DESIGN AND IMPLEMENTATION OF FREELANCE MARKETPLACE

LAWAND HASSAN MAHMOD

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : LAWAND HASSAN MAHMOD

Date of Birth : 14 July 2000

Title : Design and Implementation of freelance marketplace

Academic Session : 2022-2023/7

I declare that this thesis is classified as:

☐ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

☐ **RESTRICTED** (Contains restricted information as specified by the organization where research was done)*

☑ **OPEN ACCESS** I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

Twana H. Shareef

SIGNATURE OF STUDENT | **SIGNATURE OF SUPERVISOR**

QU182SCSJ102 | MR. TWANA HUSSEIN

MATRIX NUMBER | **NAME OF SUPERVISOR**

Date: 26 JULY 2012 | Date: 26 JULY 2022

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

May 2023

QIU Library

Sir,

CLASSIFICATION OF THESIS AS OPEN
DESIGN AND IMPLEMENTATION OF FREELANCE MARKETPLACE
LAWAND HASSAN MAHMOOD

Please be informed that the above-mentioned thesis entitled “DESIGN AND IMPLEMENTATION OF FREELANCE MARKETPLACE” be classified as OPEN ACCESS.

Thank you.

Sincerely yours.

TWANA HUSSEIN SHAREEF, As Sulaymaniyah Iraq, +964 770 197 4090

“I hereby declare that we have read this thesis and in my opinion this thesis is suffcient in term of scope and quality for the award of the degree of BSc of Computer Science (Software Engineering)”

Signature	:	Twana H. Shareef
Name of Supervisor	:	TWANA HUSSEIN SHAREEF
Date	:	29 JUNE 2022

DESIGN AND IMPLEMENTATION OF FREELANCE MARKETPLACE

LAWAND HASSAN MAHMOD

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JUNE 2022

DECLARATION

I declare that this thesis entitled *"DESIGN AND IMPLEMENTATION OF THE FREELANCE MARKETPLACE"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	.....
Name	:	LAWAND HASSAN MAHMOOD
Date	:	29 JUNE 2022

DEDICATION

This thesis is dedicated to my lovely parents, who had my back and supported me through every phase of my life, and anything good that happened to my life has been because of their example, guidance and, love.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Mr. Twana Hussein, for his encouragement, guidance, critics, and friendship. Without the continued support and interest, this thesis would not have been the same as presented here.

My fellow student should also be recognized for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance on various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members.

ABSTRACT

The purpose of this study is to investigate the development of creating a marketplace that combines both clients who are searching for someone to work for them online or freelancers who are searching for a client to make money on the skills that they have this system will act as an online community where users can contact each other and see the available resources that have been provided to them also throughout this study the analysis of the current system that is already in the market has been done to develop a system that satisfies user needs and achieves every goal that has been set by the stakeholders this system will focus on addressing and solving the issues that the current freelancers have and to provide a better solution to all the freelancers around the world. In the system that is in development agile methodology has been chosen because agile is the best-suited methodology for the proposed system the advantages are it can get feedback for users while the system is still developing and adjust the system based on the changes that were required by the stakeholders. And this proposed system can be used as an alternative for the current methods that users use and act as a replacement model for those systems.

ABSTRAK

Tujuan kajian ini adalah untuk menyiasat perkembangan mewujudkan pasaran yang menggabungkan kedua-dua pelanggan yang sedang mencari seseorang untuk bekerja untuk mereka dalam talian atau pekerja bebas yang sedang mencari pelanggan untuk membuat wang berdasarkan kemahiran yang mereka miliki sistem ini akan bertindak sebagai komuniti dalam talian di mana pengguna boleh menghubungi satu sama lain dan melihat sumber yang ada yang telah disediakan kepada mereka juga sepanjang kajian ini analisis sistem semasa yang sudah berada di pasaran telah dilakukan untuk membangunkan sistem yang memenuhi keperluan pengguna dan mencapai setiap matlamat yang telah ditetapkan oleh pihak berkepentingan sistem ini akan menumpukan pada menangani dan menyelesaikan isu yang ada pada pekerja bebas semasa dan untuk menyediakan penyelesaian yang lebih baik kepada semua pekerja bebas di seluruh dunia. Dalam sistem yang dalam pembangunan metodologi tangkas telah dipilih kerana tangkas adalah metodologi yang paling sesuai untuk sistem yang dicadangkan kelebihannya ialah ia boleh mendapatkan maklum balas untuk pengguna semasa sistem masih membangun dan menyesuaikan sistem berdasarkan perubahan yang diperlukan. oleh pihak berkepentingan. Dan sistem yang dicadangkan ini boleh digunakan sebagai alternatif untuk kaedah semasa yang digunakan oleh pengguna dan bertindak sebagai model gantian untuk sistem tersebut.

Table of Contents

LIST OF TABLES

TABLE NO. **TITLE** **PAGE**

LIST OF FIGURES

FIGURE NO. **TITLE** **PAGE**

LIST OF APPENDENCIES

Chapter 1

INTRODUCTION

1.1. Introduction

We all know that Covid-19 has affected many lives especially for business owners who had employees who were working for them daily, those who were supposed to attend the company or the place that they worked for but because of Covid-19, they were forced to start hiring freelancers even if they previously hadn't considered it at all. And those companies are shifting from steady traditional work relationships to increased heterogeneity of work relationships and related tasks [3]. As a result, many organizations around the world have recognized the advantages of remote working.

People who are working as a freelancer are enjoying flexible schedules and improved life balances. And the new generation seems to be preferring freelancing over the old-style work ethic, which is staying in a 9-5 job. But the problem that nowadays freelancers face is there is no marketplace for them to find jobs easily. They have to find clients through advertising and doing promos to find jobs in the field that they want to work.

The solution that is proposed is that this system is going to combine all the freelancers in one place so that customers who are searching for a freelancer in a certain field can easily find them and contact them through that system, this system will be in the form of website it will support all the platforms and will be available all the time for users to use it will also act as a marketplace where users will be able to contact each other directly and easily get their job done in the fastest way possible.

1.2. Problem background

Nowadays 71% of workers in the United States are doing their job at home because of Covid-19 and if they are ever given a choice, they would continue to work at home even after the pandemic, according to Pew Research Center survey. Based on those surveys more people are shifting their work ethic into working at home and the number of temporary workers or freelancers and independent workers have arisen, people tend to be more into the flexibility lifestyle where they can be their manager and have control over the task that they want to achieve (Parker, Juliana, Minkin, 2020), also “Freelancing became fairly more popular amid the Pandemic of COVID-19. Small businesses and large companies alike are taking advantage of hiring experienced professionals online to handle much of their growing to-do lists” [2]. but the issue that is generated from those required demands in the market is that people don’t know where to head to find jobs and they don’t have a marketplace where they can showcase their work and find job opportunities in their specified area, also there is another type of consumer who are people or companies that they look for a person that is specialized in a certain area to give a task to finish, but consumers usually get confused searching for people to work for them and they will have trouble finding the best person suited for the position that they want. The system that is proposed will create a marketplace and act as a bridge to connect both types of the consumer's users who are using this system will no longer have issues with finding or searching for the best person suited for their needs.

1.3. Project aim

This project aims to develop an efficient and categorized marketplace for the people who are trying to find a worker to work for them at their home and also develop a place where workers or freelancers have the ability to showcase their work and make money without leaving home.

1.4. Objectives

The objectives of the project are:

i. To review the existing approaches for the existing systems and the components that are used in the process of creating those existing systems.
ii. To design and develop the required components of the website using the tools that are best suited for the system to be created.
iii. To achieve the goals and try to improve the system to compete with already existing systems that are in the market.
iv. To test and evaluate the developed system and to make sure it reaches the standard levels of publishing the website and to reach the goals that are set accurately and effectively.

1.5. Scopes

This project will be run within the following scopes.

1. The system will focus on the development and the growth of the relationship between the person who tries to make money from the skill that they got and the persons that are looking for a worker.
2. The system will focus on categorizing different sections of freelancers (e.g., there will be a graphic design section for the freelancers who work in graphic design, as well as for people who are working in scrip writing, etc.)
3. Users must pay a small monthly fee to create a business account to work on the website.
4. This payment will be required because it will be needed for the system to run and to be continued to work.
5. The system will be available for every platform so people can visit it by using whatever device that they have and the only constraint this system will have it

will not run offline so users must have an internet connection to access the website.

6. The system will include payment methods and communication methods as well so people can communicate with each other easily and send/ receive money after they get their work done.

1.6. Importance of the Project

Global competition and the race for demand and innovation and the tendency towards organizing work in projects and assigning teams to work tasks have made the world of work more fast-speed and demanding [4]. based on those requested demands a place is needed for the workers and the customers to come together and find their needs for freelancers constantly looking for a job to land also improve their knowledge and experience in their specified field they need a place where they can find customers constantly in order to make living and get benefits from the knowledge that they have this trend has been around for many years. The project that is been in development will fill the gap that is in the market where people need a community or a marketplace to find each other and reach their needs in one unified place through this system people will be able to contact each other by using the components of the system they can easily see different categories also see the available freelancers based on their reviews and reputation they can choose who is the best suited for their needs.

1.7. Organization of the Report

In this chapter 1, it will showcase an introduction to the system that will be created, and then it will discuss what is the problem that caused this system to be created and after that, the aims and the objectives of the system that will be included and what will not be included in this system have been discussed then finally, then it

goes down to the importance of this project why this system is important to be created it has all been described in details.

1.8. Gantt Chart (PSM 1)

Figure 1-1 (Gantt Chart of PSM 1)

Chapter 2

LITERATURE REVIEW

2.1. Introduction

Online freelancing has created many great opportunities all around the world for freelancers who can easily work from home through their internet. With the growing rate of demand for online works nowadays and people searching for jobs to do independently the term freelancer has dramatically increased. A freelancer is a person who works independently without committing to a certain client for a long period of time. Those freelancers have the freedom to do whatever job they want and whenever they want to, it will also enable the worker to earn money without the traditional obligations that office jobs provide (Anikó Hannák & Claudia Wagner & David Garcia & Alan Mislove & Markus Strohmaier & Christo Wilson, 2017). Companies and organizations have seen a lot of benefits from the online marketplaces because now they can hire clients from all around the world, because of the demand for freelancers and due to the growth rate of freelancers many marketplaces have come into creation to fulfill that demand and combine all of the different categories in one unified place. To create a system that achieves all the required goals and satisfy user needs first the already existing systems has to be studied and understood then the system that is been in development will be created based on the best features and non-existing functionalities of the already available systems in the market in this chapter the currently available systems will be studied then those systems have a comparison between them to identify the needs and improvements to be done to make the current systems better for users and to achieve the user needs in every aspect.

2.2. Current System Analysis

There are plenty of web services that serve the freelance marketplace and they are constantly competing with each other in order to provide the best service for their customers also their constant users, recent stats show that there are 57 million freelance workers in the United States (Keegan, 2021). with the growth rate of the market, more alternatives are been created in order to fill the gap that is been created by the demand of the users to understand what are the similarities and what are the differences between those web services we have to identify the important characteristics of the current systems that are popular and used among users on a daily basis. The methodology that has been have used to identify the user characteristics are:

a) Whether the web services have detailed user profiles.
b) Secure payment processing.
c) Business building opportunity.
d) Job Categories based on user preferences.
e) Classifying the websites by popularity.
f) Transaction Fees between users.
g) Availability of multiple people working on the same project.

There are many websites that serve in this niche but only a few of them will have a comparison between them and those websites that are very similar to the system that is in development and those websites include many features that can be compared to the requirements of the current system.

2.2.1. Fiverr

Fiverr has 3.24 million active users and it was founded in Israel (Dean, 2021). Fiverr is a marketplace that serves freelancers and business owners in various categories those services include content creation, translation, website design,

voiceovers, graphic design, etc. Fiverr is used by both freelance and customers and they can interact with each other through this system. Freelancers can create packages for the customers so they can decide which package they can choose between standard and premium. Freelancers can withdraw their money after 14 days and they can use a PayPal debit card and also have a wire transfer. Fiverr current revenue is around 189.51 million USD in the United States (Dean, 2021). The cons of Fiverr are that they are currently banning Iraqi users from using their website and users in that area cannot access the website. The commission charged is also very high and they have a long processing payment time (V., 2021).

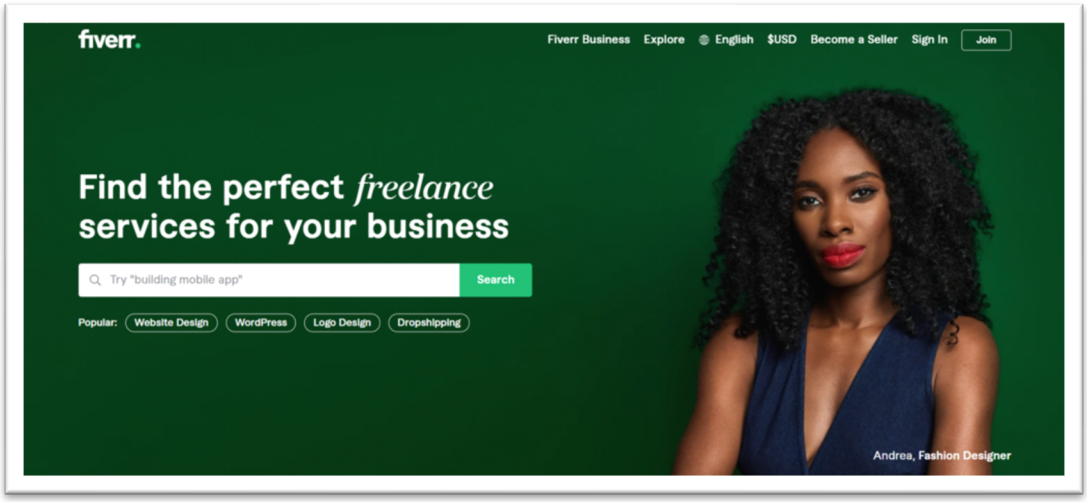


Figure 2-1 Fiverr

2.2.2. Upwork

Upwork is an American freelancing website that was founded in 2015 it was formerly known as Elance-oDesk then it was rebranded as Upwork. Their services include web design, software development, accounting, and programming issues. They also have budget-based projects that users can choose among the freelancers and they also provide with secure payment and credible clients. Their revenue in 2020 was 373.6 million USD (Dean, 2021). The cons of this website are that they have a lengthy selection process and a high service fee.

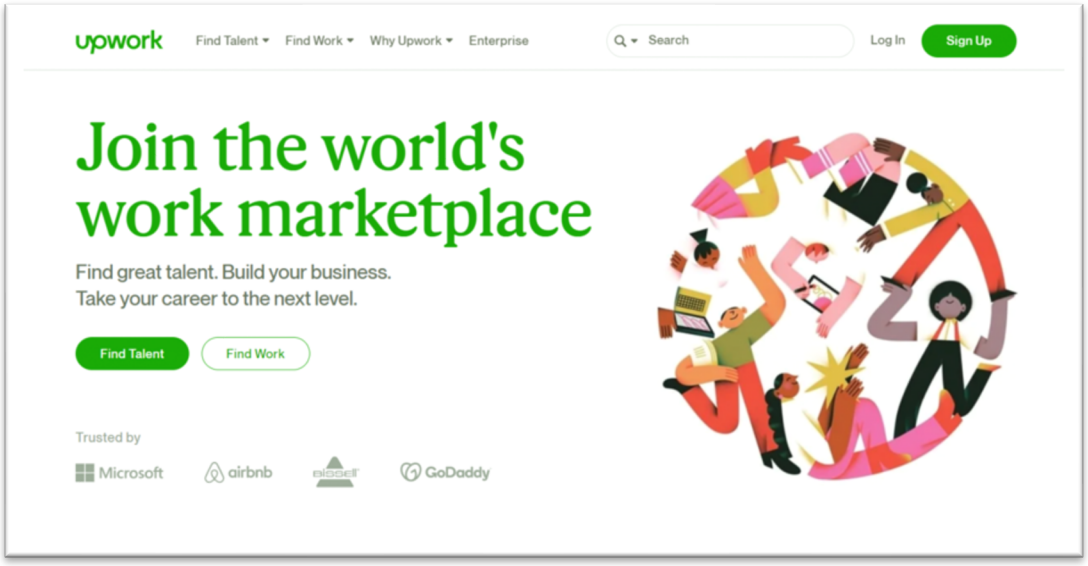


Figure 2-2 Upwork

2.2.3. Toptal

Toptal is a global remote company that was founded in 2010 (Grant, 2019). they offer services like filtering highly qualified freelancers in their specified industry and customers can easily contact them to work with them through this website for professional work, Toptal has a wide range of categories from web developers and web designers and financial advisors also product managers, Toptal is for companies and high tier list and they have a dedicated web service for time tracking which is called TopTracker with using this service freelancer will get the total number of the price that they getting via PayPal and Payoneer also with a direct local bank transfer. The cons of this web service are that they have an extensive screening process, and it's for big projects only.

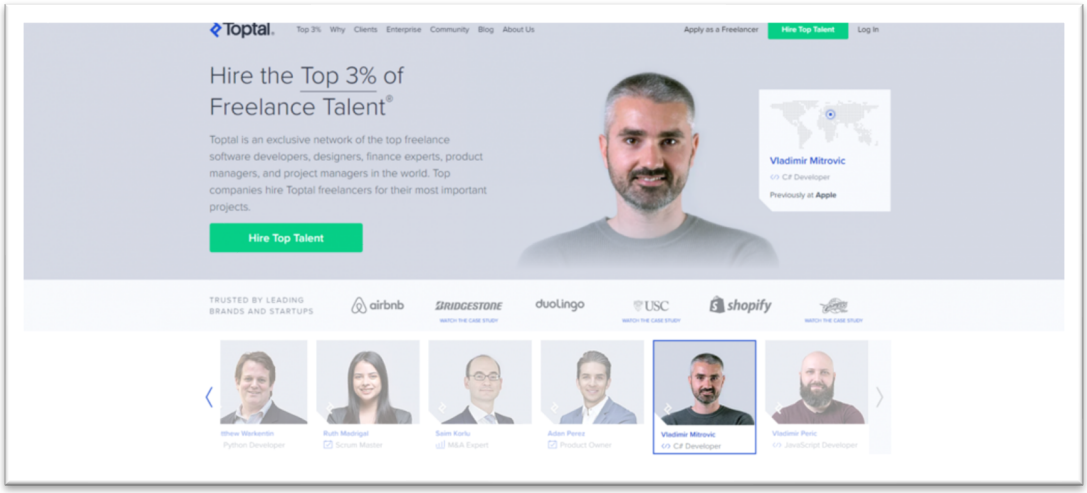


Figure 2-3

2.2.4. Freelancer.com

Freelancer is an Australian marketplace for freelancers that was founded in 2009. (Lakhani, 2015). Their revenue is 192.9 million (Velebny, 2021). With this website employees can post jobs through this system then freelancers can bid on that job to complete it, through this system different freelancers can work on the same project. Every freelancer will receive their payment through the system after completion of the project or the job that they are working on and the website will take 10% percent of that payment that they are receiving, and that payment can be transferred through PayPal and wire transfer to their bank accounts (V., 2021). The cons of this website are that they have a complex interface also spam applicants and fake clients.

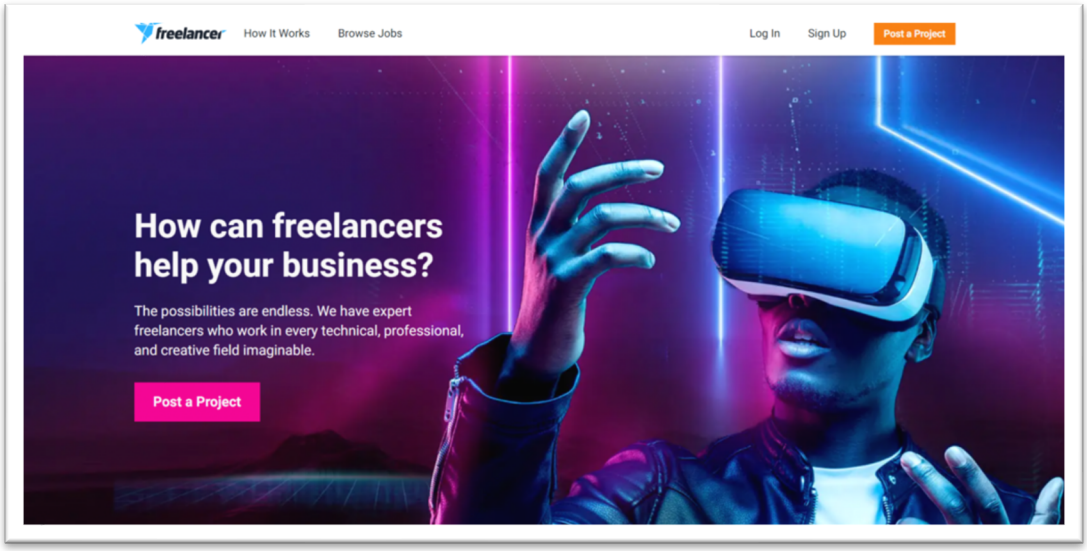


Figure 2-4 Freelancer.com

2.2.5. Flexjobs

Flex jobs is another web service that serves in the freelance community marketplace, it was founded in 2007 (Clark, 2021). Workers on this website work as part-time or remote jobs in the flexible timelines they offer broad categories from education to coaching programs and journalism. Flex jobs ensures that every job that is included in the website is legitimate and not fake unlike other web services and they also don't allow advertisers to show up on their website. The cons of this website are that customer has to subscribe for a weekly fee in order to find freelancers and hire them the amount that they have to pay is around $6.95/week this subscription will give the users unlimited access to the users and they can use PayPal or prepaid card to pay their subscription.

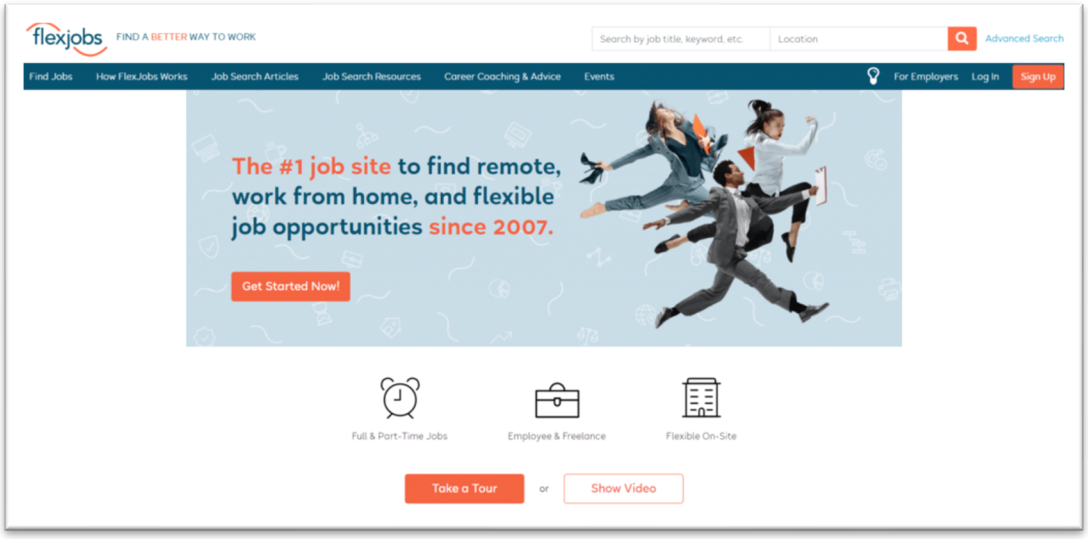


Figure 2-5 FlexJobs

2.3. Compare between existing systems

Table 2-1 List of all current systems that are in the market

Name	URL	Year Founded	Revenue	Categories	Transaction Fee	Online Community	Communication Tools	Type of market place	Project Manager	Work Monitoring
Freelance Marketplace (System in Development)	Freelancemarketplace.com	2021	N/A	Design, Marketing, Video & Animation, Music, etc.	5%	Yes	Yes	Microtask, outsourcing	Yes	No
Fiverr	www.fiverr.com	2010	$189.51 million	Design, Marketing, Video & Animation, Music …	20%	Yes	Yes	Microtask	No	No
Upwork	www.upwork.com	2015	$373.6 million	Copywriting, Editing, Proofreading …	3%	Yes	Yes	Outsourcing	No	Yes
Toptal	www.toptal.com	2010	$200 million	Programming	0%	No	No	Outsourcing	Yes	Yes
Freelancer.com	www.freelancer.com	2009	$192.9 million	Accounting, Technical,	10%	Yes	Yes	Outsourcing	No	Yes

				Software Development …						
FlexJobs	www.flexjobs.com	2007	$201,000 million	Writing, Customer Service, Marketing	Membership Only	Yes	Yes	Microtask	Yes	Yes
99designs	www.99designs.com	2008	$10 - $50 million	Design	5%	Yes	No	Crowdsourcing	No	No
PeoplePerHour	www.peopleperhour.com	2007	$70 million	Marketing, Design, Website SEO …	Hourly Rate	No	Yes	Outsourcing	Yes	No
Guru	www.guru.com	1998	$118 million	Programming, Design, Writing, Sales …	2.5%	No	Yes	Outsourcing	No	Yes
Behance	www.behance.com	2005	$254.9 million	Design	Subscription Only	Yes	No	Crowdsourcing	No	No
Dribble	www.dribble.com	2009	$60 million	Design	Subscription Only	**Yes**	**No**	Crowdsourcing	**No**	**No**

2.3.1. Drawbacks and issues in the current systems

As has been known with the existing systems that have been around for many years comparing a system that is new to the industry is a hard challenge but coming up with new ideas and bringing something new to the industry it's always a game-changer for the users of that specific category with all the stats and information that is available on the current system that are already in the market it's clear that preferred in the industry and what do users like in their specific system that they use our system. Freelance Marketplace will include all the good features that the existing systems have but the thing that needs to be changed is the drawbacks and the bad features that the current systems have so based on user complaints and feedback of the users based on the current system and to fix those issues in Freelance Marketplace, some of the bad features that users have major complains about in the current systems are:

1. High Charges and additional Costs
2. Prevalence and fraud sellers
3. Commission Charges
4. Fake Reviews about sellers
5. Poor Insights on past performance
6. Communication Gap

Those are some common issues that users face in the freelance community and those issues have been around for many years (kumari, 2021). The system that is in development will focus on specifically solving those issues and creating a system that will include all the solutions to those flaws in the existing system.

2.4. Literature Review of Technologies Used

2.4.1. Web Front-end Framework

For the framework of the front end using ReactJs is the best choice and the best suited for system that is in development because ReactJs has better performance than other frameworks it helps in preventing updating the DOM that means that it will be faster and has better looking UX and its open source and flexible, it is also can handle complex interfaces that is why it's the best suited for our system (Insignares, 2021). There are many features that makes react more suitable for this application and those characteristics are:

a) Virtual DOM
b) JavaScript XML or JSX
c) React Native
d) One Way Data Binding
e) Declarative UI

f) Server-Side Rendering
g) Components Based Approach

But as ReactJS has good features it has also some drawbacks including having a large size library also not using isomorphic approach to exploit applications and this leads to having index problems (Insignares, 2021).

2.4.2. Web Back-end Framework

For the backend framework the NodeJS is the best suited for our system and it is required for ReactJS, it is also created for enterprise apps because it's light, scalable and open-source language it will also fill the gap between frontend and backend because it's efficiency and handling different services at the same time with different response times. The key functions of NodeJS are that is can collect data from user it can also deletes, and change data in the database, it will also help with generating dynamic page content (Chowdhury, 2021). The features of NodeJS including:

a) Cross Platform Compatibility
b) Having One Coding Language
c) Facilities quick deployment and microservice development
d) Scalable
e) Active Open-Source Community
f) Advanced Hosting Ability
g) Fast Data Steaming

As has been known as all other technologies NodeJS has disadvantages too, those including

a) Reducing performance when handling with heavy tasks.
b) Heavy code changes due to unstable API.

c) Lack of library Support.

High Demand with a few experienced NodeJS Developers.

2.4.3. Cloud Provider (Server)

With all this data that is being collected there has to be somewhere to store those data and collect them for this one the best choice is Firebase cloud server it can be used on demand servers to run the website and it's secure and efficient than the other cloud providers. The key characteristics of amazon cloud server (Firebase) are:

a) On Demand Self-Service
b) Flexibility
c) Cost-Effective
d) Secure
e) Broad Network Access
f) Resource Pooling
g) Rapid Elasticity
h) Measured Service

Those were the key characteristics that define why choosing Firebase Cloud is the best option for our system (Wikins, 2021).

2.5. Chapter Summary

In this chapter a lot have been discussed and studied why there are many systems that have been created then after that those systems have identified other systems that are currently available and why those systems are needed in order to

satisfy the number of users that this niche has then the system that is in development has been compared to those systems based on some specific characteristics and analytics by using charts and tables to show the strength and weaknesses of the current systems and to decide what to include in the system that is in development then research has been done on what technology should be used in order to create a system that is efficient and reliable to reach the goal requirements of the stakeholders.

Chapter 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1. Introduction

Software development has become a highly demanded activity for business owners also the companies that work in the tech industry. Those companies use different methods based on the category of the project to maximize the profit and to minimize time and costs on the operation they work on. Most of the projects that are not choosing a proper methodology fail because it is hard to achieve project goals and to provide stability for operation without having the right methodology. A software development methodology is a structured process that determines which tasks need to be done to achieve the goals that are required by the stakeholders. The selection of the right methodology will be based on the project duration also its budget, type of software whether the requirements of the stakeholders change or not, with consideration of size also the teams that will be working on this software. In this system, the freelance marketplace it's is important to choose the suitable methodology that will make ease for the flow of the development for the website choosing the right steps can make it easier to reach the goals that have been determined by the stakeholders.

3.2. Agile methodology and justification

Reducing the development cost and delivering efficient and effective software is the primary demand for most organizations, for the project that it's been in development choosing agile is the best and the most suited methodology that can be used in this particular software, not only because agile uses iterative and incremental approaches it's also because the constant changes or improvements that might appear

during the development of the project this improved connection between the client and the project staff it's because customer inputs will be received throughout the process of creating the project and by using agile it will help in those changes that will be demanded by stakeholders or the users that will use this service in the future, other features that make using agile more suited for this project are having, improved project predictability with increased visibility of the project development predicting risk and coming up with an alternative plan will be much easier (Agile, 2021). Another aspect that will help the project development is that it will have increased flexibility because agile divide the projects into short phases and sprints which those phases can be between 1 to 4 weeks based on whether the individual or the team that works on that project has experience or not that will give enough time for them to make improvements or take into consideration the feedbacks that were sent by the users during survey session. With getting feedback from users that might potentially use this website, the system will be more user-focused and it will deliver what does the user needs.

3.3. Phases within the agile methodology

3.3.1. Phase 1: Requirement Gathering & System Analysis

The first phase in agile methodology is the requirements gathering phase in this phase the needs and project goals need to be determined and it's not easy just to ask the stakeholders and the users what they want and easily implemented in the system it's not that straightforward sometimes the users do not know what they want and what are the alternatives and the choices that they have when it comes to setting up goals and deciding whether what should be included in the system those users must be going through some filters or some phases to get the preferences of the users and what should satisfy their needs, those filters will include:

- **Interviews**: setting up some potential future users and asking them questions about what they want to be included in the system is a great way to start the requirement process those interviews will cover a diverse range of users and they will be asked open-ended and follow up the question this will help them to go through the details of their needs this will provide contexts for validating requirements.
- **Questioners & Surveys**: to get the majority belief on certain topics and those surveys can be done online through online websites like SurveyMonkey which currently has over 40 million users worldwide (Pickavance & Turner, 2022). By using SurveyMonkey after finishing with setting up questions the website will give you charts and diagrams based on the answers of the volunteers who have used the survey.
- **Document analysis**: document analysis should be done after getting the results for surveys and interviews but it's also possible to look at other currently available system data that have already done interviews or surveys, sometimes other systems have user manuals which can be very beneficial for the development of the current system.
- **Use cases & Scenarios**: once the early steps of the requirement are finished then it's time to create user stories and use cases based on the available data that has been collected from the users, and well-written use cases provide the analysis and the testing team an invaluable guidebook that they can work on easily (Goss , 2007).

Those are the main methods that are used during the requirement gathering phase and in agile methodology, those requirements can be changed during other phases because sometimes there are improvements that need to be done during the other phases and those needs must be taken into consideration to achieve the user needs.

3.3.2. Phase 2: Prototype & Design

The second phase in agile methodology is design this phase is also very important because it will define the structure of the objects and give you an idea of what the interface will look like for the users before jumping right into the code in this phase the methods will be identified based on the user stories and scenarios the users can test out the interface before implementing it into code, this is a much better way for the users also the developers by giving them an idea how to work on the app and how to walk through the website without the interference of any issue regarding the user experience and the user interface which is known as UX/UI this term is now very popular among business owners because it has the primary goal of increasing sales and increasing growth of the business (D., 2019). The design also has two phases which are:

- **Conceptual design**: In this phase, the objects and the functions of the system will be put together in a non-technical manner and the methods that has being applied in this phase will include wireframes and mockups also flow charts with component diagrams, those will be done with using tools like (Lucid Chart, Figma, Sketch, Invision, etc.) (Alam, 2019).
- **Technical design**: after finishing up with the conceptual design then the next phase is technical design this part will cover how those prototypes will be implemented and how they will interact with the server and the database that it has been used and it will cover the layout of the system, all those designs will be implemented by using (Class Diagrams, Activity Diagrams, and Sequence Diagrams, State Diagram) (Alam, 2019).

3.3.3. Phase 3: Development & Testing

The third phase is development and testing, after finishing on deciding what the user wants and how the design of the system will be implemented then in this phase goes under two steps to make a fully functional system that reaches the user

requirements and goals the first step is developing the system based on the requirements and the data that has been collected through the previous phases and the system will undergo into development, in this phase the developers will work on making the user interface into code the goal is to make a functional system that works and reaches the standard level of operation and if everything went well then it's time to move on to the next step which is testing in this part the system must go through some standard tests before deploying the website and it's being released to the public and some of those tests include (van der Hoek, 2021):

- Checking whether the code is clean
- Addressing bugs and errors
- Performing trial runs.

Final Testing and acceptance of the complete project should be carried out by the quality assurance (QA) detectors and in this phase, some end users can be tested to see how they will perform a certain task that has been developed and whether they will face any issues or not by using the website (Windsor, 2020).

3.3.4. Phase 4: Implementation & Deployment

After finishing with development and testing if the system has successfully passed the testing phase, then it goes down to the implementation of the code and making it ready to be deployed to servers and making it available online. This phase is combined in a complex set of procedures that require detailed planning to ensure to reach the requirements and to achieve the success of the system and agile project are driven by the need for continuous progress and to make those progress that has been done visible and to understand what has been done and what is not finished yet then one the project management tools needs to be used such as Trello or TFS for this project Trello will be used (Wijetunge, 2017). For the deployment part, the system must use DevOps tools which will make the deployment part much easier by doing automation, and DevOps is rooted in agile methodologies, the tools that will be used

in the system such as (Git, Docker, GitHub, etc.) this will make deployment phase much faster because it does not have to deploy every part of the system over and over again with using those tools it will automate the whole process into a faster pace development.

3.3.5. Phase 5: Review & Feedback

Finally, after finishing with the deployment and the implementation the final phase is getting review and feedback from users this is a part of the agile methodology steps it helps with regularly identifying areas for improvement (Chervenikova, 2020). In the feedback section as data has been collected in the requirement phase in this phase also survey will be conducted first from the stakeholders and the clients that are invested in the system after getting those results if any issues occurred or any improvement is needed to be done it will be sent back to the requirements phase to be solved and implemented in the system if there was not any problem with stakeholders and the clients then after that the end-users must be tested out to see if a user who doesn't have any previous information on the system will face any issues or will have problems by using the website and that feedback will be collected by using to methods which it has already been mentioned the first one is doing interviews with stakeholders and the clients the second one is doing survey and questionnaires from the end-users.

3.3.6. PSM: Gantt Chart (PSM 2)

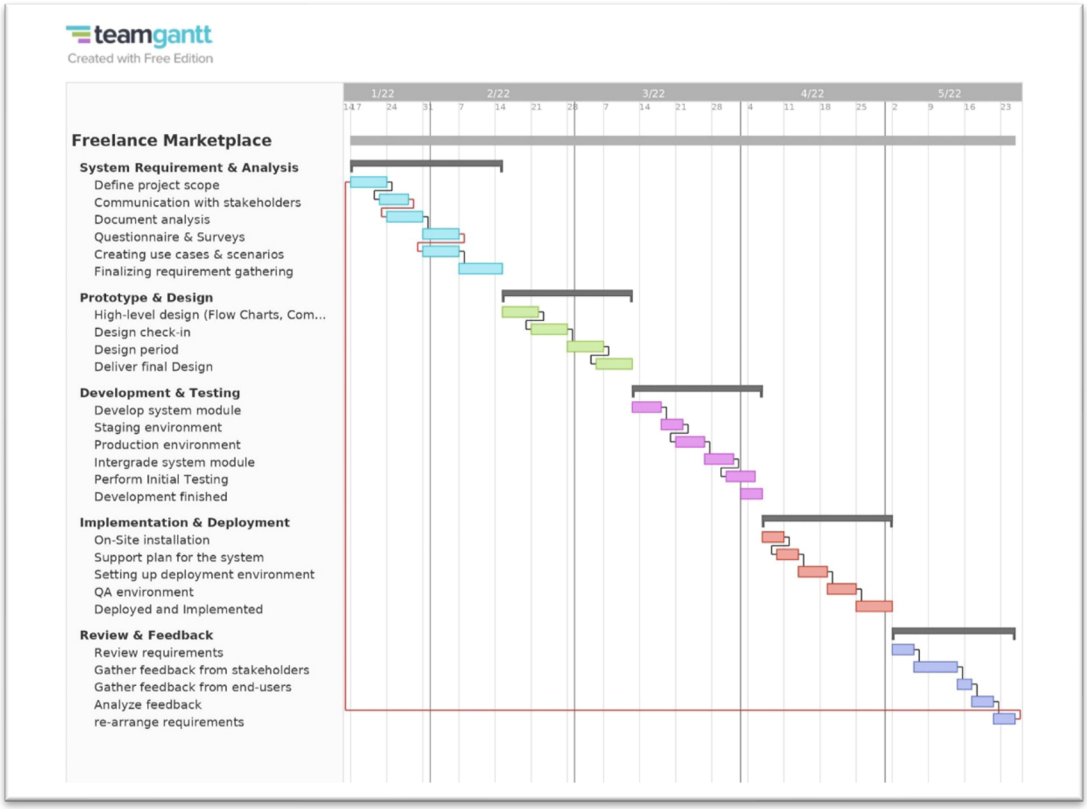


Figure 3-1 PSM 2 Gantt Chart

3.4. Describe briefly the technology or tools used to develop the system

3.4.1. ReactJS

Is the best choice and the best suited for our system that we are working on because React.Js has better performance than other frameworks it helps in preventing updating the DOM which means that it will be faster and has better-looking UX and its open source and flexible, it is also can handle complex interfaces that is why it's the best suited for our system (Insignares, 2021).

3.4.2. NodeJS

NodeJS is the best suited for our system and it is required for ReactJS, it is also created for enterprise apps because it's a light, scalable and open-source language it will also fill the gap between frontend and backend because it's efficient and handling different services at the same time with different response times. The key functions of NodeJS are that it can collect data from users it can also delete, and change data in the database, it will also help with generating dynamic page content (Chowdhury, 2021).

3.4.3. Firebase Cloud Server

With all this data that is being collected it has to have somewhere to store those data and collect them for this one the best choice is the Firebase cloud server it can be used on-demand servers to run the website and it's secure and efficient than the other cloud providers.

3.5. System requirements analysis: hardware and software

3.5.1. Hardware Requirements

Table 3-1 Hardware Requirements

Components	Minimum	Recommended
Processer	1.9 gigahertz (GHz) x86- or x64-bit dual-core processor with SSE2 instruction set	3.3 gigahertz (GHz) or faster 64-bit dual-core processor with SSE2 instruction set
Memory	2-GB RAM	4-GB RAM or more
Display	Super VGA with a resolution of 1024 x 768	Super VGA with a resolution of 1024 x 768 or Higher

3.5.2. Software Requirements

Table 3-2 Software Requirements

	Windows Requirements	**macOS Requirements**	**Linux Requirements**
Operating System	Windows 7 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Internet Connection	Required		

3.6. Chapter Summary

In this chapter the methodology that will be used in the system that is currently in development has been introduced and simplified why choosing this type of methodology is the best option that fits the current system after that the phases that will be included in the agile methodology which is the chosen method has been described phase by phase including the tools that will be used the specified phase and what should be done in a certain phase to complete the requirements that were set at the beginning of the chapter, and after that, the Gantt chart has been created to show the user the dates and how many weeks it will take to finish the project, after completing with the chapter then the technology that is going to be used in the system has been described and introduced what is going to be used in the system. Then the minimum and recommended system requirements have been identified to give the user what device or what software they should be using to use the website fluently.

Chapter 4

REQUIREMENTS ANALYSIS AND DESIGN

4.1. Introduction

Requirements can be either known or unknown. The purpose of requirements gathering is to collect as many known requirements as possible. The process of requirements gathering is both critical and difficult (Phillips, 2000). That is why having poor requirement management can lead to project failure, in every project initiation knowing every aspect of the project and having a clear view on what to do is very essential for both the developer and the end-users that will use the system in the future for the proposed system that is in development because the project will be released based on a worldwide service gathering requirements has to be based on worldwide users and every decision that will have a huge effect on the users, so the majority of the results have to come from surveys that have been conducted by worldwide organizations for example (Global Survey, IntelliJ Survey, and Logit Group Inc., etc.) those organizations provide B2B digital data that covers multicultural nations like North America, Europe, Asia also South America (Staff, 2021). The freelancing marketplace will also cover the places where the current system has already been banned and it will also focus on the countries that do not have a unified payment system like PayPal or they do not have use credit cards and Mastercard so research has to be done to know which payment is available in the local countries and cities. There are plenty of requirement analysis methods that help to explain how the system should be and how it will behave in certain processes those methods include using use cases, sequence diagrams, and activity diagrams in this part the system functionalities and how the system behaves will be explained in the proposed system, another part of requirement gathering is design in this part the overall architecture and the database will be showcased by using class diagrams and ERD tables for the database design the final part of the design is also interface design in this part the

overall look of the system will be introduced and it will show what the final product of the system.

4.2. Requirements Analysis

4.2.1. Use Case Diagram

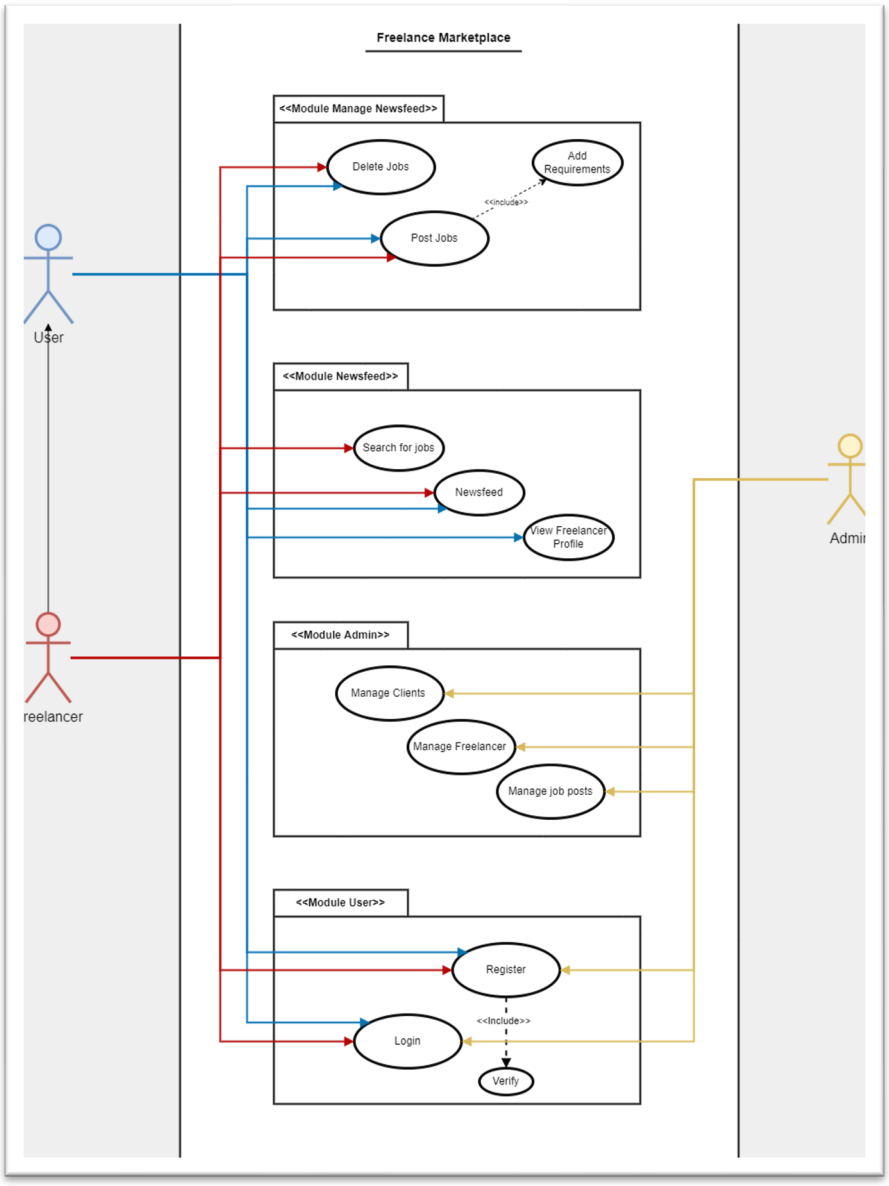


Figure 4-1 Use Case Diagram

Use cases are a very essential part of system development as shown in the use case above that describes what will be included in the system and what are the components and functionalities of the proposed system. As shown in the use case diagram there are 3 types of users are freelancers, client, and admin those 3 has the biggest impact on the system and each of them has their task and functions that they can operate in the system, starting with the client which is one of the main characters in the system, the client can search for freelancers and set reviews on those freelancers based on the job that they have performed the client can also hire those freelancers and view their profile, and one of the main functionalities of the system is posting jobs without this functionality the system will be pointless this part also is the job of the client it's their duty to provide jobs to the system and for security measures, each client has their username and password to login into the system using the tools that are provided.

For the freelancers, once they receive jobs from clients they can see the requirements of that specific job, they can either accept the job or decline it based on whether they were available or had time to accomplish that specific task, and once the freelancers accepted the job they can also update the status of the jobs that they are currently working on, whenever they finished the job the clients can set reviews based on the job that they have finished to let the other users know what is the actual ability of that freelancer, they can also send messages between each other and create groups to communicate and update each other on the latest versions of the project that the freelancers are working on, another feature of the system is that freelancers can also promote their profile to reach more people and get more jobs in the future.

Admin has access over all the clients and the freelancers that are currently in the system the admin can either edit or delete their profile based on standards and regulations of the system, also the admin can review posts that have been posted in the newsfeed by clients and freelancers.

4.2.2. Sequence Diagram (Freelancer)

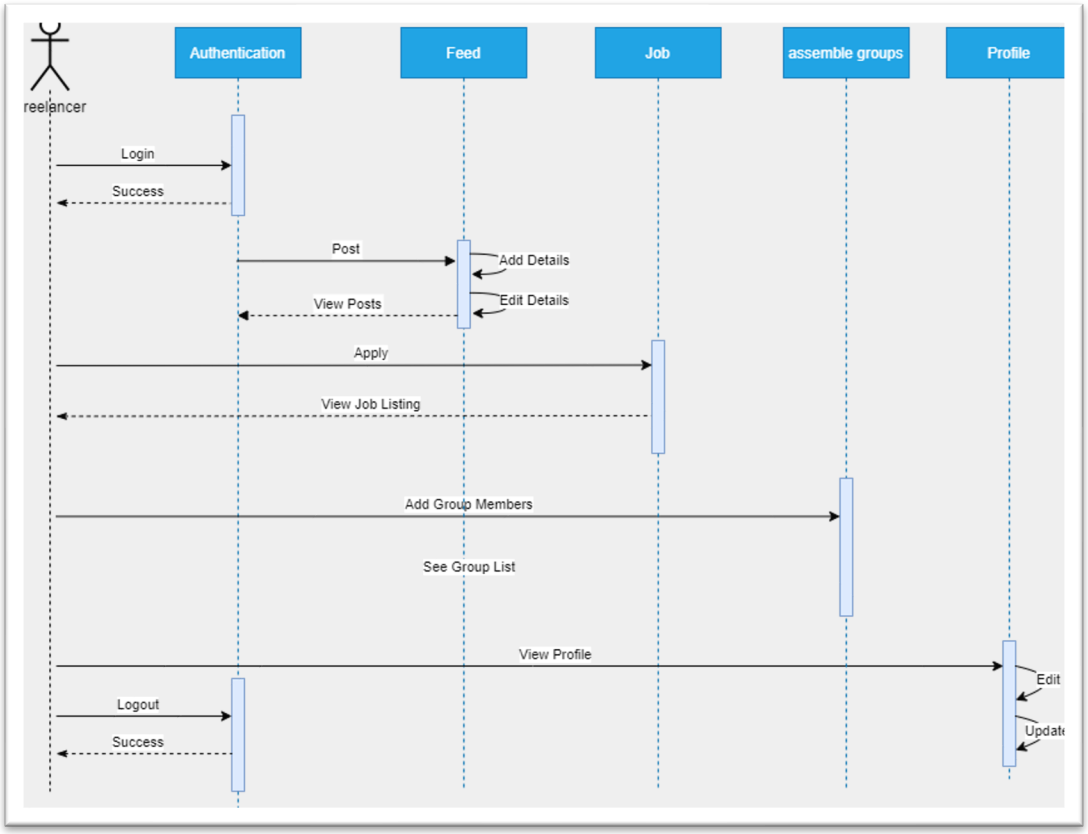


Figure 4-2 Sequence Diagram for Freelancer

Sequence Diagram is also another part of UML diagrams family in sequence diagram it shows the interaction between the objects that are set in the system for the system that is in development there are 3 sequence diagrams which are based on the users of the system for each of the users there are one sequence diagrams which shows the interaction for that specific user. The first user is the freelancer and the first object that they go through is authentication in this part the freelancer has to log in and enter their credentials if their credentials are correct, they can move on to the next object which is the post list in this part they can see what are the available post to apply for and if they want to apply, they can send the resume to that specific client.

Another object of the sequence diagram is creating groups, freelancers can create groups with other freelancers to work with each other to finish a certain task that they are assigned.

4.2.3. Sequence Diagram (Client)

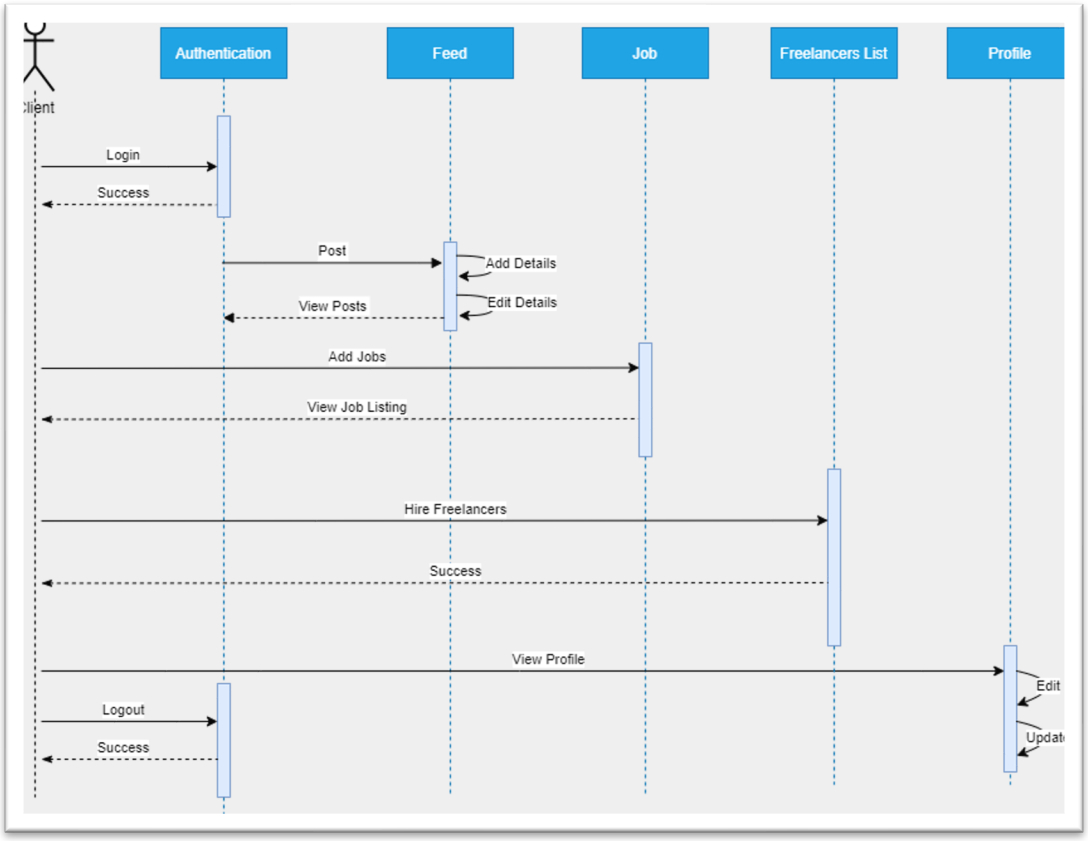


Figure 4-3 Sequence Diagram for Client

In the client part as shown the client also has to go through the authentication and login into the system after the client's logs in into the system they can view the jobs that they have listed and they can also add jobs for the freelancers but they have to enter the details and the requirements of the jobs in order to post their jobs in a proper way, the clients they can also hire freelancers without posting jobs all they have to do is to select the freelancers that they want and send a direct message to them to let them know about the jobs that they have to offer.

4.2.4. Sequence Diagram (Admin)

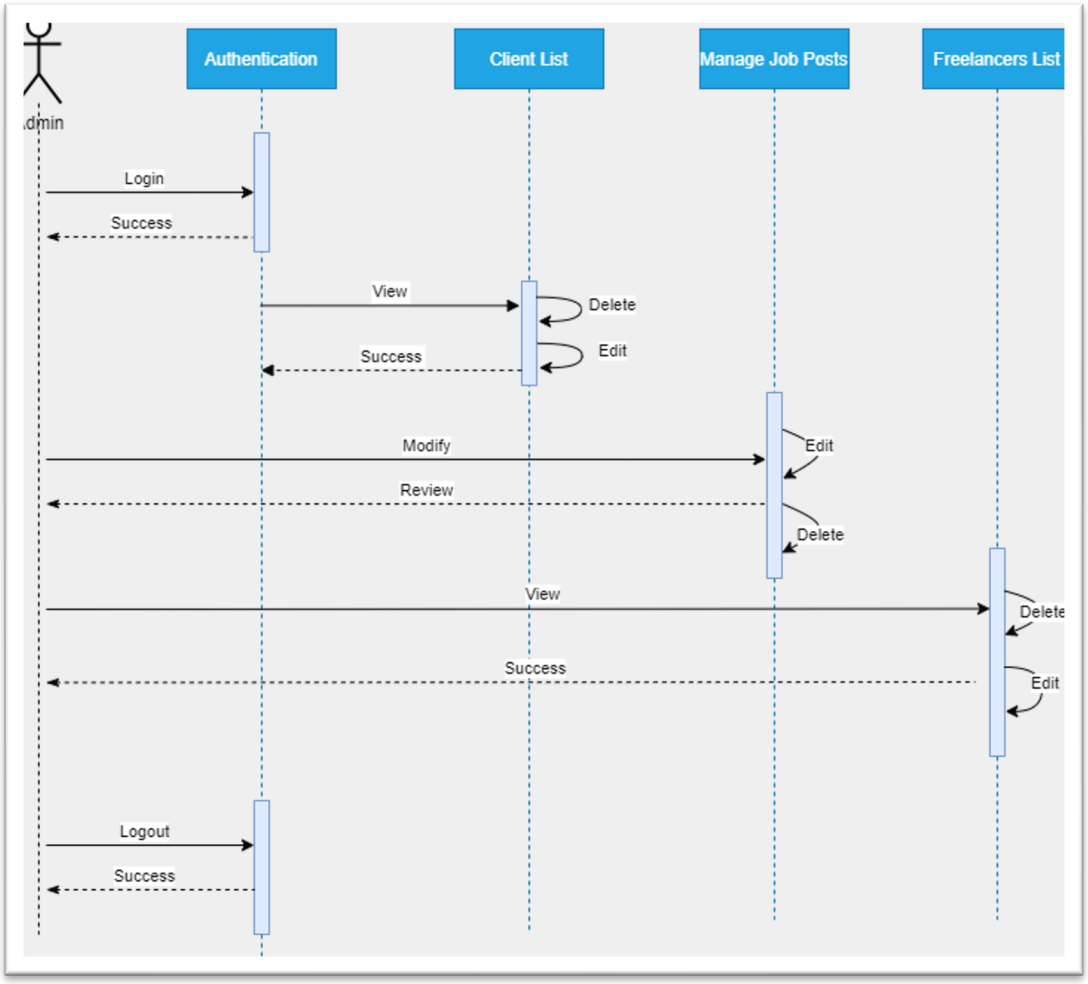


Figure 4-4 Sequence Diagram for Admin

The sequence diagram for the admin part is including also an authentication for the admin to log in and in the admin part, the admin has access to all the clients and the freelancers of the system the admin can edit and delete the clients and the freelancers and modify them the way that they want too and also, they can manage the posts that have been submitted by the users.

4.2.5. Activity Diagram (Freelancer)

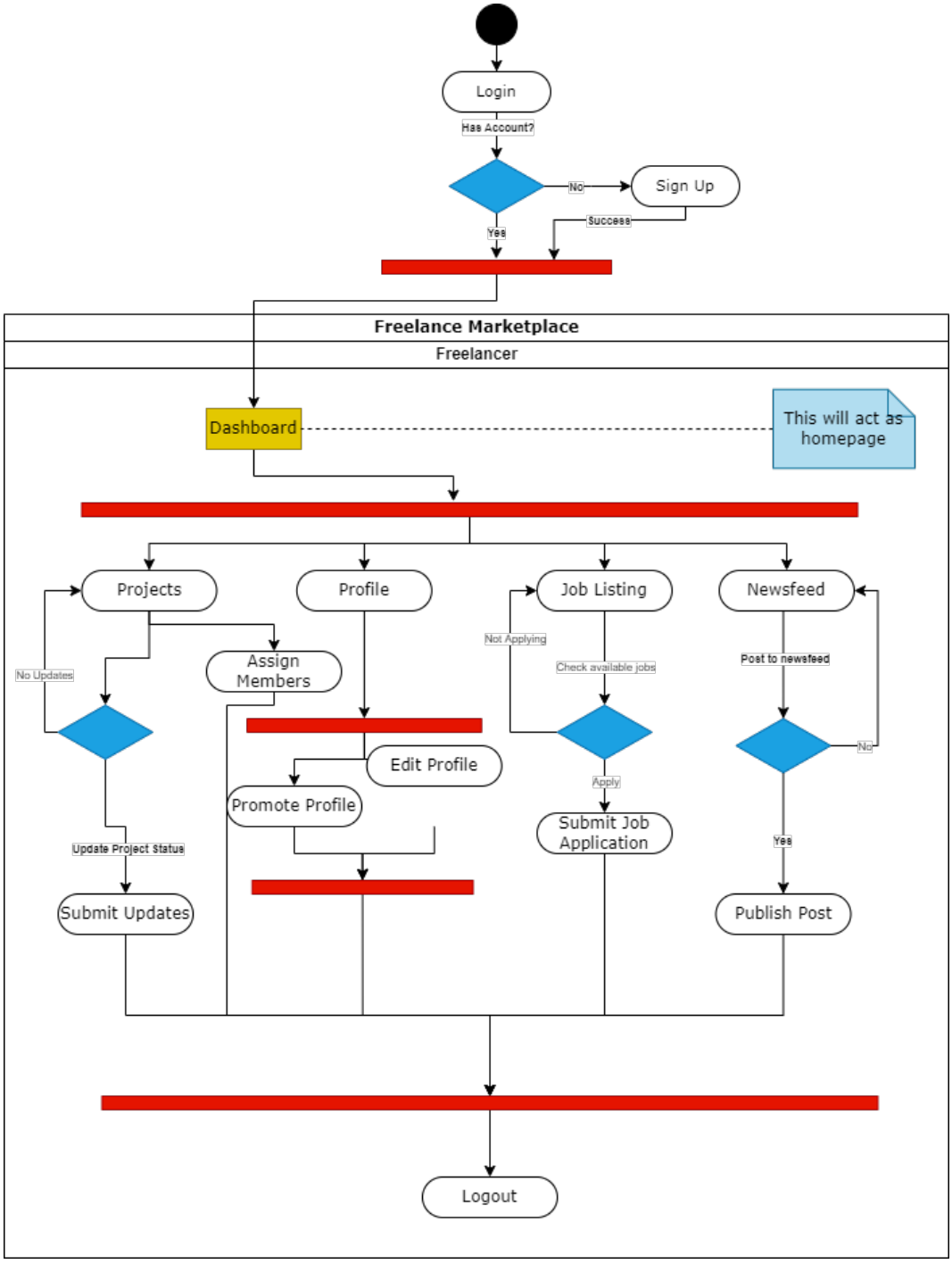


Figure 4-5 Activity Diagram for Freelancer

4.2.6. Activity Diagram (Client)

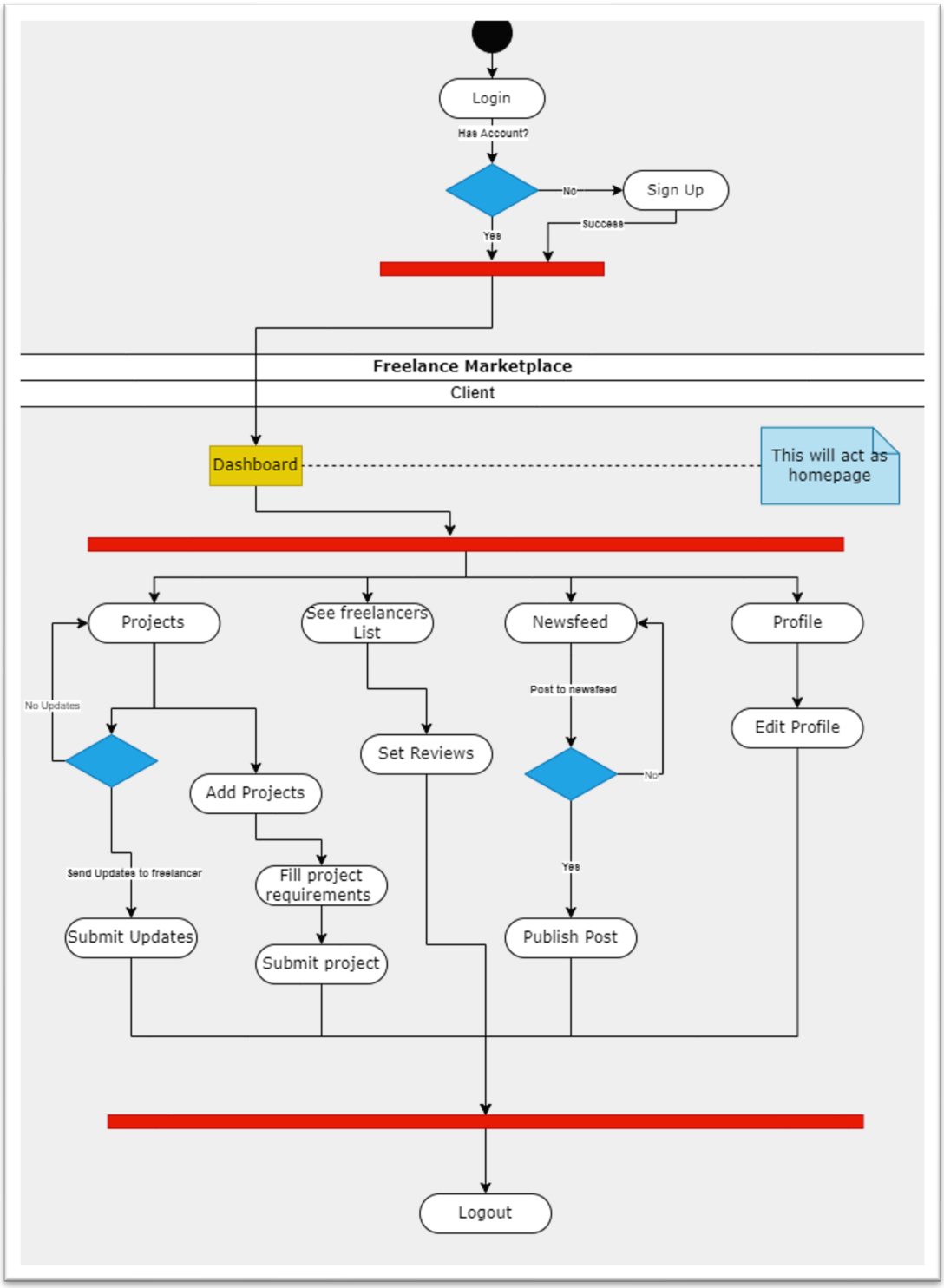


Figure 4-6 Activity Diagram for Client

4.2.7. Activity Diagram (Admin)

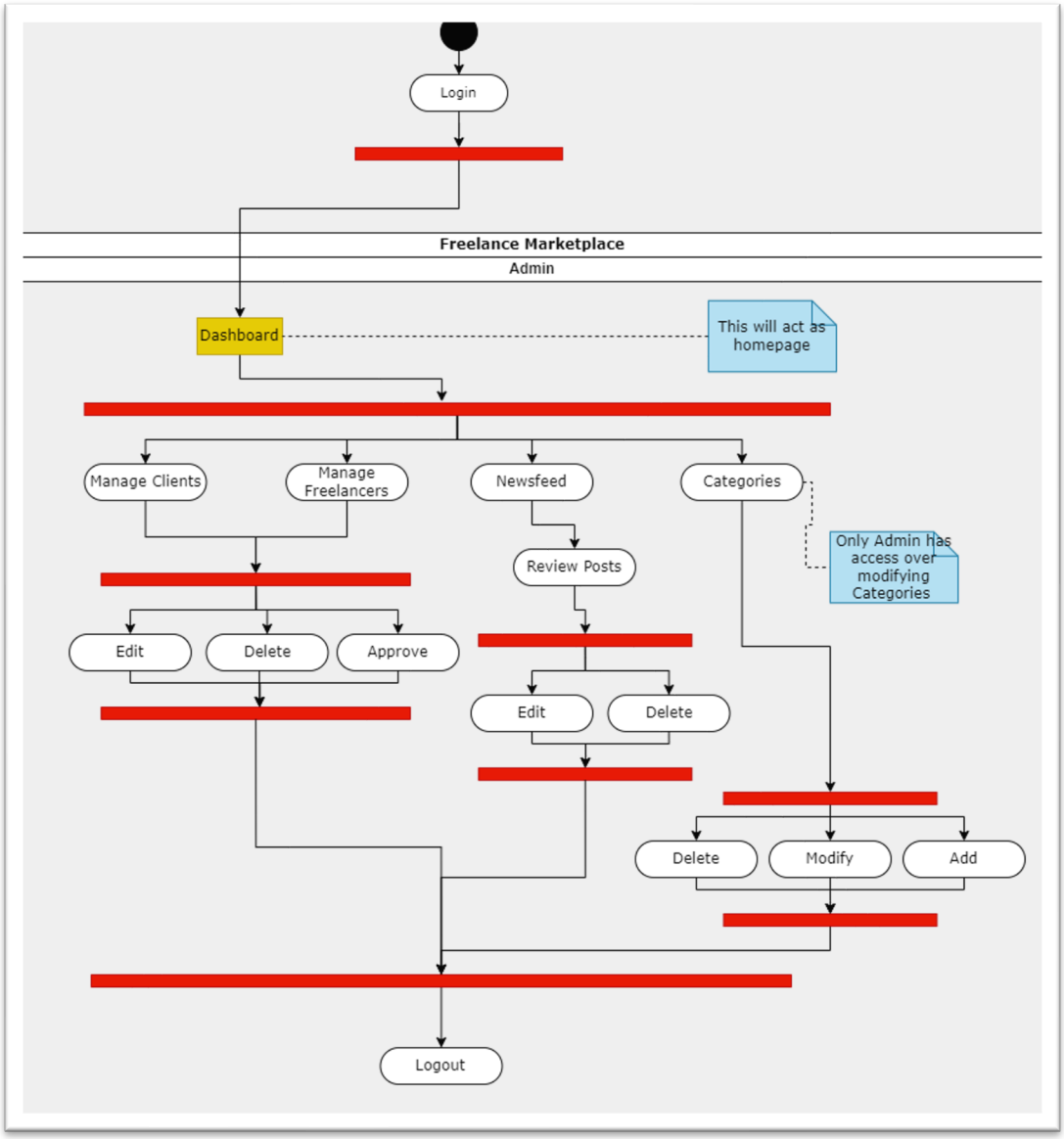


Figure 4-7 Activity Diagram for Admin

The activity diagram is been used to understand the flow of the components on a high level in the activity diagrams as shown in the figures above there are 3 main activity diagrams that represent the whole system the first one is for the freelancer which shows how the freelancers will flow between the components of the system but they all have one stage in common which is the starting state where they have to log in to the system and go through the authentication then after that stage every user has different flow based on the components that have been set for them and the stages change based on what the users have in their own specific diagrams.

4.3. Design

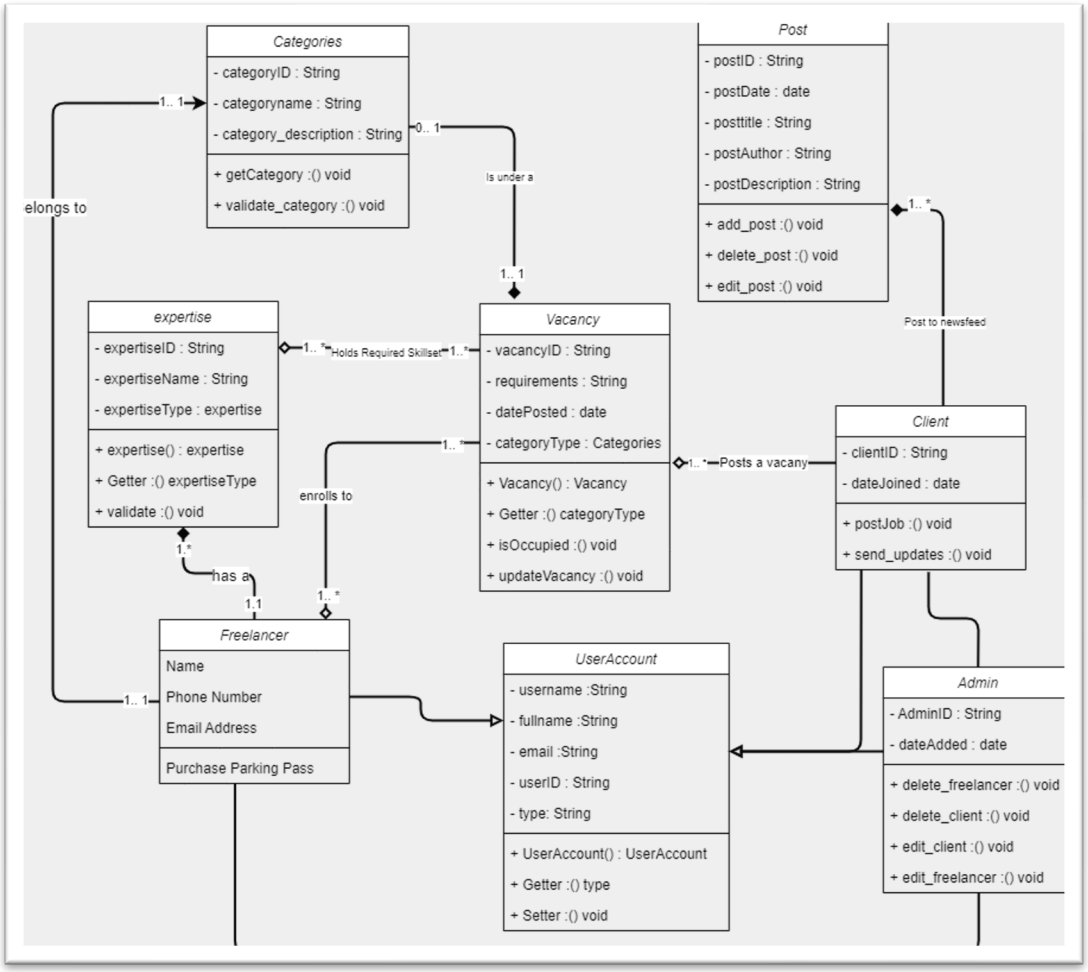


Figure 4-8 Class Diagram

The class diagram will help for the system when it implemented that it will apply all the class inside the system in the user account which is the parent that has several children each of them have their functions and parameters for the user and the different user types the composition because in each client form new user will complain and they will be assigned to the vacancy finally the client and the admin class has an association relation with assign project.

4.4. System Architecture

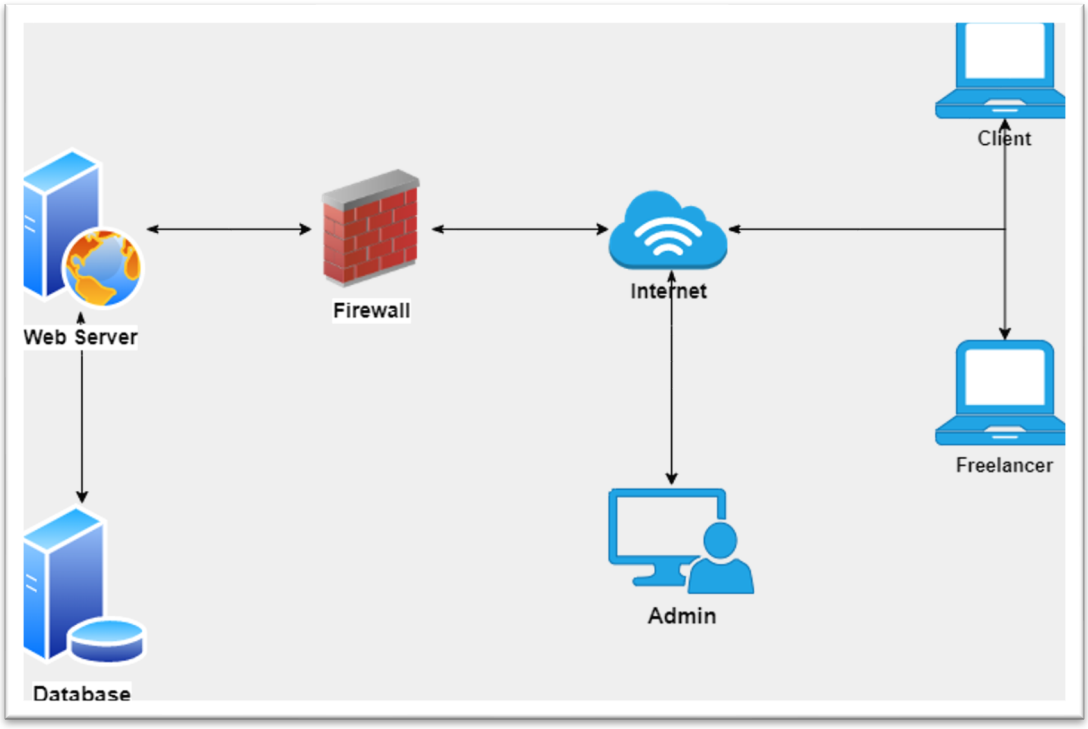


Figure 4-9 System Architecture

System architecture is very important for the stakeholders to understand how the architecture of the system will be created and where the components will go in the system in the system that is in development as shown in the figure above the end-users are freelancers and the client also the admins those are all connected to the internet where it provides the interface in the device that they use, the internet is also connected to a firewall which is the main key for protection, it will protect the databases from being attacked by the non-users of the system, after the firewall it comes down to the webserver which here the operation of the system is being worked and all the components of the system including the interface and functionalities is being saved and operated, the last part which is the database in this part the data of all the users is being stored and collected in order for the user to save the data and the changes that they made in the system to be stored somewhere safe without any interruptions.

4.5. Database Design

4.5.1. Entity Relations Diagram

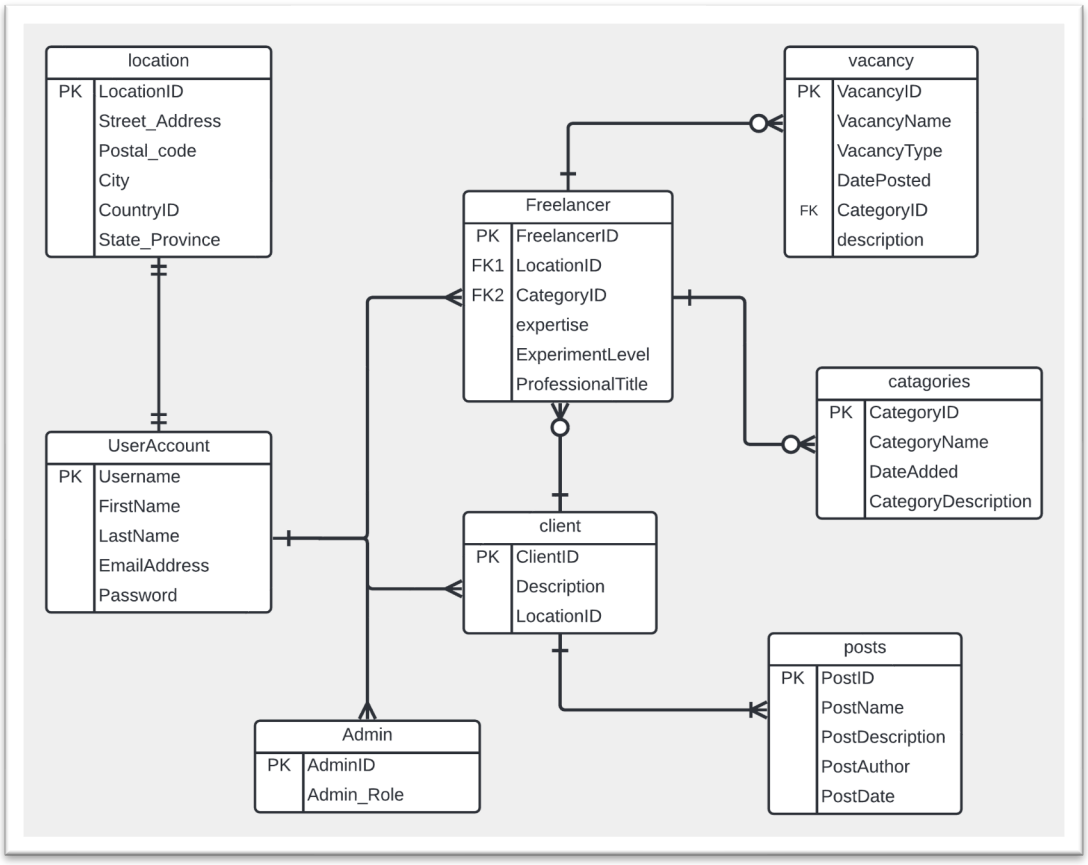


Figure 4-10 Entity Relations Diagram

In order to understand the database and know how the structure of the system will be created entity relationships diagram has to be created in this diagram the tables that store the data is being structured in the system that is in development there eight tables which represent how the data is being collected and every table has its own entities for example in the freelancers table the ID and the expertise will be stored and that table has a relation with the location where it stores the locations of every freelancers and client that are in the system, the relations are based on one to many and many to many relations and it's calculated based on if that table is used by many other tables or not.

4.6. Chapter Summary

In the conclusion, the functionality of every entity of the system that is in development has been covered throughout this chapter starting off with the use case diagram which has been used to analyze how the components will be structured and what will be included in the system in relation with the interactions with the users after finishing with the use case the next step is the sequence diagram which shows how those components for every different user will interact with each other, the next step after the sequence diagram is the activity diagram which shows how those components will flow step by step, and the next section which is the design phase in this part the class diagrams and the entity relations of the system will be created to understand the overall design of the system, the first step is the class diagram in this part which is mostly for the developers to understand what will be the classes and how the database will be structured and how the system architecture will be structured in the overall design of the system.

Chapter 5

Implementation, and Testing

5.1. Introduction

The implementation and testing is a process of discussing how the Freelance Marketplace has been created and what approach has been used to achieve the goals of the stakeholders and that tests has been applied in order to know the user is satisfied or not with the system that has been created for the public.

5.2. System Implementation

The architectural design has been used to make the process of implementing the system easier and also to deliver the best version of the system to the user and to satisfy the user needs and also the stakeholders that set the requirements.

5.2.1. System Architecture Implementation

Model-View-ViewModel (MVVM) is a software design style that separates program logic from user interface controls. Model-view-binder, sometimes known as MVVM, was invented by Microsoft architects John Gossman and Ken Cooper. By segmenting programs into manageable modules, MVVM, like many other design patterns, speeds up the creation, updating, and reuse of code. The design is often used in presentation software for Windows and online graphics.

These figures down below show the code of the system from the view model to model to view how it was creating the model and interacting with the view and the languages that has been used in this system are React, JavaScript, and tailwind library also firebase for the backend and to store data.

src > pages > Home.jsx > Home

```jsx
import React from "react";
import { AiFillGithub } from "react-icons/ai";
import { useAuth } from "../context";
import login from "../assets/login.svg";

const Home = () => {
  const { authenticateWithGitHub } = useAuth();
  return (
    <div>
      <nav className="flex justify-between px-8 pt-6 items-center">
        <h1 className="text-2xl font-bold">
          Freelance<span style={{ color: '#f59e0b' }} className="text-button">Marketplace</span>
        </h1>

      </nav>
      <main>
        <img
          src={login}
          alt="landing-img"
          className="m-auto w-[28rem]"
        />
        <div className="text-2xl font-bold text-center mb-6 md:text-4xl">
          <p>
            Find interesting <span style={{ color: '#f59e0b' }} className="text-button">remote jobs</span>
          </p>
          <p>and collaborate with people.</p>
        </div>
        <button
          className="bg-black mx-auto text-xl text-white flex items-center gap-2 px-8 py-2 rounded hover:opacity-90"
          onClick={authenticateWithGitHub}
```

Figure 5-1 Home Page

src > pages > ErrorPage.jsx > ErrorPage

```jsx
import { AiFillHome } from "react-icons/ai";
import { Link } from "react-router-dom";
import errorPage from "../assets/errorPage.svg";

export default function ErrorPage() {
  return (
    <div>
      <nav className="flex justify-between px-8 pt-6 items-center">
        <h1 className="text-2xl font-bold">
          Freelancer<span className="text-button">Marketplace</span>
        </h1>

      </nav>
      <main>
        <img
          src={errorPage}
          alt="landing-img"
          className="mt-10 mx-auto md:m-auto w-[28rem]"
        />
        <div className="font-bold text-center mb-6">
          <p className="text-2xl md:text-4xl">
            404<span className="text-button text-md"> Page not found</span>
          </p>
        </div>
        <Link to="/">
          <button className="bg-black mx-auto text-xl text-white flex items-center gap-2 px-8 py-2 rounded hover:opacity-90">
            <AiFillHome />
            Go Home
          </button>
        </Link>
```

Figure 5-2 Error Page

Figure 5-3 Project Form

Figure 5-4 Project Listing

src > pages > UserListing.jsx > UserListing

```jsx
import React from "react";
import {
  ProjectListingNavbar,
  ProjectListingSidebar,
  UserListingCard,
} from "../components";
import EmptyPage from "../components/EmptyPage";
import { useData } from "../context";

const UserListing = () => {
  const { dataState } = useData();
  const { collabUserList } = dataState;
  return (
    <div>
      <ProjectListingNavbar />
      <main className="w-11/12 m-auto">
        <ProjectListingSidebar />
        <h1 className="text-4xl text-heading font-bold text-center">
          Find teammates!
        </h1>
        <div className="flex flex-wrap justify-center gap-4 my-6">
          {collabUserList.length > 0 ? (
            collabUserList.map((user, index) => {
              return <UserListingCard key={index} userInfo={user} />;
            })
          ) : (
            <EmptyPage />
          )}
        </div>
```

Figure 5-5 User Listing

src > pages > UserProfile.jsx > UserProfile

```jsx
import { ToggleSwitch, ProjectCard, ProjectListingNavbar } from "../components";
import { BsGlobe, BsTwitter } from "react-icons/bs";
import { useAuth } from "../context";
import EmptyPage from "../components/EmptyPage";
function UserProfile() {
  const { currentUser } = useAuth();
  const { avatar, displayName, bio, blog, twitter } = currentUser;

  return (
    <>
      <ProjectListingNavbar />
      <div className="">
        <section className="flex flex-col justify-center md:flex-row md:items-center md:justify-between md:mx-4 bg-gradien
          <div className="flex flex-col items-center md:flex-row md:gap-4">
            <div className="h-44 w-44 rounded-full overflow-hidden">
              <img src={avatar} alt="user-pic" />
            </div>
            <div className="flex flex-col text-center w-60 md:text-left">
              <h1 className="font-bold text-xl">{displayName}</h1>
              <p className="text-sm ">{bio}</p>
            </div>
          </div>
          <div className="socials flex flex-col-reverse items-center mt-2 gap-3 md:items-end">
            <ToggleSwitch />
            <div className="socials flex gap-3 mt-4 justify-end text-3xl">
              {blog && (
                <a
                  href={blog}
                  target="_blank"
                  className="flex items-center gap-2 hover:text-button"
```

Figure 5-6 User Profile

src > components > ProjectCard.jsx > ProjectCard

```jsx
import { MdOutlineKeyboardArrowRight } from "react-icons/md";
import { AiFillDelete } from "react-icons/ai";
import { useAuth } from "../context";
import { doc, updateDoc } from "firebase/firestore";
import { db } from "../firebase/firebase";
import {
  getUser,
  notifyError,
  notifyProjectDeletion,
} from "../helper-functions";

function ProjectCard({ projectInfo }) {
  const {
    projectName,
    gitHubLink,
    category,
    CompanyName,
    tagArray,
    description,
    techStackArray,
  } = projectInfo;

  const { currentUser, setCurrentUser } = useAuth();

  const deleteHandler = async (notifySuccess, notifyError) => {
    try {
      const docRef = doc(db, "users", currentUser.uid);
      const projectArr = currentUser.projects;
      const filteredProjectArr = projectArr.filter(
        (project) => project.projectName !== projectName
```

Figure 5-7 Project Card

src > components > ProjectListingCard.jsx > ProjectListingCard

```jsx
import React from "react";
import { AiOutlineLink } from "react-icons/ai";
import { BsBoxArrowUpRight } from "react-icons/bs";

const ProjectListingCard = ({ projectInfo }) => {
  const {
    projectName,
    gitHubLink,
    contactLink,
    CompanyName,
    category,
    tagArray,
    description,
    techStackArray,
  } = projectInfo;

  return (
    <div className="p-8 md:w-2/5 shadow-lg flex flex-col gap-2 rounded-lg">
      <div className="flex justify-between items-center gap-4 flex-wrap">
        <div className="flex items-center gap-2">
          <p className="text-3xl font-semibold text-heading">{projectName}</p>
          <a href={gitHubLink} target="_blank" rel="noreferrer">
            <AiOutlineLink className="text-2xl" />
          </a>
        </div>
        <p className="text-button text-xs font-bold border-2 border-button p-2 rounded-full">
          {category.toUpperCase()}
        </p>
      </div>
      <div className="flex gap-1 flex-wrap">
```

Figure 5-8 Project Listing Card

src > components > ProjectListingSidebar.jsx > ...

```jsx
import React from "react";
import { NavLink } from "react-router-dom";

const SidebarLink = ({ name, to }) => {
  return (
    <li className="nav-item flex-auto text-center">
      <NavLink
        to={to}
        className={({ isActive }) =>
          isActive ? "sidebar-link active" : "sidebar-link"
        }
      >
        {name}
      </NavLink>
    </li>
  );
};

const ProjectListingSidebar = () => {
  return (
    <ul className="nav nav-tabs flex flex-row flex-wrap list-none border-b-0 pl-0 mb-4">
      <SidebarLink name={"Projects"} to="/listing/projects" />
      <SidebarLink name={"Users"} to="/listing/users" />
    </ul>
  );
};

export { ProjectListingSidebar };

```

Figure 5-9 Project Listing Sidebar

src > components > Search.jsx > Search

```jsx
import React from "react";

const Search = ({ setSearchInput, searchInput }) => {
  return (
    <div>
      <input
        className="bg-gray-50 border-2 h-12 text-base m-auto border-gray-100 outline-none text-gray-900 rounded-lg
        onChange={(e) => setSearchInput(e.target.value)}
        placeholder="Search for Jobs"
        value={searchInput}
      />
    </div>
  );
};

export { Search };

```

Figure 5-10 Search

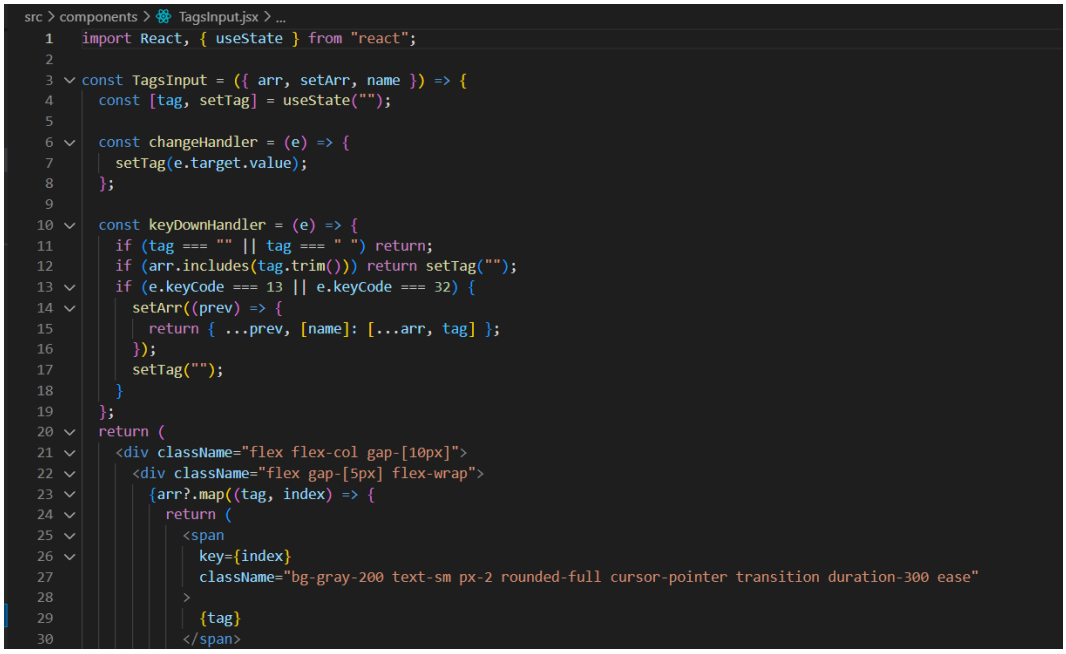


Figure 5-11 Tag Input

5.2.2. Database Management System (DBMS) Implementation

Data is a group of discrete, distinctive pieces of information. It may be used as text, numbers, media, bytes, etc., among other things. It might be stored electronically, on paper, etc. The Latin word "datum," which means "a single piece of information," is where the word "data" originates. Here, "datum" is used in the plural. Data are pieces of information that may be quickly sent and processed by computers. Changes can be made to data. To save the data, utilize them for daily communication, and store the tasks in the system, Freelance Marketplace Firebase has been used in this system.

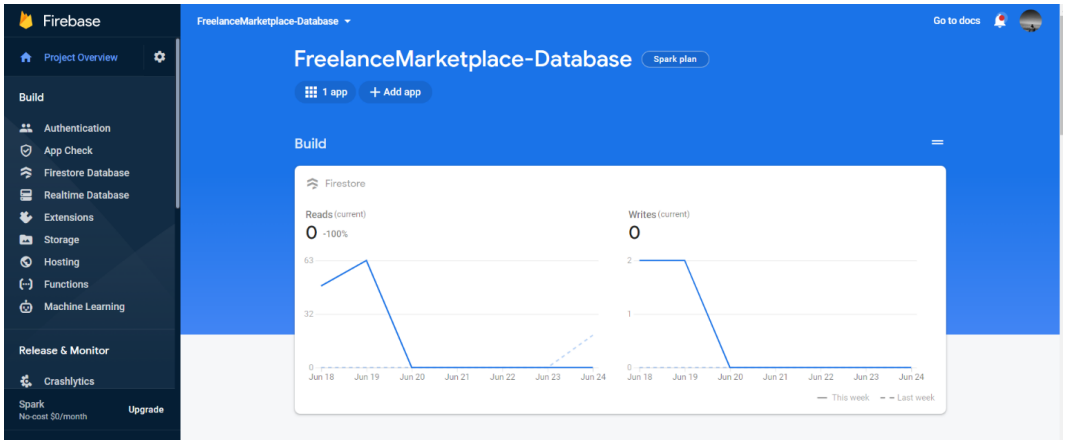


Figure 5-12 Firebase Cloud Store

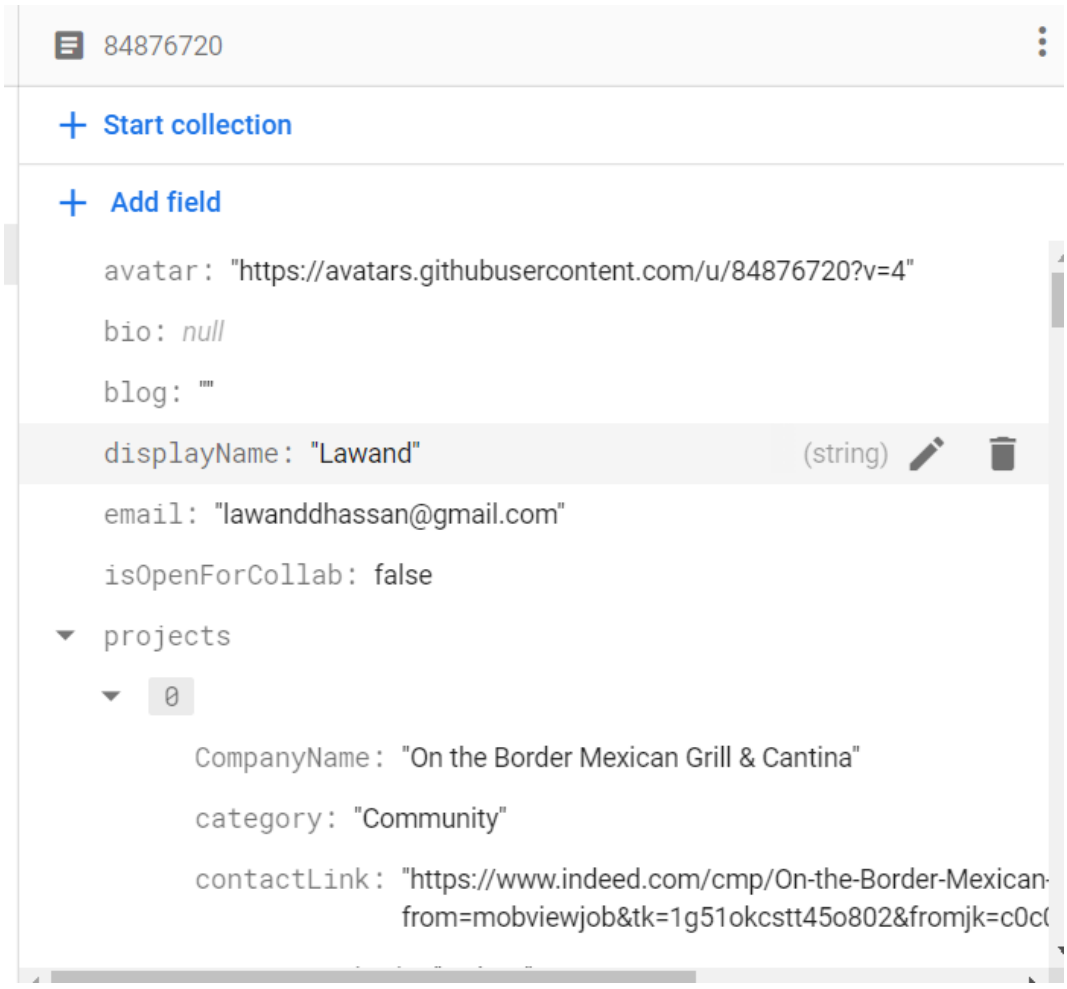


Figure 5-13 Cloud Data

src > firebase > firebase.js > ...

```
1   import { initializeApp } from "firebase/app";
2   import { getAuth } from "firebase/auth";
3   import { getFirestore } from "firebase/firestore";
4
5   const firebaseConfig = {
6       apiKey: "AIzaSyAs75xrKaOsDTZwjmvbaryAwsEtdrzGu-s",
7       authDomain: "freelancemarketplace-database.firebaseapp.com",
8       projectId: "freelancemarketplace-database",
9       storageBucket: "freelancemarketplace-database.appspot.com",
10      messagingSenderId: "614851547180",
11      appId: "1:614851547180:web:6cb84397b9db4fd037c189"
12  };
13
14  // Initialize Firebase
15  const app = initializeApp(firebaseConfig);
16  const db = getFirestore(app);
17  const auth = getAuth(app);
18
19  export { app, auth, db }
```

Figure 5-14 Firebase Authentication

5.3. System Testing

Because this project employs agile, each function that will be built needs to be tested in order to publish the highest quality system to the consumers of this freelance marketplace system. System testing is a process that checks and verifies for all technical requirements. In order to test the system with a large number of users, two strategies—user acceptability testing and black box testing—will be used.

5.3.1. Black-Box Testing

Software testing that assesses an application's functioning without looking at its internal components is known as "black-box" testing. Practically every level of software testing, including unit, integration, system, and acceptance testing, may be conducted using this test methodology. The main purpose of Black Box is to

concentrate more on the input and output of the software system. This test may be used at several stages of software testing, and one benefit of this approach is that it can be applied to the entire system, from the beginning to the finish. The consumer won't have to worry about any troubles with the program that will be supplied to them if they utilize this technique, and they will expect

5.4. Chapter Summary

In the end the implementation went successfully and the system had no issues with delivering the requirements that were set by the user and the architecture also the design pattern that has been chosen were suitable for the system that was in development and overall, performance of the system run smoothly as expected.

Chapter 6

Conclusion

6.1. Introduction

Through the project that is on the freelance marketplace which was designed to act as a community for the freelancers of the system and the clients that are looking for someone to work for them as a freelancer employee, the goal of the system was to develop an efficient and categorized marketplace for those who are searching for a freelancer and also a place for those freelancers to showcase their work to get reviews based on the client's feedback and rank their performance the objectives of this system was to develop a system that can compete with the already existing system and try to include the features that are not included in the currently existing systems, and as one of the main objects of the system was to evaluate the current systems and analyze them to understand what needs to be included in the system that is in development also to try and improve the system in the weak areas of the current system.

6.2. Achievement

As one of the parts of the objectives was to analyze the current system that is already in the market and to create a system that has most of the main features that already exist in the current systems so far, the first stage has been completed which was the requirement analysis this also counts as the first sprint of the selected methodology which was agile in the agile development which is a user center-based methodology the requirements can be changed in the future based on the user inputs from surveys and questioners if there was an improvement that was seen to take into consideration, this counts as one the main advantages of agile methodology.

6.3. Suggestions for the project

In the future based on the feedback from the users and the assessment that the users set many changes might come into consideration so far based on the analysis of the current systems there are many features that need to be added to the system for example this system shall provide the proper payment methods based on the countries that the users use this application especially for the local countries this will benefit them a lot because they cannot access to the global methods that are provided, for example, many services like PayPal have been banned in Iraq which is the country that the development of the system has been involved so adding the proper payment will increase the rate of users in this local area and this will imply for the other countries as well.

REFERENCES

Kim Parker, Juliana Horowitz and Rachel Minkin. (2020, December 9). How the Coronavirus outbreak has- and hasn't-changed the way Americans work. Retrieved from Pew Research Center: https://www.pewresearch.org/social-trends/2020/12/09/how-the-coronavirus-outbreak-has-and-hasnt-changed-the-way-americans-work/

Michael Dunn, Fabian Stephany, Steven Sawyer, Vili Lehdonvirta. (2020, June 23). When Motivation Becomes Desperation: Online Freelancing During the COVID-19 Pandemic. Retrieved from SocArXiv: 10.31235/osf.io/67ptf

Sherry Sullivan & Lisa Mainiero & Siri Terjesen. (2007). What Motivates Entrepreneurs? An Exploratory Study of the Kaleidoscope Career Model and Entrepreneurship. The Journal of Applied Management & Entrepreneurship, 4-19.

Tobin Turner & W. Wesley Pennington III. (2015). Organizational networks and the process of corporate entrepreneurship: how the motivation, opportunity, and ability to act affect firm knowledge, learning, and innovation. Small Business Economics, 447-463.

Anikó Hannák & Claudia Wagner & David Garcia & Alan Mislove & Markus Strohmaier & Christo Wilson. (2017). Bias in Online Freelance Marketplaces: Evidence from TaskRabbit and Fiverr. Association for Computing Machinery, 1914 - 1933.

Chowdhury, K. (2021, March 26). Top 10 features of NodeJS for Web Developers. Retrieved from Kunal Chowdhury: https://www.kunal-chowdhury.com/2021/03/top-ten-nodejs-features.html?session=202216164&pos=4062.39990234375&utm_source=linkedin.com&utm_medium=social

Clark. (2021, September 28). 4 Things To Know Before You Sign Up for FlexJobs. Retrieved from Clark.com: https://clark.com/employment-military/flexjobs-work-from-home-jobs/

Dean, B. (2021, April 20). Fiverr Usage and Growth Statistics: How Many People Use Fiverr in 2021? Retrieved from Backlinko: https://backlinko.com/fiverr-users

Dean, B. (2021, May 13). Upwork Revenue and Client Stats (2021). Retrieved from Backlinko: https://backlinko.com/upwork-users

Ellingwood, J. (2015, January 28). Apache vs Nginx: Practical Considerations. Retrieved from Digital Ocean: https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

Grant, C. (2019, August 16). As Toptal employees and investors demand equity, founder points to the fine print. Retrieved from Hustle: https://thehustle.co/toptal-equity-convertible-note/

Insignares, A. (2021, March 10). React Pros and Cons: What are the Advantages and Disadvantages of ReactJS? Retrieved from koombea: https://www.koombea.com/blog/react-pros-and-cons-what-are-the-advantages-and-disadvantages-of-reactjs/

Keegan, L. (2021, October 12). Freelance Statistics 2022. Retrieved from SkillScouter: https://skillscouter.com/freelance-statistics/

Lakhani, H. (2015, August 27). How freelancer.com became the biggest marketplace for freelancers in 6 years- Startup Story #19. Retrieved from JungleWorks: https://jungleworks.com/how-freelancer-com-became-the-biggest-marketplace-for-freelancers-in-6-years-startup-story-19/

V., A. (2021, December 03). 15 Best Freelance Websites to Find Work in 2022. Retrieved from Hostinger Tutorials: https://www.hostinger.com/tutorials/best-freelance-websites

Velebny, R. (2021, April 22). Freelancer.com Q1 FY21: Record breaking-quarter. Retrieved from AimGroup: https://aimgroup.com/2021/04/22/freelancer-com-q1-fy2021-a-record-breaking-quarter/

Wikins, M. (2021, March 29). What is the Public Cloud and AWS? Retrieved from inform it: https://www.informit.com/articles/article.aspx?p=2982115&seqNum=4

Agile. (2021, September 6). The 9 Key Benefits of Using the Agile Methodology. Retrieved from Kissflow: https://kissflow.com/project/agile/benefits-of-agile/

Alam, A. (2019, November 27). Importance of Software Design. Retrieved from Start it up: https://medium.com/swlh/importance-of-software-design-7ffea48ede17#:~:text=It%20is%20about%20being%20able,from%20headaches%20down%20the%20line.

Chervenikova, M. (2020, July 14). What Are Feedback Loops and Why You Need to Implement Them. Retrieved from kanbanize: https://kanbanize.com/blog/feedback-loops/#:~:text=In%20Agile%2C%20feedback%20loops%20help%20us%20regularly%20identify%20areas%20for%20improvement.&text=Feedback%20loops%20help%20teams%20to,improved%20team%20performance%2C%20and%20agility.

Chowdhury, K. (2021, March 26). Top 10 features of NodeJS for Web Developers. Retrieved from KunalChowdhurry: https://www.kunal-chowdhury.com/2021/03/top-ten-nodejs-features.html?session=202216164&pos=4062.39990234375&utm_source=linkedin.com&utm_medium=social

D., D. (2019, June 14). The Importance of UI/UX Design. Retrieved from igexsolutions: https://www.igexsolutions.com/blog/the-importance-of-ui-ux-design/#:~:text=The%20UX%2FUI%20Design%20of,users%20of%20the%20specific%20application.&text=The%20UI%20and%20UX%20Design,what%20they%20are%20looking%20for.

Ellingwood, J. (2015, January 28). Apache vs Nginx: Practical Considerations. Retrieved from digitalocean: https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

Goss , J. (2007, November 8). 10 reasons why use cases are indispensable to your software development project. Retrieved from TechRepublic: https://www.techrepublic.com/blog/software-engineer/10-reasons-why-use-cases-are-indispensable-to-your-software-development-project/

Insignares, A. (2021, March 10). React Pros and Cons: What are the Advantages and Disadvantages of ReactJS? Retrieved from koombea: https://www.koombea.com/blog/react-pros-and-cons-what-are-the-advantages-and-disadvantages-of-reactjs/

Pickavance, M., & Turner, B. (2022, January 17). Best survey tools in 2022. Retrieved from TechRadar: https://www.techradar.com/best/best-survey-tools

van der Hoek, J. (2021, October 11). Agile Explained: The 5 Stages of the Agile Development Lifecycle. Retrieved from mendix: https://www.mendix.com/blog/agile-software-development-lifecycle-stages/

Wijetunge, R. (2017, June 30). ‘Ready’ for Implementation, ‘Done’ for Deployment!! Retrieved from knowledgehut: https://www.knowledgehut.com/blog/agile/ready-implementation-done-deployment

Windsor, G. (2020, February 28). 5 Stages of the Agile System Development Life Cycle. Retrieved from medium: https://medium.com/brightwork-collaborative-project-management-blog/5-stages-of-the-agile-system-development-life-cycle-brightwork-com-a207bdf61696

Phillips, D. (2000). The Software Manager's Handbook. Los Alamitos: IEEE Computer Society.

Staff, Q. (2021, April 30). 17 Top Online Survey Research Companies. Retrieved from Quirk's Media: https://www.quirks.com/articles/17-top-online-survey-research-companies

Appendix A

Software Requirements Document (SRS)

Software Requirements Specification

Freelance Marketplace System

Version 1.0

25th June 2022

Department and Faculty
Software Engineering
Prepared by: Lawand Hassan Mahmood

Revision Page

a. Overview

This is the first draft of SRS for the Design and Implementation of Freelance Marketplace.

b. Target Audience

Stakeholders.

c. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	Lawand Hassan	First Draft	24th June 2022

2. Introduction

Design And Implementation of Freelance Marketplace is described in detail in this Software Requirements Specification (SRS), which includes both functional and non-functional requirements. With a collection of use cases, activity diagrams, and sequence diagrams, it depicts the interaction between the user and the system to represent the flow of the system and the declared objects both an activity diagram and sequence diagram are being elaborated.

2.1 Purpose

The purpose of this SRS is shown as below:

i. To identify and analyse both functional and non-functional requirements of the system.

ii. To describe the overall flow of the system and the interaction with the user.

iii. To serve as an input to software Design Document (SDD).

iv. To act as the validation check of the product that is being built.

2.2 Scope

This software system to be developed is a Freelance Marketplace, it is a website that offer users to post jobs online so others can see it and collaborate with it. The targeted user of the system are people around the globe. There are two categories of user, which is Admin and normal user.

The system works specifically for both type of user which are the people who are searching for jobs and also for the ones that trying to find someone to collaborate with them and both users can log in and post jobs they can also search for jobs using the categories that has been provided and the admin can delete jobs, posts and users. The system will be created as website using React which is JavaScript framework and its an open-source UI software that allows users to create any web-based applications for all the platforms. The system main goal is to provide a centralized job portal for users to interact with each other and to post and see the available jobs that has been offered.

2.3 Definitions, Acronyms and Abbreviation

Table 1 Definitions, Acronyms and Abbreviation

Term	Definition
FR	Functional Requirement
NFR	Non-Functional Requirement
SDD	Software Design Document
SRS	Software Requirement Specification
React	React is a JavaScript library for building user interfaces.
API	API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function

2.4 Overview

This document is divided into three sections, each section contains several subsections and the main sections are shown below:

i. Introduction

   This section describes an overview of the SRS

ii. Overall Description

   This section provides the factors that affect the software product. It acts as the background for the requirements of the system.

iii. Specific Requirements

   This section provides the requirements of the system to be developed in details with the interaction between user and the system.

3. Overall Description

There are 2 main actors in the system which are User and Admin and Freelancer is inherited from the User. The user case is divided into 4 modules and those modules are Manage Newsfeed Module, Module Newsfeed, and Admin Module, also the User Module which contains the authentication in the use case. Figure 1 shows the use case diagram of the system.

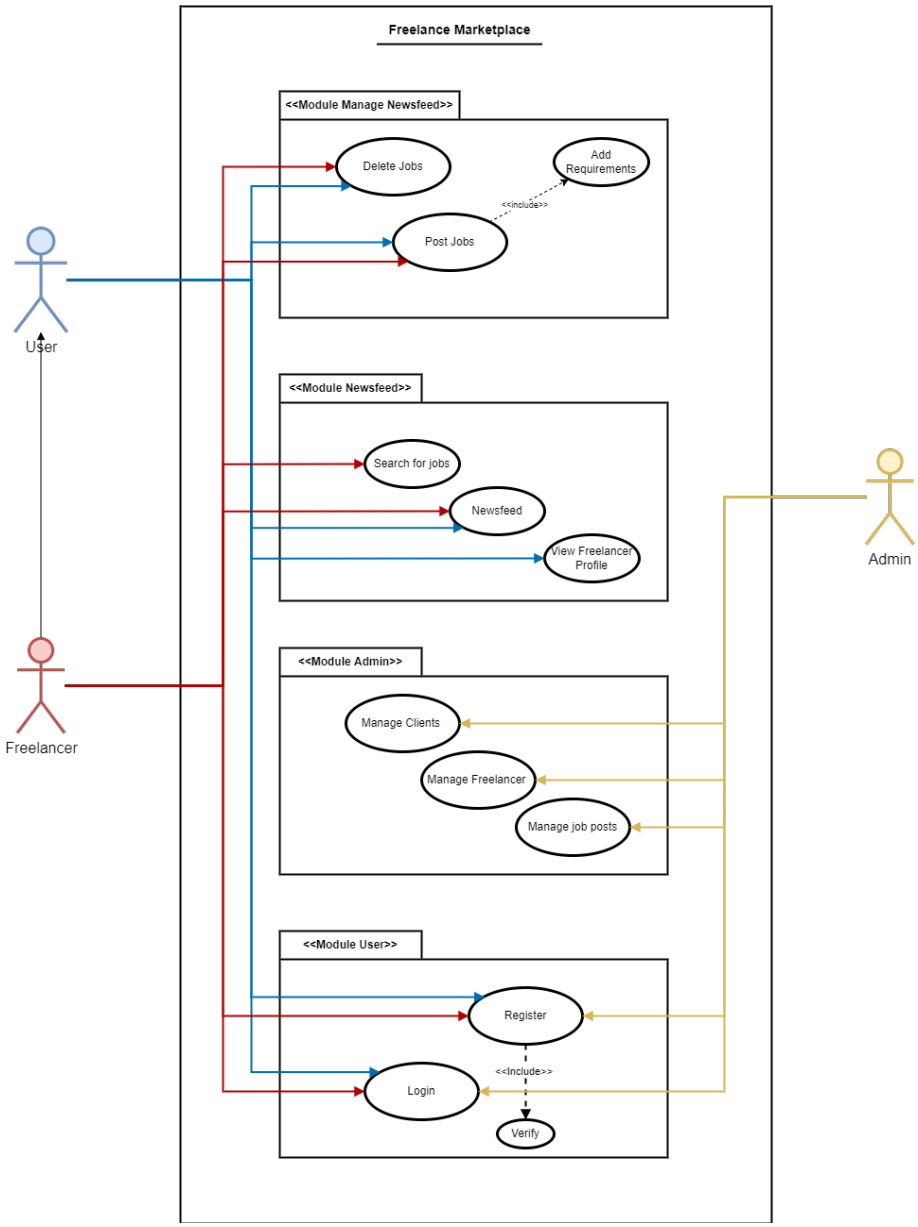


Figure 1: Use Case Diagram of <Freelance Marketplace>

3.1 Product Perspective

Freelance Marketplace is a web-based system that will be built to run on all operating systems, Freelance Marketplace is created so that a user whether an admin, a user or a freelancer, can post and search for online jobs in professional manner and on a centralized platform, Admin can manage the data that is available in the database and the user authentication and the user can see the available jobs that has been posted

and they can collaborate with the author of the job owner. This website is very beneficial during this pandemic for all people who are looking to find jobs remotely without leaving their house, and the technology that has been built with relies on cloud-based database to centralize and synchronize data across all operating systems.

3.2 Product Functions

The functionalities provided by the system are shown as follows:

a) Module User

i. UC01 – Register

This use case allows the user to register a new account using their GitHub on the Freelance Marketplace application, the user can sign up the account as Freelancers or any other user.

ii. UC02 – Login

The use case allows users to login into the system by using their already registered account which is the GitHub account.

b) Module Manage Newsfeed

i. UC03 – Post to Newsfeed

This use case allows users that are logged in into the system to post into the newsfeed, but they have to fill the form of requirements before submitting the post.

ii. UC04 – Post Jobs

This use case allows users to post jobs into the system so other users can see it and interact with it, they also have to fill a form that ask for the details of the job that is required.

c) Module Newsfeed

i. UC05 – Search for Jobs

This use case allows users to search for jobs that are available in the system by using the search bar that is provided in the dashboard.

ii. UC06 – Newsfeed

This use case allows users to access to the dashboard and to see the available data that is in the system with having access to this one the user cannot see the posted jobs or the available users.

d) Module Admin

i. UC07 – Manage User

This use case allows admins to manage the information that is provided by the users and they can manipulate it as they want too.

ii. UC08 – Manage

Since the admin has access to the database so this use case allows the admin to edit any data of the posts that has been made in the system.

3.3 User Characteristics

This section describes all the actors of the system. The main actors of the Freelance Marketplace are User, Admin we have also Freelancer but it is a sub user because it does the same job as user.

3.3.1 User

User is a parent actor that can do many functionalities and any normal user has to be registered into the system in order for them to have access to the Freelance Marketplace the user will login into the system using their GitHub account they can manage their profile information by using their GitHub account and they can also post jobs and search for other people's jobs, also they can see other people that are open for collaboration.

3.3.2 Admin

Admin is a user who does not has to be registered because they have a different interface and the admin has access to all user information and user data also, they can change any data that they want through the database which is firebase.

4. Specific Requirements

This section describes the functional and non-functional requirements of the system, use case specifications and respective sequence diagram and activity diagram for each use cases are also presented in this part of the SRS. Figure 2 shows the class diagram or the domain model of the system.

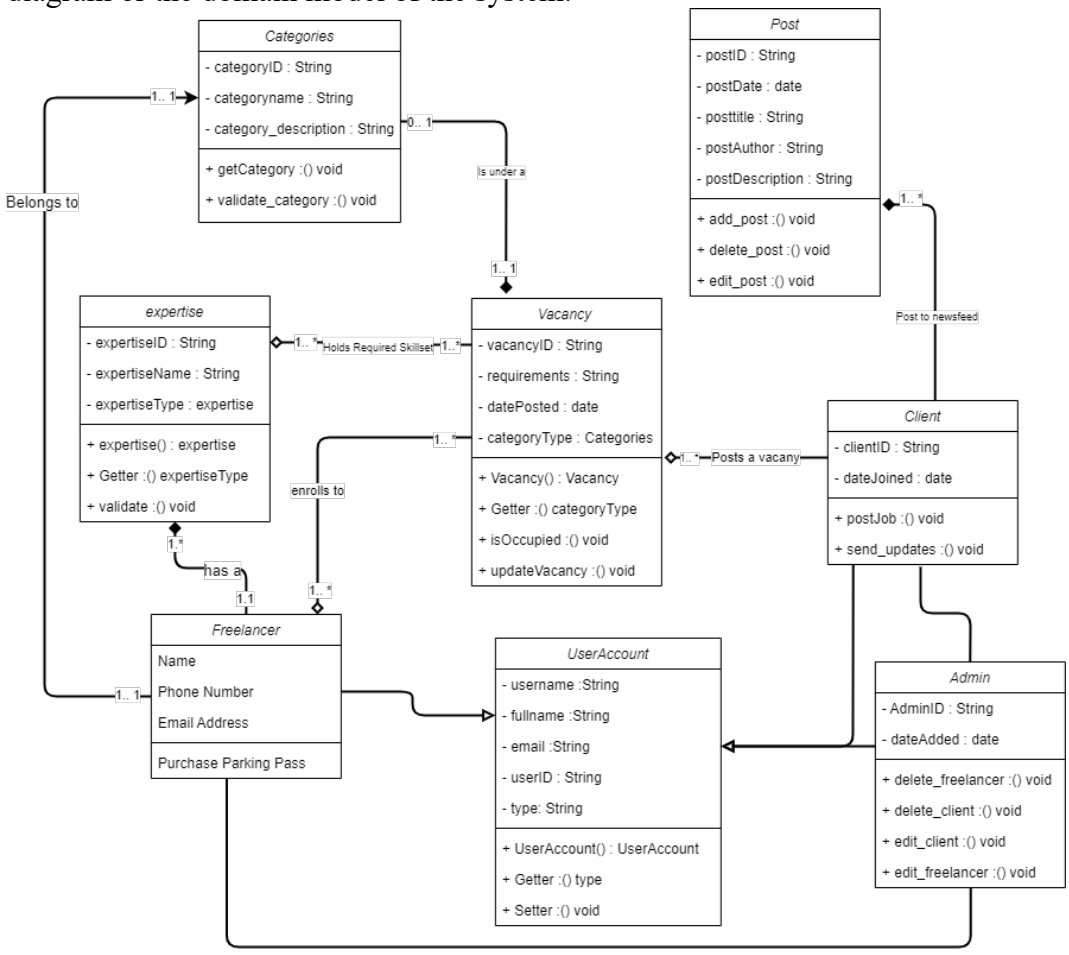


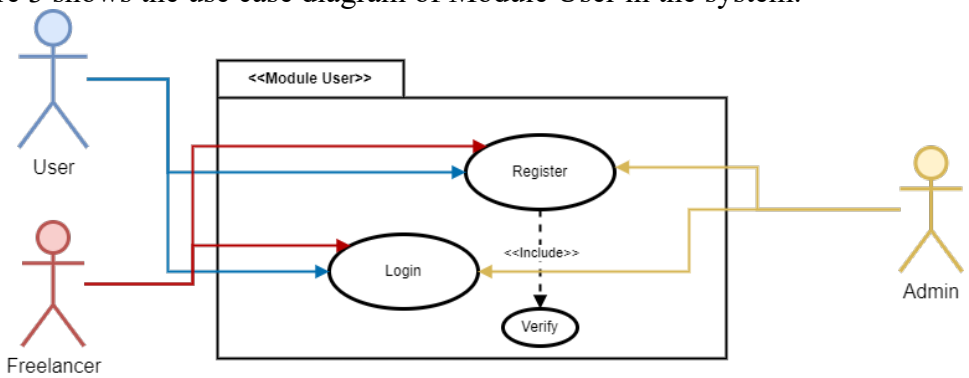
Figure 2: Domain Model of <Freelance Marketplace>

4.1 System Features

This subsection describes the function requirements of each module with respective use case specification, sequence diagram and activity diagram.

4.1.1 Module <User>

Figure 3 shows the use case diagram of Module User in the system.



UCD01: <Use Case Diagram of Module User>

The functional requirements of Module User are listed as below:

i. FR001 – Register

The system shall allow user to register an account with the system to have access to the systems data.

ii. FR002 – Login

The system shall allow user to login into the system to gain access of the system functionalities.

iii. FR003 – View Profile Information

The users should be able to see their profile picture on the home screen.

iv. FR004 – Validate Account

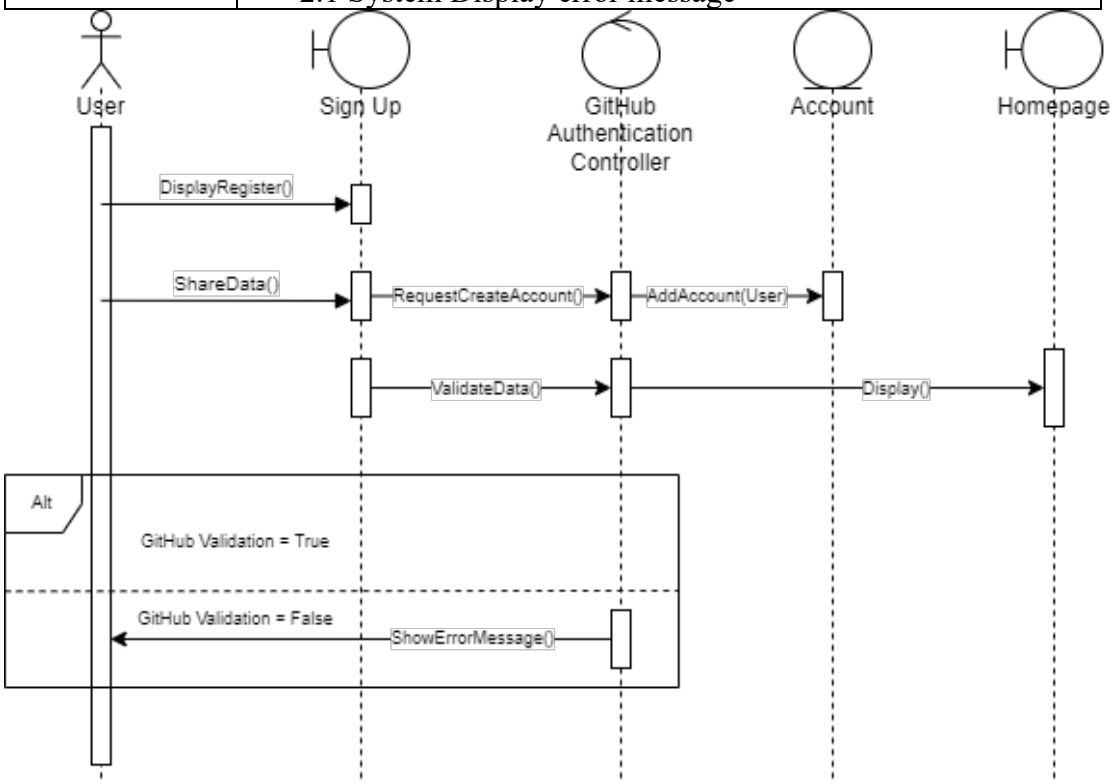
The system authentication must be able to authenticate the account by using the GitHub API that is provided by firebase.

4.1.1.1 UC01: Register

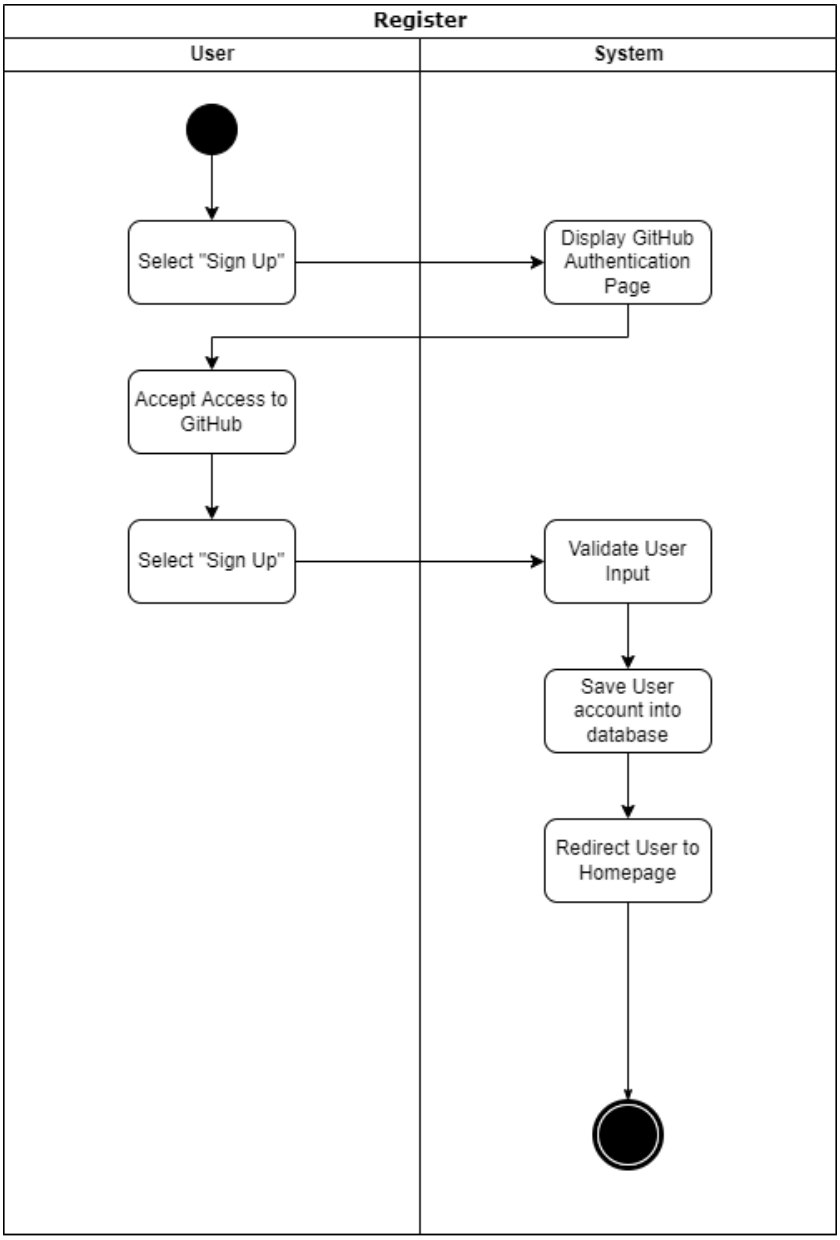
Table 2: Use Case Description for Register

Use Case ID	UC001
Use Case Name	Register
Brief Description	This use case allows user to create an account on the system
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step
Normal Flow	1. On the login page user can select Sign up 2. System will redirect user to the authentication page by GitHub. 3. User agree to share their personal data with system. 4. Users click on sign Up 5. The system will check validation of the form if wrong it will start from flow 2 again else continue 6. System will save the user account on the database 7. System will redirect user to home page. 8. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User account is added in account database.

	1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message



SD01: Sequence Diagram of Register



AD01: Sequence Diagram of Register

4.1.1.2 UC02: Login

Table 3: Use Case Description for Login

Use Case ID	UC002
Use Case Name	Login
Brief Description	This use case allows user to login into the system
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step. 2. The User must be registered into the system.
Normal Flow	1. On the login page user can select Login in by GitHub. 2. System will redirect user to the authentication page by GitHub.

	3. The system will check validation of the form if wrong it will start from flow 2 again else continue 4. System will redirect user to home page. 5. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User successfully login into the system. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message

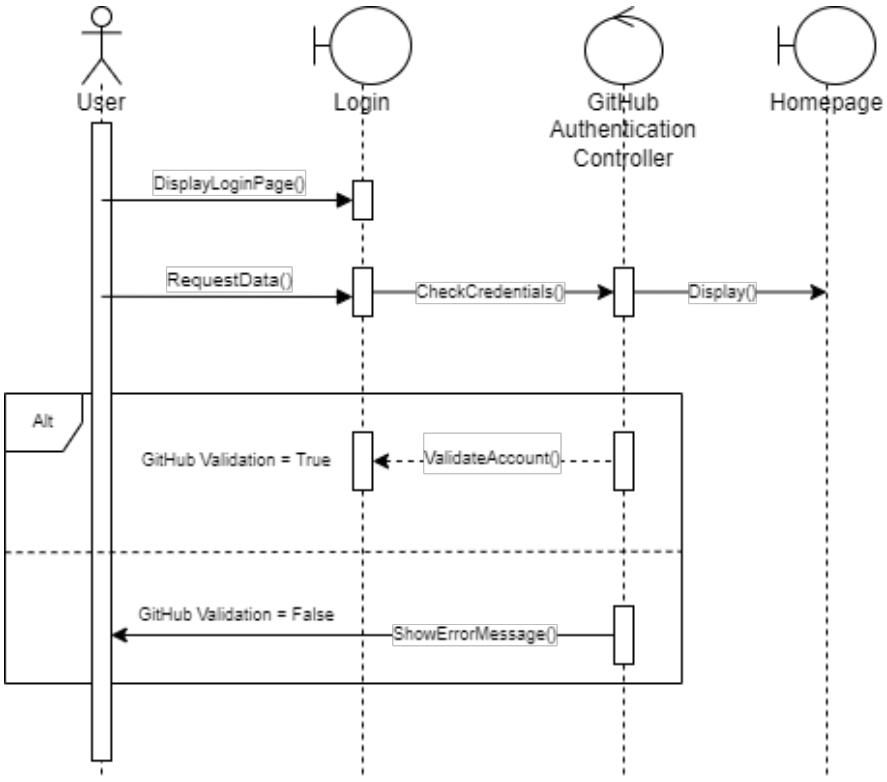


Figure 3 System Sequence Diagram for Login

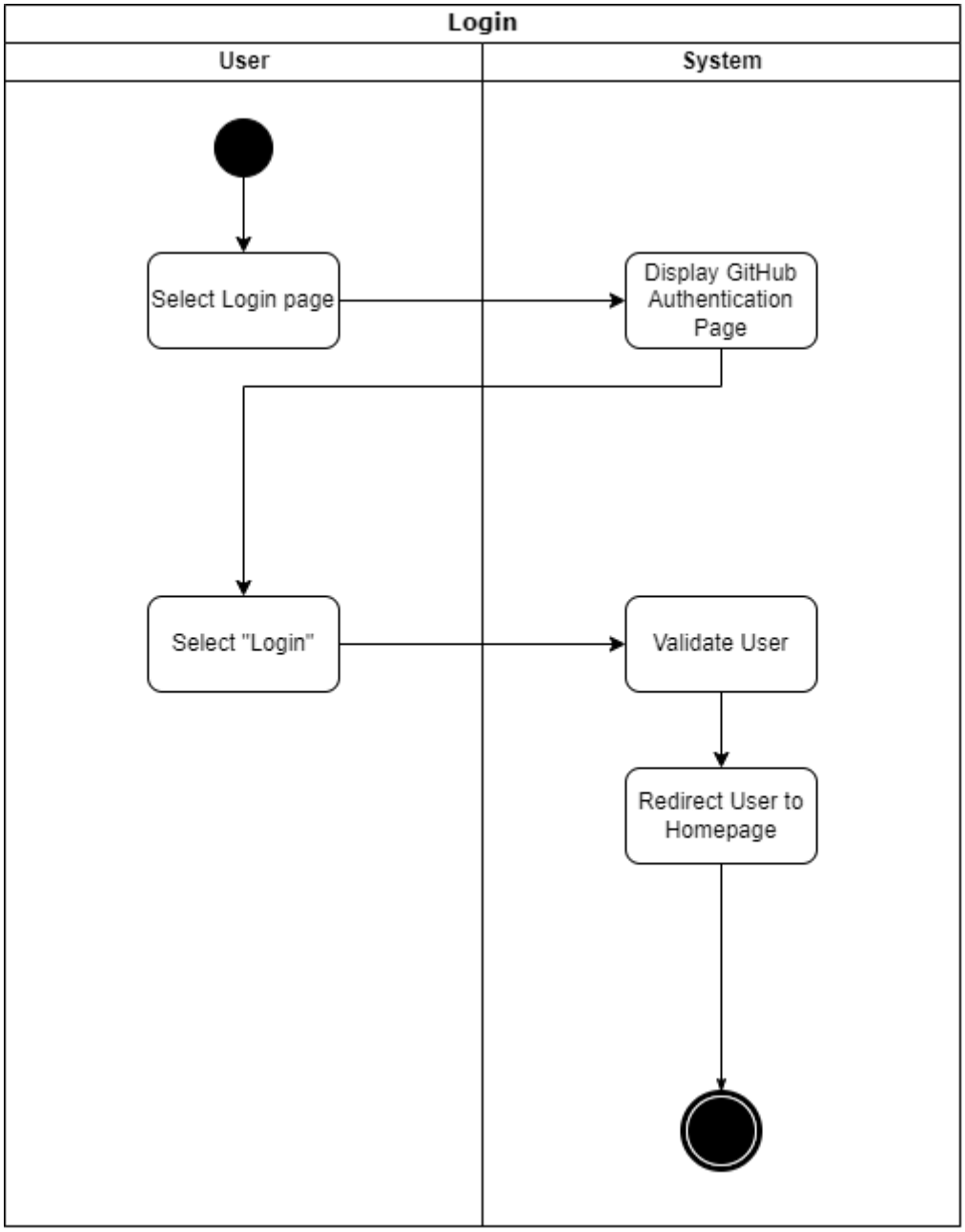
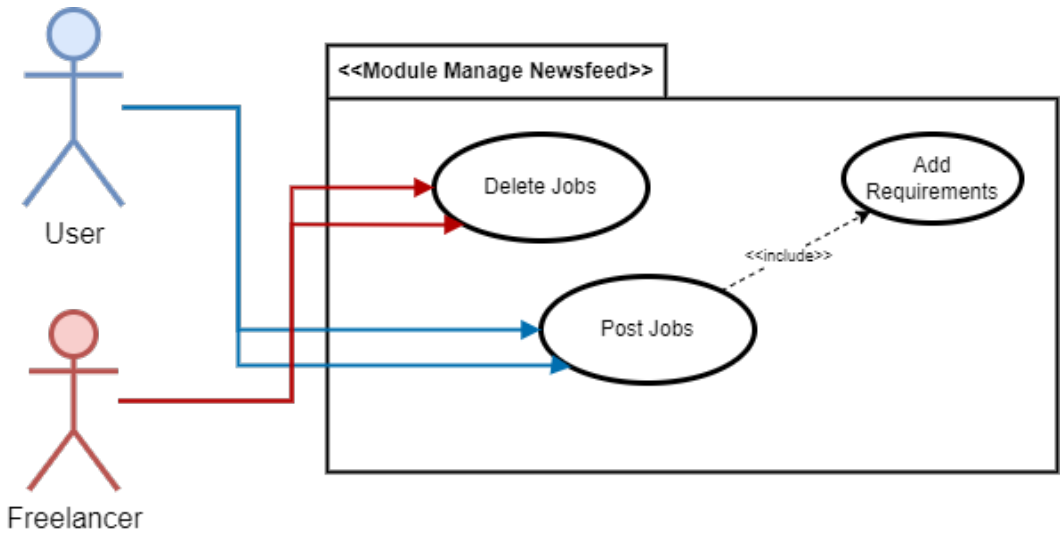


Figure 4 Activity Diagram for Login

4.1.2 Module Manage Newsfeed

Figure 4 shows the use case diagram of Module Manage Newsfeed in the system.



UCD02: <Use Case Diagram of Module Manage Newsfeed>

The functional requirements of Module Manage Newsfeed are listed as below:

i. FR005 – Post to Jobs

The system shall allow the User to create post and publish it on the newsfeed, this will be seen by other users.

ii. FR006 – Delete Posts

The system should let the users delete the post that they've made by seeing the list of the posts in their profile.

iii. FR007 – Add Requirements

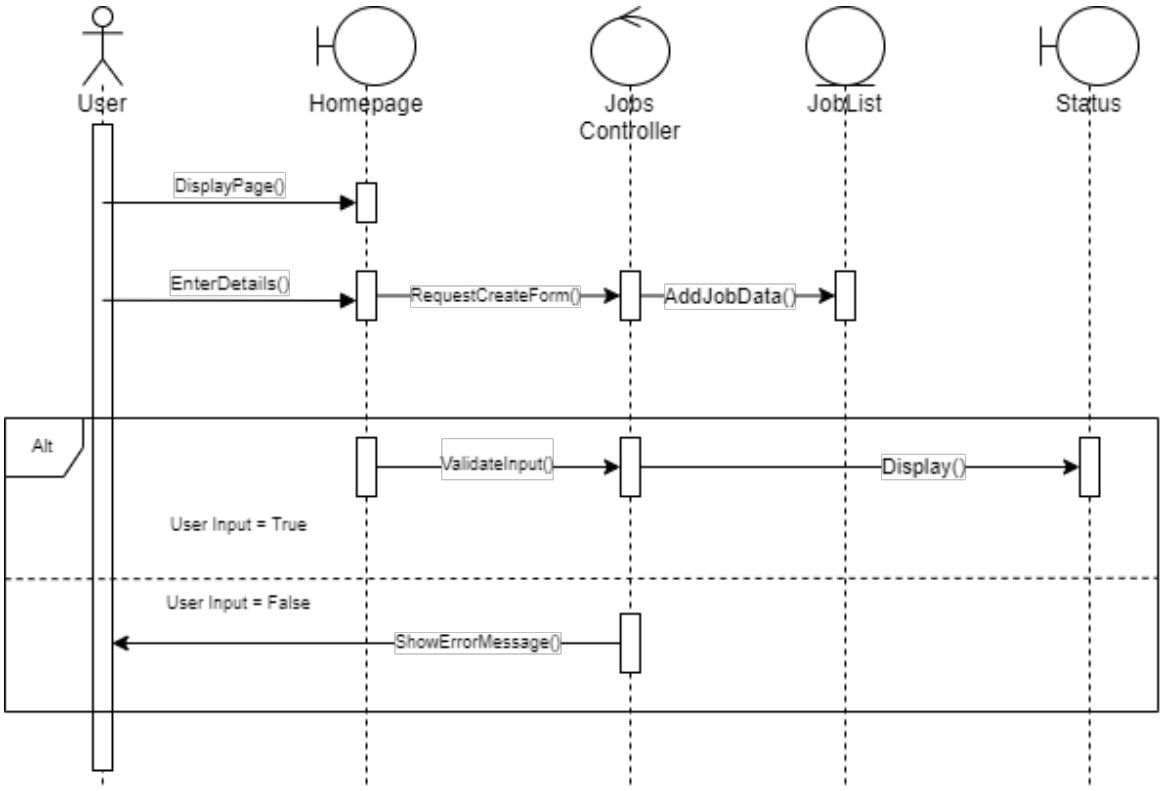
The system shall let the users to add requirements to the post that they are publishing without those data the job will be empty and it will not show the full information.

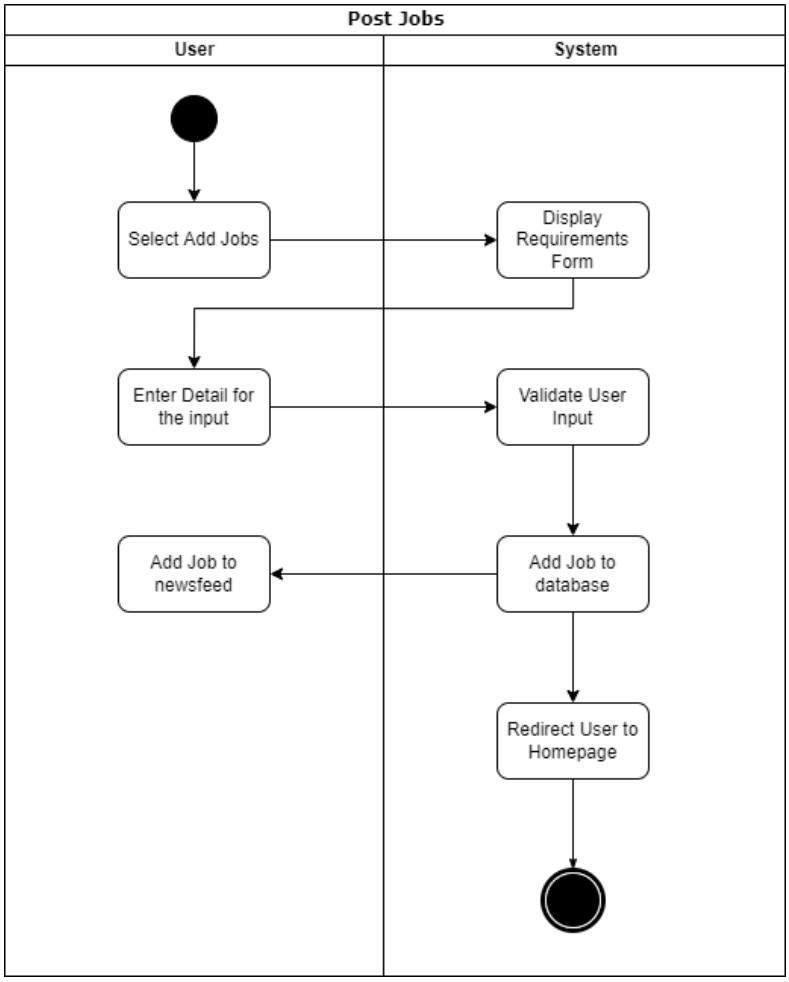
4.1.2.1 UC03: Post Jobs

Table 4: Use case Description for Post Jobs

Use Case ID	UC003
Use Case Name	Post Jobs
Brief Description	This use case allows user to post jobs into the system
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step. 2. The User must be registered into the system.
Normal Flow	1. The user login into the system. 2. The user will be redirected to the homepage. 3. The user clicks on the post jobs button which is the "+" button.

	4. The user enters the detail of the jobs that need to be published. 5. The user clicks on publish then the post will be published. 6. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User successfully post the job into the system. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message



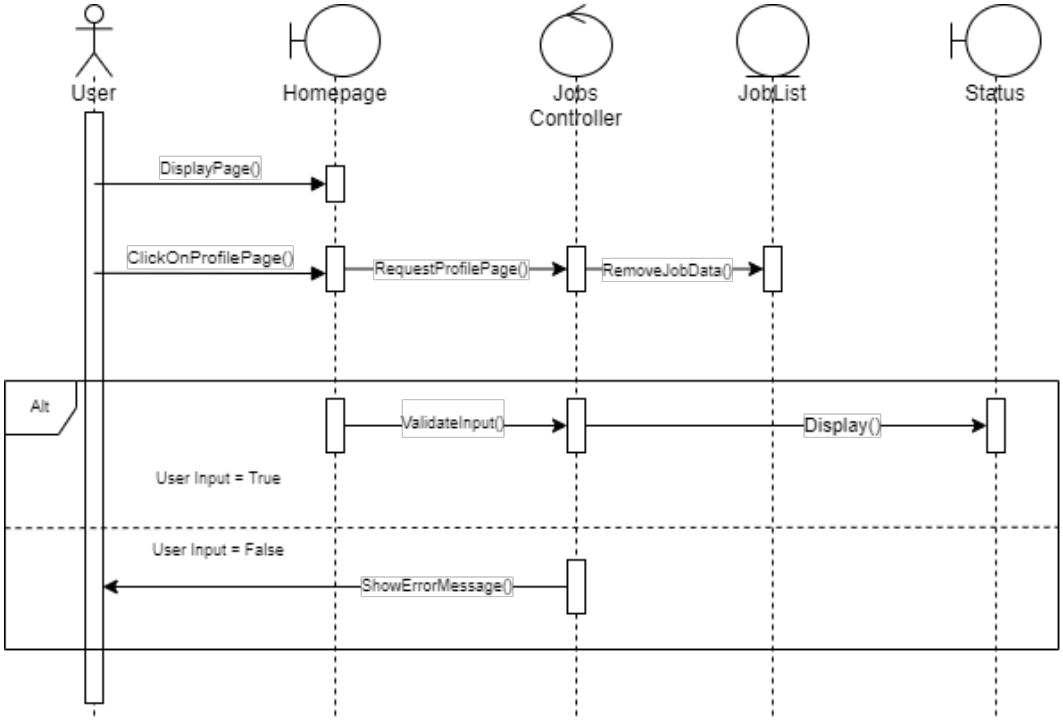


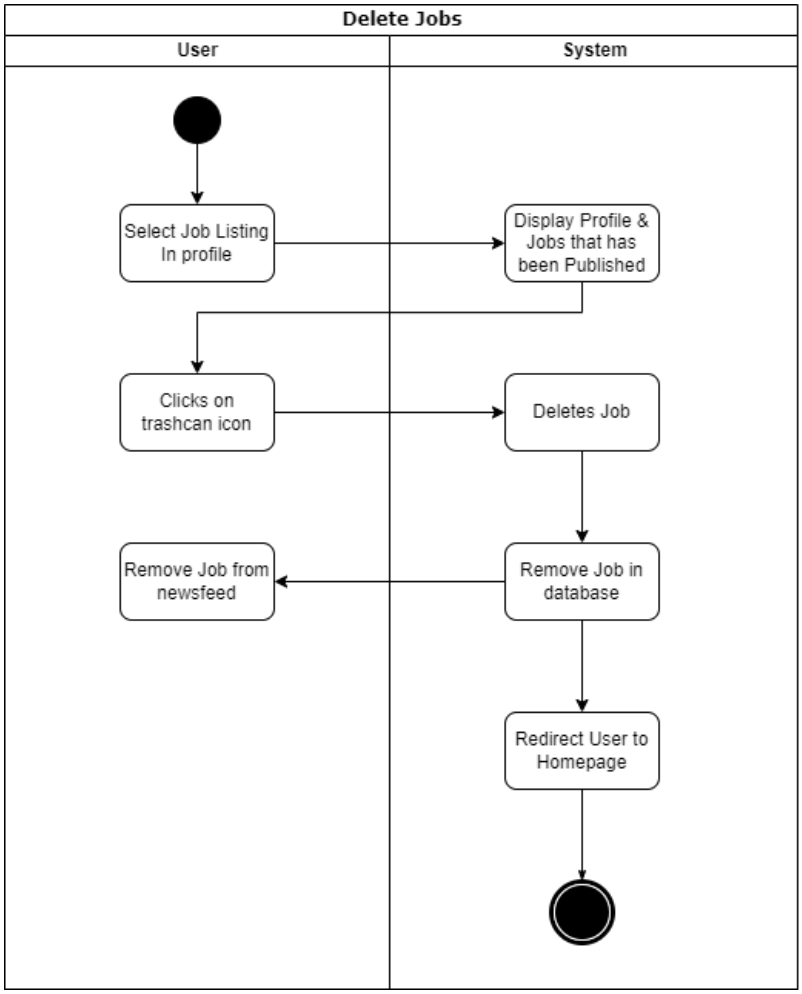
4.1.2.2 UC03: Delete Jobs

Table 5: Use case Description for Deleting Jobs

Use Case ID	UC004
Use Case Name	Delete Jobs
Brief Description	This use case allows user to delete jobs into the system
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step. 2. The User must be registered into the system. 3. The User must have posted a job before.
Normal Flow	1. The user login into the system. 2. The user will be redirected to the homepage. 3. The user must click on their profile picture. 4. The user must click on profile in the dropdown menu. 5. The user will be redirected to their profile page 6. The user clicks on the trashcan icon next to their post. 7. The post will be deleted.

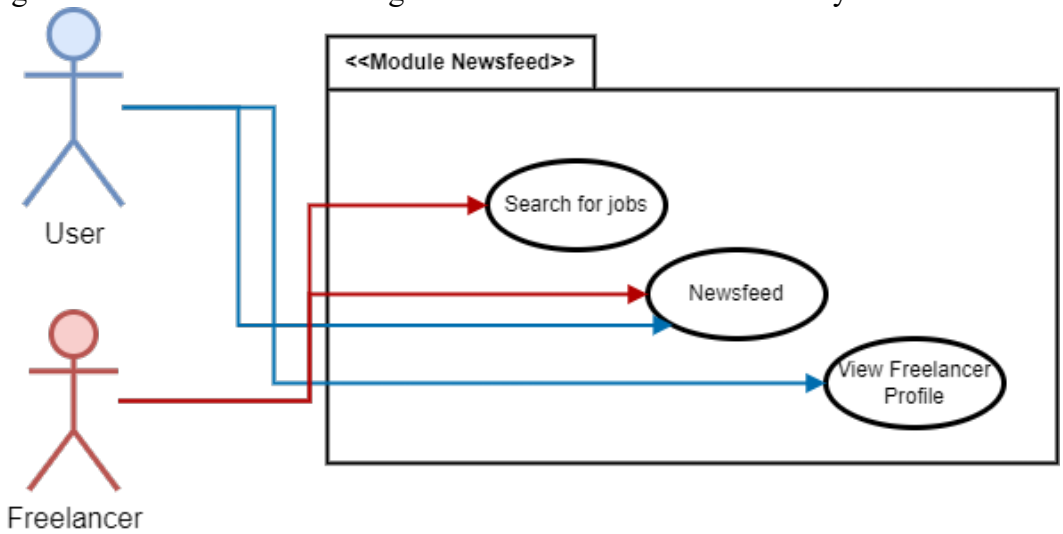
	8. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User successfully deletes a job in the system. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message





4.1.3 Module Newsfeed

Figure 4 shows the use case diagram of Module Newsfeed in the system.



The functional requirements of module newsfeed are listed as below:

i. FR008 – Search for Jobs

The user shall be able to search for jobs that has been posted by the other users and they can filter it based on their need.

ii. FR009 – Newsfeed

The users must have a unified place where they can see all the post that has been made in one place that they can interact with it.

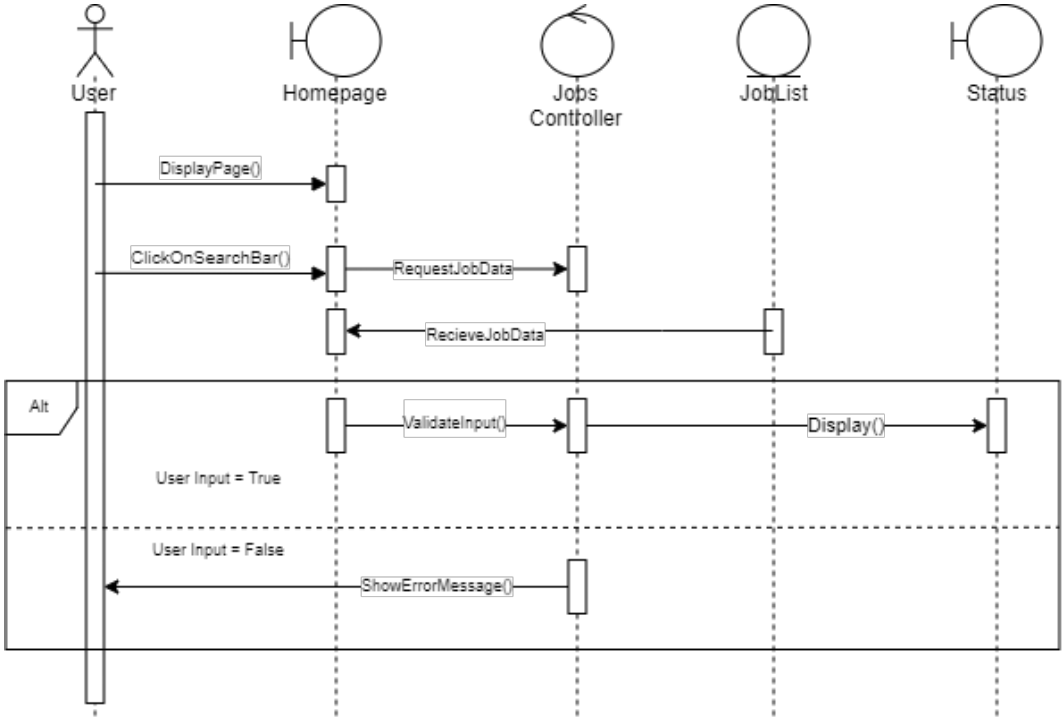
iii. FR010 – See Collaborative Users

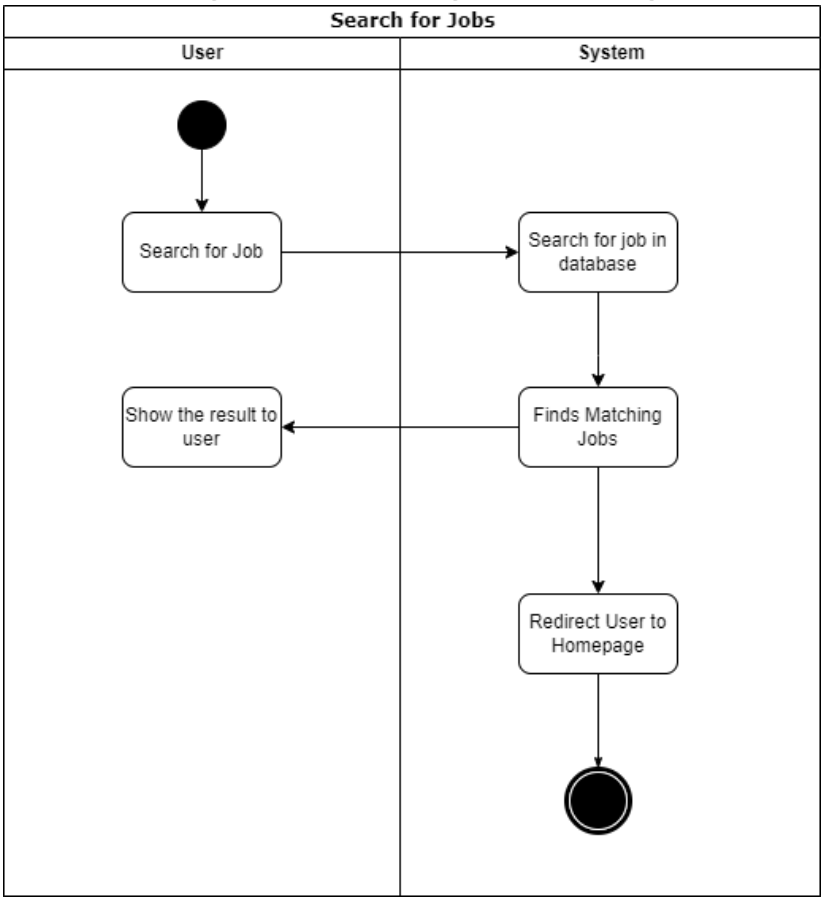
The user should see a list of users who are ready or open for collaboration this will help them to see who is ready for interaction.

4.1.3.1 UC005 Search for Jobs

Table 6: Use case Description for Search for jobs

Use Case ID	UC005
Use Case Name	Search for Jobs
Brief Description	This use case allows user to search for jobs in the system
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step. 1. The User must be registered into the system.
Normal Flow	1. The user login into the system. 2. The user will be redirected to the homepage. 3. The user will click on the search bar. 4. The user will type the name of the jobs that they want. 5. The user will see the results from the search bar. 6. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User successfully gets result from searching for a job in the system. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message



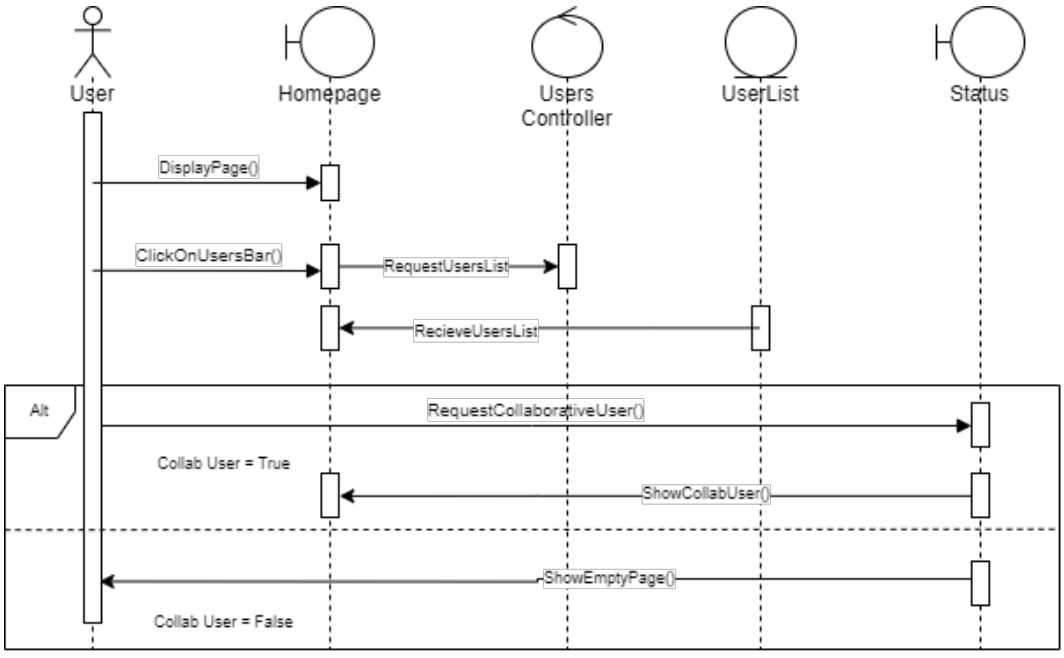


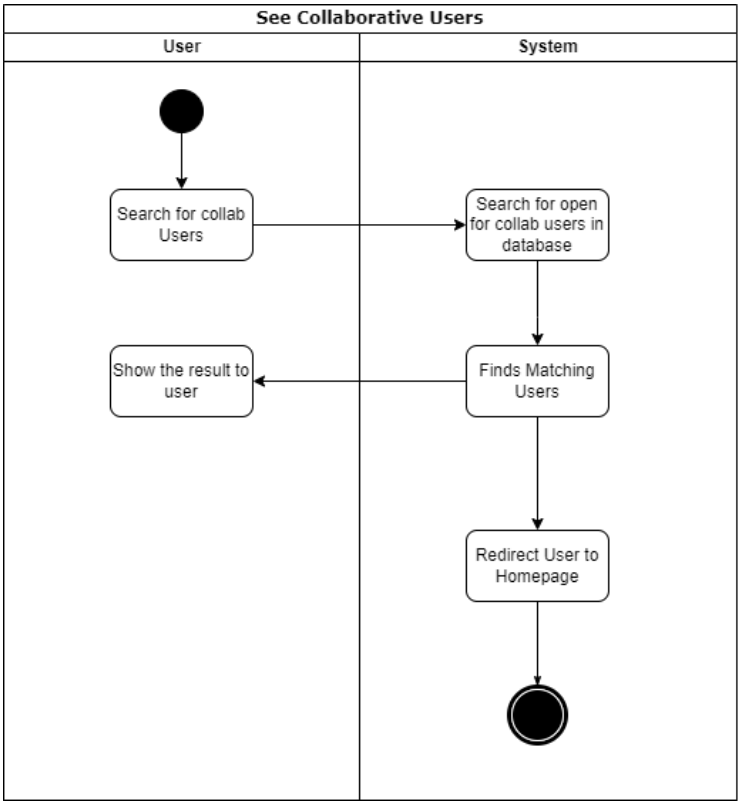
4.1.3.2 UC006 See Collaborative Users

4.1.3.3 Table 7: Use case Description for See Collaborative Users

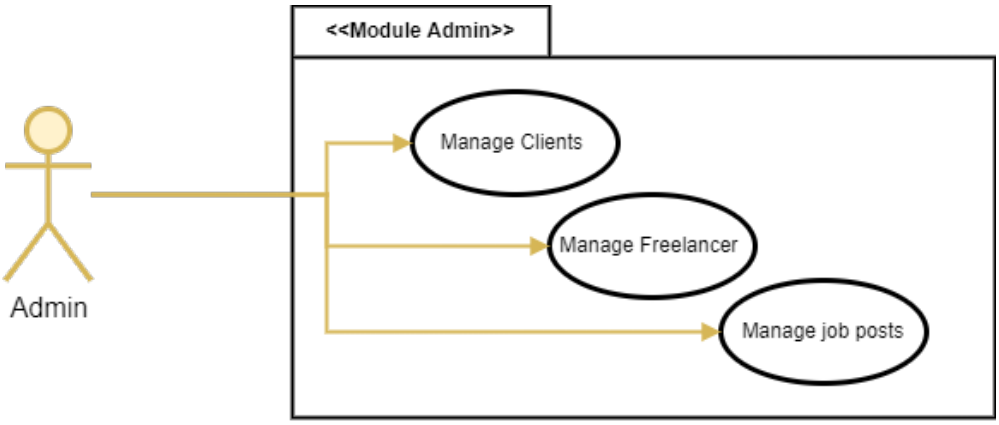
Use Case ID	UC006

Use Case Name	See Collaborative Users
Brief Description	This use case allows user to find other users who are ready to collab.
Actors	Clients / Freelancers
Pre-Condition	1. Internet is required to procced to the next step. 2. The User must be registered into the system.
Normal Flow	1. The user login into the system. 2. The user will be redirected to the homepage. 3. The user will click on the Users Bar. 4. The user will see a list of users who are ready to collab. 5. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 2
Post-Condition	1. Successful Completion 1.1 User successfully gets result from searching for a job in the system. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message





4.1.4 Module Admin



The functional requirements of module admin are listed as below:

i. FR011 – Manage Posts:

The admin must be able to have access to the data of posts that has been made by the other users.

ii. FR012 – Manage Users:

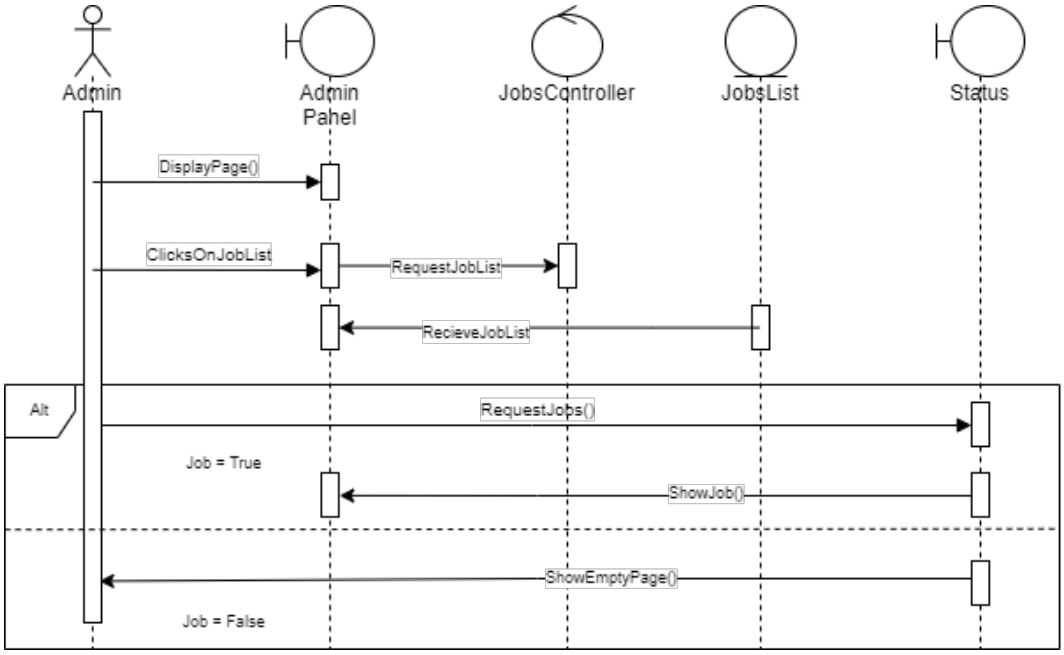
The admin should be able to have access to the data of users that has been registered in the system.

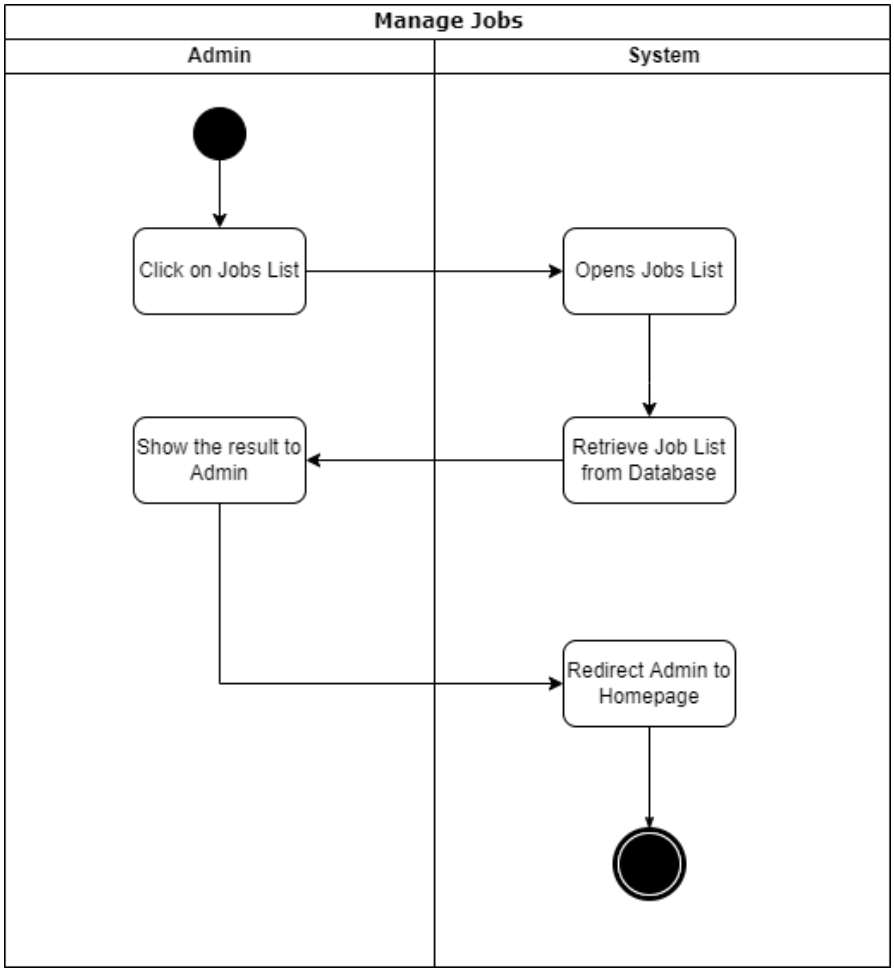
3.1.4.1 UC007 Manage Posts

Table 8: Use case Description for Search for jobs

Use Case ID	UC007
Use Case Name	Manage Posts
Brief Description	This use case allows admins to have access to the post that has been saved into the system.

Actors	Admin
Pre-Condition	1. Internet is required to procced to the next step. 2. The admin must be registered into the system.
Normal Flow	1. The admin opens the site. 2. The admin will be redirected to the admin panel. 3. The admin will click on the Jobs Bar. 4. The admin adjusts the jobs that they want. 5. The use case ends.
Alternative Flow	1. The system redirects to the homepage
Exception Flow	1. Validation fails 1.1 System displays error message. 1.2 The use case resumes at Normal Flow 1
Post-Condition	1. Successful Completion 1.1 Admin successfully manages posts. 1.2 User redirect to homepage 2. Failure Condition 2.1 System Display error message



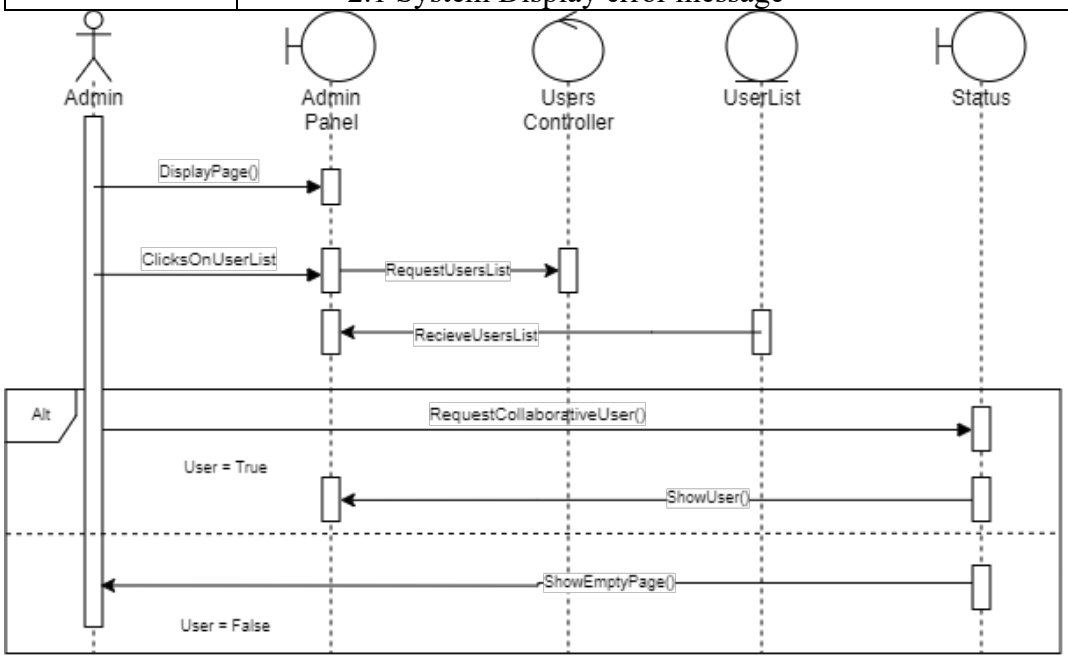


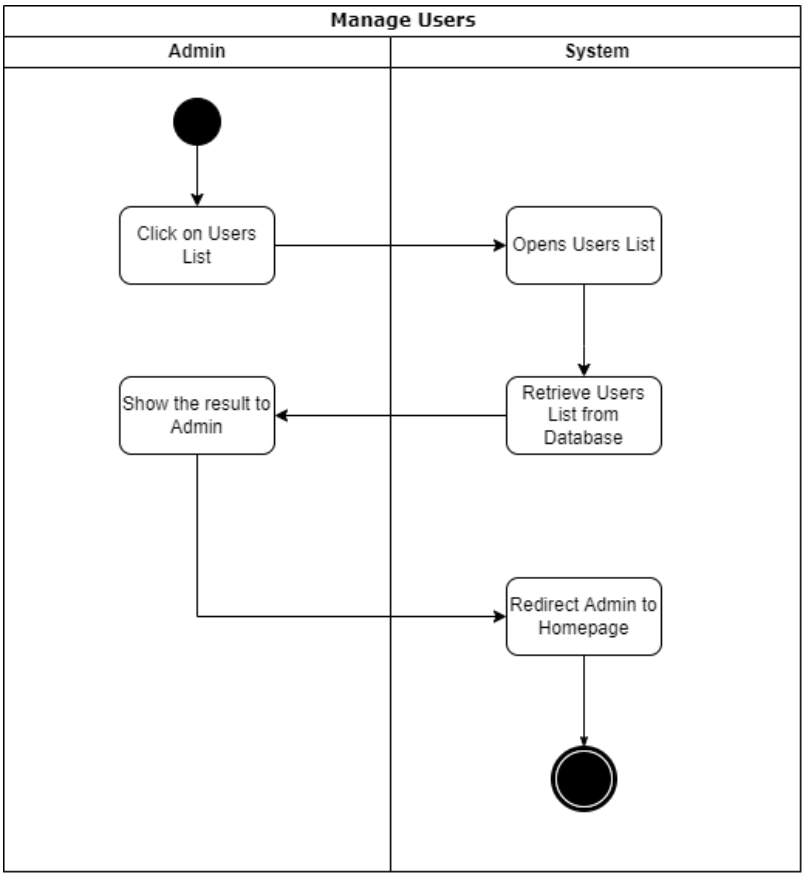
3.1.4.2 UC008 Manage Users

Table 9: Use case Description for Search for jobs

Use Case ID	UC008
Use Case Name	Manage Users
Brief Description	This use case allows admins to have access to the users that has been saved into the system.
Actors	Admin
Pre-Condition	3. Internet is required to procced to the next step. 4. The admin must be registered into the system.
Normal Flow	6. The admin opens the site. 7. The admin will be redirected to the admin panel. 8. The admin will click on the Users List. 9. The admin adjusts the users that they want. 10. The use case ends.
Alternative Flow	2. The system redirects to the homepage
Exception Flow	2. Validation fails 2.1 System displays error message. 2.2 The use case resumes at Normal Flow 1

Post-Condition	3. Successful Completion 3.1 Admin successfully manages Users. 3.2 User redirect to homepage 4. Failure Condition 2.1 System Display error message





4.2 Performance Requirements

i. Usability

   i. NFR001 – Learnability

      All the users must know how to use the system without any issue and without any one teaching them how to use it or tutorial.

ii. Reliability

   i. NFR002 – Data Synchronization

      The system shall be able to use one database to all devices across different platforms.

iii. Maintainability

   i. NFR003 Maintenance

      The system shall be maintained based on need and bugs that will be reported back from users who has complains.

iv. Security

   i. NFR004 – Data Protection

      The system must be secured and no data could be leaked because those data belong to the users and cannot be published to other users.

4.3 Design Constraints

i. Portability

   i. NFR005 – Android & IOS

      The system will not be available on those platforms that will be a constraint to the system but they can still access it because it's a website and it's accessible by most of the devices.

Appendix B

Software Testing Documentation (STD)

Software Testing Documentation

Freelance Marketplace System

Version 1.0

25th June 2022

Department and Faculty
Software Engineer
Prepared by: Lawand Hassan Mahmood

Revision Page

d. Overview

This is the first draft of STD for Freelance Marketplace System

e. Target Audience

Stakeholders

f. Project Team Members

Lawand Hassan Mahmood

g. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	**Lawand Hassan Mahmood**	**First Draft**	

1. Introduction

This Software Testing Document describes all of the activities that have been carried out for the Freelance Marketplace System. This document provides software testers with comprehensive documentation structure for recording the testing process. This will be very beneficial and it will allow for proper testing process management and provides a clear reference in the future. Black Box and User Acceptance Testing are two testing techniques that has been prioritized in this project and it is the best suited for this situation.

1.1 Purpose

Software testing has many purposes but those are the main ones that has been focused on:

i. To evaluate the system functions by testing the Freelance Marketplace system.

ii. To demonstrate the two-approach test that has been used in the Freelance marketplace.

iii. To document those tests for later reuse and ease of use.

1.2 Scope

This software system to be developed is a Freelance Marketplace, it is a website that offer users to post jobs online so others can see it and collaborate with it. The targeted user of the system are people around the globe. There are two categories of user, which is Admin and normal user.

The system works specifically for both type of user which are the people who are searching for jobs and also for the ones that trying to find someone to collaborate with them and both users can log in and post jobs they can also search for jobs using the categories that has been provided and the admin can delete jobs, posts and users. The system will be created as website using React which is JavaScript framework and it's an open-source UI software that allows users to create any web-based applications for all the platforms. The system main goal is to provide a centralized job portal for users to interact with each other and to post and see the available jobs that has been offered.

2. Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. Black Box is mostly used to be more focused on the input and the output of the software system this test can be used in many levels of software testing, and one of the advantages of this method is that it can be used from the beginning to the end of the system. By using this method, the user won't have to worry about any issues

regarding the software that is going to be delivered to them and they will except a very professional system.

2.1 Unified Interface Section

This section showcases how black box testing is implemented in the Freelance Marketplace for all the users and the interfaces.

2.1.1 Register

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Email	-	Lawand	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com
Password	-	Lawa123	1234454	Lawa123	Lawa123	Lawa123
Account Type	User	User	User	User	User	User
First Name	-	-	Lawand	-	Lawand	Lawand
Last Name	-	-	Hassan	Hassan	-	-
Images	Uploaded	-	Uploaded	-	-	Uploaded
Expected Result	Actual Result					
Error Message on invalid email address	✓	✓				
Error Message on empty password	✓		✓			
Error message on empty images		✓		✓	✓	
Error message on	✓	✓		✓	✓	✓

last name or first name						
Redirect user (User) to dashboard Page			✓			
Test Results						
	Fail	Fail	Pass	Fail	Pass	Pass

2.1.2 Login

Test Case ID	TC 01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Email	-	Lawand	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com
Password	-	Lawa123	-	Lawa123	Lawa123	Lawa123
Account Type	User	User	User	User	User	User
Expected Result	Actual Result					
Error Message on invalid email address	✓	✓				
Error Message on empty password	✓		✓			
Redirect user (User) to						

dashboard Page						
Test Results						
	Fail	Pass	Fail	Pass	Pass	Pass

2.2 Newsfeed Section

This section showcases how black box testing is implemented in the Freelance Marketplace for all the users and the interfaces of newsfeed section which contains Search for jobs, View Collaborative Users.

2.2.1 Search for Jobs

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Title of the job	-	Software Engineers	-	-	Developer	Backend Developer
Category	-	-	Remote	-	Remote	On-Site
Tags	UX / UI	-	-	-	-	C#
Expected Result	Actual Result					
Error Message on invalid title of the job address	✓		✓	✓		
Error Message on empty Search bar	✓		✓	✓		
Error message on invalid category		✓	✓	✓	✓	
Redirect user (User) to search result			✓			✓
Test Results						
	Pass	Pass	Pass	Fail	Pass	Pass

2.1.2 View Collaborative Users

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06

Inputs and Actions						
Turned On Ready for Collab	-	ON	-	ON	-	-
Turned Off Ready for Collab	-	-	OFF	-	OFF	-
Expected Result	Actual Result					
Error Message on empty users	✓					✓
Redirect user (User) to Users Page		✓	✓	✓	✓	
Test Results						
	Fail	Pass	Pass	Pass	Pass	Fail

2.3 Manage Newsfeed Section

This section showcases how black box testing is implemented in the Freelance Marketplace for all the users and the interfaces of manage newsfeed section which contains Posting Jobs, Deleting Jobs as well.

2.2.1 Post Jobs

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Title	-	Software Engineers	-	Designer	Developer	Backend Developer
Category	-	-	Remote	-	Remote	On-Site
Tags	UX / UI	-	-	-	-	C#
Description	Uploaded			Uploaded		-
External Links	Uploaded	-	Uploaded	-	-	Uploaded
Company Name	Google	-	Meta	-	Facebook	Google
Expected Result	Actual Result					
Error Message on invalid title of the job title	✓		✓			
Error message on	✓	✓		✓		

invalid category						
Error Message on invalid Company Name		✓		✓		
Error Message on invalid External Link		✓		✓	✓	
Post the job to the dashboard						✓
Test Results						
	Pass	Pass	Pass	Pass	Pass	Pass

2.4 Admin Section

This section showcases how black box testing is implemented in the Freelance Marketplace for all the users and the interfaces of admin section which contains Manage jobs, Manage User.

2.2.1 Manage Jobs

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Title	-	Software Engineers	-	Designer	Developer	Backend Developer
Category	-	-	Remote	-	Remote	On-Site
Tags	UX / UI	-	-	-	-	C#
Description	Uploaded			Uploaded		-
External Links	Uploaded	-	Uploaded	-	-	Uploaded
Company Name	Google	-	Meta	-	Facebook	Google
Expected Result	Actual Result					
Change title of the job		✓		✓	✓	✓
Change category of the job			✓		✓	✓
Change tags of the job	✓					✓

Change description of the job	✓			✓		
Change Company name of the job	✓		✓		✓	✓
Test Results						
	Pass	Pass	Pass	Pass	Pass	Pass

2.1.2 Manage Users

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04	TC01-05	TC01-06
Inputs and Actions						
Email	-	Lawand	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com	Lawand@gmail.com
Account Type	User	User	User	User	User	User
First Name	-	-	Lawand	-	Lawand	Lawand
Last Name	-	-	Hassan	Hassan	-	-
Images	Uploaded	-	Uploaded	-	-	Uploaded
Expected Result	Actual Result					
Change Name			✓		✓	✓
Change Email			✓	✓	✓	✓
Change User type	✓	✓	✓	✓	✓	✓
Change Image	✓		✓			✓
Test Results						
	Pass	Pass	Pass	Pass	Pass	Pass

3. User Acceptance Testing

User Acceptance Testing (UAT), which is performed on most UIT projects, sometimes called beta testing or end-user testing, is a phase of software development in which the software is tested in the "real world" by the intended audience or business representative. It's used to validate the software system before publishing the system to the production environment this method will be use when everything has been finished with the application.

3.1 User Section

Tables below show the list of what will the user will see throughout the process of using the website and what are the task that they will interact with the system.

Table 3.1: Task List of User Interfaces

Task	Description
1	Register User account with valid information
2	Login to the system using the website as User
3	Post Jobs
4	Update Profile Information
5	View Jobs
6	View Dashboard
7	Delete Jobs (Personal Jobs)
8	Logout from the account

Table 3.1: Task List of Admin Interfaces

Task	Description
1	Login to the system using the website as Admin
2	Add User
3	Update Profile Information
4	Add Jobs
5	View Dashboard
6	Delete Jobs
7	Remove Jobs
8	Logout from the account

3.2 UAT Results

The result from testers was collected directly from the tester also the time that they spend on each task after that for the system review how affective the system was and many more all those were collected and down below will be showed as graphical chart.

Appendix C

Software Design Document (SDD)

SCSJ3323: Software Design and Architecture

Software Design Document

Freelance Marketplace System

Version 1.0

25th June 2022

Department and Faculty
Software Engineering

Prepared by: Lawand Hassan Mahmood

Revision Page

h. Overview

This is the first draft of SDD for Freelance Marketplace System

i. Target Audience

Stakeholders

j. Project Team Members

Lawand Hassan Mahmood.

k. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	**Lawand Hassan Mahmood`**	**First Draft**	

1. Introduction

Design And Implementation of Freelance Marketplace is described in detail in this Software Design Document (SDD), which includes architecture design, database design, and user interface design with a collection of use cases, activity diagrams, and sequence diagrams, it depicts the interaction between the user and the system to represent the flow of the system and the declared objects both an activity diagram are being elaborated.

1.1 Purpose

i. To describe the system architectural design of the freelance marketplace system.
ii. To demonstrate how the database design has been layout.
iii. To document and illustrate the user interface design in the freelance marketplace

1.2 Scope

This software system to be developed is a Freelance Marketplace, it is a website that offer users to post jobs online so others can see it and collaborate with it. The targeted user of the system are people around the globe. There are two categories of user, which is Admin and normal user.

The system works specifically for both type of user which are the people who are searching for jobs and also for the ones that trying to find someone to collaborate with them and both users can log in and post jobs they can also search for jobs using the categories that has been provided and the admin can delete jobs, posts and users. The system will be created as website using React which is JavaScript framework and it's an open-source UI software that allows users to create any web-based applications for all the platforms. The system main goal is to provide a centralized job portal for users to interact with each other and to post and see the available jobs that has been offered.

1.3 Definitions, Acronyms and Abbreviation

Term	Definition
FR	Functional Requirement
NFR	Non-Functional Requirement
SDD	Software Design Document
SRS	Software Requirement Specification
React	React is a JavaScript library for building user interfaces.

1.4 Overview

This document is divided into five sections, each section contains several and also several subsections and those sections are:

i. System Architectural Design

This section explains how the architecture the freelance marketplace has be created and how it has been layout.

ii. Database Design

This section explains the design of the database for the Freelacne Marketplace. It contains a data dictionary that explains all the data attributes that will be used in the system.

iii. Interface Design

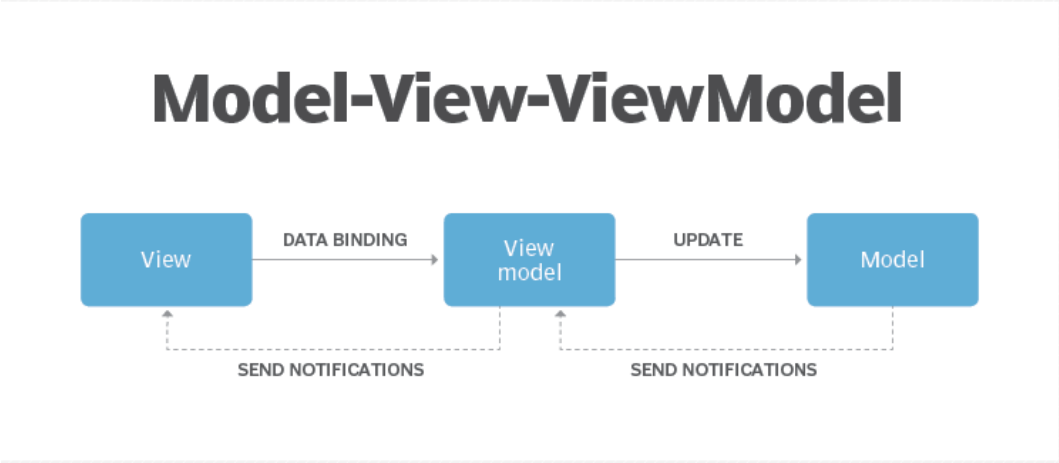
This section explains the chosen system architecture for Freelance Marketplace.

2. System Architectural Design

In this part of the software design document the best system architecture to design is going to be implemented in the freelance marketplace, this system uses MVVM (Model View-Viewmodel).

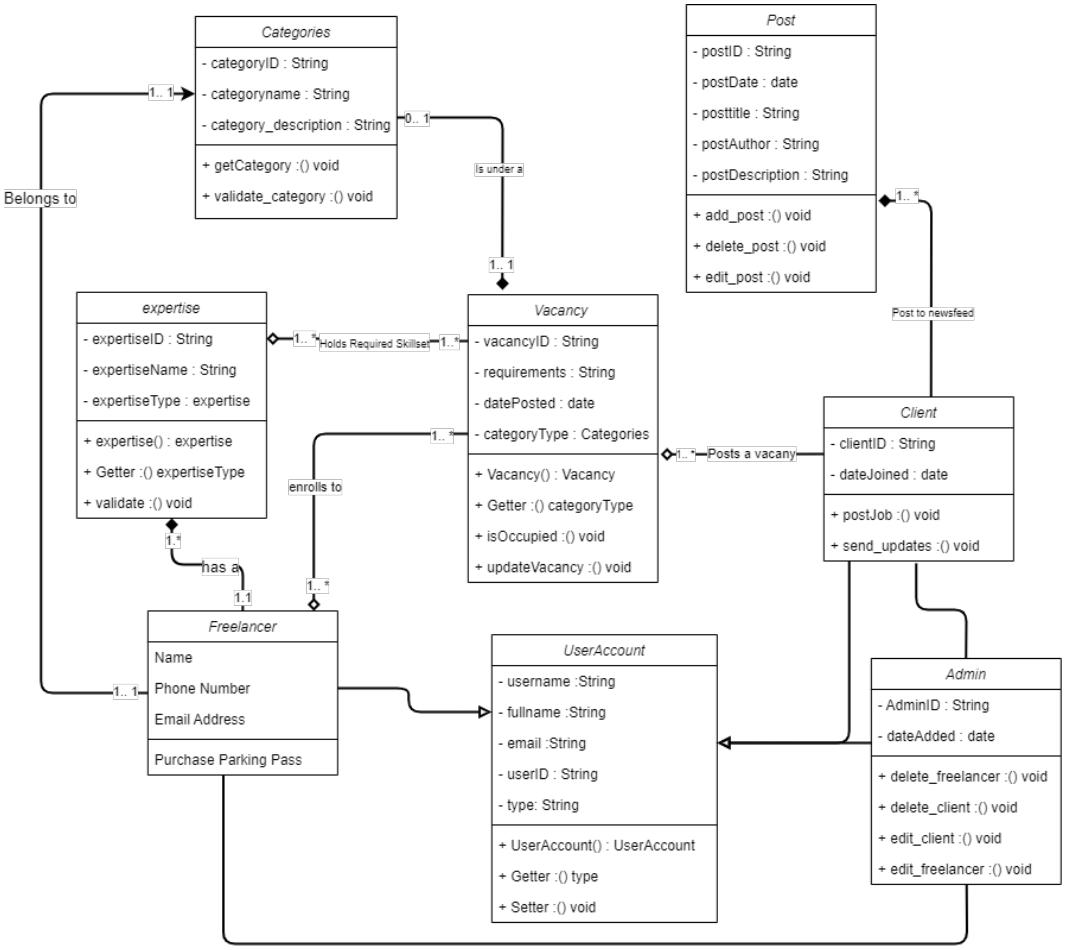
2.1 Model-View-ViewModel

A software architecture pattern called Model-View-ViewModel (MVVM) is organized to keep program logic and user interface controls apart. Microsoft architects Ken Cooper and John Gossman developed MVVM, commonly referred to as model-view-binder. Like many other design patterns, MVVM streamlines code creation, updates, and code reuse by breaking applications into manageable modules. In Windows and web graphics presentation software, the pattern is frequently utilized.



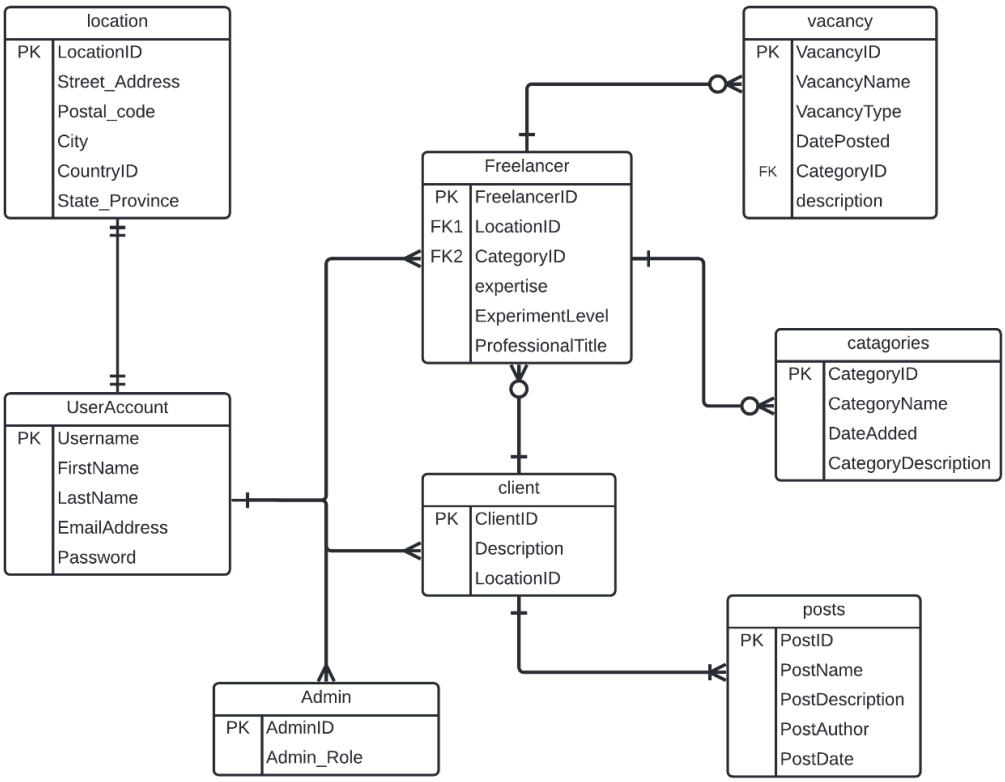
The separation of the code in MVVM is divided into View, ViewModel and Model:

1. View is the collection of visible elements, which also receives user input. This includes user interfaces (UI), animations and text. The content of View is not interacted with directly to change what is presented.
2. ViewModel is located between the View and Model layers. This is where the controls for interacting with View are housed, while binding is used to connect the UI elements in View to the controls in ViewModel.
3. Model houses the logic for the program, which is retrieved by the ViewModel upon its own receipt of input from the user through View.


Categories
- categoryID : String
- categoryname : String
- category_description : String
+ getCategory :() void
+ validate_category :() void
Post
- postID : String
- postDate : date
- posttitle : String
- postAuthor : String
- postDescription : String
+ add_post :() void
+ delete_post :() void
+ edit_post :() void
expertise
- expertiseID : String
- expertiseName : String
- expertiseType : expertise
+ expertise() : expertise
+ Getter :() expertiseType
+ validate :() void
Vacancy
- vacancyID : String
- requirements : String
- datePosted : date
- categoryType : Categories
+ Vacancy() : Vacancy
+ Getter :() categoryType
+ isOccupied :() void
+ updateVacancy :() void
Client
- clientID : String
- dateJoined : date
+ postJob :() void
+ send_updates :() void
Freelancer
Name
Phone Number
Email Address
Purchase Parking Pass
UserAccount
- username :String
- fullname :String
- email :String
- userID : String
- type: String
+ UserAccount() : UserAccount
+ Getter :() type
+ Setter :() void
Admin
- AdminID : String
- dateAdded : date
+ delete_freelancer :() void
+ delete_client :() void
+ edit_client :() void
+ edit_freelancer :() void
1.. 1
Belongs to
0.. 1
Is under a
1.. 1
1.. *
Post to newsfeed
1.. *
Holds Required Skillset
1.. *
1.. *
enrolls to
1.. *
1.. *
Posts a vacany
1.. *
has a
1..1
1.. 1

3. Database Design

Database design is very important in software design and implementation because all data that was involved into making the system database and will store those data that was retrieved by the users and been collected.



3.1 Data Dictionary

This part explains all the data that has been stored inside the system and it's been saved in the database.

3.1.1 Account

Attributes	Data Type	Description
Id	number	A unique id for all users
EmailAddress	String	Email address to use it for authentication
FirstName	String	First Name of the user
LastName	String	Last name of the user
accountType	String	Which type of user it is whether freelancer or client

3.1.2 Jobs

Attributes	Data Type	Description
Id	number	A unique id for all users

Job title	String	A job title to differentiate from other jobs
Company Name	String	Company that provided the job
Category	String	The Category of the job
Tags	String	Tags related to the job
External Links	String	External link like guidelines

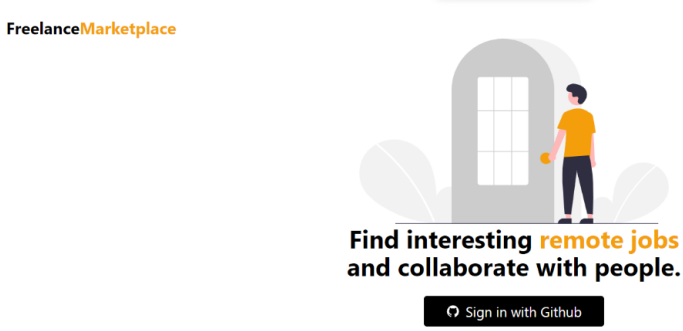
4. User Interface Design

This section is to explain what will be included in the interface and how the user will interact with the it.

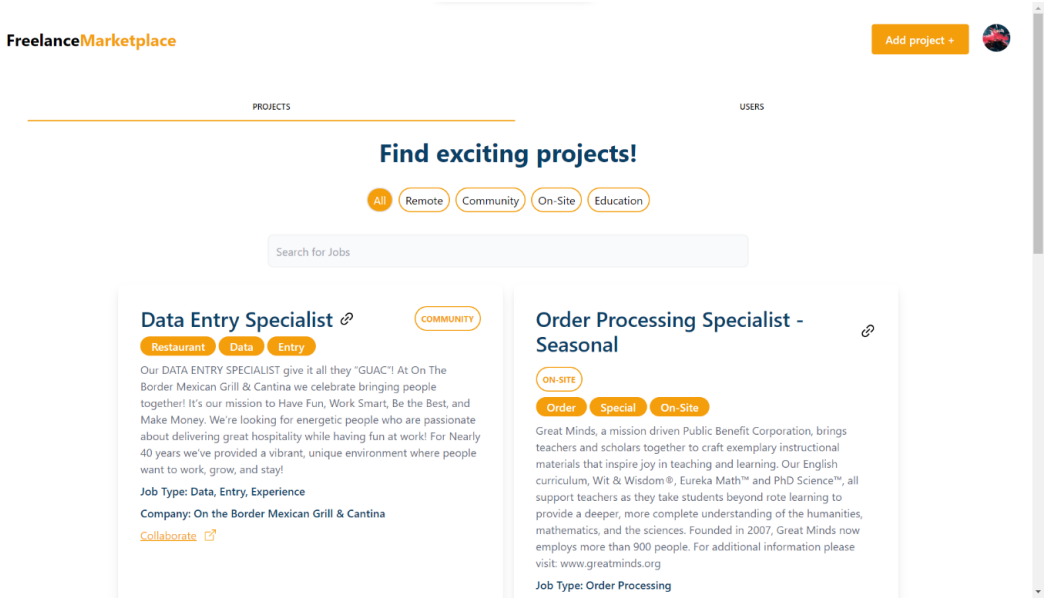
4.1 User Interface

The user interface includes a login page and after login in successfully the user will be redirect to the dashboard or the homepage and then the user can navigate between other screens like Post list, User List, Profile, and then logout.

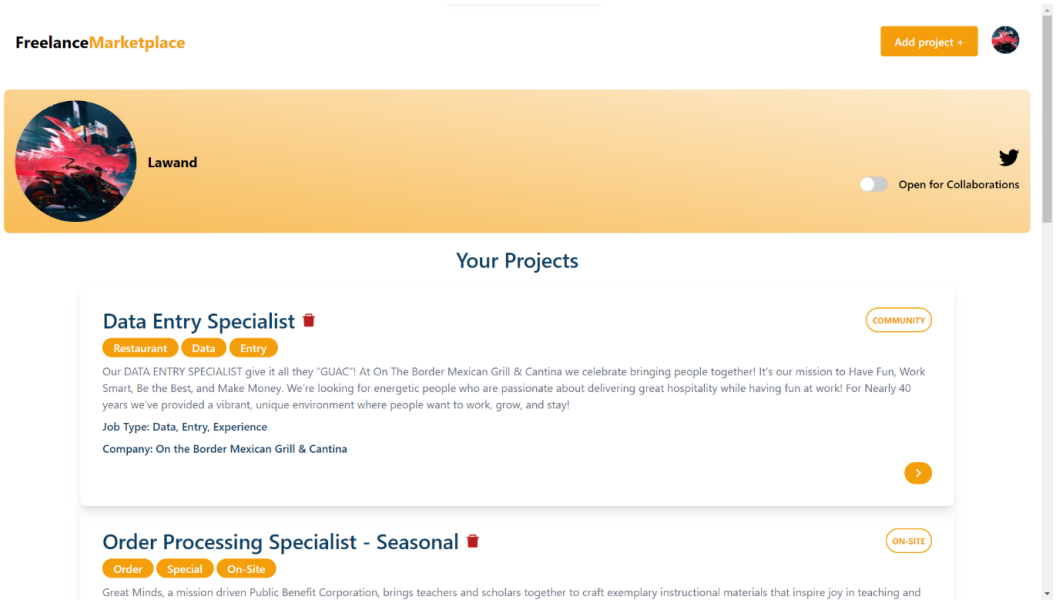
4.1.1 Sign Up Page



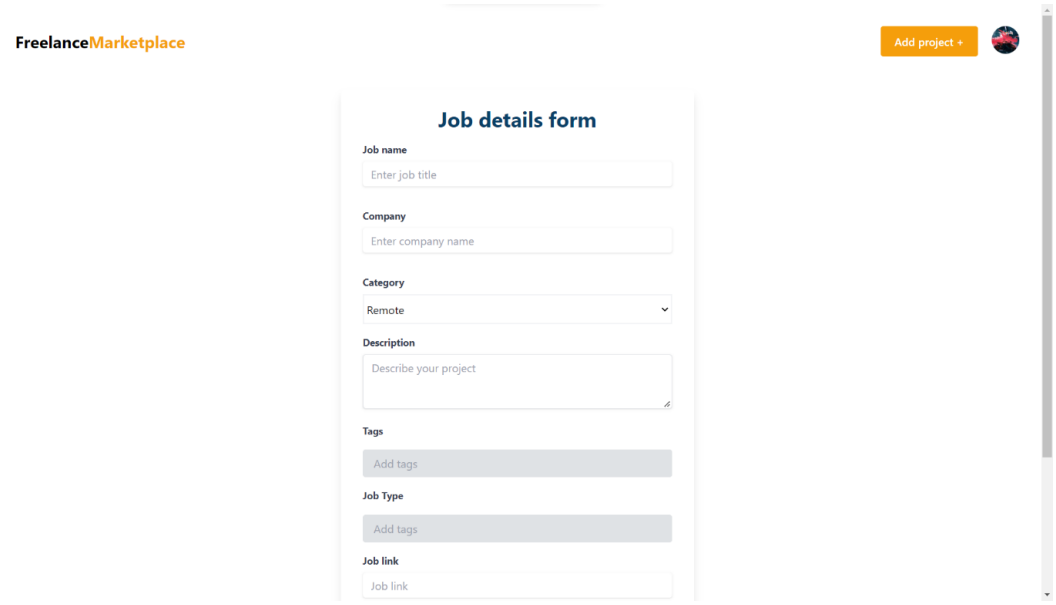
4.1.2 Homepage

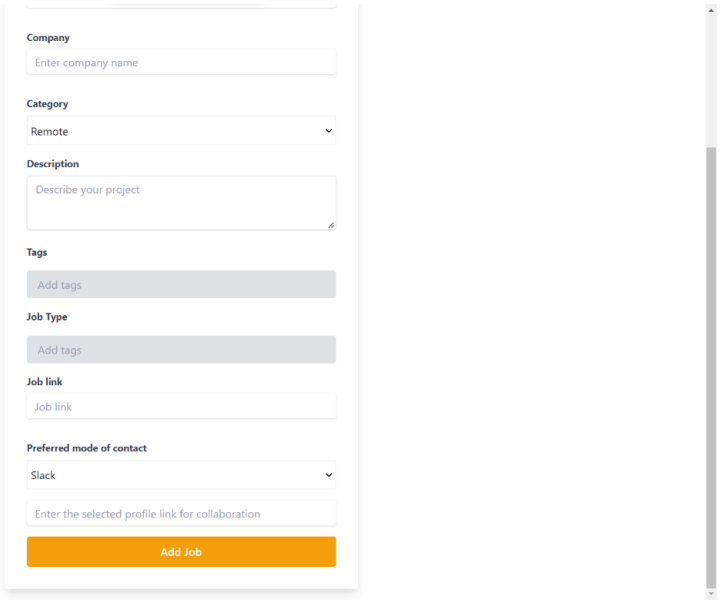


4.1.3 Profile


FreelanceMarketplace
Add project +
Lawand
Open for Collaborations
Your Projects
Data Entry Specialist
COMMUNITY
Restaurant
Data
Entry
Our DATA ENTRY SPECIALIST give it all they "GUAC"! At On The Border Mexican Grill & Cantina we celebrate bringing people together! It's our mission to Have Fun, Work Smart, Be the Best, and Make Money. We're looking for energetic people who are passionate about delivering great hospitality while having fun at work! For Nearly 40 years we've provided a vibrant, unique environment where people want to work, grow, and stay!
Job Type: Data, Entry, Experience
Company: On the Border Mexican Grill & Cantina
Order Processing Specialist - Seasonal
ON-SITE
Order
Special
On-Site
Great Minds, a mission driven Public Benefit Corporation, brings teachers and scholars together to craft exemplary instructional materials that inspire joy in teaching and

4.1.4 Add Jobs


FreelanceMarketplace
Add project +
Job details form
Job name
Enter job title
Company
Enter company name
Category
Remote
Description
Describe your project
Tags
Add tags
Job Type
Add tags
Job link
Job link



4.1.5 User List