

LaborWorld



By:

Zain ul Abdin

19683

Abdul Hadeed

11061

Muhammad Azeem Khan

19862

Supervised by:

Prof. Muhammad Saud Khan

Faculty of Computing

Riphah International University, Islamabad

Fall 2023

A Dissertation Submitted To

**Faculty of Computing,
Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Award of
the Degree of
Bachelors of Science in Software Engineering**

**Faculty of Computing
Riphah International University, Islamabad**

Date: 07-November-2023

Final Approval

This is to certify that we have read the report submitted by ***Muhammad Azeem Khan, Zain ul Abdin and Abdul Hadeed (19862, 19683, 11061)***, for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Software Engineering (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Software Engineering (BSSE).

Committee:

[Mr. Muhammad Saud Khan]

(Supervisor)

[Mr. Musharif Ahmed]

(Head of Department/chairman)

Declaration

We hereby declare that this document “**LaborWorld**” neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanying report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **Mr. Muhammad Saud Khan**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

Muhammad Azeem Khan, Zain ul abdin and Abdul Hadeed (19862, 19683, 11061),

Zail ul Abdin

19683

Muhammad Azeem Khan

19862

Abdul Hadeed

11061

Dedication

We wholeheartedly dedicate this project to the divine presence that has guided us throughout our journey - God Almighty. We are grateful for the unwavering support and encouragement of our parents, whose belief in us has been a constant source of motivation. We would like to express our sincere gratitude to our esteemed supervisor, Prof. Muhammad Saud khan, for his invaluable support, collaboration, and guidance, without which this project would not have come to fruition. Additionally, we extend our appreciation to all those who have provided assistance and support during our academic endeavors. May Almighty Allah (S.W.T) bless us with the strength and wisdom to successfully complete the development of this project as envisioned – Ameen.

Acknowledgement

First of all, we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We would like to express our sincere appreciation to our supervisor, Prof. Muhammad Saud Khan, for his unwavering support, invaluable guidance, and constructive feedback. His expertise and dedication have played a pivotal role in shaping the direction of this project.

We are grateful to our parents and families for their love, encouragement, and understanding. Their unwavering belief in our abilities has been a constant source of motivation.

We also extend our thanks to our friends and classmates who have supported us throughout this journey. Their collaboration and shared experiences have enriched our understanding and enhanced the project's outcomes.

Additionally, we acknowledge the contribution of our teachers and faculty members for their knowledge, patience, and willingness to provide assistance whenever needed.

Finally, we would like to express our gratitude to all those who have directly or indirectly contributed to the success of this project. Your support has been invaluable.

Zail ul Abdin

19683

Muhammad Azeem Khan

19862

Abdul Hadeed

11061

Abstract

LaborWorld is a comprehensive web application designed to connect customers with skilled workers for their home service needs. Inspired by successful online platforms, LaborWorld aims to streamline the process of finding and hiring workers for a variety of services, including home repairs, renovations, and maintenance. The platform provides a user-friendly interface where customers can easily browse and connect with professional workers in their local area. By incorporating features such as job posting, bid management, and secure communication channels, LaborWorld offers a convenient and reliable solution for individuals seeking reliable worker services. With a focus on trust, convenience, and efficient service delivery, LaborWorld revolutionizes the way people access and hire worker services for their homes.

Table of Contents

Chapter 1: Introduction	14
1.1 Opportunity & Stakeholders	15
1.2 Motivations and Challenges	17
1.3 Goals and Objectives	17
1.4 Solution Overview	19
1.4.1 User-Friendly Web and Mobile Interface	19
1.4.2 Extensive Service Categories and Listings	19
1.4.3 Direct Communication and Task Management	20
1.4.4 Group and Skill Development	20
1.4.5 Enhanced Security and Emergency Support	21
Chapter 2: Literature / Market Survey	23
2.1 Introduction	23
2.2 Literature Review/Technologies Overview	24
2.2.1 Fori Mazdoori	24
2.2.2 Super Tasker	24
2.2.3 Labour Square	24
2.2.4 Mahir App	24
2.2.5 Wonoloo	24
2.2.6 Gigsmart	24
2.2.7 Veryable	24
2.2.8 HireDay Labor	25
2.2.9 Thumbtack	25
2.2.10 Existing Systems	25
2.3 Summary	26
Chapter 3: Requirement Engineering	27
3.1 Introduction	28
3.2 Problem Scenarios	28

3.2.1 Problem Statement 1	29
3.2.2 Problem Statement 2	29
3.2.3 Problem Statement 3	30
3.2.4 Problem Statement 4	31
3.2.5 Problem Statement 5	31
3.3 Functional Requirements	32
3.3.1 Worker's Requirements	32
3.3.2 Client's Requirements	33
3.3.3 Admin's Requirements	34
3.4 Non-Functional Requirements	35
 Chapter 4: System Design	37
4.1 Introduction	38
4.2 Architectural Design	39
4.3 Detailed Design	39
4.3.1 Use Case Diagrams	39
4.3.1.1 Client Use Case Design	39
4.3.1.2 Worker Use Case Design	40
4.3.1.2 Common Use Case Design	41
4.3.2 Detail Use Cases	42
4.3.2.1 Common Use Cases	42
4.3.2.2 Client Use Cases	46
4.3.2.3 Worker Use Cases	55
4.3.3 Activity Diagrams	65
4.3.3.1 Client Activity Diagram	65
4.3.3.2 Worker Activity Diagram	66
4.3.4 Entity Relationship Diagram	67
 Chapter 5: Implementation	68
5.1 Endeavour (Team + Work + Way of Working)	69
5.2 Components, Libraries, Web Services and stubs	70

5.3 IDE, Tools and Technologies	71
5.4 Best Practices / Coding Standards	71
5.4.1 Development Practices & Standards	71
5.5 Development Environment	74
5.6 Summary	75
 Chapter 6: Testing and Evaluation	 76
6.1 Introduction	77
6.2 Test Cases	77
6.2.1 Registration	77
6.2.2 Post a Job	81
6.2.3 Reset Password	84
6.2.4 Price offer	86
 Chapter 7: Conclusions and Outlooks	 87
7.1 Introduction	88
7.2 Achievements and Improvements	88
7.3 Critical Review	88
7.4 Future Recommendations/Outlook	89
7.5 Summary	89
References	90
Plagiarism Summary	91

List of Figures

4.1: Architectural Diagram	38
4.2 Client Use Case Diagram	39
4.3 Worker Use Case Diagram	40
4.4 Common Use Case Diagram	41
4.5 Registration Activity Diagram	43
4.6 Login Activity Diagram	45
4.7 Post job(Client) Activity Diagram	47
4.8 Search Worker(Client)Activity Diagram	49
4.9 View Worker Profile Activity Diagram	50
4.10 Hire Worker Activity Diagram	52
4.11 End Job Activity Diagram	54
4.12 View Jobs(Worker)Activity Diagram	57
4.13 Give Price offer Activity Diagram	59
4.14 Cancel a job Activity Diagram	61
4.15 Send Job end Request Activity Diagram	63
4.16 Client's Activity Diagram	65
4.17 Worker's Activity Diagram	66
4.18 Entity Relationship Diagram	67

List of Tables

2.1 Comparison of Existing Systems.	25
3.1 Problem Statement 1	29
3.2 Problem Statement 2	29
3.3 Problem Statement 3	30
3.4 Problem Statement 4	31
3.5 Problem Statement 5	31
4.1 Common Use Case	42
4.1.1 Registration Use Case	42
4.1.2 Login Use Case	44
4.2 Client Use Cases	46
4.2.1 Post Job Use Case	46
4.2.1 Accept Job Request	46
4.2.3 Search Worker	48
4.2.4 Review Worker Profile	49
4.2.5 Hire worker	51
4.2.6 End Job	53
4.2.7 Profile Management	54
4.3 Worker Use Cases	56
4.3.1 View Job	56
4.3.2 Request on Job	59
4.3.4 Cancel a job	62
4.3.3 Profile Management	64
5.1 Endeavor	69
6.1 Registration test data	77
6.2 Registration Test Case	78
6.3 Post a Job test data	82
6.4 Post a job test case	83
6.5 Reset password test data	85

6.6 Reset password test case	86
6.7 Price offer test data	87
6.8 Price offer test case	87

Chapter 1:

Introduction

Chapter 1:

Introduction

LaborWorld is a comprehensive web platform that aims to bridge the gap between customers and skilled workers by providing a seamless and efficient solution for hiring worker services. In today's fast-paced and demanding world, finding reliable and competent workers for various tasks can often be a daunting and time-consuming process. LaborWorld is designed to address this challenge by offering a user-friendly platform where customers can quickly and easily connect with skilled workers for their specific needs. Whether it's household chores, repairs, or specialized services, LaborWorld provides a convenient and reliable way to find the right workers based on reputation, work expertise, and location. Workers can create profiles, manage their availability, and send requests for different jobs. The platform also offers emergency assistance to both customers and workers in case of any unforeseen issues during the work process. Through this documentation, we will explore the objectives, features, and implementation details of LaborWorld, providing a comprehensive guide to its functionalities and benefits. Join us as we embark on this exciting journey to simplify the process of finding and hiring skilled workers, ultimately enhancing the experience for both customers and workers in the LaborWorld ecosystem.

1.1 Opportunity & Stakeholders:

In today's world technology is involved in every field from Medical to aeronautical to farming almost everywhere. There is still a need for technology to reach some areas of different fields more effectively. Workers and service providers are one of them. In Pakistan, we can call a driver at our doorstep to take us to our destination. Likewise, we also want to call service providers at our doorstep to solve the various different tasks. In this way, work opportunities for the service providers could also increase with the

increasing opportunity of getting more work. There are some opportunities that we identified that could be solved using the technology.

1. Workers have to wait long outside for the clients to come and give some work. Due to less demand and high supply of workers, most of the workers could not get the work.
2. Since, it's not feasible for the workers to wait throughout the day for the work and they leave the place after a few hours of waiting which means they could not provide services at the different span of time in a day.
3. Workers gather to a specific place and most of them walk to the place, sometimes they could not see that someone is planning to get some work done. And they walk away from the opportunity around them.
4. Similarly, clients who need a service provider have to visit place to place to find a relevant service provider.
5. Sometimes finding different types of service providers becomes difficult for a client, because the client has to visit different locations to find relevant service providers.
6. Visiting multiple places is also a time and energy wasting activity.

1.2 Motivations and Challenges:

The rapid proliferation of mobile apps has revolutionized the way people interact with technology, creating a significant impact on their daily lives. While our project may not be highly advanced due to time constraints, the overarching goal we are pursuing holds immense importance as a fundamental step towards achieving more sophisticated tasks in the future.

In today's fast-paced world, individuals have diverse needs for various home services. Often, they rely on traditional methods such as seeking recommendations from friends and family, which can lead to challenges and inefficiencies. To address these issues, we aim to introduce an online home services platform that serves as a centralized hub for connecting customers with skilled workers .

Initially, our platform may not immediately capture the attention of users, as they may already be accustomed to conventional advertising methods or personal referrals. However, we strive to create an interface that is visually appealing, user-friendly, and highly functional. By incorporating attribute-level functionality for easy job posting and addressing the limitations of existing systems, we aim to make our platform highly productive and improve the overall experience for both customers and workers .

1.3 Goals and Objectives:

Our project, LaborWorld, aims to revolutionize the way people find and avail home services by providing a comprehensive online platform. The following goals and objectives have been defined to achieve this vision:

- 1. Develop an intuitive and user-friendly website application** that allows users to easily search for a wide range of home services and connect with qualified professionals.

2. **Implement a robust content management system** to ensure accurate and up-to-date information about the services offered and the professionals available on the platform.
3. **Create a reliable search module** that enables users to search for services based on their specific requirements, location, and preferred professionals.
4. **Integrate Google Maps functionality** to provide users with geolocation services, enabling them to find nearby professionals and visualize their locations on a map.
5. **Enable users to post their home service jobs** and allow professionals to bid on these projects, facilitating a transparent and competitive marketplace.
6. **Implement a secure user registration and login system**, ensuring the privacy and confidentiality of user information.
7. **Enable users to rate and review the services** provided by professionals, promoting accountability and helping other users make informed decisions.
8. **Incorporate social media sharing features** to allow users to easily share their positive experiences with LaborWorld, thereby increasing awareness and user engagement.
9. **Provide language translation functionality** to accommodate users who prefer different languages, enhancing accessibility and inclusivity.
10. **Develop a comprehensive dashboard for users and professionals**, allowing them to manage their profiles, view posted projects/bids, and track the progress of ongoing projects.
11. **Implement a reliable and efficient messaging system** to facilitate communication between users and professionals, ensuring seamless collaboration throughout the service delivery process.
12. **Enable professionals to showcase their services**, skills, and qualifications through detailed profiles, including service descriptions, working hours, photos, and other relevant information.

13. **Implement a robust review management system**, allowing users to provide feedback on professionals' performance, fostering transparency and trust.
14. **Integrate SMS-based service request functionality**, enabling users to directly request services from professionals with a simple tap on their mobile devices.
15. **Provide tools for professionals to manage their availability status**, indicating their free and busy time slots to help users schedule appointments effectively.
16. **Enable users and professionals to manage their profiles** and account settings, allowing them to update personal information, change passwords, and modify service preferences.

These goals and objectives form the foundation of LaborWorld, aiming to provide a reliable, efficient, and user-centric platform for home or office services, ultimately improving the overall experience for both users and professionals in the industry.

1.4 Solution Overview:

The proposed solution is a web-based platform that aims to streamline and enhance the process of finding and availing home services. It serves as a direct communication channel between professionals and consumers, offering convenience, efficiency, and a wide range of features to meet their needs. The key components of the solution are as follows:

1.4.1 User-Friendly Web and Mobile Interface

The platform provides a user-friendly interface accessible via web devices. The interface is designed to be intuitive and visually appealing, ensuring a seamless experience for both professionals and consumers.

Users can easily navigate through the platform, access various features, and perform tasks with ease.

1.4.2 Extensive Service Categories and Listings

The solution encompasses a comprehensive database of service categories and listings, catering to the diverse needs of consumers. Professionals can showcase their skills and expertise by creating profiles and listing their services under relevant categories. Consumers can browse through these listings, view detailed information about the services offered, and make informed decisions based on their requirements.

1.4.3 Direct Communication and Task Management

The platform facilitates direct communication between professionals and consumers, eliminating the need for intermediaries. Professionals can send job requests or bids for posted tasks, while consumers can easily review and select suitable professionals based on their profiles, ratings, and reviews. The platform also offers features for task management, allowing consumers to post specific tasks, and professionals to manage their assigned tasks efficiently.

1.4.4 Group and Skill Development

To support skill development and collaboration, the platform includes features that enable beginners to request to join groups and gain experience. This helps them kick start their careers and learn from more experienced professionals. Additionally, professionals can create their own groups, collaborate on larger tasks, and train other workers, fostering a supportive community.

1.4.5 Enhanced Security and Emergency Support

The solution prioritizes user safety and security. It incorporates measures such as account verification for professionals, ensuring the authenticity of their profiles. In case of any emergencies or safety concerns, both professionals and consumers have access to an emergency button, enabling them to seek immediate assistance. The platform's administration promptly responds to such alerts and takes appropriate action to provide the necessary support and security.

Overall, the proposed solution aims to revolutionize the home or industry services by providing a user-friendly platform that connects professionals and consumers, streamlines the process of service availing, fosters skill development, and ensures a safe and secure environment for all users.

Chapter 2:

Literature / Market Survey

Chapter 2:

Literature / Market Survey

2.1 Introduction

In this chapter, we present a comprehensive literature and market survey conducted in the area of providing home services. This survey plays a crucial role in providing valuable insights into the existing body of knowledge and the current landscape of the home services industry. By reviewing relevant literature and exploring market trends, we gain a deeper understanding of the challenges, opportunities, and advancements in this field. This knowledge not only helps us position our project within the context of providing home services but also serves as a foundation for developing an innovative and effective solution. Through this survey, we aim to identify gaps, trends, and best practices that will guide our project and contribute to the overall advancement of the home services sector.

2.2 Literature Review/Techologies Overview

In this section, we provide a description of the existing systems in the home services sector based on the comparison table. We analyze the features and capabilities of each system to understand the current landscape and identify areas for improvement. Here is an overview of the existing systems:

2.2.1 Fori Mazdoori:

Fori Mazdoori is a platform that offers job posting services for labor work. It allows users to post job requirements but lacks features such as payment integration, in-app calling and chatting, and job scheduling.

2.2.2 Super Tasker:

Super Tasker is a platform that serves as a trusted marketplace for outsourcing tasks. It provides payment integration, in-app calling and chatting, and job

scheduling features. However, it does not offer flexible contracts or a web platform.

2.2.3 Labour Square:

Labour Square is a system that focuses on connecting workers and customers. It offers in-app calling and chatting functionality but lacks payment integration, job scheduling, and a web platform.

2.2.4 Mahir App:

Mahir App is a platform that enables users to find skilled professionals for various home services. It offers features such as payment integration, in-app calling and chatting, and job scheduling. However, it does not provide flexible contracts or a web platform.

2.2.5 Wonoloo:

Wonoloo is a system that connects customers with service providers. It offers payment integration, in-app calling and chatting, and job scheduling features. However, it lacks flexible contracts and a web platform.

2.2.6 Gigsmart:

Gigsmart is a platform that focuses on providing on-demand staffing solutions. It offers features such as payment integration, in-app calling and chatting, and a web platform. However, it does not provide flexible contracts.

2.2.7 Veryable:

Veryable is a system that aims to connect businesses with flexible workers. It offers payment integration, in-app calling and chatting, and a web platform. However, it does not offer flexible contracts.

2.2.8 Hire Day Labor:

HireDay Labor is a platform that facilitates the hiring of day laborers. It offers features such as payment integration, in-app calling and chatting, and job scheduling. However, it does not provide a web platform.

2.2.9 Thumbtack:

Thumbtack is a system that connects customers with local professionals. It offers payment integration, in-app calling and chatting, and job scheduling features. However, it does not provide flexible contracts.

Based on the comparison, it is evident that no single existing system encompasses all the desired features. Our research aims to address this gap and develop a comprehensive solution that integrates the best features from existing systems while incorporating additional functionalities to enhance the user experience and efficiency in the home services industry.

2.2.10 Existing System/ Description of the Current Situation:

	Fori mazdoori	Super Tasker	Labour Square	Mahir	Wonoloo	Gigsmart	verable	HireDay Labor	Thumb tack
Job Posting	✗	□	✗	✗	✗	□	□	□	□
Payment Integration	□	□	✗	□	□	□	✗	✗	✗

In-App calling & chatting	□	□	✗	□	✗	□	□	□	□
Job Scheduling	✗	✗	✗	□	✗	□	□	□	✗
flexible contracts	✗	✗	✗	✗	✗	□	✗	✗	✗
Online Contract	□	□	□	□	✗	✗	□	□	✗
Web Platform	□	□	✗	✗	□	□	✗	✗	□
Mobile App	✗	□	□	□	□	□	□	□	□

Table 2.1 Comparison of Existing Systems.

2.3 Summary

This chapter provides an overview of LaborWorld, our labor-focused platform. We discuss its key features and functionalities, highlighting its unique approach to addressing labor market challenges. We also compare LaborWorld with existing systems to identify opportunities for improvement. Additionally, we present findings from a market survey to understand market trends and preferences. This chapter sets the foundation for developing a user-centric solution that meets the evolving needs of workers and employers within the LaborWorld platform.

Chapter 3:

Requirement Engineering

Chapter 3:

Requirement Engineering

3.1 Introduction

This chapter provides an overview of the requirement engineering process for the development of LaborWorld. It highlights the significance of understanding the needs and expectations of different stakeholders involved in the labor industry.

3.2 Problem Scenarios

In this digital world some people are still working with their old methods which could be made more efficient digitally. One of them is related to the worker hiring process. Finding workers is also a very messy type of work in the beginning. Workers mostly gathered in the specific places of the specific areas where they wait for the client who comes and offers them work. Whereas, most of the workers do not get hired and go back home helpless. Also, there's a problem that a worker sometimes does not get the client because of the lack of communication skill despite having a decent work experience.

It's also not possible for the worker to wait the whole day in the search of work. This causes them to fail the chance of getting part time work as well because they leave the place in the afternoon. Meanwhile, a client also suffers problems like he faces difficulty in finding the right worker for his job.

Because of the inflation and poverty in Pakistan everyone tries to get some kind of work to earn some money and this causes them to work on whatever opportunity they are getting. This affects the client and worker's experience ending up in losing the client for the worker. Also, there's no standard rate of workers as well which end up in losing more money either from clients or workers. There's also a lack of part time opportunities for the workers while some people have time to hire a worker in the evening based on their availability.

3.2.1 Problem Statement 1:

The Problem of	There are many inexperienced and unqualified workers who are working but experienced and qualified workers are not getting the proper work. Whereas, clients hire inexperienced workers.
Affects	Workers, clients
The result of which	There's no track of workers' work and the worker cannot express the qualification of his work to satisfy the client. Also, the client cannot see the track record of worker work to satisfy himself.
Benefits of	Our System will enforce the clients and workers to provide suitable feedback after completing the work, maintaining the track record of workers' work and clients which will be prominently visible on the profile.

Table 3.1 Problem Statement 1

3.2.2 Problem Statement 2:

The Problem of	The workers also face difficulty understanding the past behavior of clients with workers.
Affects	Workers

The result of which	There are no proper feedback systems for the clients who hired the workers and didn't behave well in terms of contracts.
Benefits of	Our system will provide the feedback mechanism of two ways so that workers can also express their opinions about the clients they have worked with.

Table 3.2 Problem Statement 2

3.2.3 Problem Statement 3:

The Problem of	Client spent his time finding a worker near him.
Affects	Client
The result of which	Client wastes his time finding suitable workers for his work.
Benefits of	Our system will provide a worker profile and their experienced work with our client previously and provide a worker that is near to the client order to save time.

Table 3.3 Problem Statement 3

3.2.4 Problem Statement 4:

The Problem of	The workers can't get jobs easily at any time when they want to work. Stand out from the home full day and return empty handed.
Affects	Workers , Client
The result of which	Flexibility of time to the worker. Workers can work any time they want.
Benefits of	Clients can take any service at any time from workers.

Table 3.4 Problem Statement 4

3.2.5 Problem Statement 5:

The Problem of	Some workers pay commissions to the shops (like sanitary or electrician etc.) in order to get jobs from their customers.
Affects	Worker
The result of which	Workers lose more money by giving commission money to the shops.

Benefits of	workers can get hired for a work through our system without any commission money to pay. So, this saves workers money.
--------------------	--

Table 3.5 Problem Statement 5

3.3 Functional Requirements:

Functional requirements are articulated based on the various user categories within the system. The system accommodates the following user types:

3.3.1 Worker's Requirements:

FR.01: Worker shall be able to register using his/her first name, last name, password, gender, role, city, phone number, email and picture.

FR.02: Worker shall be able to sign in (login) by using email address or phone number and password.

FR.03: Worker shall be able to logout from the system.

FR.04: Worker shall be able to reset its account password.

FR.05: Workers shall be able to view the jobs listings.

FR.06: Workers shall be able to add the skills he works in.

FR.07: Workers shall be able to view the job posted by clients.

FR.08: Workers shall be able to request a job posted by clients.

FR.09: Worker shall be able to send an offer price when requesting for the job.

FR.10: Workers shall be able to set prices for hourly, full-time service.

FR.11: Workers shall be able to chat with clients for job dealing.

FR.12: Workers shall be able to update his profile

FR.13: Workers shall be able to see his feedback on the profile.

FR.14: Workers shall be able to edit skills in the profile.

FR.15: Worker shall be able to give feedback to client's job which he has done.

FR.16: Workers shall be able to view contracts he has assigned to.

FR.17: Workers shall be able to notify the client when he reaches the job location.

FR.18: Workers shall be able to view all the previous work he/she has done.

FR.19: Workers shall be able to view the client's profile before applying to the job.

FR.20: Workers shall be able to get notified by sms or call when a relevant work is posted nearby.

FR.21: Workers shall be able to see in-app notifications about the new job posted.

FR.22: Workers shall be able to see his requests and status of request.

FR.23: Workers shall be able to see the previously completed jobs.

FR.24: Workers shall be able to see feedback on previously completed jobs.

FR.25: Workers shall be able to give feedback on previously completed jobs.

3.3.2. Client's Requirement:

FR.26: Clients shall be able to register himself using his/her first name, last name, password, gender, role, city, phone number, email and picture.

FR.27: Client shall be able to sign in (login) by using email address and password.

FR.28: Client shall be able to logout from the system.

FR.29: Client will be able to recover his/her password if he/she forgets through mobile number.

FR.30: Clients shall be able to search workers.

FR.31: Clients shall be able to search workers by skills, name or city.

FR.31: Clients shall be able to see nearby workers.

FR.32: Clients shall be able to view the worker profile.

FR.33: Clients shall be able to see the feedback given on his job.

FR.34: Clients shall be able to message workers using in app chatting.

FR.35: Clients shall be able to post the job.

FR.36: Clients shall be able to edit/update the job.

FR.37: Clients shall be able to delete the job.

FR.38: Client shall be able to see his open jobs.

FR.39: Client shall be able to view workers' requests on his job.

- FR.40:** Client shall be able to accept the worker request.
- FR.41:** Client shall be able to reject the worker's request.
- FR.42:** Client shall be able to see the price and message with the worker request.
- FR.43:** Client shall be able to see his contracts.
- FR.44:** Client shall be able to see updated price of job after accepting worker offer.
- FR.45:** Client shall be able to update the price before starting the job.
- FR.46:** Client shall be able to start the job after deciding with the worker.
- FR.47:** Client shall be able to cancel the contract before starting the job.
- FR.48:** Client shall be able to end the job after starting.
- FR.49:** Client shall be able to give feedback on the job after ending the job.
- FR.50:** Client shall be able to update his profile.
- FR.51:** Client shall be able to view his posted jobs.
- FR.52:** Client shall be able to change his password.
- FR.53:** Client shall be able to receive worker sms when he reaches the job location.
- FR.54:** Client shall be able to receive the sms when worker requests to end the job.

3.3.3. Admin's Requirement:

- FR.55:** Admin shall be able to register itself using his/her username , email , gender , role and password.
- FR.56:** Admin shall be able to login by using email and password.
- FR.57:** Admin shall be able to logout from the system.
- FR.58:** Admin shall be able to create new users.
- FR.59:** Admin shall be able to delete existing users.
- FR.60:** Admin shall be able to assign roles to users.
- FR.61:** Admin shall be able to update users.
- FR.62:** Admin shall be able to create a new job.
- FR.63:** Admin shall be able to close the job.

FR.64: Admin shall be able to edit the jobs.

FR.65: Admin shall be able add categories of jobs.

FR.66: Admin shall be able to change feedback if requested by the user.

FR.67: Admin shall be able to a new user role.

3.4 Non-Functional Requirements:

In our Final Year Project (FYP), we aim to develop a comprehensive worker-hiring platform that connects workers and clients. To ensure the successful implementation and performance of our system, we have identified a set of functional requirements that specify the core functionalities to be developed. Now, we will address the corresponding non-functional requirements (1s) that define how the system should behave and what limits exist on its functionality. These NFRs are essential to guarantee the overall performance, security, usability, and reliability of the system.

Goal: Security

NFR 1: Access Control and Data Protection:

The system shall enforce strict access controls, ensuring only authenticated and authorized users can access resources.

Goal: Performance

NFR 2: Responsive Query Handling

The system shall respond to user queries within 2 seconds under normal load conditions.

During peak times, the average response time for user queries shall not exceed 3 seconds to maintain a seamless user experience.

Goal: Scalability

NFR 3: Future-proof Architecture

The system's architecture shall allow seamless integration of new features without disrupting existing functionalities.

Goal: Maintainability

NFR 4: Simplified Maintenance

The system shall employ modular and well-documented code practices, facilitating easy maintenance, updates, and bug fixes.

Chapter 4:

System Design

Chapter 4:

Systems Design

4.1 Introduction

This chapter provides an introduction to the systems design phase of the LaborWorld platform. It highlights the importance of designing a robust and scalable system architecture to meet the requirements identified in the previous chapters.

4.2 Architectural Design

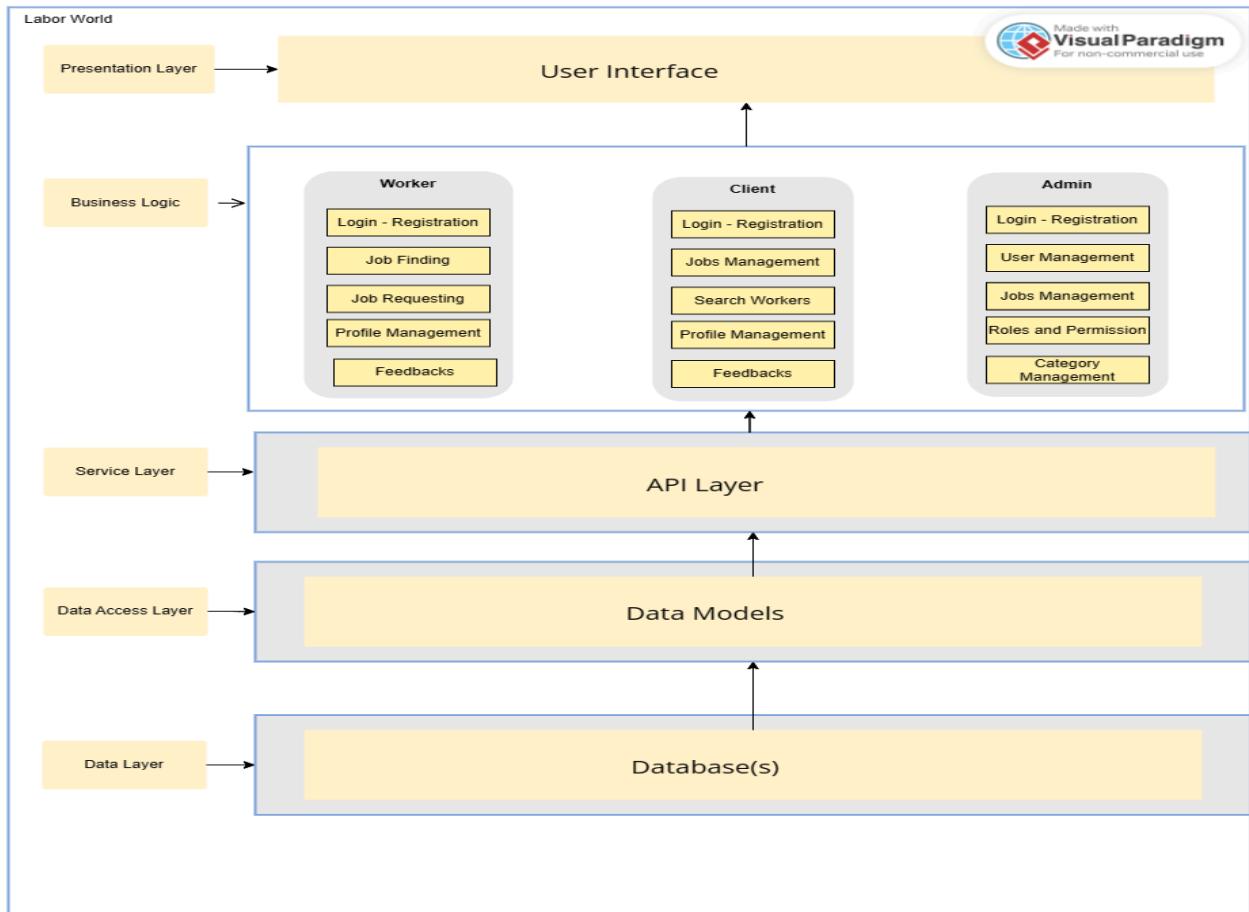


Figure 4.1: Architectural Diagram

4.3 Detail Design

The detail design phase focuses on the specific design aspects of LaborWorld. It involves designing the various modules, components, and interfaces required for the platform. This section provides a comprehensive overview of the detailed design considerations and decisions made to support the functionality and usability of the system.

4.3.1 Use Case Diagram

4.3.1.1 Client Use Case Diagram

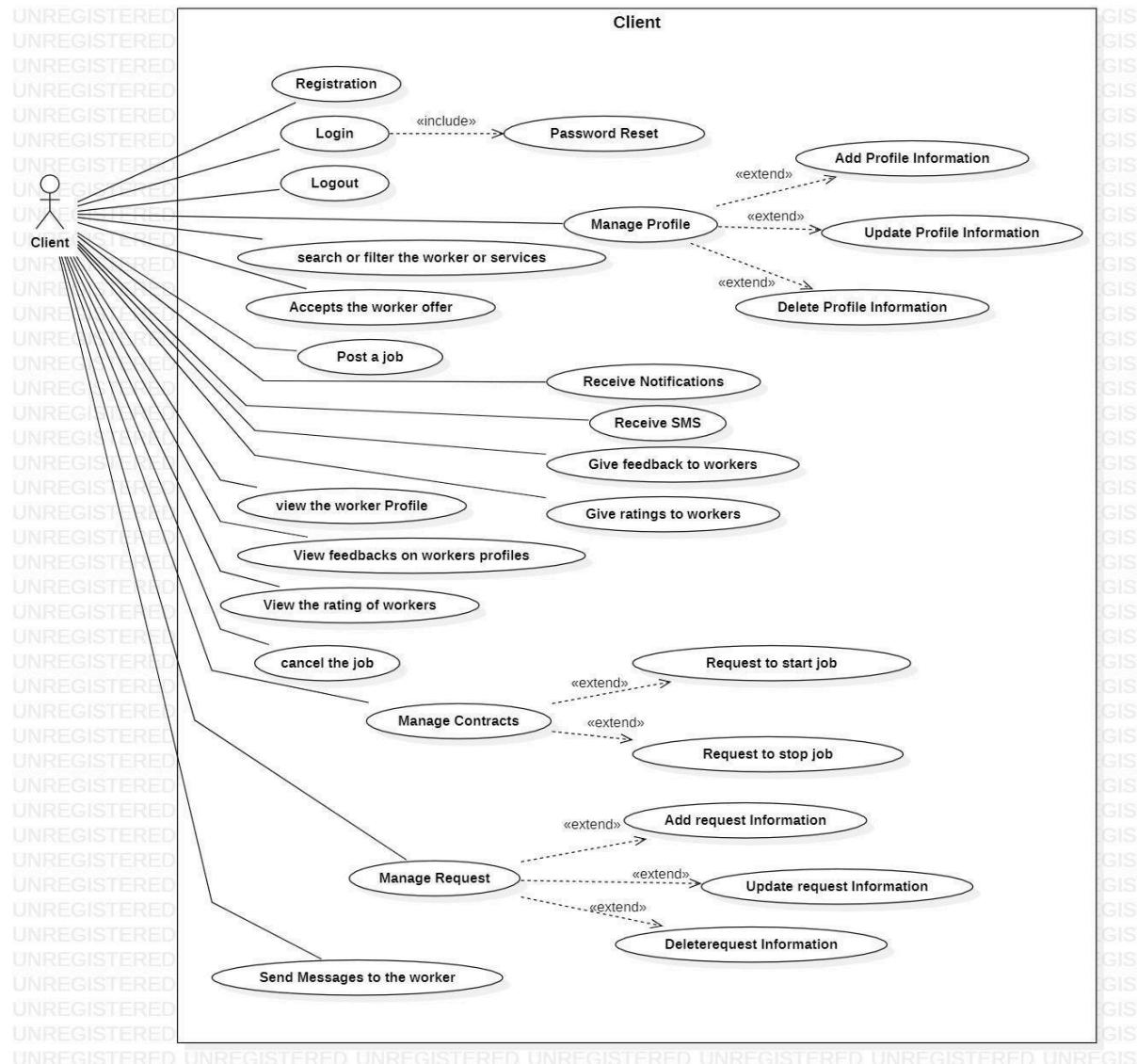


Figure 4.2 Client Use Case Diagram

4.3.1.2 Worker Use Case Diagram

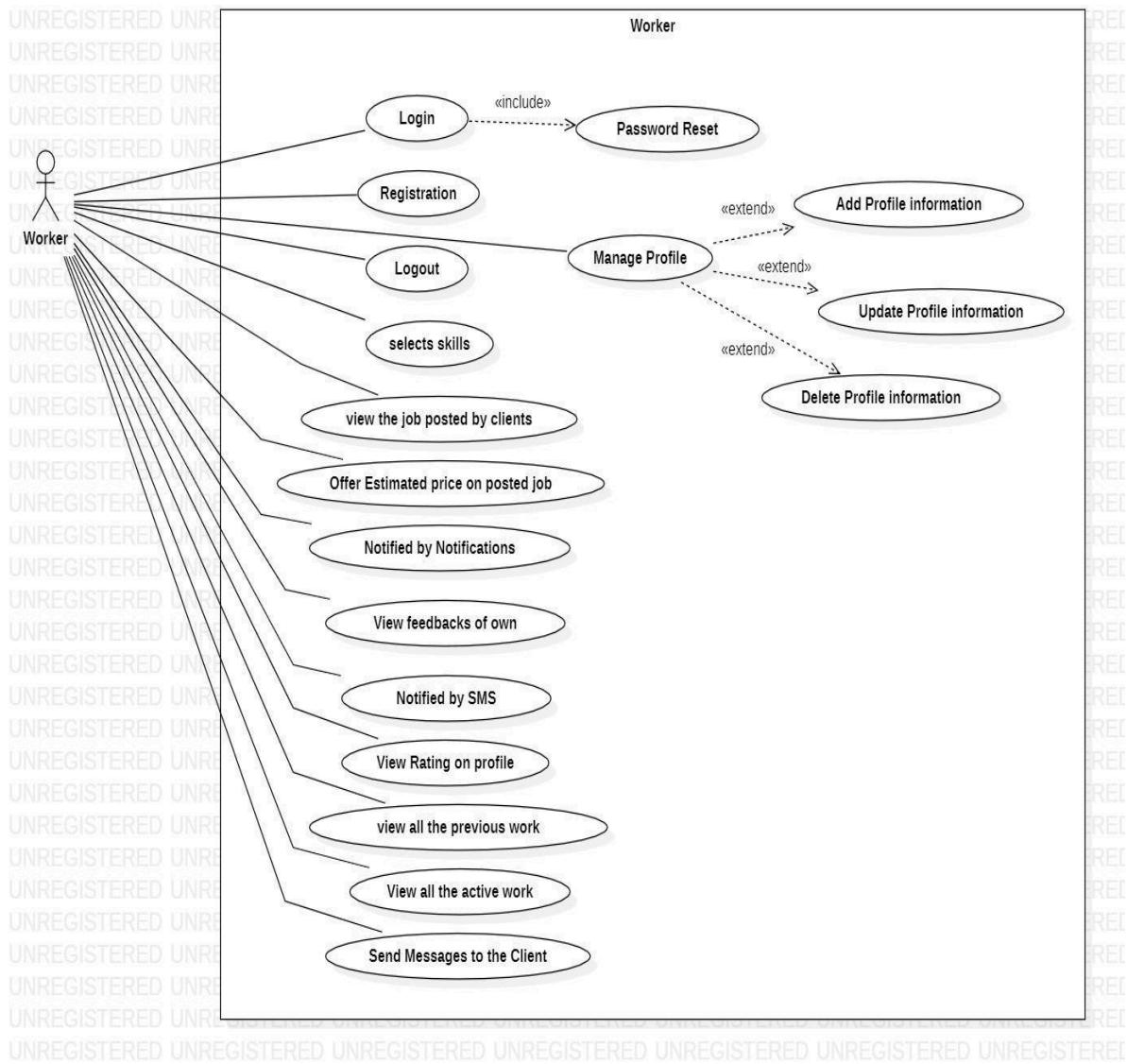


Figure 4.3 Worker Use Case Diagram

4.3.1.3 Common Use Case Diagram



Figure 4.4 Common Use Case Diagram

4.3.2 Detail Use Case Design

A Detailed Use Case Diagram provides a comprehensive and detailed view of the interactions between actors and use cases in a system. It illustrates the step-by-step collaboration between users or external systems (actors) and the specific functionalities (use cases) of the system.

4.3.2.1 Common Use Cases:

4.3.2.1.1 Registration

Use Case ID	User_001
Use Case Name	Registration
Actors	Client , Worker
Description	Users will create an account on our platform.
Precondition	<ul style="list-style-type: none">● The user has a valid email address.● The user has access to a computer or mobile device with an internet connection.
Postcondition	The user has created a new account on the platform.
Normal Flow	<ol style="list-style-type: none">1. The user visits the platform's registration page.2. The user enters their first name, last name, age, phone number, email address, password, confirm password, role, and profile picture.3. The user clicks on the "Register" button.4. The platform validates the user's input.

	5. Message will display “ Success! You have been registered successfully ”
Alternative Flow	A1: If validation fails at step-3, the message will display “Error! Registration Failed”

Table 4.1 Registration Use Case

4.3.2.1.1 Activity Diagram for Registration:

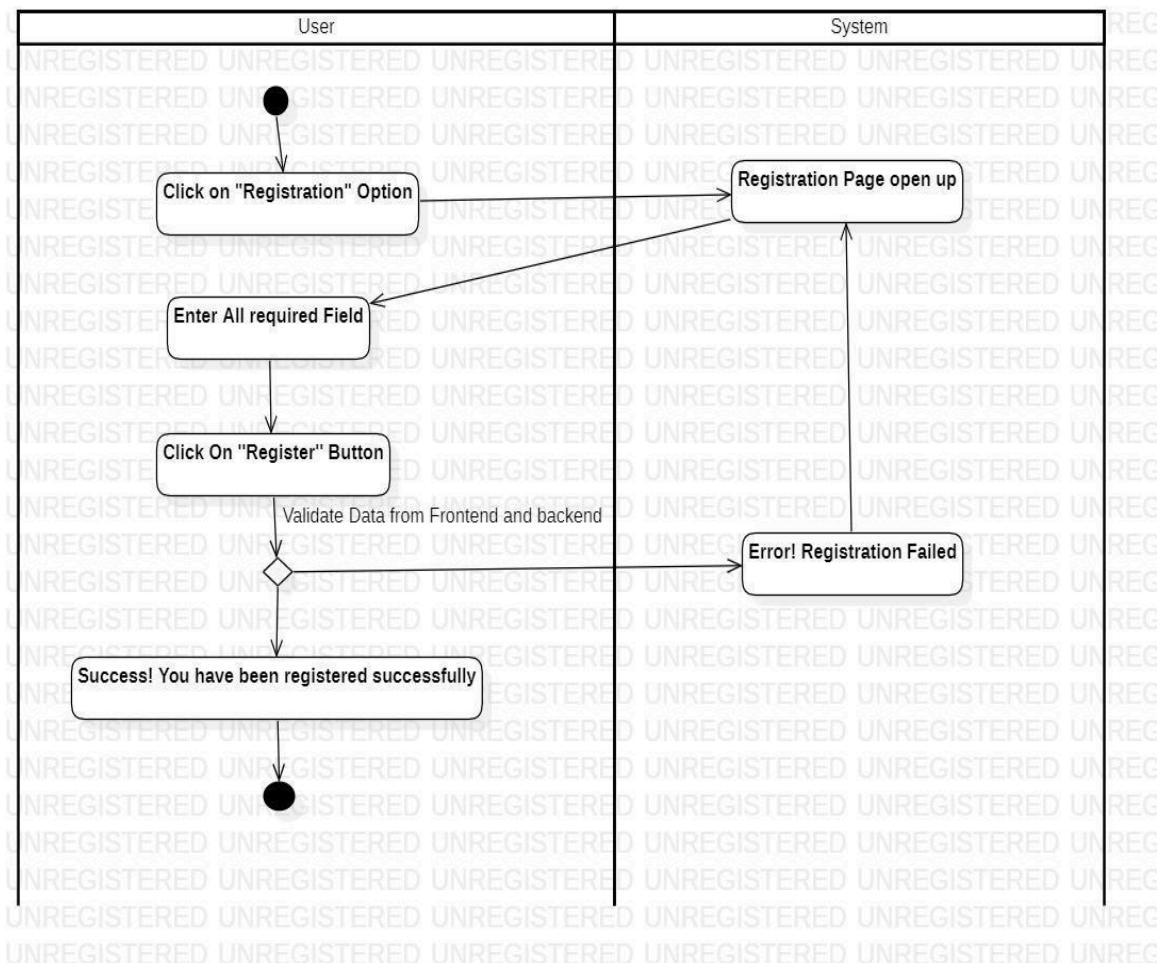


Figure 4.5 Registration

4.3.2.1.2 Login

Use Case ID	User_002
Use Case Name	Login
Actors	Client , Worker
Description	Users will log in to their account on the platform.
Precondition	<ul style="list-style-type: none"> ● The user has a valid email address associated with an existing account on the platform. ● The user has access to a computer or mobile device with an internet connection.
Postcondition	<ul style="list-style-type: none"> ● Upon successful login, the user is redirected to the platform's dashboard. ● The user's session is maintained, allowing them to access various platform features.
Normal Flow	<ol style="list-style-type: none"> 1. The user navigates to the platform's login page. 2. The user enters their registered email address and registered password in the input fields. 3. The user clicks on the "Login" button. 4. The platform validates the user's credentials by comparing the entered email address and password against the stored credentials for that account. 5. If the credentials are valid, the user's session is established, and they are redirected to the platform's dashboard. 6. the system displays a confirmation message: "Success! Login Successful."

Alternative Flow	<p>A1: Invalid Email: If the entered email address does not match an existing account, the system displays an error message: "Error! Invalid email address. Please check your email and try again."</p> <p>A2: Incorrect Password: If the entered password is incorrect, the system displays an error message: "Error! Incorrect password. Please enter the correct password and try again."</p>
-------------------------	--

Table 4.2 Login Use Case

4.3.2.1.2 Activity Diagram for Login:

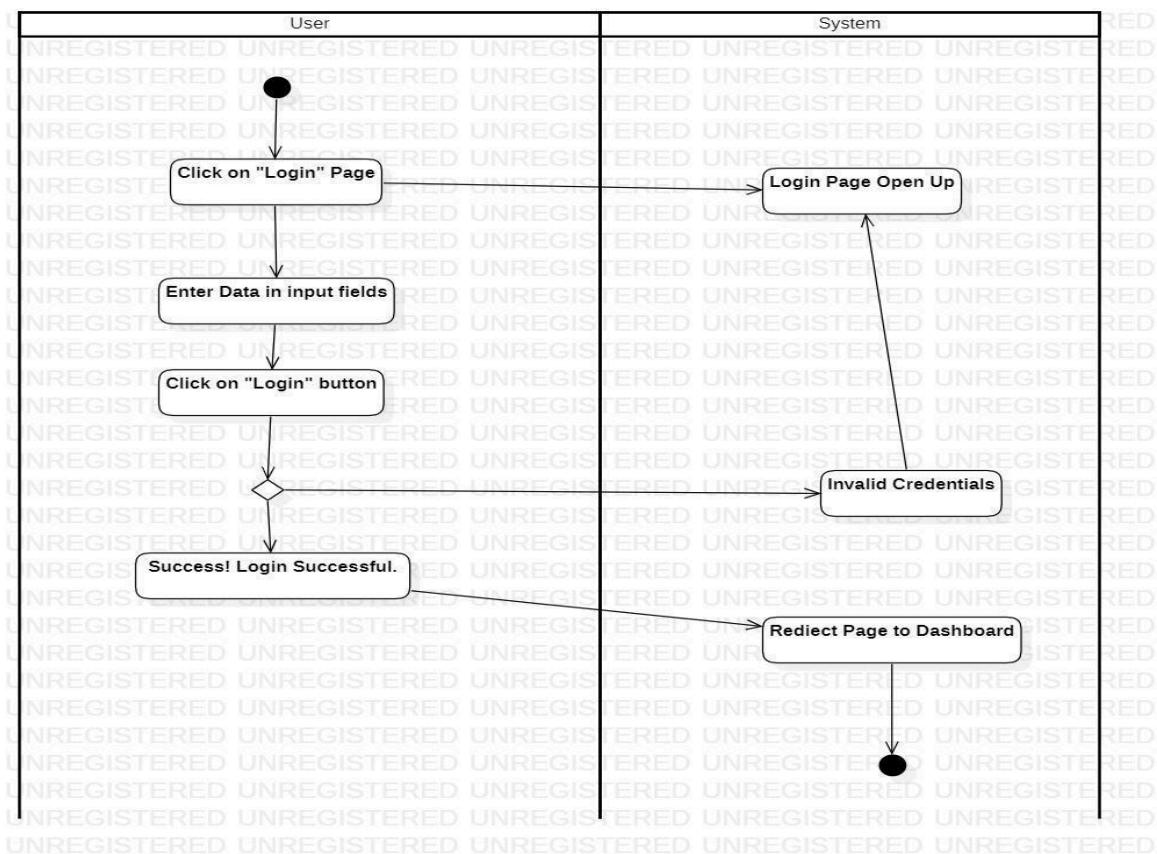


Figure 4.6 Login

4.3.2.2 Client Use Cases:

4.3.2.2.1 Post a Job

Use Case ID	Client_001
Use Case Name	Post a Job
Actors	Client
Description	Client will post a job.
Precondition	Client is logged into the system.
Postcondition	Client will post a job successfully.
Normal Flow	<ol style="list-style-type: none"> 1. Client login successfully. 2. Dashboard Page is opened. 3. Click on the option “Post a Job” on the header bar. 4. Client Enter job type, price for a job , job description , time duration for work details. 5. Click on the “Submit button”. 6. The platform validates the user's credentials 7. Message displayed “Post a Job Successfully” 8. Client will redirect on the dashboard Page again.
Alternative Flow	<p>A1: Client clicks on the Button “Add New Job” on the dashboard and the system will take the client to the “Post a Job” page.</p> <p>A2: If validation fails at step-6, the message will display “ Error! Job posting failed! ”</p>

Table 4.3 Post a Job

4.3.2.2.1 Activity Diagram for Post a Job:

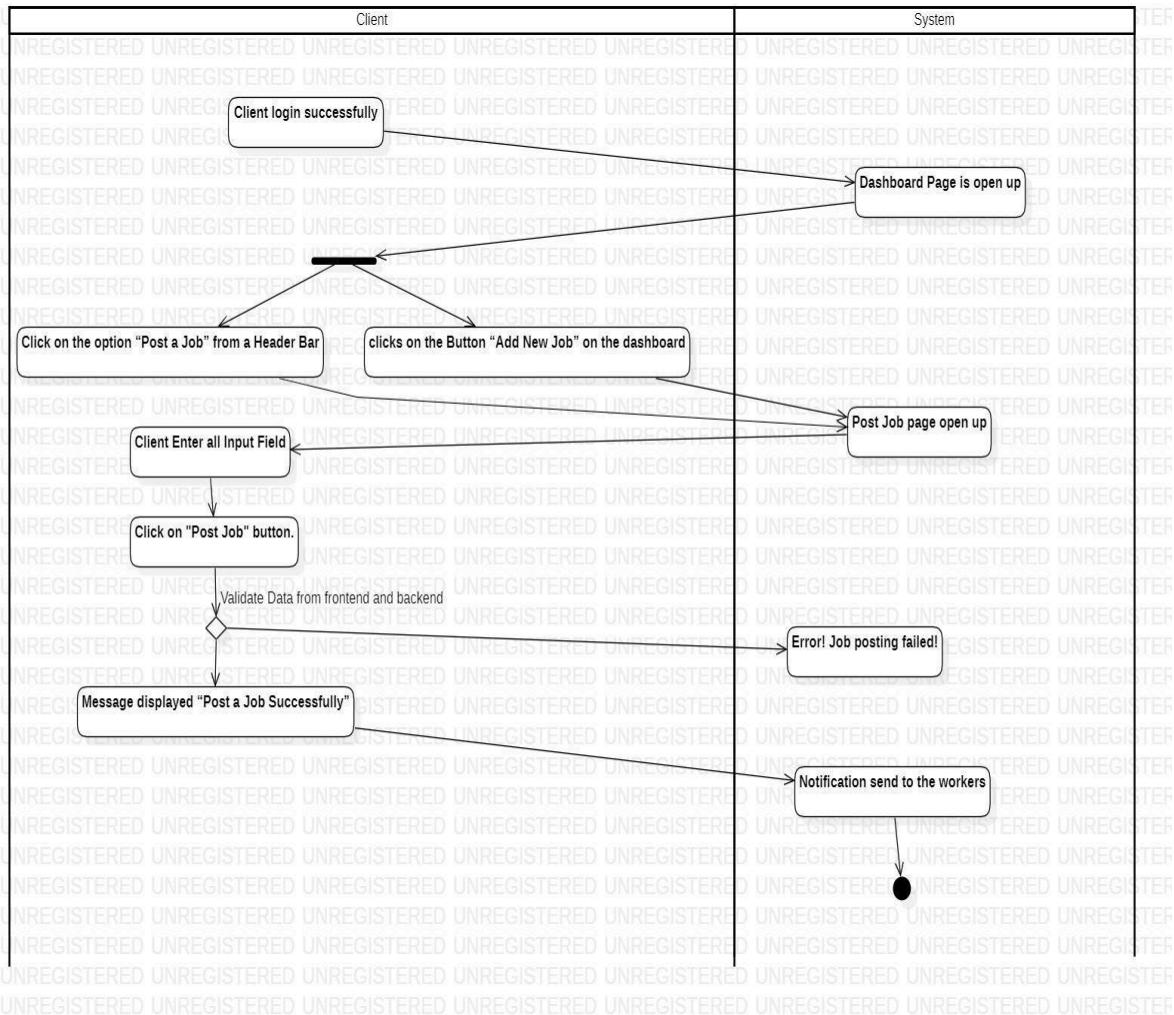


Figure 4.7 Post a Job

4.3.2.2 Client search the worker

Use Case ID	Client_002
Use Case Name	Client can search the worker
Actors	Client
Description	Clients can search the worker.

Precondition	Client is logged into the system.
Postcondition	Clients searched the Worker.
Normal Flow	<ol style="list-style-type: none"> 1. Client login successfully. 2. Dashboard page is opened. 3. Client clicks on the option “Worker” on the header bar. 4. Enter the worker information. 5. Click on the “Search Worker” button. 6. List of particular workers or specific worker gigs showed up.
Alternative Flow	<ul style="list-style-type: none"> • A1: on the step:6, if no such name and job type are found in the database. A message will display “No Worker Found”.

Table 4.4 Client can search the worker

4.3.2.2.2 Activity Diagram for Search the worker:

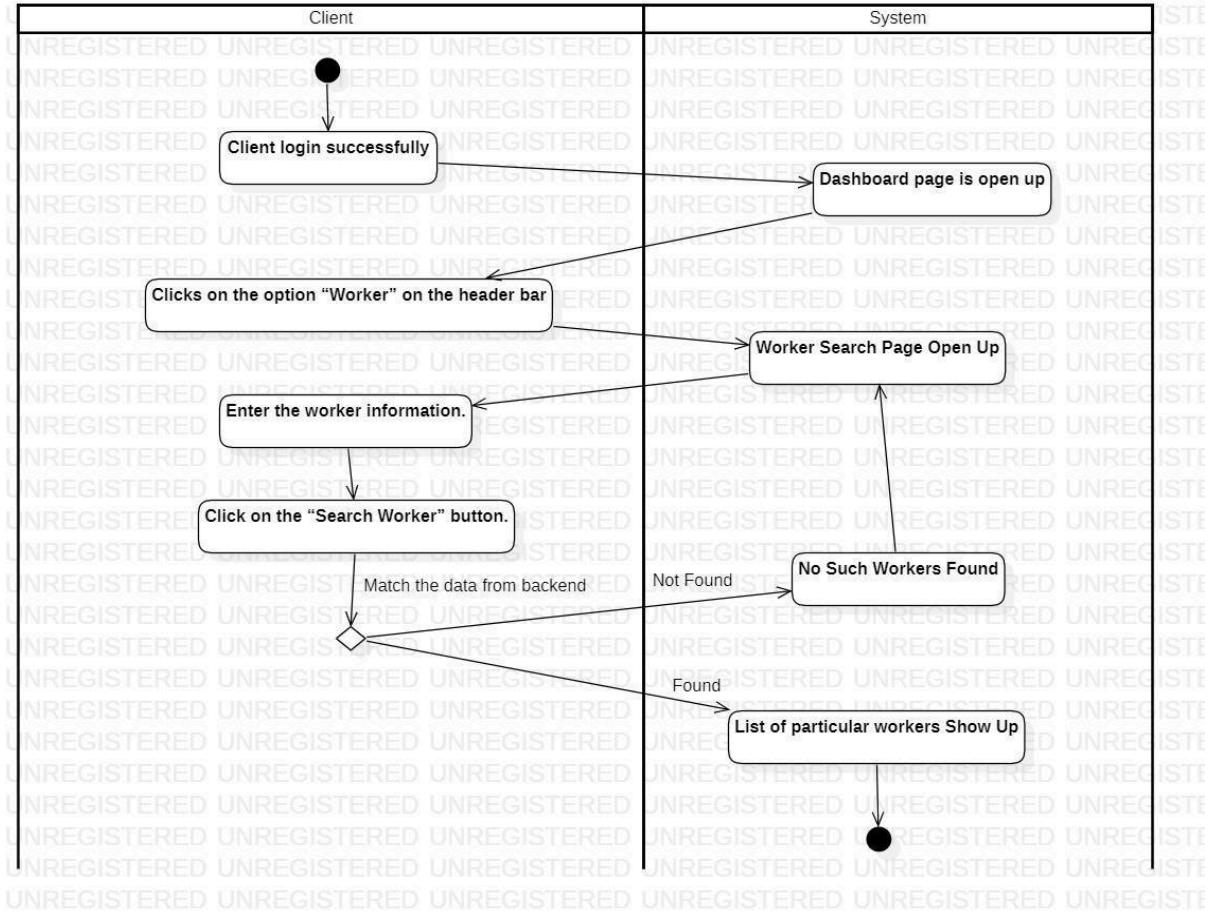


Figure 4.8 Search the worker

4.3.2.2.3 View the worker Profile

Use Case ID	Client_003
Use Case Name	View the worker Profile.
Actors	Client
Description	Clients can view the worker Profile
Precondition	Client has successfully logged in and has initiated a search for a specific worker

Postcondition	Clients are able to view the worker's profile information, including their skills, experience, portfolio, and ratings.
Normal Flow	<ol style="list-style-type: none"> 1. Clients search for workers or a worker. 2. Client click on the “Search Worker” button. 3. A list of particular workers gigs showed up. 4. Clients click on a workers gig. 5. System will redirect clients to the specific worker profile.
Alternative Flow	<ul style="list-style-type: none"> • A1: on the step:2, if no such name, location and job type found in database.A message will display “No Worker Found”.

Table 4.5 View the worker Profile

4.3.2.2.3 Activity Diagram for View the Worker Profile:

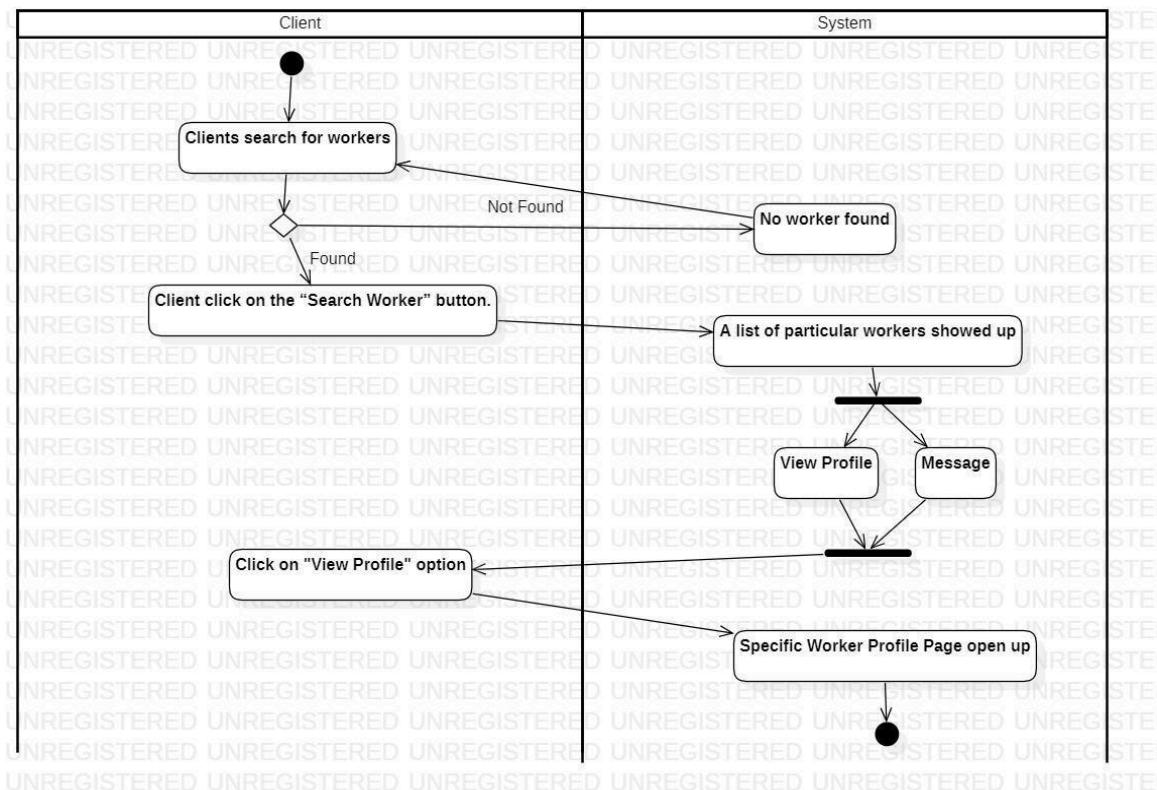


Figure 4.9 View worker profile

4.3.2.2.4 Hire a worker

Use Case ID	Client_004
Use Case Name	Hire a worker
Actors	Client , Worker
Description	Worker has given a price suggestion and the client gets this price suggestion. Client will decide whether to hire the worker or not.
Precondition	Clients get price suggestions from workers.
Postcondition	Client hired a worker successfully.
Normal Flow	<ol style="list-style-type: none"> 1. Client login successfully. 2. Dashboard page is opened. 3. Client clicks on the option “contract” on the header bar. 4. Contract page opens up. 5. Client client on “Pending” Tab. 6. List of pending jobs shows up. 7. Client selects one pending job. 8. Client updates the price on the job. 9. Client sends a price offer to the worker. 10. Workers accept the offer. 11. Hire Button becomes active (Clickable). 12. Client then clicks on the hire button. 13. Message displayed “Success! Hired Worker Successfully”
Alternative Flow	A1: on Step 10: if the worker rejects the price offer , the hire button will remain unclickable.

Table 4.6 Hire a worker

4.3.2.2.4 Activity Diagram for Hire a Worker:

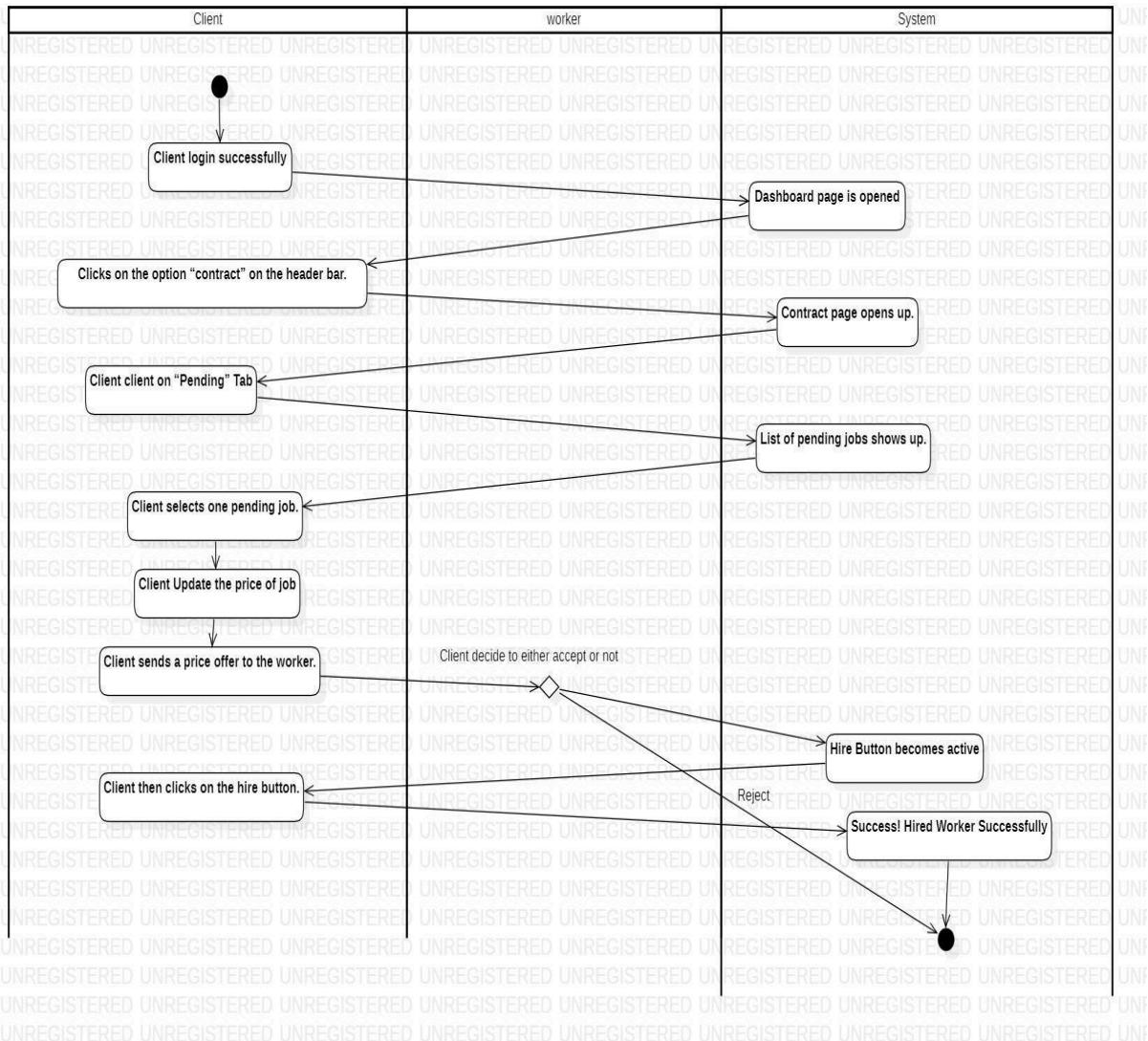


Figure 4.10 Hire a worker

4.3.2.2.5 End a job

Use Case ID	Client_005
Use Case Name	End a Job
Actors	Client
Description	Clients terminate a job after the worker has completed the work.

Precondition	<ul style="list-style-type: none"> • The client is logged into the system. • The worker has completed the agreed-upon work for the job.
Postcondition	<ul style="list-style-type: none"> • The client has successfully ended the job. • The job status is changed to "Completed," and the payment process is initiated.
Normal Flow	<ol style="list-style-type: none"> 1. Client login successfully. 2. Dashboard page is opened. 3. Client clicks on the option “contract” on the header bar. 4. The client navigates to the "Active" tab to view a list of ongoing jobs. 5. The client selects the specific job for which they want to end a job. 6. The client clicks on the "End" button. 7. Alert Box opens up with two options, “Yes, End Job” or “Cancel”. 8. Click on the “Yes, End Job” option. 9. The system validates the client's decision and initiates the job closure process. 10. A confirmation message is displayed indicating that the job has been successfully ended.
Alternative Flow	<p>A1: On step 9: If the client clicks on the “Yes, End Job” option, job status will remain active.</p>

Table 4.7 End a Job

4.3.2.2.5 Activity Diagram for End a Job:

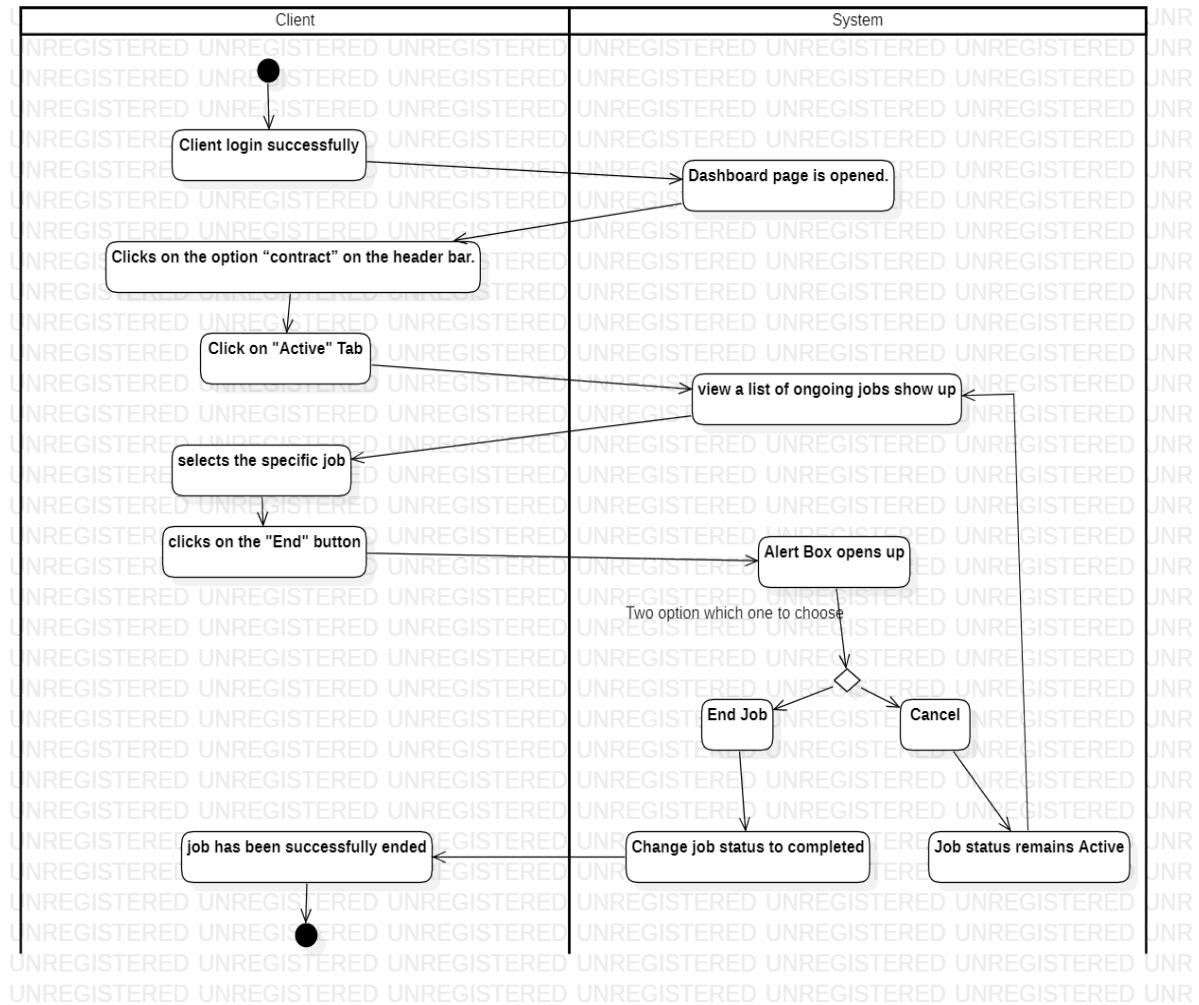


Figure 4.11 End a Job

4.3.2.2.6 Manage Profiles

Use Case ID	Client_006
Use Case Name	Clients can manage Profile.
Actors	Client
Description	Clients can add, delete and update their Profile information.

Precondition	Client is logged into the system.
Postcondition	Clients added the Profile information. Clients updated the Profile information. Clients deleted the Profile information.
Normal Flow	<ol style="list-style-type: none"> 1. Client login successfully. 2. Dashboard Page is opened. 3. Client clicks on the profile setting option on the header bar. 4. Profile page is opened up with details of name, phone number, profile picture, home address and email address. 5. Click on the “Edit” button and enter the updated values. 6. Message display “successfully changed the value”
Alternative Flow	<ul style="list-style-type: none"> • A1: Client clicks on the Button “Edit the Profile” on the dashboard and the system will take the client to the Profile page.

Table 4.8 Manage Profile

4.3.2.3 Worker Use Cases:

4.3.2.3.1 Worker can view jobs

Use Case ID	Worker_001
Use Case Name	Worker can view jobs
Actors	Worker
Description	A worker views jobs posted by clients on the platform
Precondition	Worker is logged into the system.
Postcondition	Worker successfully viewed jobs posted by Client

Normal Flow	<ol style="list-style-type: none"> 1. Worker login successfully. 2. Worker dashboard Page is opened. 3. The platform displays a list of jobs related to the worker's skills. 4. The worker browses the list of jobs and reviews the job details, including the job description, requirements, and budget. 5. The worker can click on specific jobs to view more detailed information, such as the client's profile and contact information.
Alternative Flow	A1: On step 3: if skill is not added by the worker, then a nearby job will be shown.

Table 4.9 Worker can view jobs

4.3.2.3.1 Activity Diagram for View all Jobs:

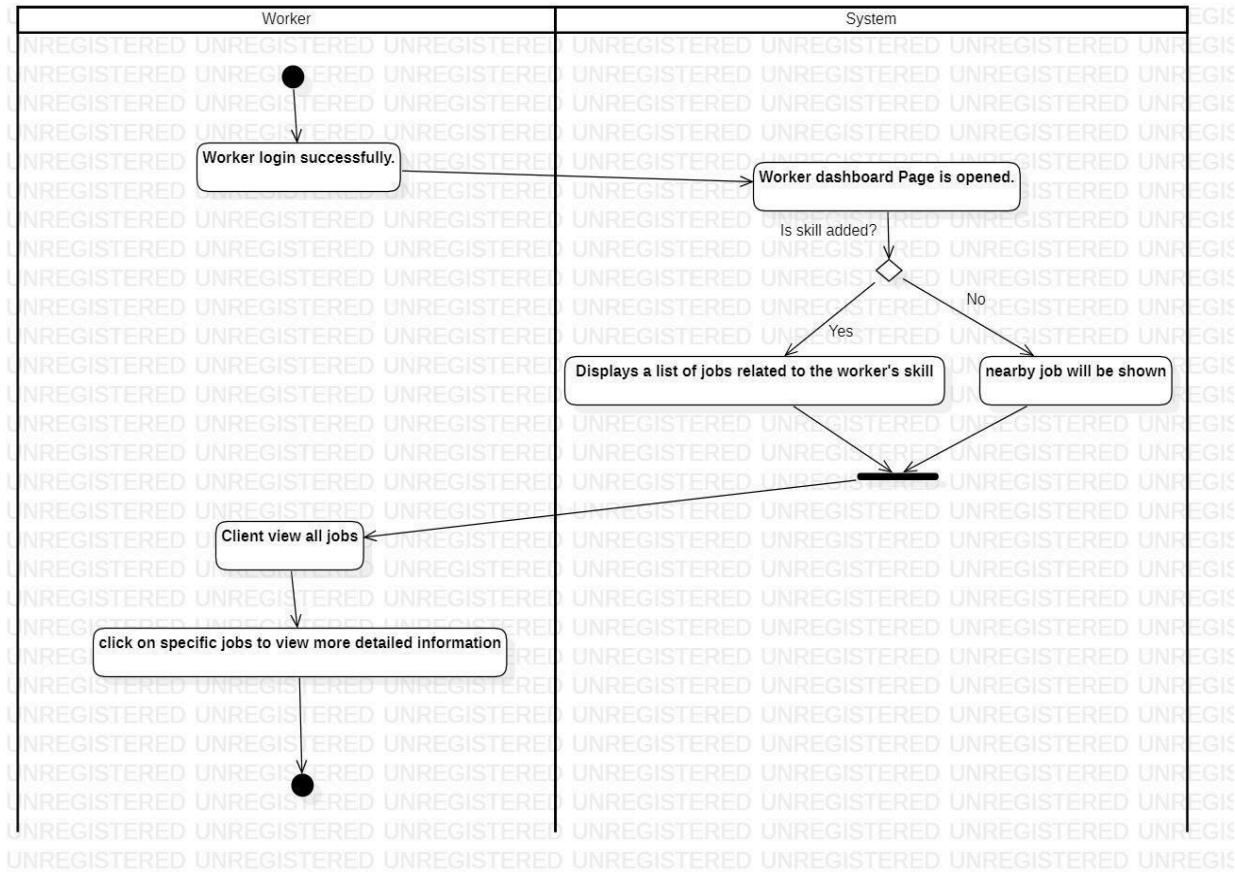


Figure 4.12 View all Jobs

4.3.2.3.2 Worker can give price offer

Use Case ID	Worker_002
Use Case Name	Worker can give price offer
Actors	Worker
Description	Workers request on the job posted by the client by giving a price offer for a job.
Precondition	worker has logged in and viewed a specific job posted by a client.

Postcondition	Workers request a price offer on a job posted by Client successfully and the amount of detail sent to the client.
Normal Flow	<ol style="list-style-type: none"> 1. Worker login successfully. 2. Worker dashboard Page is opened. 3. Worker view list of jobs. 4. Workers select one of the jobs. 5. Worker clicks on the “Send Request” on the job. 6. Enter a price suggestion and any relevant message to the client. 7. Click on the send button. 8. System validates the price offer(e.g., positive value) and sends it to the client 9. Message shows up “Success! Price Request Sent Successfully”.
Alternative Flow	<ul style="list-style-type: none"> • A1: on step 8, If a negative value is entered in price input, the system shows Message “Price Offer failed: Please enter a valid positive price.”.

Table 4.10 Worker can give price offers

4.3.2.3.2 Activity Diagram for Price Offer:

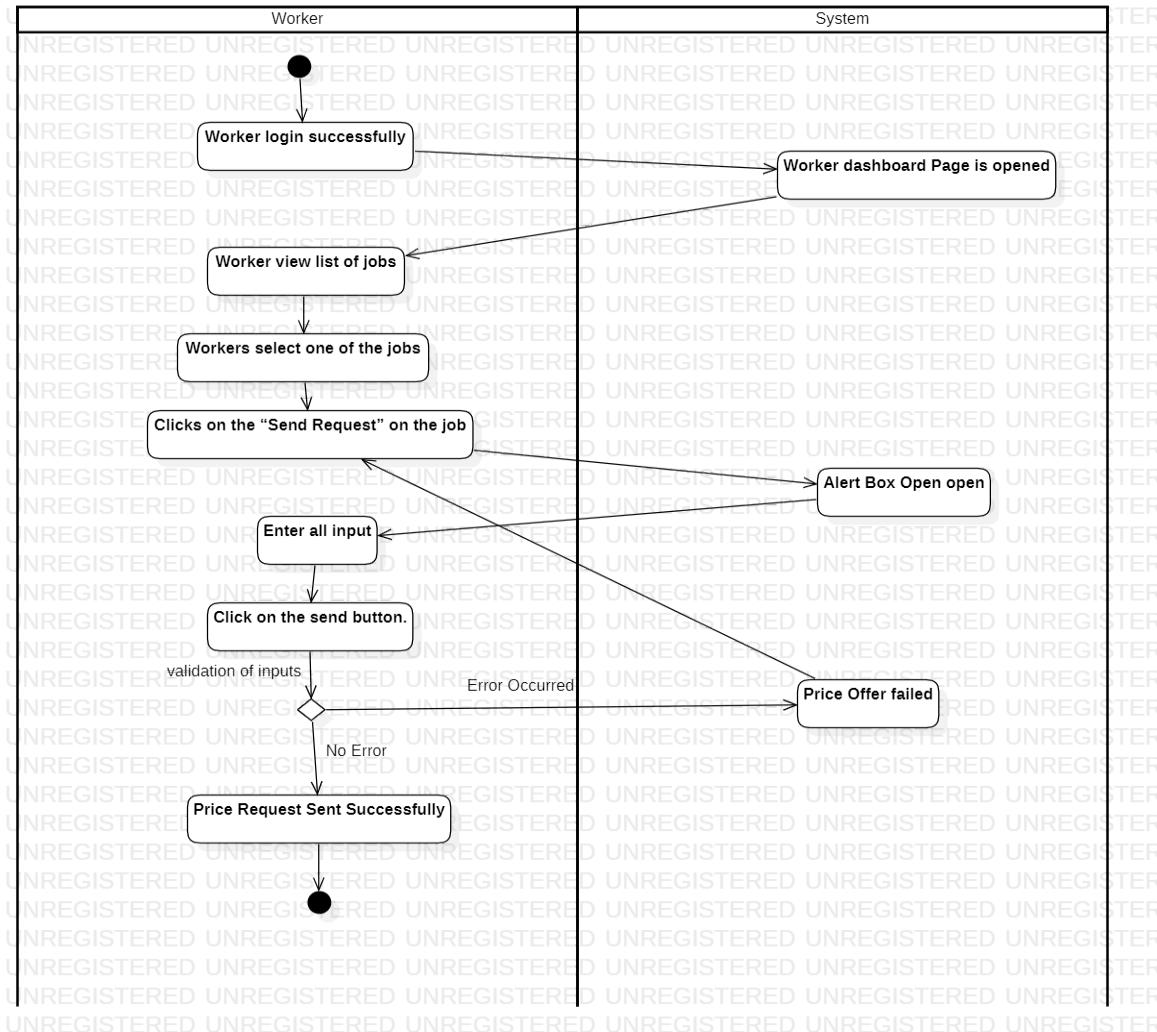


Figure 4.13 Price Offer

4.3.2.3.3 Worker cancels a job

Use Case ID	Worker_003
Use Case Name	Worker Cancels a Job
Actors	Worker

Description	A worker can cancel a job and the job will open again for other workers.
Precondition	<ul style="list-style-type: none"> • The worker is logged into the system. • The worker has accepted a job request from a client.
Postcondition	<ul style="list-style-type: none"> • The worker has successfully canceled the job. • The job post is reopened for other workers to send price requests.
Normal Flow	<ol style="list-style-type: none"> 1. Worker login successfully. 2. Dashboard Page is opened. 3. Worker clicks on the "Contract" option from the Header bar. 4. The worker navigates to the "Pending" tab to view a list of jobs they have accepted. 5. The worker selects the specific job they want to cancel. 6. The worker clicks on the "Cancel Job" button. 7. A confirmation modal appears, prompting the worker to confirm their choice. 8. The system immediately cancels the job and displays a confirmation message "Job Canceled Successfully" indicating successful cancellation. 9. The job post becomes available again for other workers to view and bid on.
Alternative Flow	<p>A1: Cancel Confirmation: On step 7, instead of clicking "Yes, Cancel," the worker can click the "Cancel" button within the confirmation modal, which closes the modal and retains the job's open status.</p>

Table 4.11 Worker cancels a job

4.3.2.3.3 Activity Diagram for Cancel a Job:

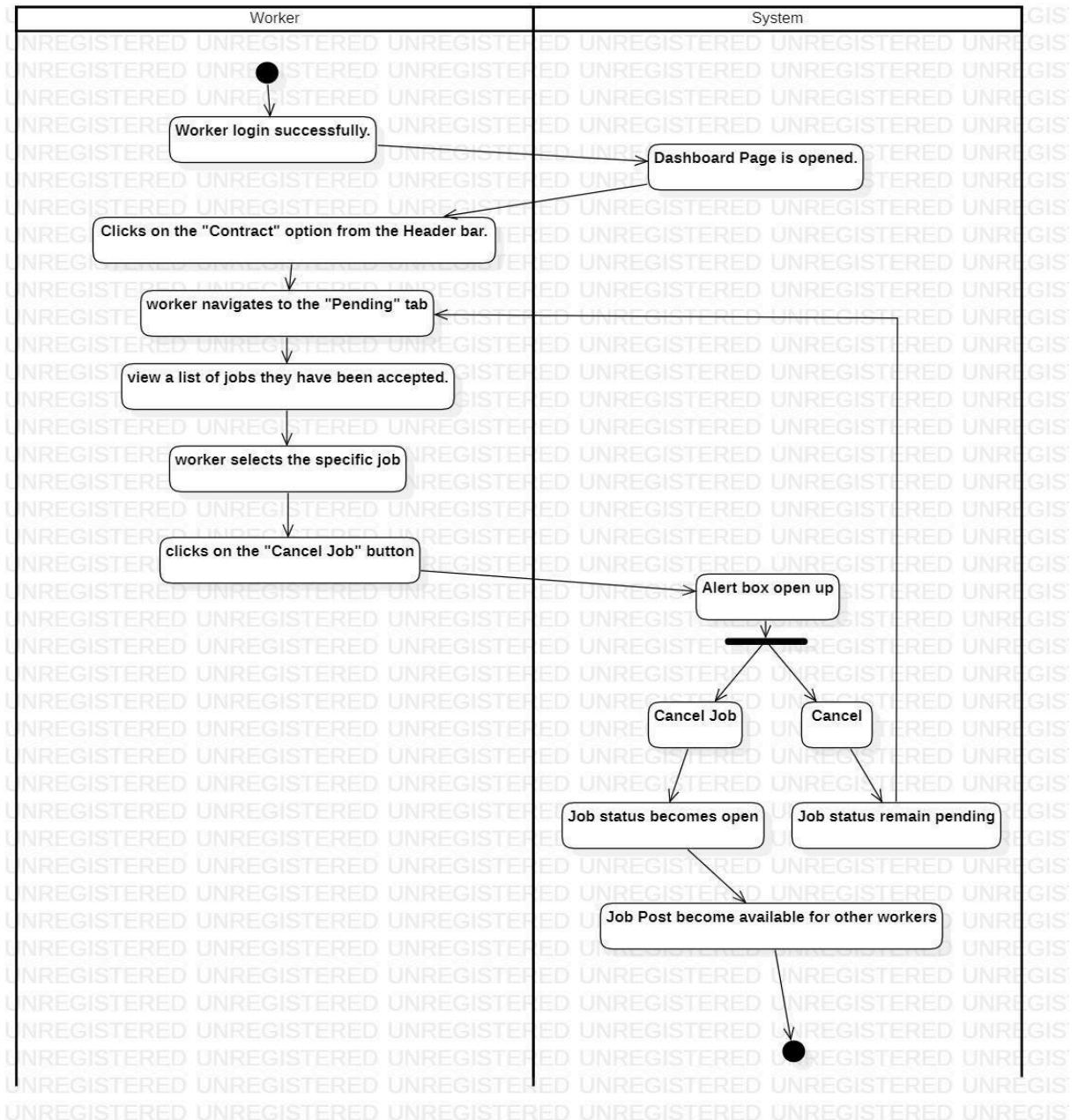


Figure 4.14 Cancel a Job

4.3.2.3.4 Worker sends job end request

Use Case ID	Worker_004
Use Case Name	Worker Sends Job End Request
Actors	Worker
Description	A worker sends a request to the client to end the job and sends a notification to the client.
Precondition	<ul style="list-style-type: none"> • The worker is logged into the system. • The worker has accepted a job request from a client. • The worker has completed their work on the job.
Postcondition	<ul style="list-style-type: none"> • The worker has successfully requested the end of the job. • The client receives a notification about the worker's request.
Normal Flow	<ol style="list-style-type: none"> 1. Worker login successfully. 2. Dashboard Page is opened. 3. Worker clicks on the "Contract" option from the Header bar. 4. The worker selects the "Active" tab to view currently active jobs. 5. The worker selects the specific job they want to end. 6. The worker clicks on the "End Job" button. 7. The worker confirms the request by clicking "Yes, Send Request" 8. The system sends a notification to the client informing them that the worker has requested to end the job.

Alternative Flow	<p>A1: Cancel Confirmation: In step 6, instead of clicking "Yes, Send Request" the worker can click the "Cancel" button within the confirmation modal.</p> <p>This closes the modal and retains the job's active status.</p>
-------------------------	---

Table 4.12 Worker send job end request

4.3.2.3.4 Activity Diagram for Sends Job End Request:

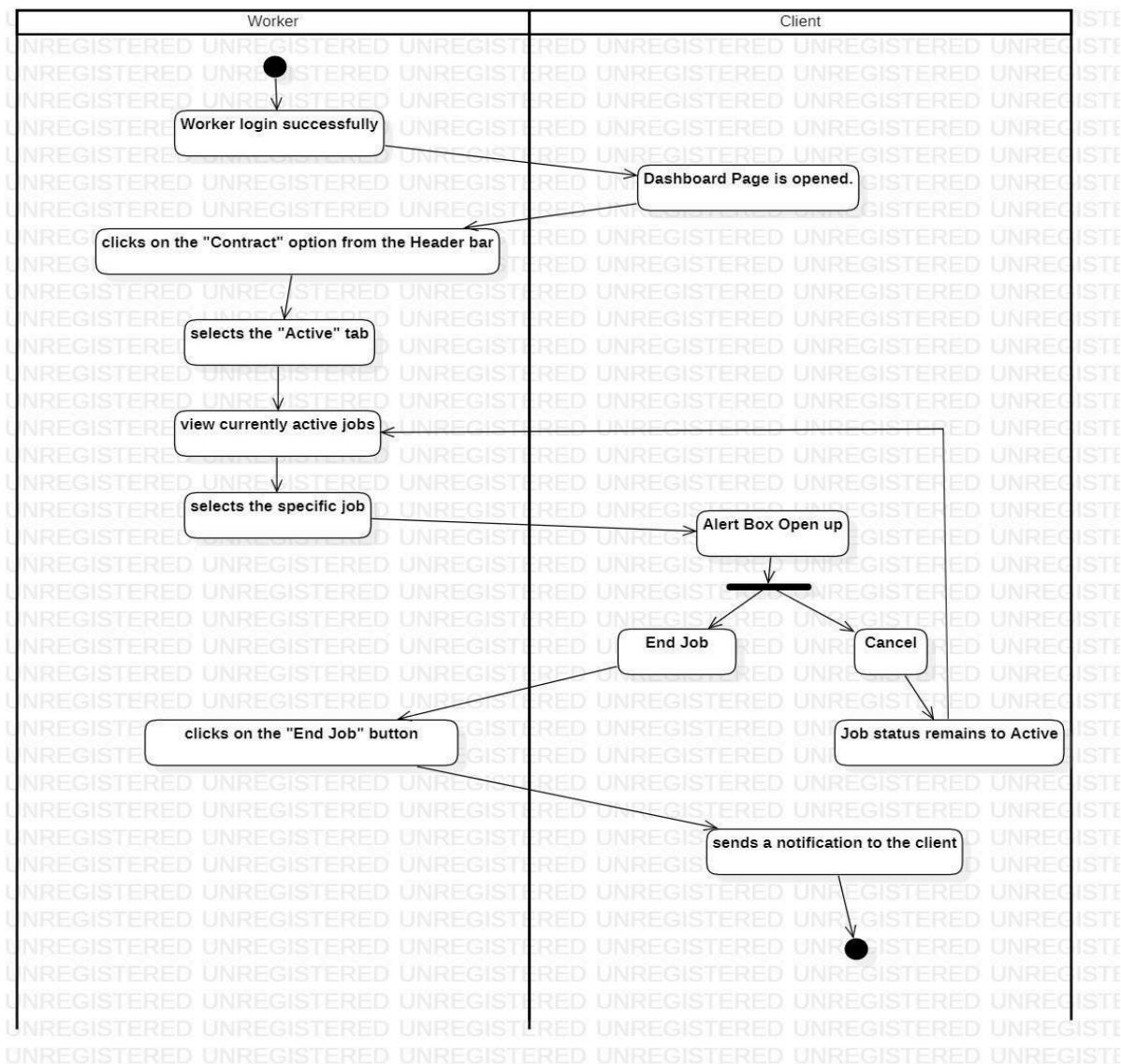


Figure 4.15 Send Job End Request

4.3.2.3.5 Worker can manage profile

Use Case ID	Worker_005
Use Case Name	Workers can manage Profile.
Actors	Worker
Description	Workers can add, delete and update their Profile information.
Precondition	Worker is logged into the system.
Postcondition	Worker added the Profile information. Worker updated the Profile information. Worker deleted the Profile information.
Normal Flow	<ol style="list-style-type: none"> 1. Worker login successfully. 2. Dashboard Page is opened. 3. Worker clicks on the profile setting option on the header bar. 4. Profile page is opened up with details of name, phone number, profile picture, home address and email address. 5. Worker clicks on the “Edit” button and enters the updated values. 6. Message display “successfully changed the value”
Alternative Flow	<ul style="list-style-type: none"> • A1: Worker clicks on the Button “Edit the Profile” on the dashboard and the system will take the Worker to the Profile page.

Table 4.13 Worker can manage profile

4.3.3 Activity Diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.3.3.1 Client Activity Diagram

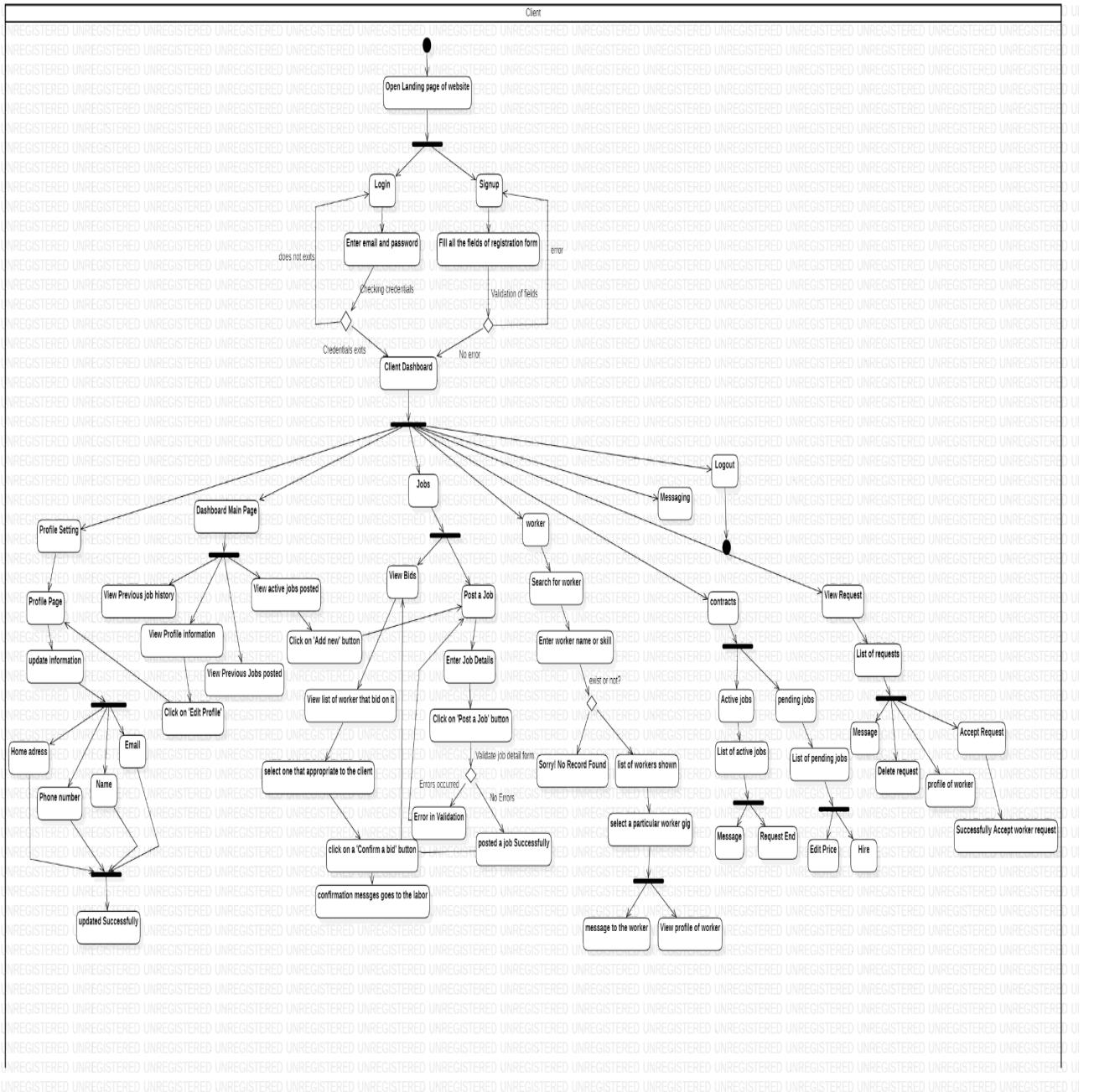


Figure 4.16 Client's Activity Diagram

4.3.3.2 Worker Activity Diagram

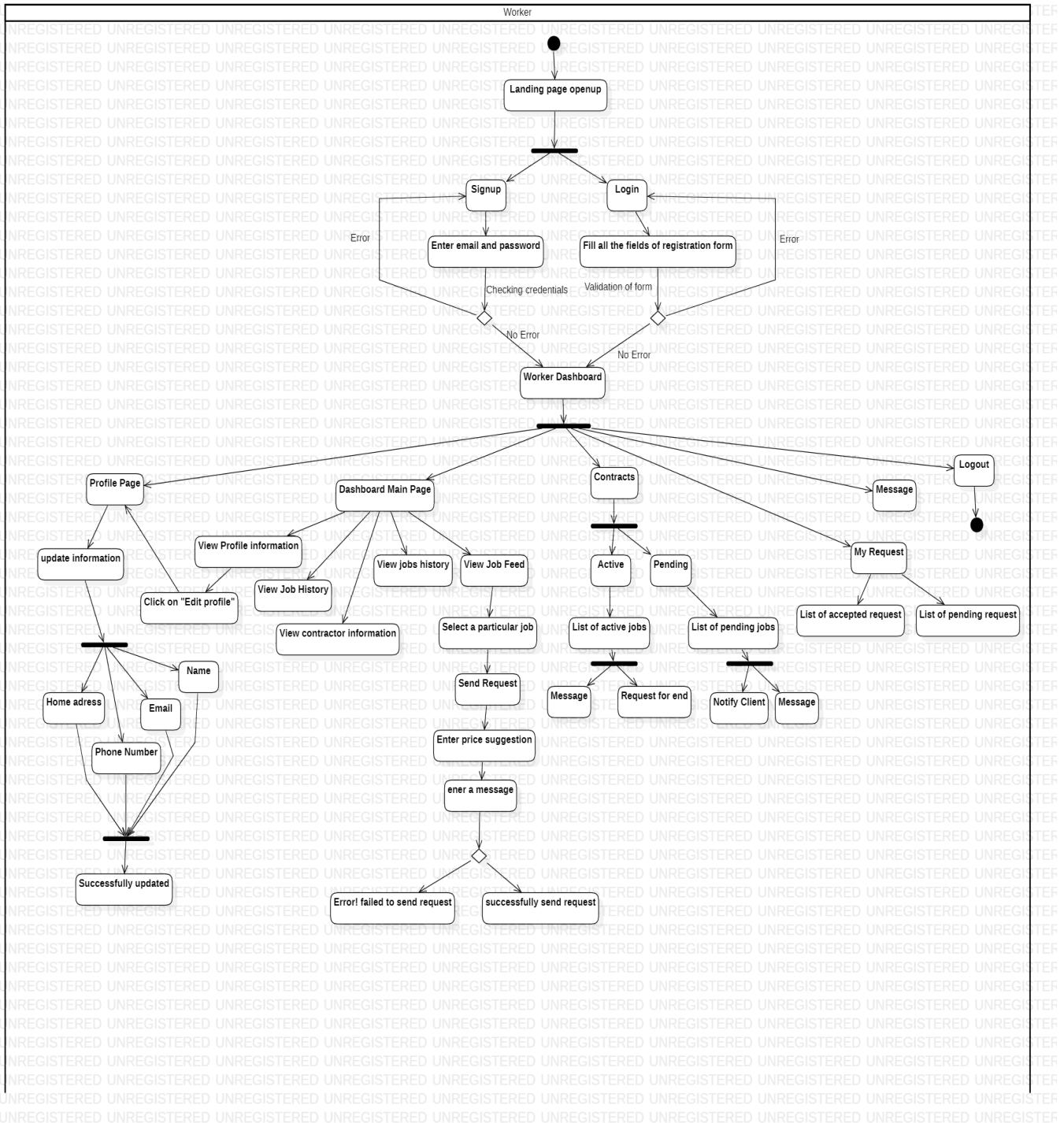
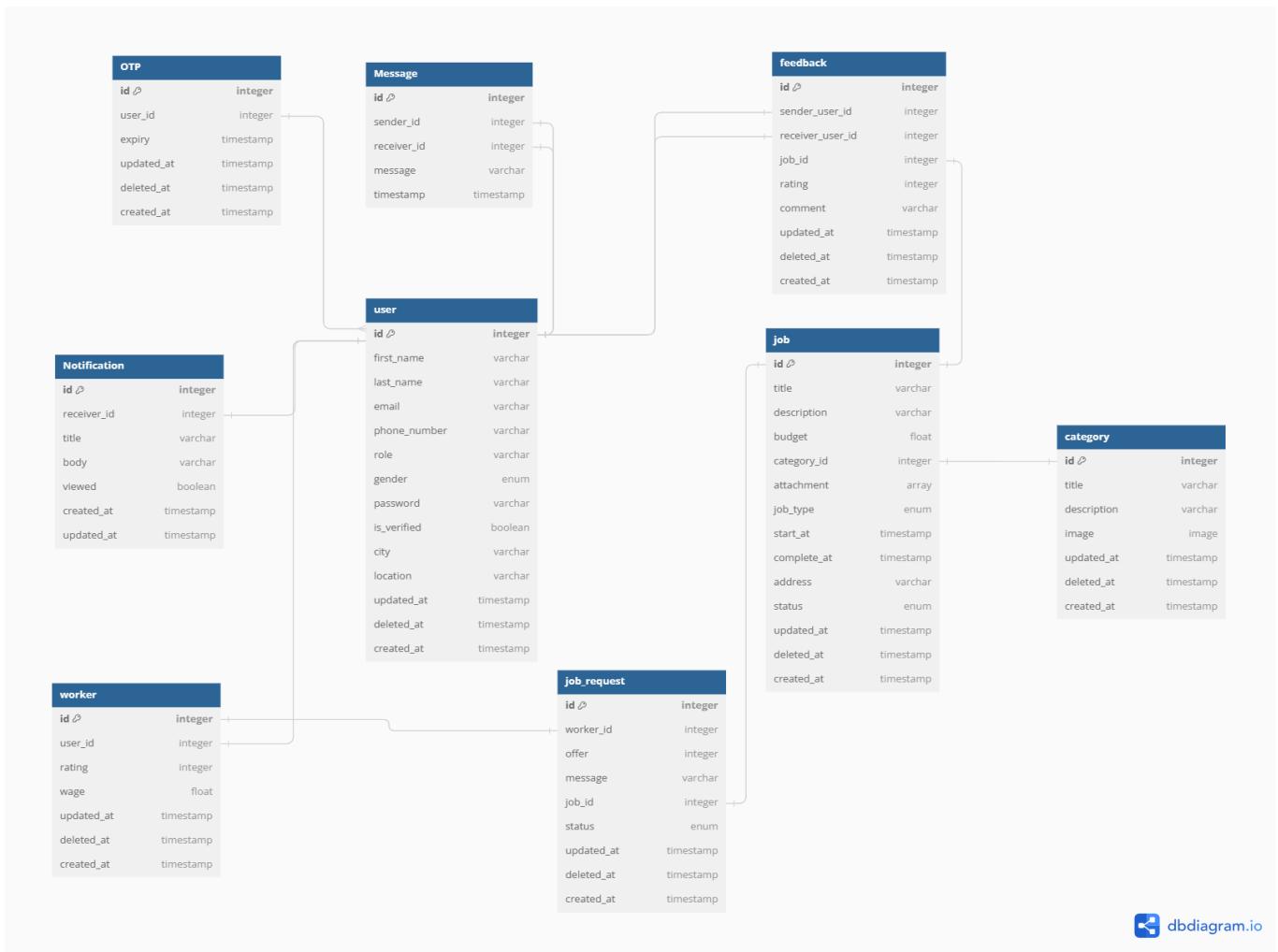


Figure 4.17 Worker's Activity Diagram

4.3.3 Entity Relationship Diagram



Link: [Labor World DB Schema - dbdiagram.io](https://dbdiagram.io)

Figure 4.18 Entity Relationship Diagram

Chapter 5:

Implementation

Chapter 5:

Implementation

5.1 Endeavour

Team Members	Responsibilities
Zain Ul Abdin	Documentation <ul style="list-style-type: none">• Comparison of existing systemsFunctional and Non-Functional requirements• ERD diagram Implementation <ul style="list-style-type: none">• Front end
Abdul Hadeed	Documentation <ul style="list-style-type: none">• Comparison of existing systems Implementation <ul style="list-style-type: none">• Backend
Azeem Khan	Documentation <ul style="list-style-type: none">• Problem scenarios• Use case diagrams and fully addressed format• Architecture diagram• Activity diagram• Test cases Implementation

	<ul style="list-style-type: none"> ● Backend ● Front end
--	--

Table 5.1 Endeavor

5.2 Components, Libraries, Web Services and stubs

Front-end:

- **Axios:** To fetch API
- **Vue.js:** Used as front-end framework
- **Pusher API:** For real time chatting.
- **Mapbox API:** For showing Maps

Database:

PostGreSQL: For storing the data

Back-end:

- **Python:** Language used for the backend
- **Django:** Backend framework of Python
- **Rest Framework:** Django Framework to create restful APIs

5.3 IDE, Tools and Technologies

- ❖ Pycharm [IDE]
- ❖ visual studio [Editor]
- ❖ Python [Backend]
- ❖ Django + Rest Framework [Python Framework]
- ❖ Vue.js [Frontend Framework]
- ❖ Pusher API [For Real Time Messaging]

5.4 Best Practices / Coding Standards

Coding Standards are an important factor for achieving a high code quality. A common visual style, naming conventions and other technical settings allow us to produce a code, which is easy to read and maintain.

5.4.1 Development Practices & Standards:

Python Best Practices:

PEP 8 Style Guide: Follow the guidelines outlined in Python Enhancement Proposal 8 (PEP 8) for code style and formatting. Consistent and readable code improves collaboration and maintainability.

Modularity and Reusability: Write modular code by breaking it into small, reusable functions or classes. This promotes code reuse, testability, and maintainability.

Error Handling: Implement proper error handling and exception handling to ensure graceful handling of errors and prevent unexpected crashes. Use try-except blocks to catch exceptions and provide meaningful error messages.

Documentation: Document your code using docstrings and provide clear comments to improve code comprehension for yourself and other developers. Use tools like Sphinx to generate documentation from docstrings.

Django Rest Framework practices:

Serializer Usage:

Use DRF serializers to convert complex data types, such as models, into JSON or other formats. Serializers handle serialization, deserialization, and validation of data, making it easier to work with complex data structures.

Viewsets and Routers:

Utilize DRF's viewsets and routers to simplify the creation of CRUD (Create, Retrieve, Update, Delete) APIs. Viewsets provide a concise way to define API endpoints, while routers handle URL routing for you.

Pagination:

Implement pagination for large datasets to improve performance and user experience. DRF provides built-in pagination classes that you can easily integrate into your views.

Testing:

Write comprehensive unit tests and integration tests using DRF's testing utilities. Test your API endpoints to ensure they function as expected and handle various scenarios correctly.

API best practices:

RESTful Design:

Adhere to the principles of Representational State Transfer (REST) for designing your API. Use clear and consistent URLs, HTTP verbs, and response codes to make your API intuitive and easy to use.

Versioning:

Consider implementing API versioning to manage changes and backward compatibility. This allows you to introduce new features without breaking existing integrations.

Authentication and Authorization:

Implement secure authentication and authorization mechanisms, such as token-based authentication or OAuth, to protect your API endpoints and restrict access to authorized users.

Input Validation and Sanitization:

Validate and sanitize user inputs to prevent security vulnerabilities such as SQL injection or cross-site scripting (XSS) attacks. Use libraries like Django's form validation or DRF's serializers to handle input validation.

Vue.js Best Practices:

Component-based Development: Follow Vue.js's core principle of component-based development. Break your user interface into reusable components to improve code organization, reusability, and maintainability.

Vue Single File Components (SFC): Use Vue Single File Components to keep your template, script, and styles in a single file. This promotes better separation of concerns and improves code readability.

Proper State Management: Use Vuex or Pinia, the official state management library for Vue.js, to handle complex application states. Vuex helps centralize and manage shared state across components, ensuring a predictable flow of data.

Vue Router for Routing: Utilize Vue Router for managing application routing. Define routes, handle navigation, and pass parameters between different views.

Vue Devtools: Install Vue Devtools browser extension for Chrome or Firefox. It provides helpful debugging capabilities, allowing you to inspect Vue components, state, and events.

Code Organization: Follow a consistent folder structure to organize your Vue.js project. Group related components, views, assets, and utilities in separate directories to improve code discoverability and maintainability.

5.5 Development Environment:

Installation:

- Python [version 3.11]
- Django [version 4.0.0]
- Node.js [version 17.0.0]
- npm [version 8.1]
- Vue.Js [version 3.x]
- PyCharm
- Visual Studio Code

5.6 Summary:

In this chapter we describe the basic libraries, components, web services that we used in our Software for the Easy Development and Installation requirements. We follow all the Standard rules of the coding as much as possible to make sure that our System will be maintained in the future for more development.

Chapter 6:

Testing and Evaluation

Chapter 6:

Testing and Evaluation

6.1 Introduction

The robustness and reliability of any system are contingent on rigorous testing and evaluation. In this chapter, we delve into the pivotal phase of ensuring the functionality and security of our application through a series of well-crafted test cases. Our approach involves Equivalence Partition Testing, focusing on both black-box and white-box testing methodologies.

6.2 Test Cases

6.2.1 Registration

Test data :

Features	Valid classes	Invalid classes
First name	[a-z, A-Z],[length>=3]	[0,..., 9],[!@#\$%^&*0], [0>length<3]
Last name	[a-z, A-Z],[length>=3]	[0,..., 9],[!@#\$%^&*0], [0>length<3]
Age	[+∞],[18=<length<100]	[a-z, A-Z], [-∞] , [18>length>100], [!@#\$%^&*0]
Phone Number	[0,..., 9] ,(030x-xxx-xxxx), [length=11]	[a-z, A-Z] , [!@#\$%^&*0]

Role	[client , worker , superadmin]	[0,...., 9] , [!@#\$%^&*0]
Email	[a-z, A-Z], [0,...., 9] [@gmail.com]	[!#\$%^&*0]
password	[a-z, A-Z], [0,...., 9] , [!#\$%^&*0] [length>=6]	[length<6]
confirm password	[a-z, A-Z], [0,...., 9] , [!#\$%^&*0] [length>=6] , [confirm password == password]	[confirm password != password] , [length<6]
profile picture	['jpeg', 'jpg', 'png', 'gif']	[pdf , docx]

Table 6.1 Registration Test Data

Test Case for Registration:

Test Case ID	Inputs	ECP	Actual Output
1	First name = azeem , last name = khan, age = 23, phone number = 03040002341 , role = client, password = 12345678 , confirm password = 12345678 ,email = azeem@gmail.com , profile picture	First name = [a-z, A-Z],[length>=3] , last name = [a-z, A-Z],[length>=3] , age = [+∞],[18<length<100] , phone number = [0,...., 9]	Success! You have been registered successfully

	= catImage.png	,(030x-xxx-xxxx) , [length=11] , role = [client , worker , superadmin], password = [a-z, A-Z], [0,...., 9] , [!#\$%^&*()] [length>=6] , confirm password = [confirm password == password], email = [a-z, A-Z], [0,...., 9] [@gmail.com] , profile picture = ['jpeg', 'jpg', 'png', 'gif']	
2	First name = az , last name = k, age = 17, phone number = 03040002341 , role = client, password = 1234567 , confirm password = 1234567 ,email = azeem@gmail.com , profile picture = catImage.png	First name = [a-z, A-Z],[length>=3] , last name = [a-z, A-Z],[length>=3] , age = [+∞],[18<length<100] , phone number = [0,...., 9] ,(030x-xxx-xxxx) , [length=11] , role = [client , worker , superadmin], password = [a-z, A-Z], [0,...., 9] , [!#\$%^&*()] [length>=6] , confirm password = [confirm password == password], email = [a-z, A-Z], [0,...., 9] [@gmail.com] , profile picture = ['jpeg', 'jpg', 'png', 'gif']	First name must be at least 3 characters long Last name must be at least 3 characters long Age must be between 18 and 100

3	First name = azeem , last name = khan, age = 23, phone number = 03040002341 , role =superman, password = 123 , confirm password = 123,email = azeem@gamil.com , profile picture = catDocumentary.pdf	First name = [a-z, A-Z],[length>=3] , last name = [a-z, A-Z],[length>=3] , age = [+∞],[18<length<100] , phone number = [0,...., 9] ,(030x-xxx-xxxx) , [length=11] , role = [client , worker , superadmin], password = [a-z, A-Z], [0,...., 9] , [!#\$%^&*() length>=6] , confirm password = [confirm password == password], email = [a-z, A-Z], [0,...., 9] [@gmail.com] , profile picture = ['jpeg', 'jpg', 'png', 'gif']	superman is not a valid choice. Unsupported image format. Please use JPEG, PNG, or GIF. Password must be at least 6 characters long.
4	First name = azeem , last name = khan, age = 23, phone number = 03040002341 , role = client, password = 12345678 , confirm password = 123456 ,email = azeemAtgamil.com , profile picture = catImage.png	First name = [a-z, A-Z],[length>=3] , last name = [a-z, A-Z],[length>=3] , age = [+∞],[18<length<100] , phone number = [0,...., 9] ,(030x-xxx-xxxx) , [length=11] , role = [client , worker , superadmin],	Enter a valid email address. Passwords do not match.

	<p>password = [a-z, A-Z], [0,...., 9] , [!#\$%^&*()] [length>=6] , confirm password = [confirm password == password], email = [a-z, A-Z], [0,...., 9] [@gmail.com] , profile picture = ['jpeg', 'jpg', 'png', 'gif']</p>	
--	--	--

Table 6.2 Registration Test Case

6.2.2 Post a job

Test data :

Features	Valid classes	Invalid classes
Job Title	[a-z, A-Z] , [length>=3]	[0,...., 9],[!#\$%^&*()], [0>length<3]
Job Type	[on-site , remote]	[0,...., 9],[!#\$%^&*()],
Job Category	[plumber , electrician , carpenter, cleaner , painter],	[a-z, A-Z] , [!#\$%^&*()], [0,...., 9],
Estimated Price	[0,...., 9]	[!#\$%^&*()] , [a-z, A-Z]
Job Description	[a-z, A-Z], [0,...., 9] , [length>=3]	[!#\$%^&*()] , [0>length<3]
Start Date:	[dd - mm - yyyy],	[mm - dd- yyyy], [yyyy - dd- mm],

Audio	['mp3', 'aiff', 'mav', 'aac']	['jpeg', 'jpg', 'png', 'gif'], ['mp4', 'mov', 'avi', 'wmv']
Video	['mp4', 'mov', 'avi', 'wmv']	[pdf , docx] ,['jpeg', 'jpg', 'png', 'gif']
Image	['jpeg', 'jpg', 'png', 'gif']	['mp4', 'mov', 'avi', 'wmv'] , [pdf , docx]

Table 6.3 Post a Job Test Data

Test Cases:

Test Case ID	Inputs	ECP	Actual Output
1	Job Title = 'Need a plumber` , Job Type = on-site , Job Category = plumber , Estimated Price = 5000 , Job Description = “ we need to a plumber that fix out taps in our house”, Start Date = 28/11/2023 , Audio = myVoice.mp3, Video = myVideo.mp4 , Image = myPic.png	Job Title = [a-z, A-Z] , [length>3] , Job Type = [on-site , remote] , Job Category = [plumber , electrician , carpenter, cleaner , painter] , Estimated Price = [0,...., 9] , Job Description = [a-z, A-Z], [0,...., 9] , Start Date = [dd - mm - yyyy], , Audio = ['mp3', 'aiff', 'mav', 'aac'] , Video = ['mp4', 'mov', 'avi', 'wmv'] , Image = ['jpeg', 'jpg', 'png', 'gif']	Success! Job posted successfully!
2	Job Title = 'Ne` , Job Type = on-site , Job Category = plumber , Estimated Price = 5000 , Job Description = “we”, Start Date = 28/11/2023 , Audio =	Job Title = [a-z, A-Z] , [length>3] , Job Type = [on-site , remote] , Job Category = [plumber ,	Job Title must be at least 3 characters long Job Description must be

	myVideo.mp4, Video = myVideo.mp4 , Image = myPic.png	electrician , carpenter, cleaner , painter] , Estimated Price = [0,...., 9] , Job Description = [a-z, A-Z], [0,...., 9] , Start Date = [dd - mm - yyyy], , Audio = ['mp3', 'aiff', 'mav', 'aac'] , Video = ['mp4', 'mov', 'avi', 'wmv'] , Image = ['jpeg', 'jpg', 'png', 'gif']	at least 3 characters long Unsupported audio format. Please use MP3 , AIFF, MAV , or GIF.
3	Job Title = `Need a plumber` , Job Type = on-site , Job Category = plumber , Estimated Price = 5000 , Job Description = “ we need to a plumber that fix out taps in our house”, Start Date = 28/11/2023 , Audio = myVoice.mp3, Video = myVoice.mp3 , Image = myVoice.mp3	Job Title = [a-z, A-Z] , [length>3] , Job Type = [on-site , remote] , Job Category = [plumber , electrician , carpenter, cleaner , painter] , Estimated Price = [0,...., 9] , Job Description = [a-z, A-Z], [0,...., 9] , Start Date = [dd - mm - yyyy], , Audio = ['mp3', 'aiff', 'mav', 'aac'] , Video = ['mp4', 'mov', 'avi', 'wmv'] , Image = ['jpeg', 'jpg', 'png', 'gif']	Unsupported video format. Please use MP4, MOV, AVI, or WMV. Unsupported image format. Please use JPEG, PNG, or GIF.
4	Job Title = `Need a plumber` , Job Type = on-site , Job Category = plumber , Estimated Price = - 5000 , Job Description = “ we need to a plumber that fix out taps in our house”, Start Date = 28/11/2023 , Audio = myVoice.mp3, Video = myVideo.mp4 , Image = myPic.png	Job Title = [a-z, A-Z] , [length>3] , Job Type = [on-site , remote] , Job Category = [plumber , electrician , carpenter, cleaner , painter] , Estimated Price = [0,...., 9] , Job Description = [a-z, A-Z], [0,...., 9] , Start Date = [dd - mm - yyyy], , Audio = ['mp3', 'aiff', 'mav',	Success! Job posted successfully!

	'aac'] , Video = ['mp4', 'mov', 'avi', 'wmv'] , Image = ['jpeg', 'jpg', 'png', 'gif']	
--	---	--

Table 6.4 Post a Job Test Case

6.2.3 Reset a Password

Test data :

Features	Valid classes	Invalid classes
Current Password	[a-z, A-Z], [0,...., 9] , [!#\$%^&*0], [length>=6]	[length<6]
New Password	[a-z, A-Z], [0,...., 9] , [!#\$%^&*0] [length>=6] ,	[length<6]
Confirm new Password	[a-z, A-Z], [0,...., 9] , [!#\$%^&*0] [length>=6] , [confirm password == password]	[confirm password != password] , [length<6]

Table 6.5 Rest a Password Test Data

Test case for Reset Password:

Test Case ID	Inputs	ECP	Actual Output
1	Current Password = password330 , New Password = password440, Confirm new Password = password440	Current Password = [length<6] , New Password =[length<6] , Confirm new Password = [confirm password !== password] , [length<6]	Success! Password Reset successfully!
2	Current Password = password330 , New Password = password440, Confirm new Password = password4400	Current Password = [length<6] , New Password =[length<6] , Confirm new Password = [confirm password !== password] , [length<6]	New password and new Password do not match.
3	Current Password = password330 , New Password = pass, Confirm new Password = pass	Current Password = [length<6] , New Password =[length<6] , Confirm new Password = [confirm password !== password] , [length<6]	New Password must be at least 6 characters long.

Table 6.6 Rest a Password Test Case

6.2.3 Price Offer by Worker

Test data :

Features	Valid classes	Invalid classes
Price	[0,...., 9]	[a-z, A-Z], [!#\$%^&*()]
Message	[a-z, A-Z], [0,...., 9] ,	[!#\$%^&*()]

Table 6.7 Price Offer by Worker Test Data

Test case for Price Offer:

Test Case ID	Inputs	ECP	Actual Output
1	Price =3000 , message = “need a raise in money”	Price =[0,...., 9] , message = [a-z, A-Z], [0,...., 9] ,	Sent! Your Request Has Been Sent.
2	Price = 40000 , message = “”	Price =[0,...., 9] , message = [a-z, A-Z], [0,...., 9] ,	Sent! Your Request Has Been Sent.
3	Price = -3500 , message = “need a raise in money”	Price =[0,...., 9] , message = [a-z, A-Z], [0,...., 9] ,	Sent! Your Request Has Been Sent.
4	Price = , message = “”	Price =[0,...., 9] , message = [a-z, A-Z], [0,...., 9] ,	Error! Please Enter Price.

Table 6.8 Price Offer by Worker Test Case

Chapter 7:

CONCLUSIONS & OUTLOOK

Chapter 7:

Conclusions & Outlooks

7.1 Introduction:

This chapter delves into the comprehensive overview of the project "LaborWorld." The application has been systematically developed and rigorously tested in alignment with the defined requirements. The entire project life cycle is meticulously documented in this report. Additionally, this chapter discusses the notable achievements, areas of improvement, and challenges encountered during the project's progression.

7.2 Achievements and Improvements:

During the whole project we have achieved different kinds of skills that include teamwork and communication, software development skills, and how to plan and execute the development process of software applications. During the project we realized the importance of teamwork and communication and how it aids in development of something remarkable. We have learned from this project how much it is important to take feedback on the software application to analyze the usability of our software application. The Labor World is an on-going project that will keep going through improvements and development phases to achieve the desired state in the public.

7.3 Critical Review:

The labor world project is designed to solve the problem of various kind of labors working around us. The project's target audience is very vast and diverse in nature. To tackle the needs of multiple laborers and workers, the system should have to cover multiple scenarios to meet the requirements of the workers and clients to ensure the smooth working of the system. This aspect of Labor world application is the biggest challenge that is faced by our industries as well. The scope of this project is very diverse and it takes a lot of research and then plans to

develop the system. Thus, it increases the budget and time for the industry to meet the requirements expected from the system.

7.4 Future Recommendations:

The labor world is an on-going project that has to cover a lot of things in the future to achieve solid status in the market. Our future recommendations for the system are as follow:

- **Third party payment gateway:**

The third party payment system will be used in the system to deduct the payment from the clients account after he posts the job. This will ensure that the payment made through the system and both parties remain on the system instead of getting direct with each other.

- **Remote jobs:**

After successful integration of the payment system the system's goal is to provide remote job facilities to freelancers.

- **Contractor Module:**

To provide the support for the contractors in the system to deal with the bigger jobs.

- **Call Live streaming Record :**

Will Maintain Records talks of client and employer for handling any miscellaneous security issues and consistent on agreements with each other.

7.5 Summary:

"LaborWorld" is a web-based application designed for both workers and clients, addressing the needs of the Pakistani job market. It caters to workers who cannot afford physical shops and offers clients the convenience of hiring skilled professionals online. Future plans include extending the project's scope to provide greater benefits to users. The success of this project owes much to the unwavering dedication and support of our supervisor, Mr. Muhammad Saud Khan, to whom we express our deepest gratitude.

Reference and Appendices

- [1] <https://pusher.com/>
- [2] <https://www.python.org/>
- [3] <https://www.djangoproject.com/>
- [4] <https://vuejs.org/>
- [5] <https://pinia.vuejs.org/>
- [6] <https://www.mapbox.com/>
- [7] <https://web.dev/explore/progressive-web-apps>
- [8] <https://supertasker.pk/>
- [9] <https://mahircompany.com/>
- [10] <https://gigsmart.com/>
- [11] <https://www.formazdoori.com/>

12%	7%	2%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

- | | | |
|----------|---|---------------|
| 1 | Submitted to Higher Education Commission
Pakistan | 3% |
| 2 | www.coursehero.com | 1% |
| 3 | Submitted to University of Hertfordshire | 1% |
| 4 | docobook.com | 1% |
| 5 | Tatar, S... "Two learning approaches for
protein name extraction", Journal of
Biomedical Informatics, 200912 | 1% |
| 6 | Submitted to Plainfield South High School | 1% |
| 7 | Submitted to University of Teesside | <1% |
| 8 | Submitted to National School of Business
Management NSBM, Sri Lanka | <1% |
-

9	nationalzoologicalgardens.wordpress.com Internet Source	<1 %
10	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %
11	www.bartleby.com Internet Source	<1 %
12	Submitted to City University of Hong Kong Student Paper	<1 %
13	www.deed-records.com Internet Source	<1 %
14	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	<1 %
15	Submitted to University of South Africa Student Paper	<1 %
16	utpedia.utp.edu.my Internet Source	<1 %
17	Submitted to Harrisburg University of Science and Technology Student Paper	<1 %
18	Submitted to Cardiff University Student Paper	<1 %
19	repository.tudelft.nl Internet Source	<1 %

20	speakerdeck.com Internet Source	<1 %
21	Submitted to St. Paul University - Tuguegarao Student Paper	<1 %
22	irigs.iiu.edu.pk:64447 Internet Source	<1 %
23	repozitorij.unizg.hr Internet Source	<1 %
24	Submitted to Loughborough University Student Paper	<1 %
25	Submitted to University of Mauritius Student Paper	<1 %
26	studentsrepo.um.edu.my Internet Source	<1 %
27	Submitted to Singapore Institute of Technology Student Paper	<1 %
28	Submitted to Korea National University of Transportation Student Paper	<1 %
29	core.ac.uk Internet Source	<1 %
30	Submitted to Napier University Student Paper	<1 %

31	Submitted to Swinburne University of Technology Student Paper	<1 %
32	Submitted to University of Northampton Student Paper	<1 %
33	Submitted to City University Student Paper	<1 %
34	communityqlik.com Internet Source	<1 %
35	www.juicingforweightloss.com Internet Source	<1 %
36	"Law of International Trade in the Region of the Caucasus, Central Asia and Russia", Brill, 2022 Publication	<1 %
37	Submitted to Al Khawarizmi International College Student Paper	<1 %
38	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
39	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
40	Submitted to University of Birmingham Student Paper	<1 %

41

dl.lib.uom.lk

Internet Source

<1 %

42

Ton Duc Thang University

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off