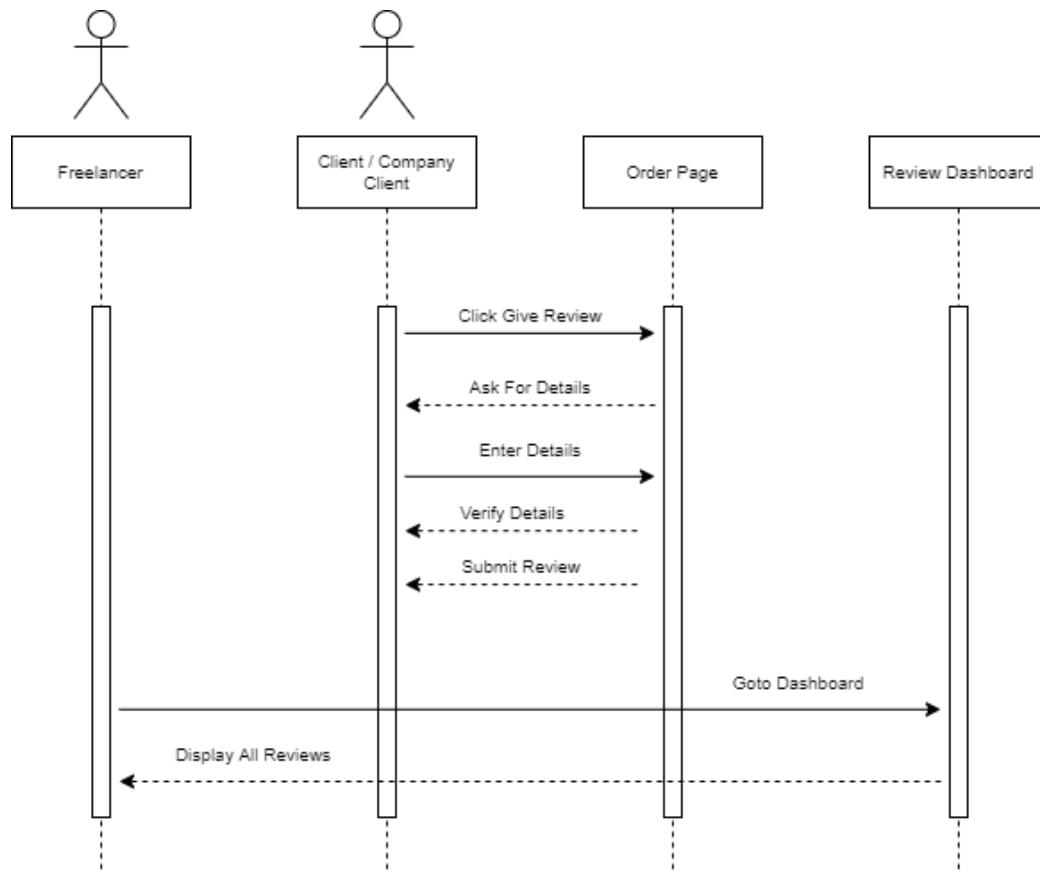
1.9 Push Notification System



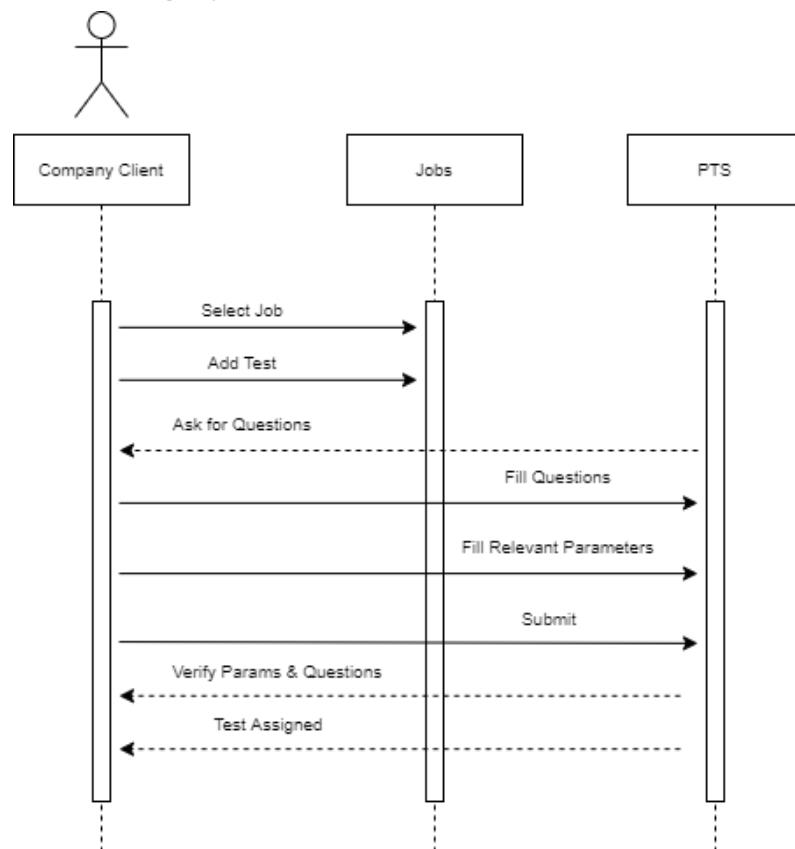
1.10 Chat Message System



1.11 Rating / Feedback System



1.12 Personalized Testing System



Software Design Specification

1. Introduction

1.1 Purpose of this document

The purpose of this document is mainly to break down the product into components to describe in detail what the purpose of each component is and how it will be implemented.

1.2 Scope of the development project

We aim to create a platform that targets freelancers with much easier policies and regional service charges. On the other hand, to tackle the issue of unprofessional and unskilled freelancers, we aim to provide a customized filtering process to ensure that only professional and skilled freelancers can be allowed to work on the platform. We also provide a customized test related to the job that is being bided by freelancers so that only matched freelancers can bid on outsourced projects by businesses, as this way we can tackle much of the filtering part we can offer to the clients. To tackle the issue of the order policy we will make sure that the freelancer has the choice to accept or decline customer job offers made by the client.

1.3 Definitions, acronyms, and abbreviations

Terms	Description
API	Application Programming Interface
ERD	Entity Relationship Diagram
SDS	Software Design Specification
SRS	Software Requirement Specification
SQL	Standardized Query Language
UI	User Interface

1.4 References

IEEE Template for System Design Specification Documents: <https://goo.gl/nsUFwy>

1.5 Overview of document

This System Design specification document contains the following information:

Section 1.0: An introduction to the document by explaining the purpose of the document, the scope providing terms and references, and a brief overview of each section within the document.

Section 2.0: An overview of the system architecture, system components and modules, relationships between the various components, and user interface issues without any technical details.

Section 3.0: A detailed description of each software component used in our product.

Section 4.0: A detailed description of the user interface including screen images.

Section 5.0: An explanation of the reusability of existing products and relationships within the product.

Section 6.0: A listing of the design decisions and tradeoffs made during the design phase. This section will help readers and users understand the reasoning for these decisions.

Section 7.0: Pseudo code implementation for the different algorithms used in the creation of neural networks and in its training and simulation.

Section 8.0: It contains an appendix.

2. System architecture description

This section provides an overview and rationale for the program's data and architectural design decisions.

2.1 Section Overview

This section describes the constraints and usability of the system. Data design are used to explain the major features of the system.

It describes the complete structure of the software and detail description of each individual features. It also describes the architecture design with major components of the software.

2.2 General Constraints

Pakwork, the main thing which is to be kept in mind is to provide flow in the performance of the system. The system will be easy to understand, easy to use, responsive, secure and the most important it will be so simple so that the user can use it smoothly. Website does not need any heavy performance computer normal system can also access this website; however, we do require an internet connection in order to use the website. The project is flexible and can modify any part of the project. This software will work on web platform. There is no more external software which is connected to this software directly or indirectly. The database of the website is placed on MySQL and MongoDB which requires internet connectivity to fetch and store data. The Mailing

System is based on Gmail so the SMTP service also requires an internet connectivity for its communication.

➤ **Hardware Interface:**

Since the tool has no designated hardware and it can run on any computing machine it does not have any direct hardware interface. Although for optimal performance and faster results the user is required to use a high-speed internet.

➤ **Software Interface:**

The software interfaces that are used in the tool are as follows:

Microsoft Visual Studio Code: Microsoft Visual Studio Code is a lightweight text editor that is widely-used for any particular programming task, mainly it is used for web development. Due to its performance many developers consider it as their daily go-to text editor when they're programming scripts or applications.

MySQL Workbench: MySQL is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network.

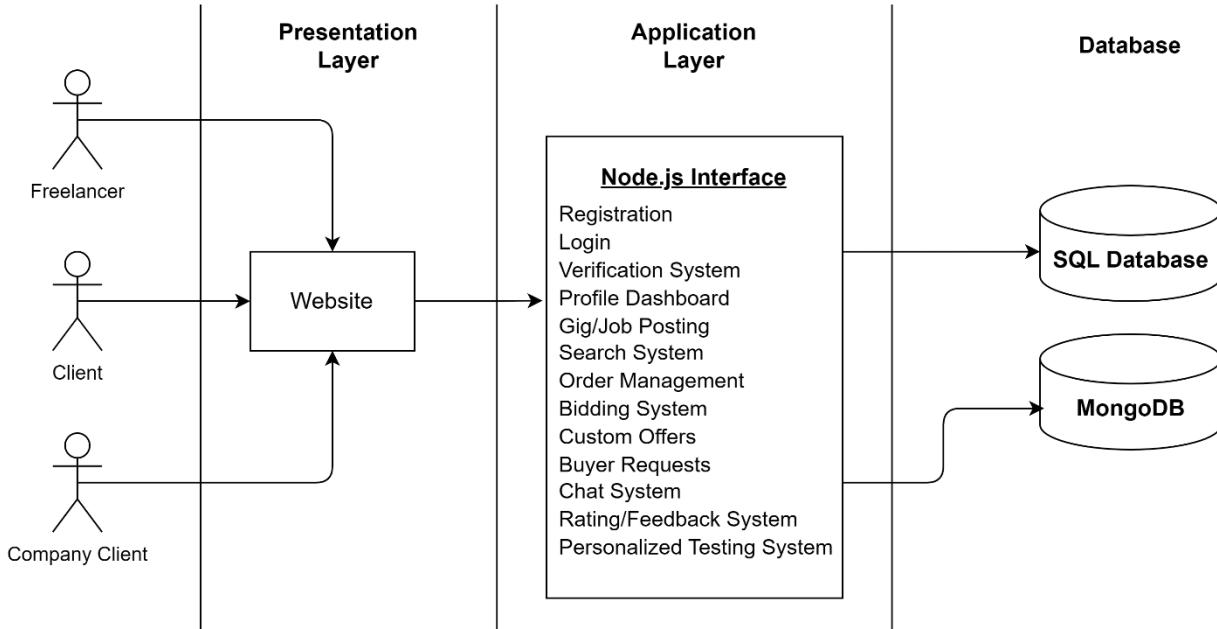
MongoDB: MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

Communication Interface: The communication between the different parts of the system is important since they depend on each other. However, the most important communication happening in the website is between web-based interface used by the user to communicate with the system and that is locally done.

2.3 Data Design

- Reference to: SRS Appendix B: Entity Relation Diagram, Domain Model

2.4 Program Structure



The major functions the product must perform are listed as follows:

- Freelancer, Client and Company Client would be able to register themselves on the platform
- Freelancer, Client and Company Client would be able to log on to the platform
- Freelancer, Client and Company Client would be able to customize their profile
- Freelancer would be able submit verification documents for verification.
- Freelancer would be able to create, edit or delete Gigs
- Client and Company Client would be able to create, edit or delete custom jobs, requests and offers
- Freelancer, Client and Company Client would be able to chat with each other.
- Freelancer would be able to accept offers sent by Client or Company Client.
- Client and Company Client would be able to start a bid on their project.
- Freelancer would be able to place bids on the projects.
- Freelancer would be able to search for active jobs.
- Client and Company Client would be able to accept the order delivered by Freelancers.
- Freelancers would be able to deliver the order placed by the Client or Company Client.
- Client and Company Client would be able to leave feedback and rate the order.
- Company Client would be able to create any personalized test for the Freelancers.

2.5 Alternatives Considered

N/A

3. Detailed description of components

This section represents the meat of your document. Be as detailed as time allows.

3.1 Section Overview

This section describes the complete description of all components in detail with their function's dependencies, their relationships with other components. It describes the interface of each component with input and output requirements.

3.2 Component of Gigs & Jobs Feature:

3.2.1 Description:

Creating Gigs and Jobs is the most important feature of our application because that's how the whole process of freelancing would work out, the freelancer makes their gig and the buyer puts out a job that matches the suitable gigs.

3.2.2 Data Members:

- Include Type:** Gigs allow clients to see potential freelancers for their work according to their gig description and jobs allow freelancers to match their appropriate buyer according to their job description
- Visibility:** Gigs are publicly viewable for all while jobs are only available to freelancers and clients who posted that job.
- Description:** Freelancer can make gigs and clients can view them and communicate with them, clients can post jobs and freelancers can view them and send particular offer to the client.

3.2.3 Methods

Identification	The Gigs/Jobs component can be accessed by the freelancers and clients to interact on the platform. It is located on the dashboard for the appropriate user
Type	It is a module having several sub components
Purpose	The freelancer makes their gig and the buyer puts out a job that matches the suitable gigs.
Function	Freelancer can make gigs and clients can view them and communicate with them, clients can post jobs and freelancers can view them and send particular offer to the client.
Subordinates	Gig id, Job id and username are used to differentiate between gigs and jobs creators
Dependencies	This component depends on the user provided details and further used in different ways in the system. User can use this component on website.
Interfaces	The user interface of course feature is situated in the user profile page. SQL queries are used to connect the interfaces. And the code which is running on back-

	end implements sessions which means user session will maintain automatically and user can view data according to the gig id or job id.
Resources	Database access and External APIs permissions is required to run this component.
Processing	Gigs and Jobs will be created and freelancers, clients can interact with them.
Data	This Component will store related Gigs and Jobs data inside the database.

3.3 Component of Order Management and Rating/Feedback:

3.3.1 Description

Order management and Rating/Feedback are one of the most important features in our application because with their interaction the client and freelancer can have a secured business on the platform without any 3rd Party involvement.

3.3.2 Data Members

- **Include type**

Clients will place an order on a particular offer made by the freelancer and the order will be started. Freelancers would have to deliver the order by sending the necessary content within the time or else they will be penalized for that. After the delivery the client has to accept the delivery or request for changes and leave feedback once the order is completed with a rating as well.

- **Visibility:**

Order management is only visible to the freelancer and the client, the rating system will be available to the client while the freelancer can only view the given rating and feedback.

- **Description**

Clients can place the order from the offers sent by the freelancers and start the order and mark it complete once it's delivered by the freelancer.

3.3.3 Methods

Identification	The order component can be accessed by the freelancers and clients to interact on the platform. It is located on the dashboard for the appropriate user and the rating/feedback component will be appeared after the completion of the order
Type	It is a module having several sub components
Purpose	Clients can place the order from the offers sent by the freelancers and start the order and mark it complete once it's delivered by the freelancer.
Function	Clients will place an order on a particular offer made by the freelancer and the order will be started. Freelancers would have to deliver the order by sending the necessary content within the time or else they will be penalized for that. After the delivery the client has to accept the delivery or request for changes and leave feedback once the order is completed with a rating as well.
Subordinates	Order id and usernames are used to differentiate between different orders.
Dependencies	The Order Component is dependent on the freelancer's given offer and the feedback/rating component is dependent on the order that has been completed.

Interfaces	The user interface of the order component is accessible from the dashboard and then clicking a particular order and the rating/feedback component will be displayed on that order if the client is viewing.
Resources	Database access and External APIs permissions is required to run this component.
Processing	Orders and Rating/Feedback will be created and freelancers, clients can interact with them.
Data	This Component will store related Order and Feedback/Rating data inside the database.

4. User Interface Design

4.1 Section Overview

This section provides a view into the set of standards which we have made use of to build a user-friendly Pakwork Website. They have been applied in the best possible way to make the interface appear pleasing to the user and not seem too difficult to understand at the same time.

4.2 Interface Design Rules

Component Reusability in React.js is to design all web interfaces. Html, CSS and bootstrapping are used to design all website pages. There is no specific standard followed for designing this website. Users should always be informed of system operations with easy to understand and highly visible status displayed on the screen within a reasonable amount of time. Interface designers should ensure that both the graphic elements and terminology are maintained across similar platforms. For example, an icon that represents one category or concept should not represent a different concept when used on a different screen. Designers should assume users are unable to understand technical terminology, therefore, error messages should almost always be expressed in plain language to ensure nothing gets lost in translation.

4.3 GUI Components

Pakwork is not planning to use any external APIs for its functionality and plans on building most of its APIs internally. As for its GUI components are concerned, they're purely built on React, while the email-templates and external verification links are rendered html directly from node-mailer and express-html.

4.4 Detailed Description

Registration: Create your account on the platform (freelancer, client, company client)

Login: Sign-in to your accounts (one login for all types of users)

Dashboard: Show profile detail of the user that is logged in according to their user type.

Profile Management: Update Profile Details such as profile picture, description, social links.

Verification System: Upload Documents to verify as a freelancer on the platform.

Gigs Page: Collection of Gigs according to the given searched text.

Gig Detail Page: Show Detail of the selected gig.

Jobs Page: Collection of Jobs according to the given searched text.

Job Detail Panel: Show Detail of the selected job.

Manage Job Panel: Show appropriate options to create, update a job.

Manage Gig Panel: Show appropriate options to create, update a gig.

Search Box: Search for any type of gigs available.

Manage Orders Page: Show details of the orders along with each's status.

Order Page: Show details of a particular order with appropriate options.

Chat Inbox Message: Interactive chat conversations with specific users.

Live Bidding Page: Shows all active live biddings that are happening.

Live Bidding Room: Show details of a particular bidding with options to interact.

Custom Offer Panel: Create a custom offer with appropriate options.

Buyer Requests Panel: Create a buyer request with appropriate options.

- For Screenshots refer to appendix in the Software Requirement Specifications

5. Reuse and relationships to other products

Reusability allows developers to be more efficient because the same code can be developed once and used in many different applications. Second, reliability can be improved by reusing previously developed, and previously tested, components. The development of new code entails additional costs in time and money for testing, validation, and verification. Much of these expenses can be avoided by using already manufactured external components.

6. Design decision and tradeoffs

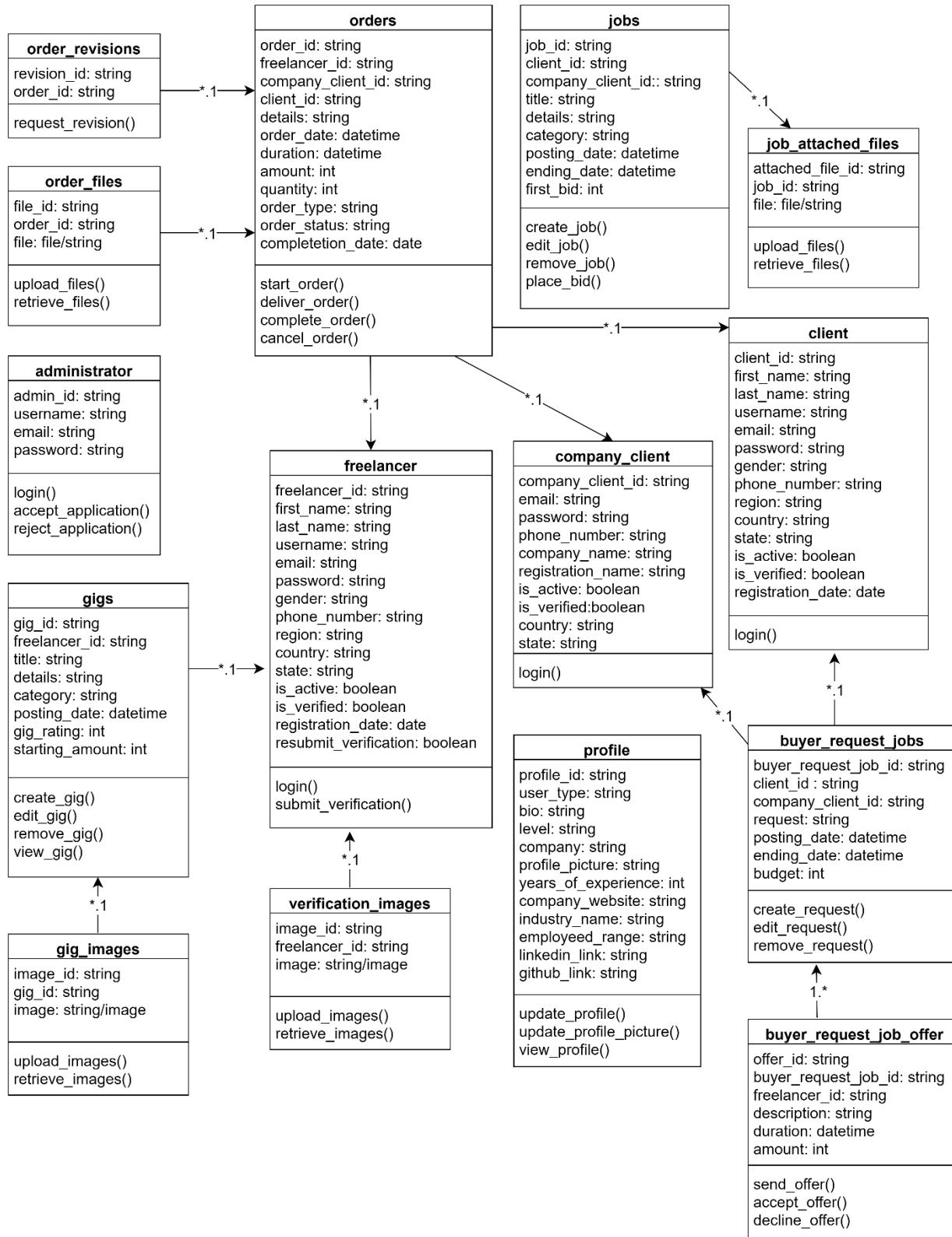
N/A

7. Pseudocode for components

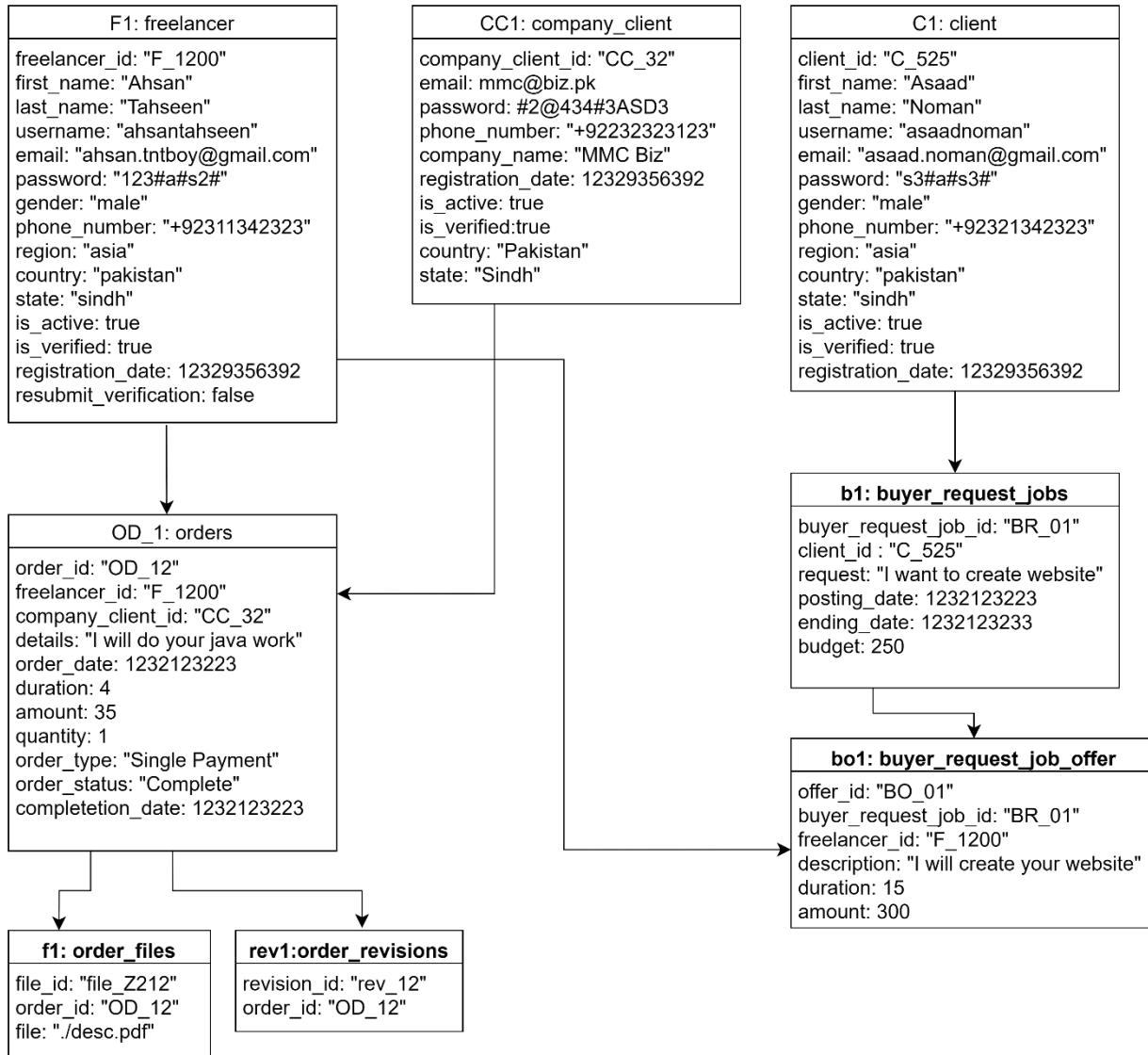
N/A

8. Appendices

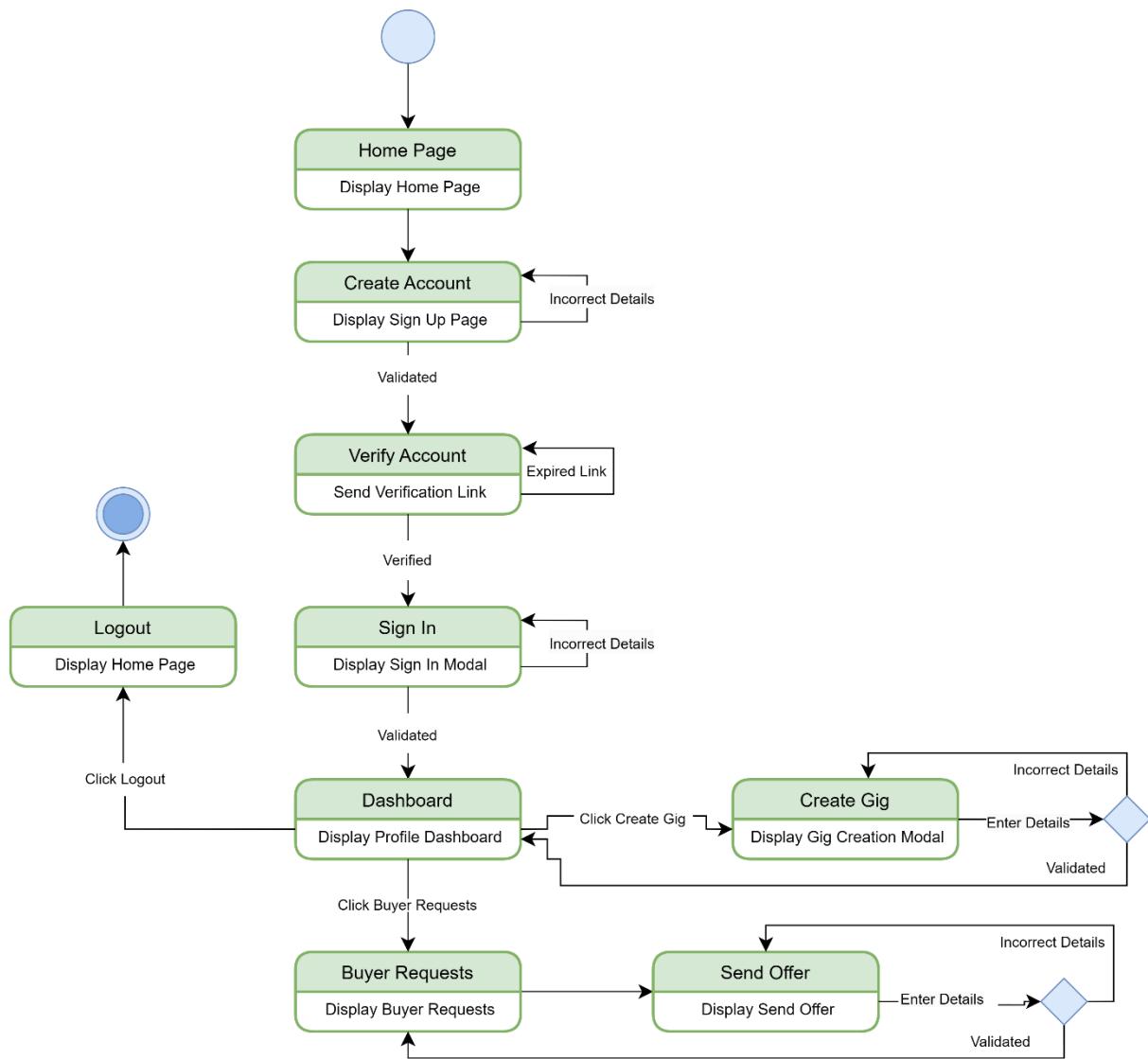
8.1 Class Diagram



8.2 Object Diagram

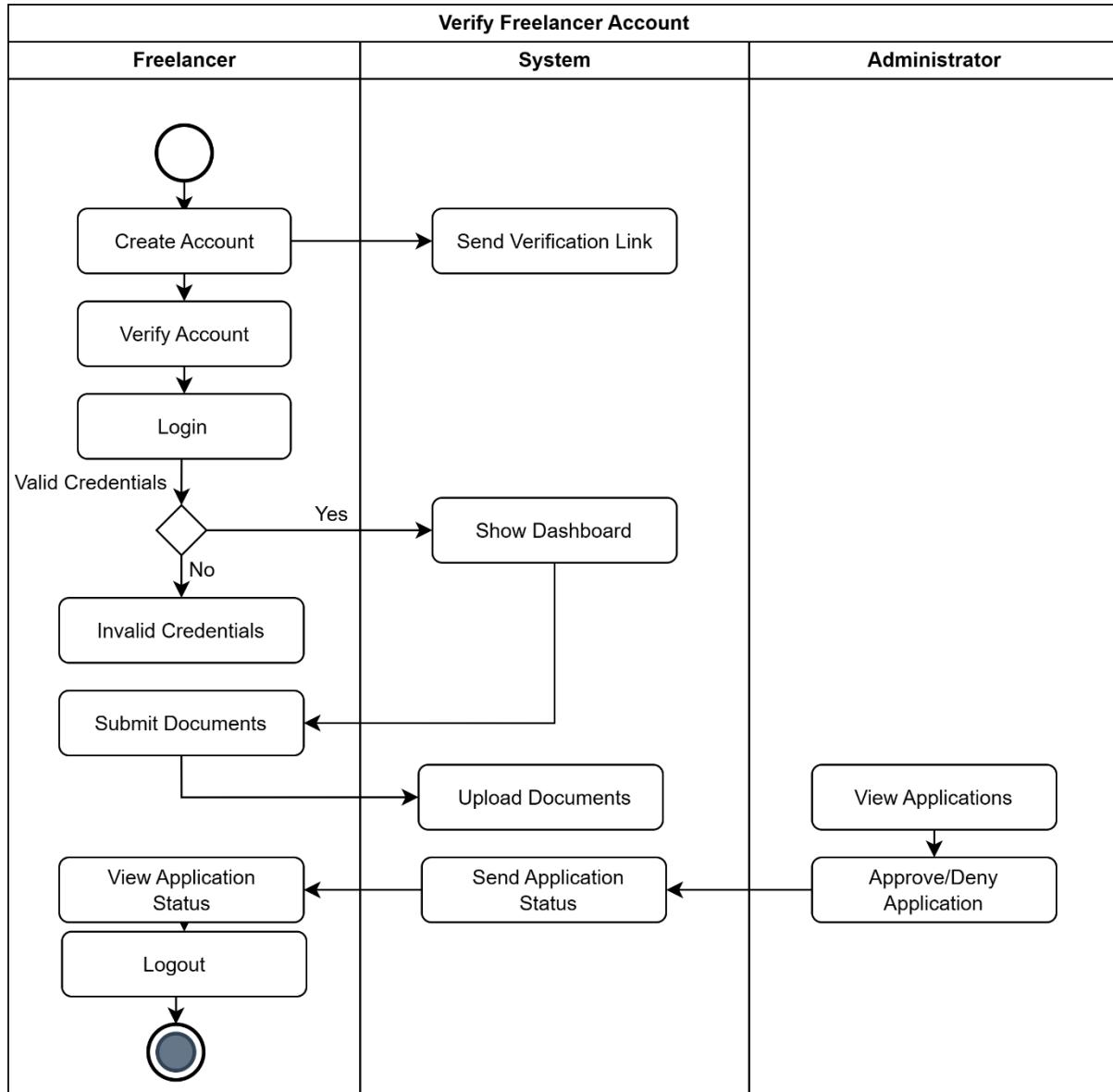


8.3 Statechart Diagram

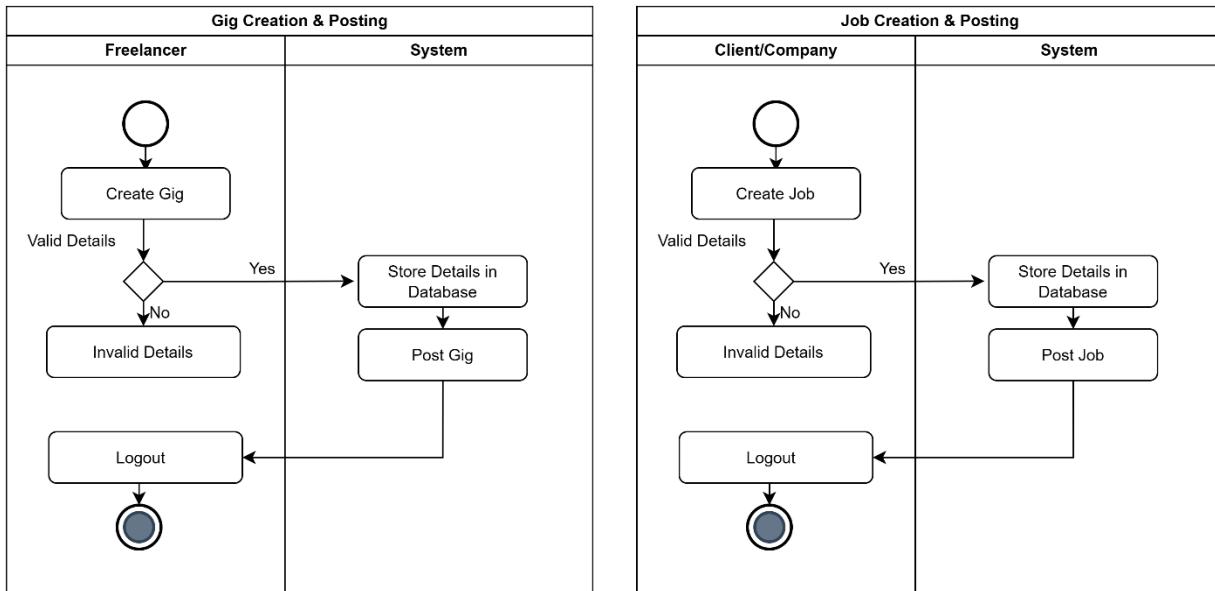


8.4 Activity Diagram

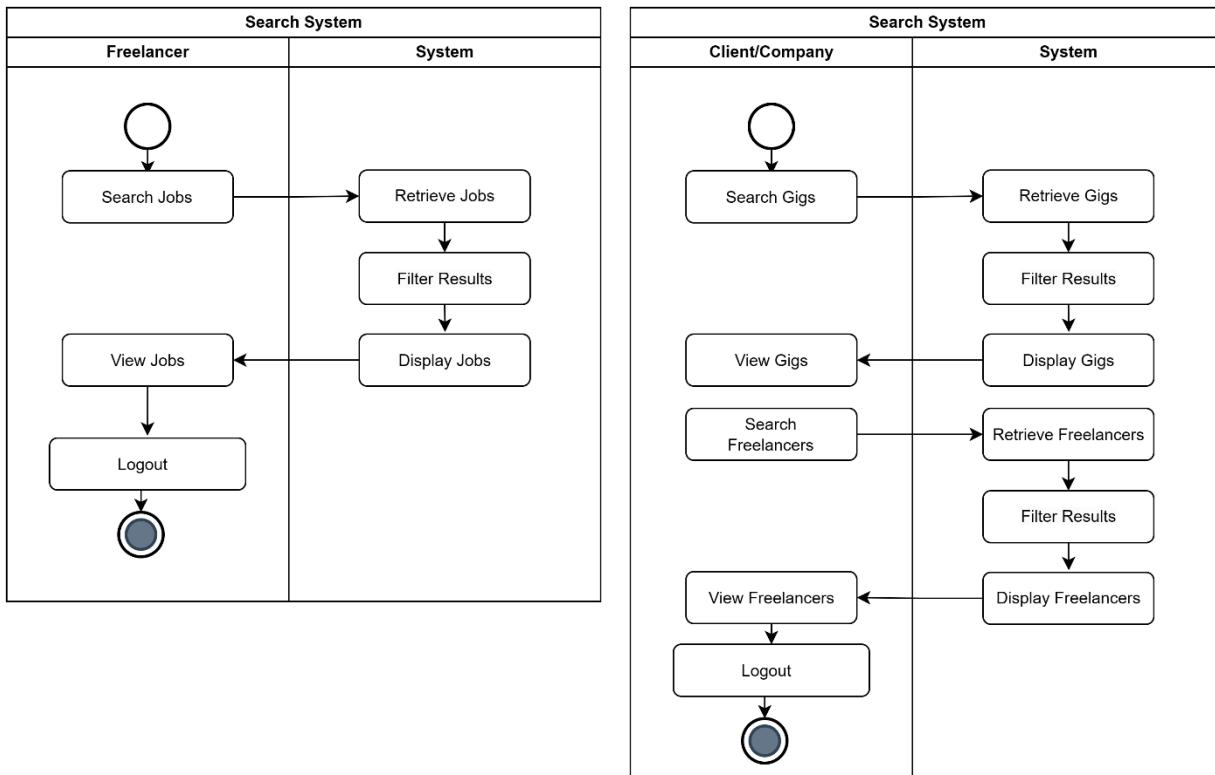
8.4.1 Freelancer Account Verification



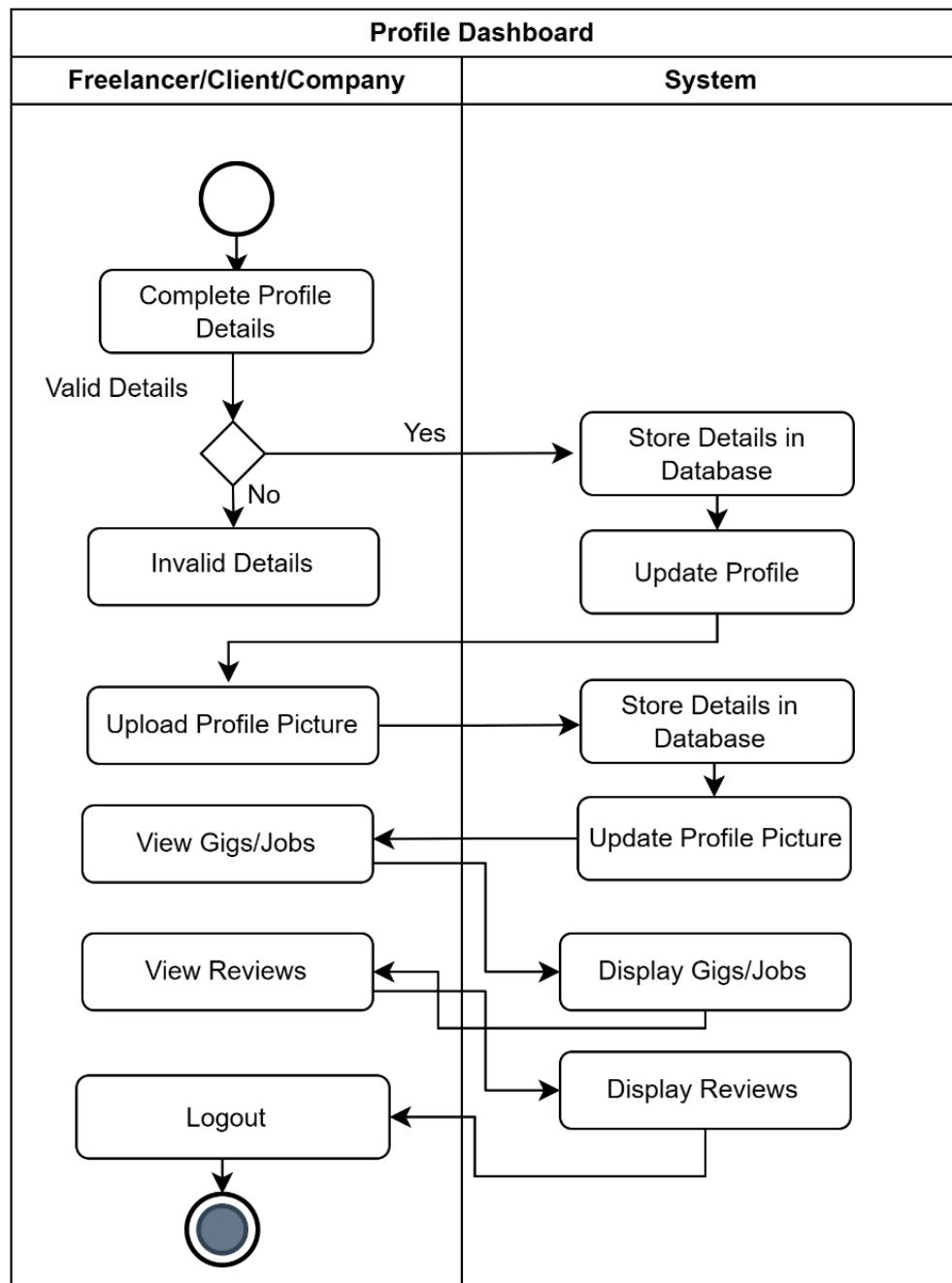
8.4.2 Gig / Job Creation & Posting



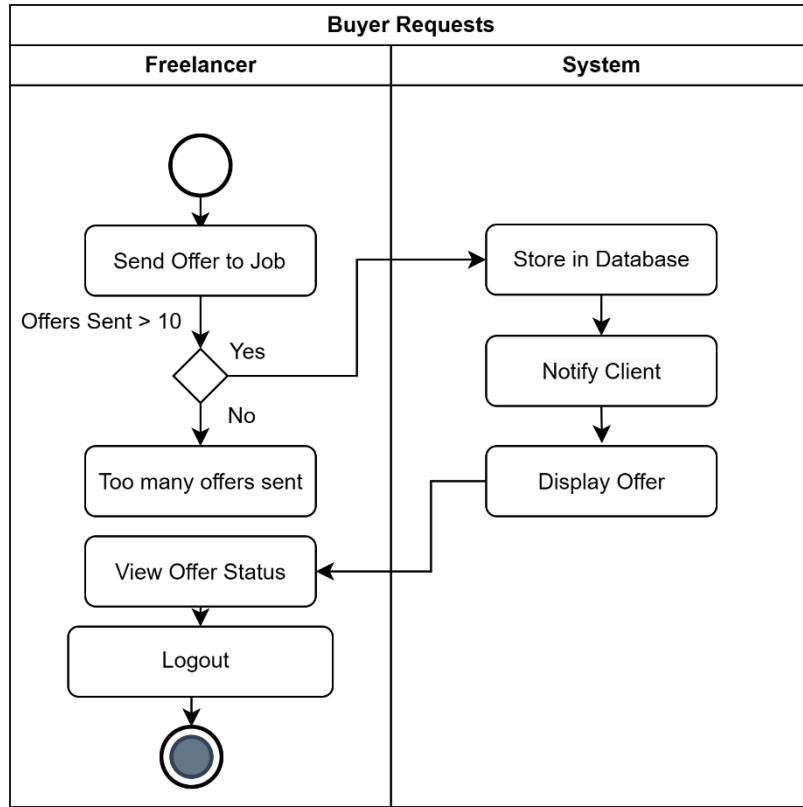
8.4.3 Search System



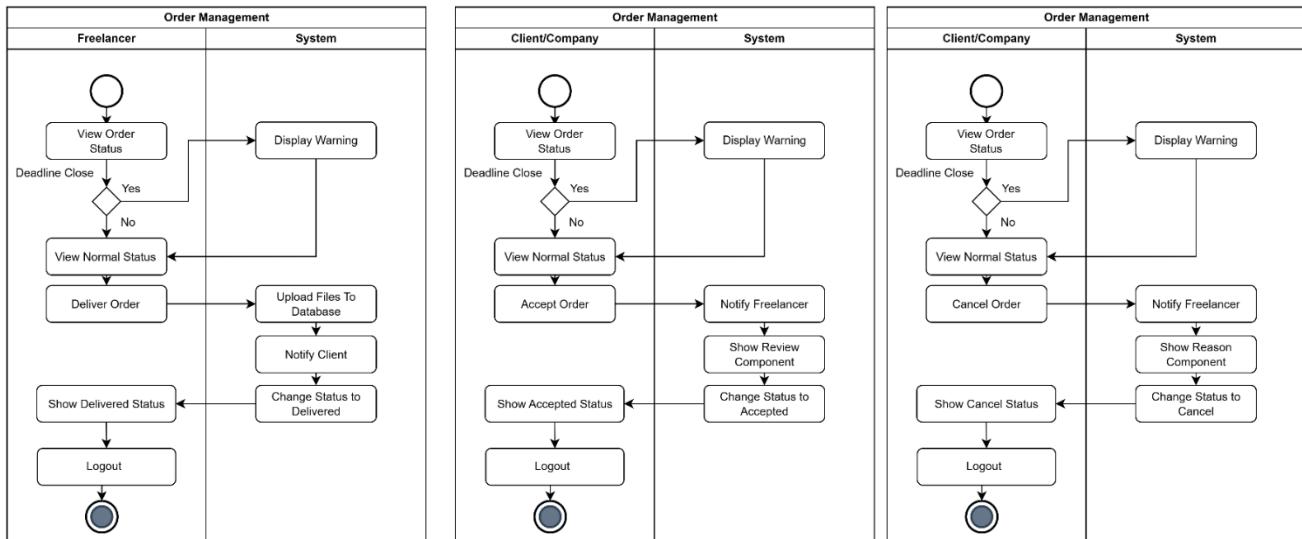
8.4.4 Profile Dashboard



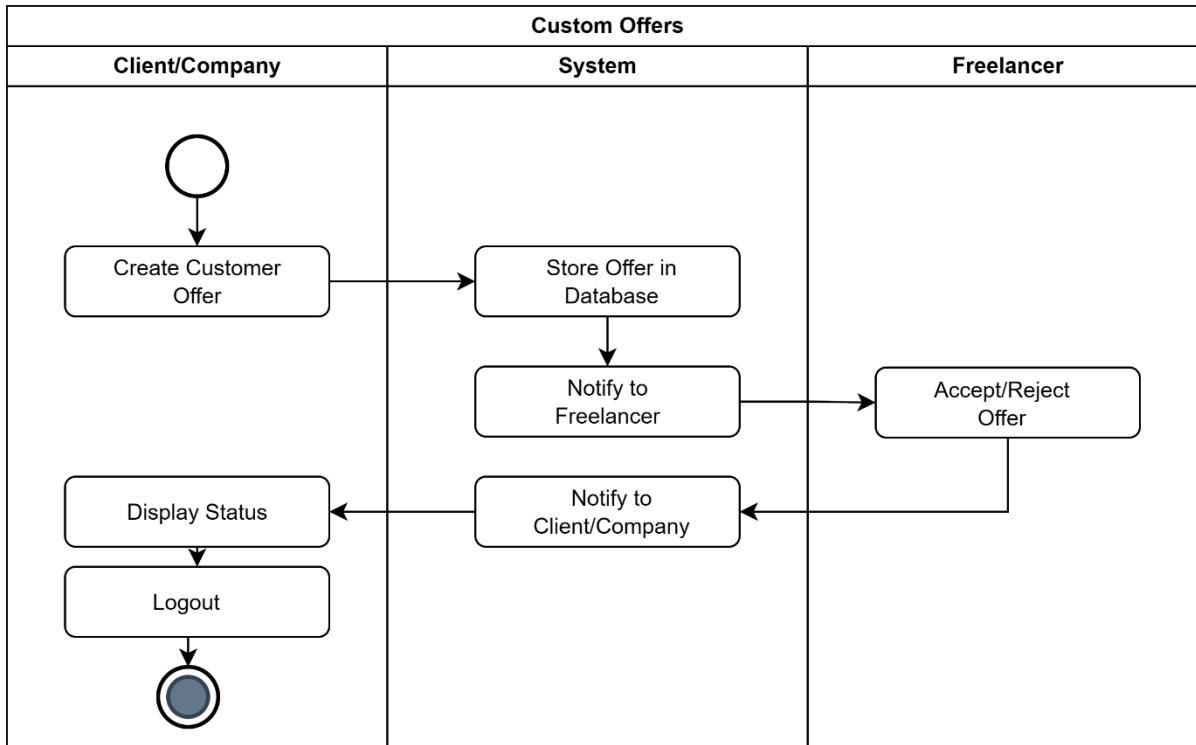
8.4.5 Buyer Request



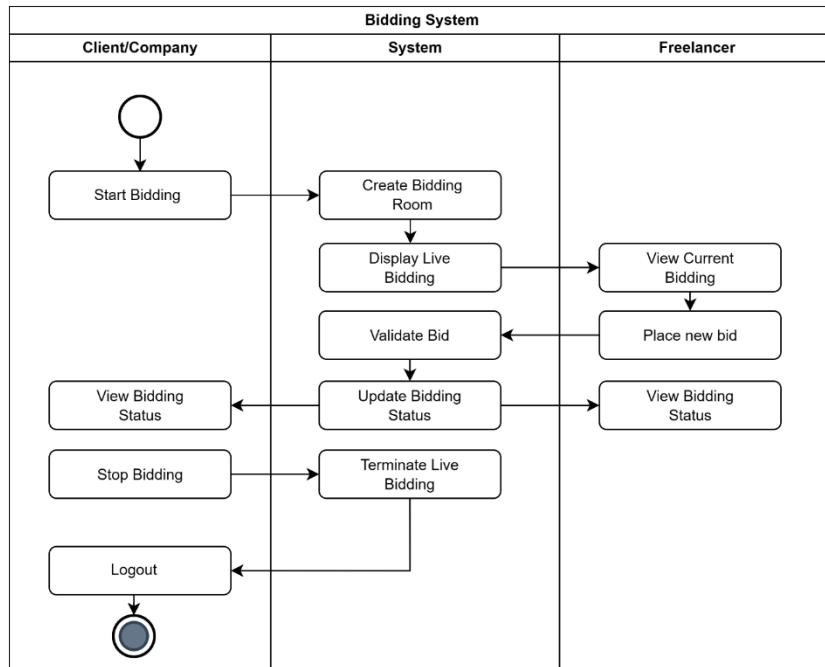
8.4.6 Order Management



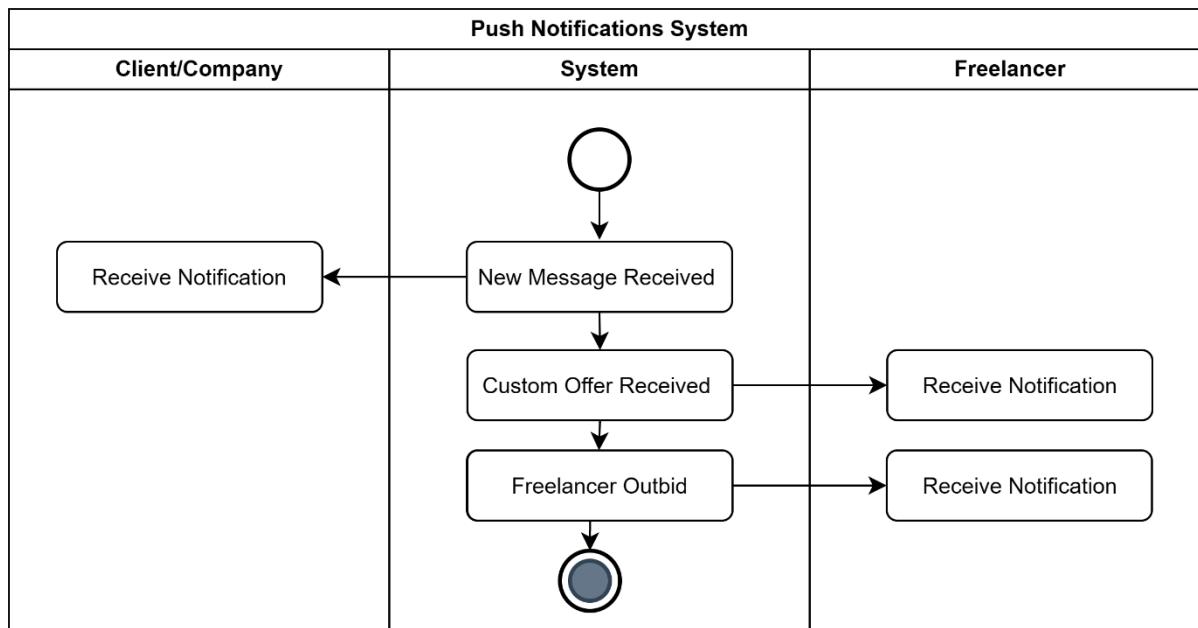
8.4.7 Custom Offers



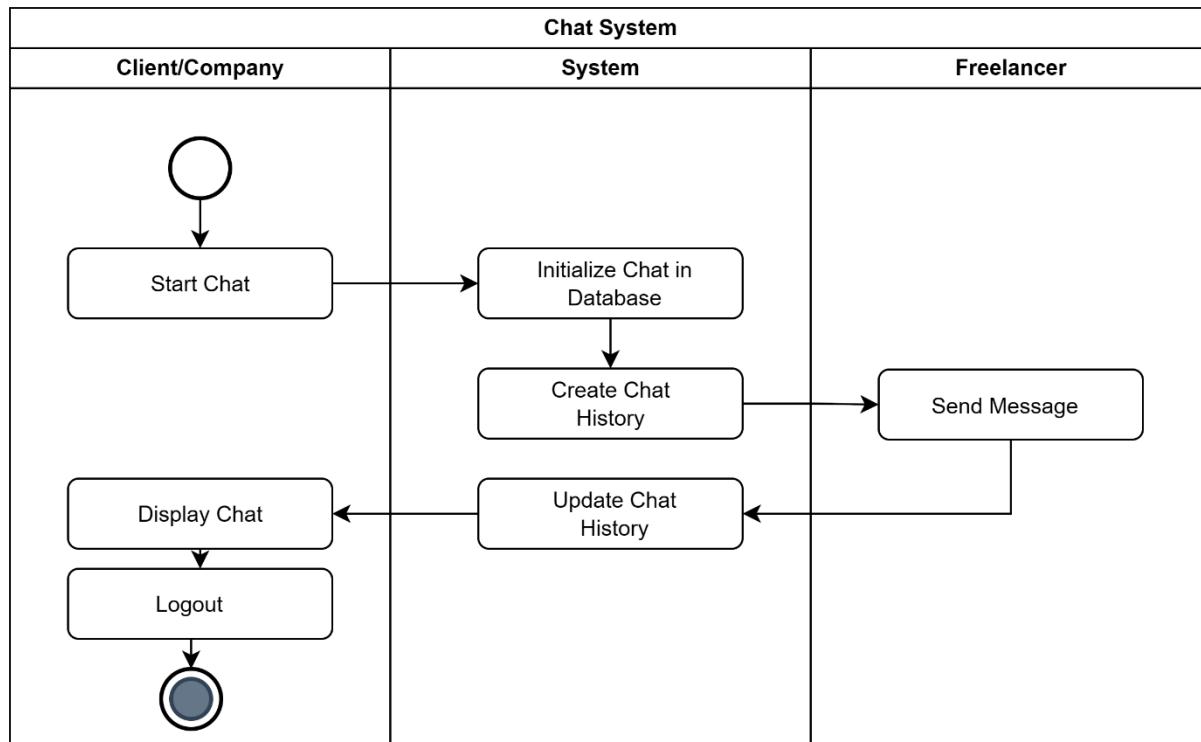
8.4.8 Bidding System



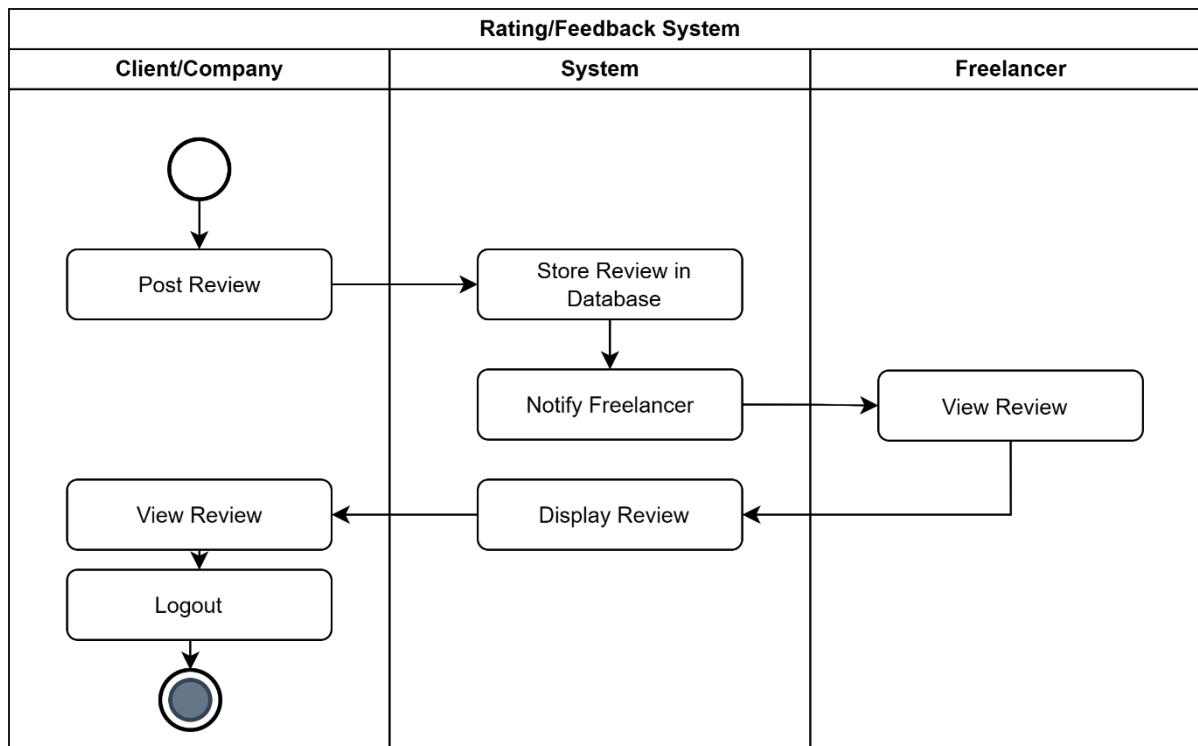
8.4.9 Push Notification System



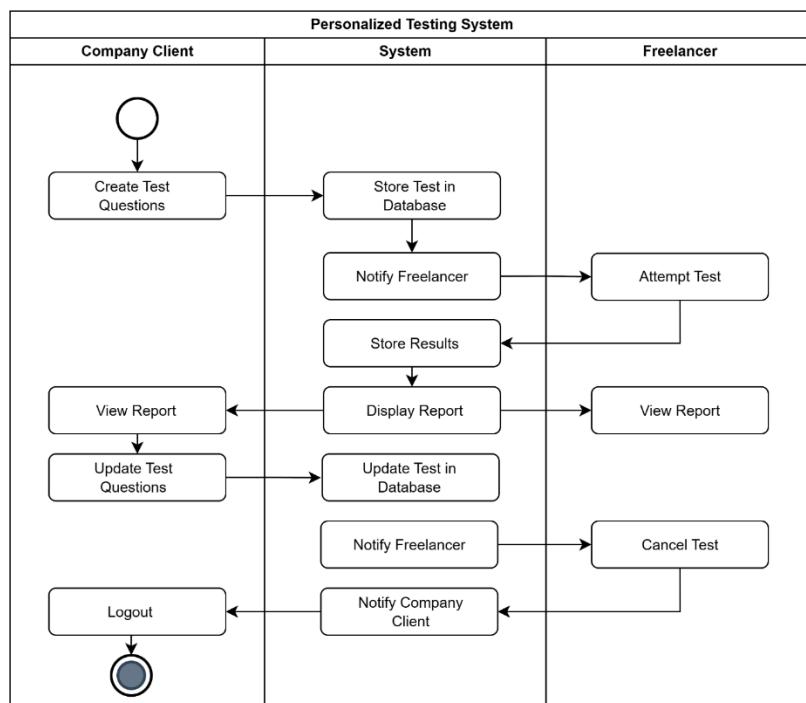
8.4.10 Chat System



8.4.11 Rating/Feedback System



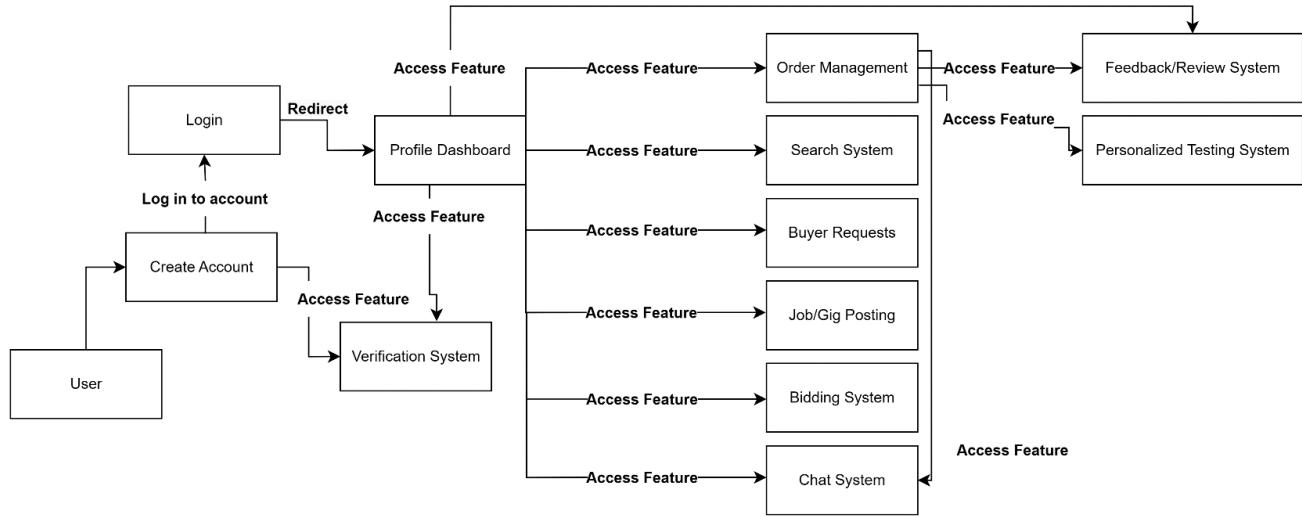
8.4.12 Personalized Testing System



8.5 Sequence Diagram

Refer to Appendix in System Requirements Specification or Page 68

8.6 Collaboration Diagram



8.7 Use Case Diagrams

Refer to Appendix in System Requirements Specification

8.8 Component Diagram

