7SENG013C.Y Software Development Project

2023/24

Gig Work Marketplace Software Project

Student: Ms. Poddiwela Keerthirathna (20274450)

Supervisor: Ms. Dileeka Alwis

Date: 02/04/2025

MSc Information Technology
School of Computer Science and Engineering
College of Design, Creative and Digital Industries
University of Westminster

Abstract

This project involves the development of a digital platform, that connects part-time job seekers with employers who need to get work done. This web application facilitates job posting, job search and applying, tracking application progress, communication between employers and workers, and rating the worker's performance. A wide range of job categories, including delivery, handyman work, babysitting, online freelancing, and more can be serviced by this application.

Inefficiencies in traditional methods of finding part-time work or hiring part-time workers are minimized by not relying on payments from users for the maintenance of it. Other available platforms for job seeking caters to a wide range of job opportunities, from full-time, contractual jobs to flexible part-time jobs. This solution enhances gig economy participation by narrowing its focus on part-time workers and employers, unlike platforms that caters to both full-time and part-time jobs, enabling employers to access a manageable number of suitable candidates and workers to efficiently access job opportunities that suit their requirements.

The aim of this project is to provide a seamless experience to job seekers and employers, to outsource work and find opportunities. The frontend of this application is implemented using HTML, CSS and JavaScript in a visually appealing, intuitive manner. The backend development is done by using PHP scripting language that provides fast processing and efficient interactions with the database. The database is developed using MYSQL, providing scalable data storage and manipulation.

Keywords

Gig Economy, Job Seeking, Hiring, Web Application, JavaScript, MYSQL

Declaration

This report is submitted in partial fulfilment of the requirements for the MSc Software Engineering (Conversion) Degree at the University of Westminster.

This report has been prepared based on my own work. Where other published and unpublished source materials have been used, these have been acknowledged in the text of the report and included in the list of references.

Word Count: **9451**
Student Name: **Ms. Poddiwela Keerthirathna**
Date of Submission: 02/04/2025

Acknowledgements

My sincere gratitude goes to my supervisor, Ms. Dileeka Alwis, who provided constructive feedback and support throughout this project. Also, I would like to express my heartfelt gratitude to my family for their understanding and encouragement to make this project a success. Additionally, my friends and colleagues provided me their moral support that contributed to the successful delivery of this project.

Contents

List of Figures

List of Tables

1 Chapter 1 Introduction

1.1 Introduction

Gig work spans across a wide range of tasks, from handyman work, babysitting to online freelancing (De Stefano and Aloisi, 2018). Digital platforms have revolutionized the way employers get their work done and how workers find matching opportunities. Gig economy is rapidly evolving with technological advancement, emergence of the Gen Y and Gen Z workforce and the Covid-19 pandemic (Ray, Sengupta and Varma, 2024). Amisdt this volatality of the gig economy, both employers and part-time workers struggle to find suitable matches efficiently as the current digital platforms present an unmanageable amount of job opportunities and candidates, which makes finding the right match challenging. Technology is called for in the gig economy for efficient work allocation.

This software development project intends to bridge this gap by developing a web application that connects employers who needs to get work done with part-time workers looking for work. This digital platform will facilitate job posting, job picking by gig workers, and seamless communication between employers and workers. Inefficiencies in traditional methods of finding part-time work or hiring part-time workers will be minimized by streamlining the gig work allocation process, saving time for both employers and part-time workers. Conventional job intermediary platforms have less flexible arrangements that require payments for job postings and commissions for sucessful matches. This digital platform does not rely on payments from either party for the maintenance of it. Other available platforms for job seeking caters to a wide range of job opportunities, from full-time, contractual jobs to flexible part-time jobs. By narrowing down the target market, this platform enables employers to access a manageable amount of suitable candidates and workers to efficiently access job opportunities that suit their requirements. This solution enhances gig economy participation by narrowing its focus on gig work, unlike platforms that caters to both full-time and flexible jobs. Employers will benefit from efficiently sourcing reliable part-time talent for numerous tasks, ranging from household work to short-term professional work. Gig employees will find the right opportunity to earn additional income by rendering the posted service.

This application will serve the needs of below users.

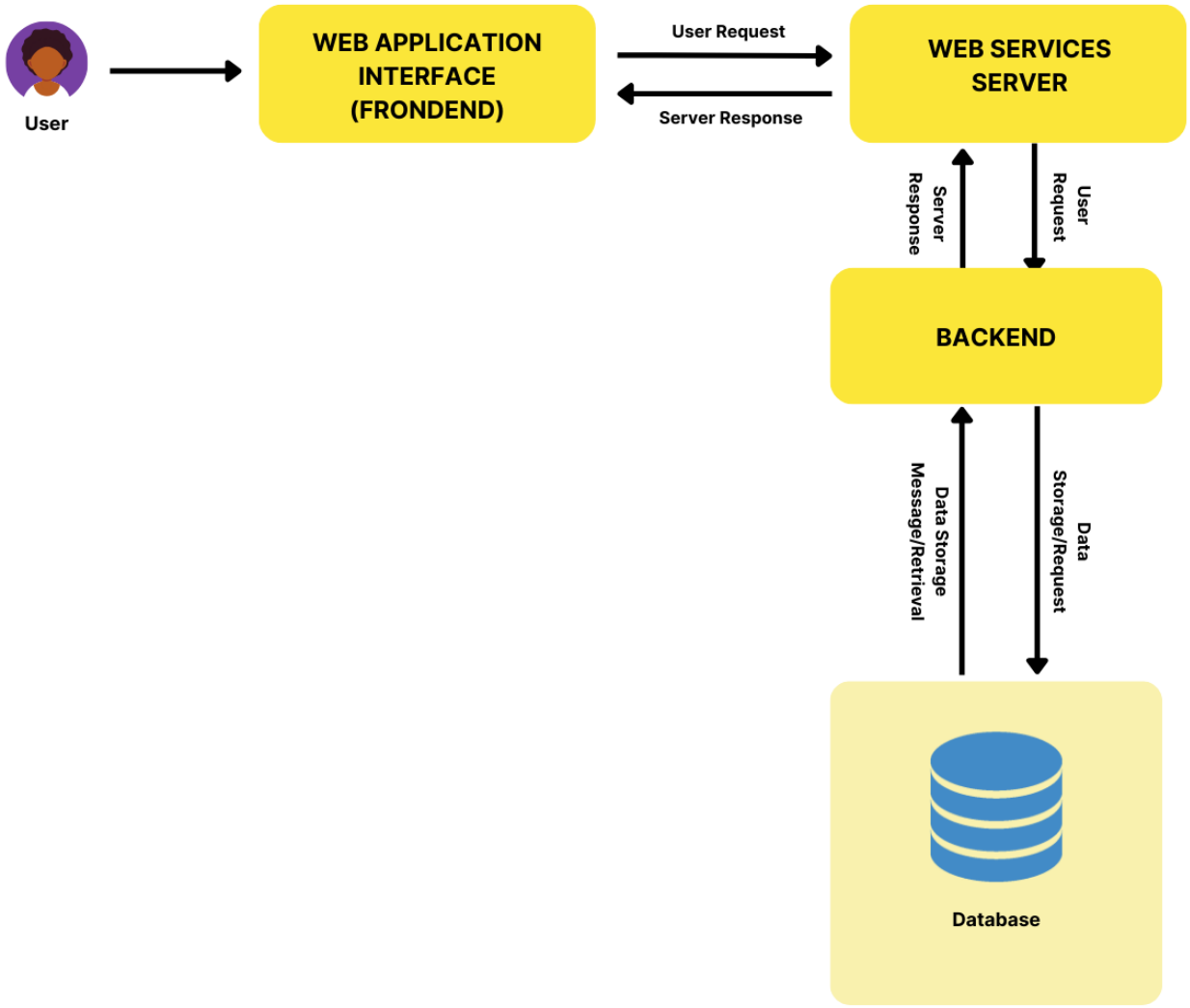
1) Employers: Businesses and individuals requiring reliable part-time workers for various tasks.
2) Gig Workers: Individuals, including students, freelancers seeking short-term job opportunities that fit their schedules.

This gig work intermediary platform will include the following core features.

- Allow employers to post part-time jobs with skills required and payment offered.
- Enable gig workers to search for gigs and apply for suitable opportunities.
- Facilitate in-app messaging for clear communication between employers and workers.
- Send notifications to users about job offers, application statuses, and messages.

A simplified system architecture diagram of this web application is shown in Figure 1.1, that provides an overview of the components of the application, their interactions and the context it operates in.

Figure 1.1 Simplified System Architecture Diagram



This report targets readers who has a basic knowledge of web application development. However, the readers are not expected to possess a specialized knowledge on the gig economy, and job matching and hiring platforms in the industry. The industry specific terms are explained in the body of each section of this report to improve clarity.

1.2 Project Aims and Objectives

The purpose of this digital platform is to streamline the flexible work allocation process by connecting employers looking for part-time workers and individuals seeking part-time employment. This project will address the current inefficiencies in the part-time job market by achieving the following aims and objectives.

1.2.1 Project Aims

Aim 1 (PA1) – Design and develop a user friendly, simple and intuitive digital platform that enables seamless task outsourcing for employers.

Aim 2 (PA2) – Design and develop a user friendly, simple and intuitive digital platform that facilitates efficient job seeking for part-time workers.

Aim 3 (PA3) - Enhance communication between employers and part-time workers to improve task outsourcing.

1.2.2 Project Objectives

The objectives for PA1 are as follows.

- (PO1.1) Implement a responsive user interface that allows employers to easily post jobs.
- (PO1.2) Develop a database to manage employer data and job postings securely and efficiently.

The objectives for PA2 are as follows.

- (PO2.1) Implement a responsive user interface that allows part-time workers to easily search and apply for jobs.
- (PO2.2) Develop a database to manage employee data and applications securely and efficiently.

The objectives for PA3 are as follows.

- (PO3.1) Implement an in-app messaging system to facilitate seamless communication between employers and part-time workers.
- (PO3.2) Develop a notification system to notify users about application statuses and messages.

1.3 Project Resources

This subsection provides an overview of hardware and software that were used to complete the development of this digital platform.

1.3.1 Software Tools

- Operating system: macOS Sonoma 14.5 is used for this project due to its stability, efficiency and scalability.
- Web development environment: XAMPP web server as the local server environment for development and testing, which includes Apache web server and MYSQL database, and can handle PHP scripts.
- Code editor: Visual Studio Code to write and edit the code efficiently.
- Frontend development: HTML is used to structure the web pages, defining their layout and content (AppMaster, 2023). CSS makes the web pages visually appealing and user-frinedly by styling the HTML elements (Abramowski, 2023). Javascript is used to add interactivity to the web pages using its features such as form validation, event handling etc (Joshi, 2023).
- Backend development: PHP scripting language to interact with database to store and retrieve data, and to handle the application logic.
- Database: MySQL as the database management system to store, query and manipulate data.

- Version control: Git for tracking changes in the source code.

1.3.2 Hardware

- Computer: MacBook Pro 13” with its Apple M2 processor, 8GB RAM and 245.11 GB storage is used due to its performance and the ability to fulfil the requirements of web application development.
- Internet connection: A high speed wifi connection was used for accessing resources, and transfer and update files.

1.4 Outline of the Report

This subsection gives a summary of each chapter in this report, and appendices.

Chapter 2 on requirements analysis provides a detailed review of similar job posting systems to gather features to be incorporated into the project. It contains the analysis of systems based on a review criteria and what aspects were selected to be included in the Gig Work Marketplace web application.

Chapter 3 is on system requirements that identifies the stakeholders, and evaluate the information flow between stakeholders and the system. From this analysis, a list of functional, non-functional & user interface requirements were identified which are categorized as essential, desirable or luxury.

Chapter 4 on system design outlines the use cases of this web application, identifies the sequence of messages passed between different entities in the system and the relationships that exists between them. It illustrates the design screens of the user interface of the system and the links between those screens.

Chapter 5 is on implementation discusses the architectural aspects of the application, covering the tools and technologies that were chosen to build the system. It details the implementation issues encountered and the solutions adopted to resolve them, and the software components used in the development.

Chapter 6 on testing describes the testing methodologies used to evaluate the behaviour of the system. It includes the test scenarios covered, defects identified and test results.

Chapter 7 provides the conclusion reflecting the achievement of project aims and objectives. It lists the requirements fulfilled by this web application and propose areas for improvement in future iterations of the project.

The bibliography lists all the sources of information used throughout this report, referenced in Harvard style.

The appendices section contain the additional design diagrams not included in chapter 4 and the software code in plain text.

2 Chapter 2 Requirements Analysis

2.1 Introduction

This chapter provides information on the detailed analytical review that was conducted of the existing systems similar to the Gig Work Marketplace application. Details of work hiring digital platforms that were selected for review and the rationale behind selecting them are described in this chapter. Several review criteria were established to evaluate the features of these system, ranging from main functionalities, user interface to the implementation methods. Each system was reviewed in sufficient detail to assess their good and bad features against the established review criteria. Then, these systems were compared against each other to identify the best features to be incorporated into the Gig Work Marketplace application.

2.2 Systems to be Reviewed

This subsection lists the systems that were reviewed to gather features that were later incorporated into the Gig Work Marketplace web platform. What each system is, and why it was selected for review is briefly described.

Existing systems that connects employers with workers, and support efficient work allocation include GigSmart, Airtasker, Indeed, TaskRabbit and Upwork.

2.2.1 GigSmart

GigSmart is a web platform that connects skilled, on-demand workers with full-time or part-time jobs. Employers in many industries can post their shifts and workers can pick suitable opportunities that matches their requirements. GigSmart collaborates with other service providers to perform background checks and motor vehicle records checks on workers (GigSmart, no date).

Rationale behind the selection: GigSmart has a visually appealing and easy to navigate interface that allows businesses to hire on-demand workers avoiding the hassle of reviewing resumes and conducting interviews. Additionally, it provides job categorizing features that assist gig workers to narrow down their job search and find suitable opportunites efficiently.

2.2.2 Airtasker

Airtasker is an Australian digital platform that connects those who need to get work done with people who are available and willing to work. Employers can post jobs and hire workers get those jobs done. Workers can search for jobs and apply for them to earn an additional income. This platform caters to a wide range of tasks from cleaning, handyman jobs to photography or building websites (Airtasker, no date).

Rationale behind the selection: Airtasker provides sophisticated task outsourcing features along with a visually appealing, intuitive web interface. Additionally, it has a engaging messaging platform for users to communicate with each other that builds trusts and transparency among employers and workers.

2.2.3 Indeed

Indeed is global job seeking and hiring platform that enables job seekers to search for matching jobs, research companies, and apply for jobs (Indeed, no date). Employers can post jobs in Indeed and attract a large pool of suitable candidates. Indeed provides user guides for both employers and job seekers to smoothly navigate their job hiring or seeking journey.

Rationale behind the selection: This application was chosen for its robust job posting, job searching, and ratings features. Additionally, Indeed grants a meaningful user experience by providing comprehensive user guides for resume creation, conducting interviews, company research and boosting visibility of job postings.

2.2.4 Taskrabbit

Taskrabbit brings people who has work that needs to be done and those who are ready and willing to work together (Taskrabbit, no date). This platform caters to a range of tasks from cleaning, moving, assembly to electrical help and plumbing.

Rationale behind the selection: This application was reviewed to get insights on its task matching feature, user-friendly interface and customer support capabilities.

2.2.5 Upwork

Upwork has been in the market for around 20 years, connecting talent remotely with opportunities to bring projects to life (Upwork, no date). It enhances talent souring in several categories, including writing, designing and video editing.

Rationale behind the selection: Upwork was chosen for review to gather information on its sophisticated communication tools, extensive user guidance and focus on remote work opportunities.

2.3 Review Criteria

This subsection lists the important features/functions of the above systems that were used as review criteria to evaluate those systems, which resulted in identifying the features that were later incorporated into the Gig Work Marketplace application.

(RC1) Job posting and Management: The process that employers follow to post or update jobs, with details such as job description, location, budget and required skills.

(RC2) Categorizing tasks: There are different categories of tasks ranging from handyman services, delivery, babysitting to online freelancing.

(RC3) Job search: The efficiency of searching for jobs using keywords, job titles and locations for gig workers.

(RC4) Job picking: Gig workers can pick jobs that match their skills, experience, and availability.

(RC5) User profiles: Both employers and gig workers can create and maintain accounts, where worker profiles include their skills, experience, reviews and preferences.

(RC6) Communication Tools: Users can message each other to obtain more information on posted jobs, negotiate terms of employment, ensure the reliability of jobs or establish trustworthiness of the workers.

(RC7) User Interface: The layout, ease of navigation, responsiveness, visual appeal and the user-friendliness of the application.

(RC8) Rating Workers: The ability to rate workers based on their job performance that assists in building credibility of the workers.

(RC9) Notification System: There are different types of notifications, including notifications for the receipt of job applications, application status changing and message receipt, that enhances user engagement with these digital platforms.

(RC10) Security: Web applications adopt security measures to protect user data from unauthorized access and to comply data protection laws and regulations.

2.4 Review of the Systems

2.4.1 GigSmart

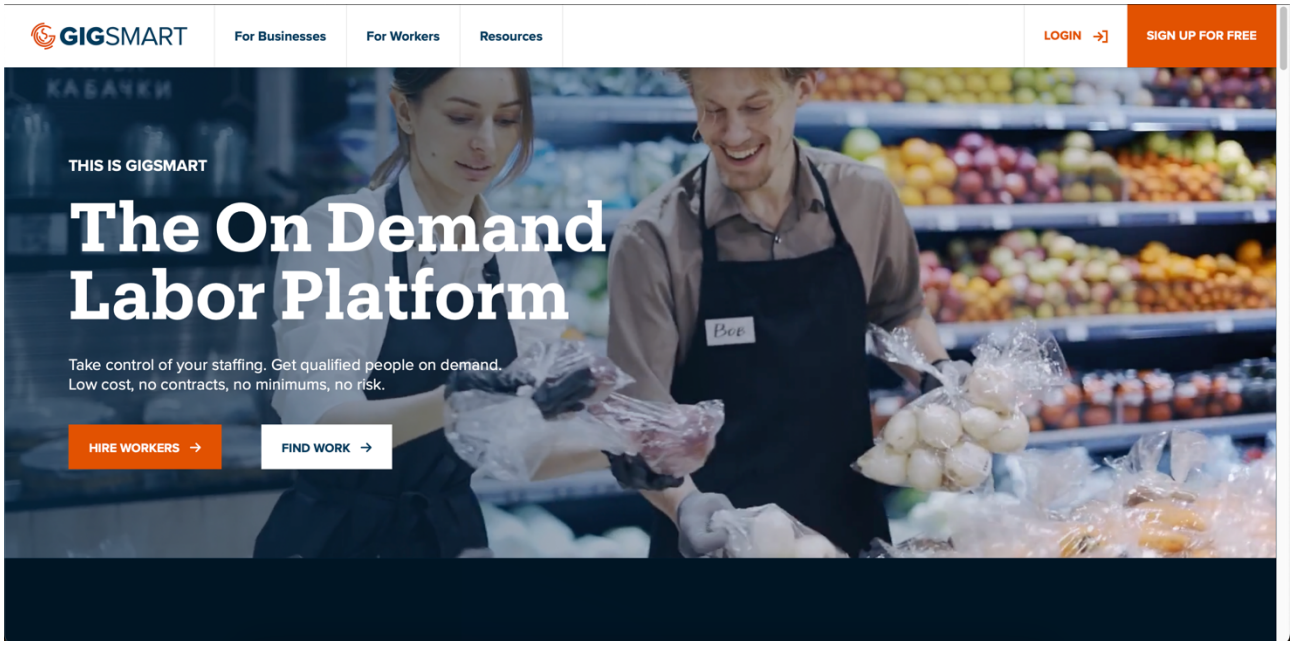
(RC1) Job posting and Management: Employers can post jobs with a job description, location, and a budget. Additionally users can update job postings with information they deem necessary at any point in the hiring lifecycle. GigSmart provides extensive background checks and motor vehicle record checks in collaboration with other service providers to ensure the reliability of hired help.

(RC3) Job search: GigSmart enables searching for full and part time jobs using keywords, job titles and locations. Search results sorting functionality is not provided for workers.

(RC4) Job picking: On-demand workers can find the right work opportunities that suits their skills, experience, availability, and preference. They are granted a comprehensive insurance package that covers occupational hazards and motor vehicles. The hassle of creating resumes for every job and facing interviews in the hiring process is minimized by the maintenance of updated worker profiles and verified ratings. The users are directed to the company website job portal to apply for jobs.

(RC7) User Interface: GigSmart's user interface is clean and structured, where users can find information easily and perform the required functions seamlessly. All the main functions of the application are easily accessible and responsive to user's actions, ensuring seamless interaction with users. However, GigSmart highlights the employer control over the hiring process which could be perceived as prioritizing employers over employees.

Figure 2.1 GigSmart User Interface



GigSmart has a strong reliance on its partners to provide different services, such as background checks and motor vehicle records checks. Also, applications for jobs can be submitted only by navigating to the relevant company website, which requires employers to maintain web applications.

2.4.2 Airtasker

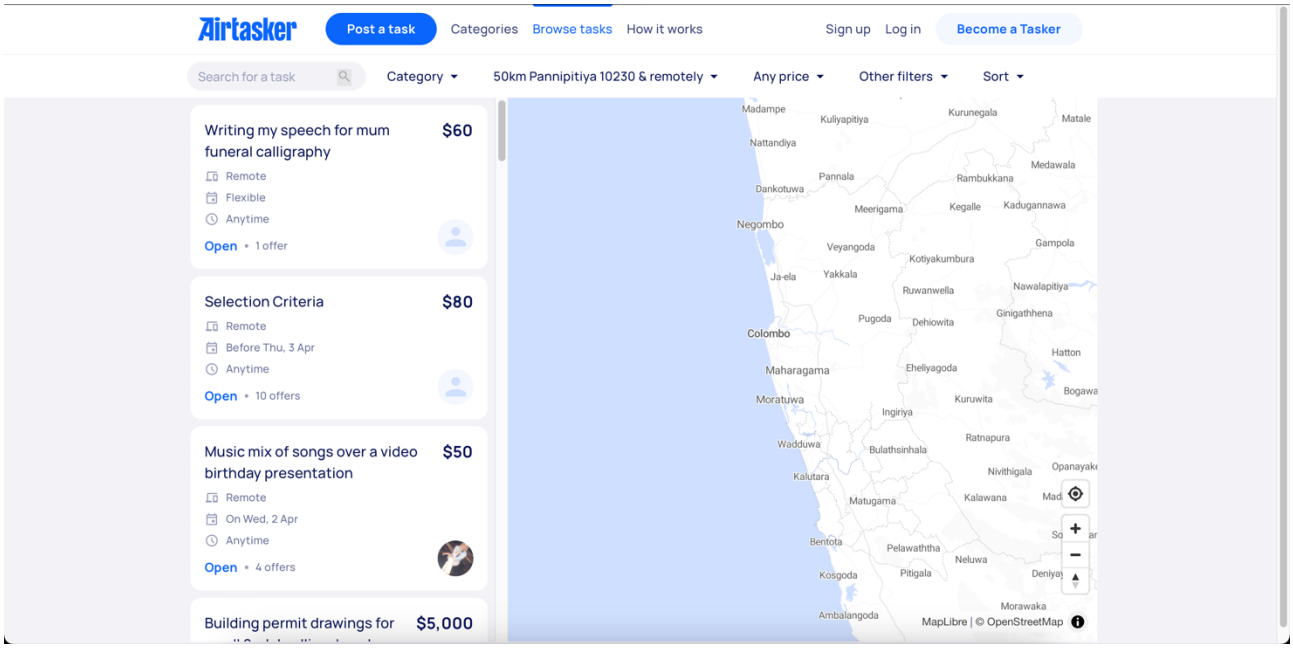
Airtasker is an Australian digital platform that connects those who need to get work done with people who are available and willing to work. Employers can post jobs and hire workers get those jobs done. Workers can search for jobs and apply for them to earn an additional income. This platform caters to a wide range of tasks from cleaning, handyman jobs to photography or building websites (Airtasker, no date).

(RC1) Job posting and Management: Employers can post jobs with a job description, location, and a budget. The visible fields for job posting are controlled by the keywords on job description which enhances the completeness of the job posting. Additionally users can update job postings with information they deem necessary at any point in the hiring lifecycle. Furthermore, employers are given the flexibility to post jobs without creating user accounts.

(RC2) Categorizing tasks: There are around 100 categories of tasks ranging from handyman services, delivery, babysitting to online freelancing in Airtasker. This categorization enables employers to post jobs with adequate level of information and workers to narrow down their job searches amidst thousands of job postings.

(RC3) Job search: Airtasker enables users search for jobs using keywords, job titles and locations for gig workers. It provides 3 system defined search parameters, and 2 user defined search parameters for search results optimization. In addition, users can sort job search results based on 5 job market specific parameters.

Figure 2.2 Airtasker Search Functionality



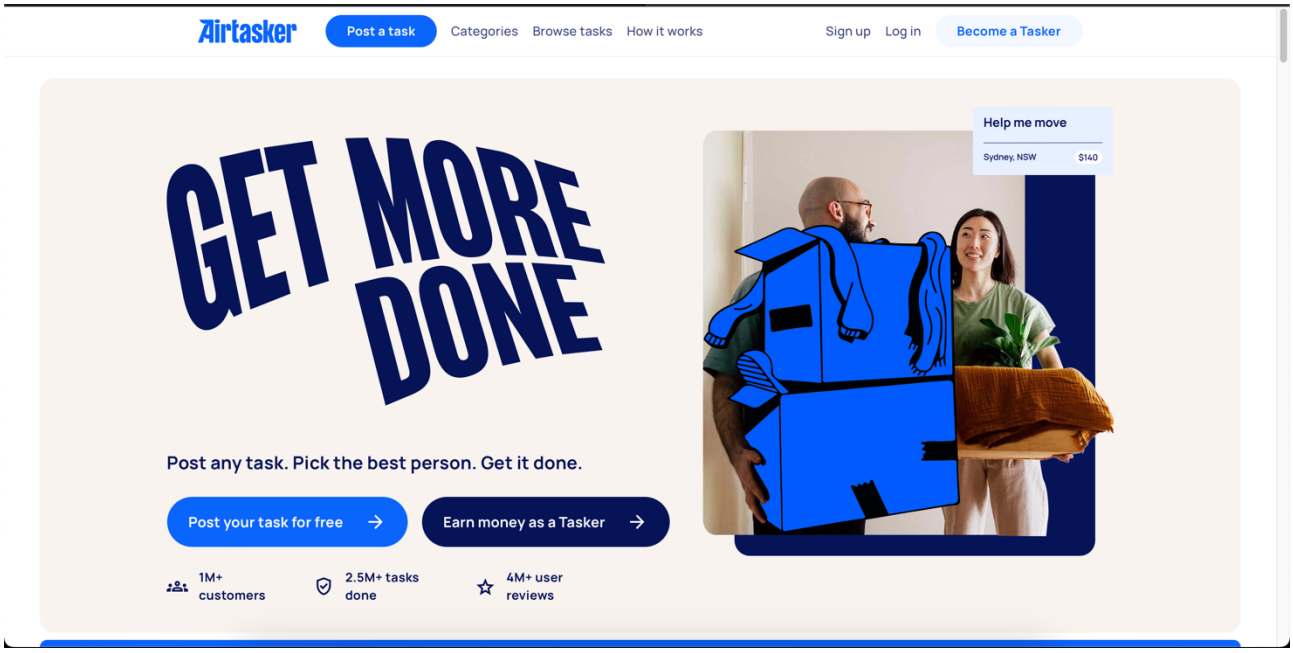
(RC4) Job picking: Gig workers can make job offers for postings that require their skills, experience, availability and match with their budget. They can also view offers made by other prospects for a job.

(RC5) User profiles: Both employers and gig workers can create and maintain accounts, where worker profiles include their skills, experience, and preferences. Additionally, they can sign up using single sign on in Google, Facebook or Apple, without creating an account.

(RC6) Communication Tools: Users can message each other to make job offers, obtain more information on posted jobs or negotiate terms of employment.

(RC7) User Interface: Airtasker has a clean layout, which is visually appealing and easy to navigate. All the main functions of the application are easily accessible and responsive to user's actions, ensuring seamless interaction with users.

Figure 2.3 Airtasker User Interface



(RC8) Rating Workers: When a worker completes a job, employers can rate workers based on their job performance, empowering prospective employers to choose reliable workers.

(RC9) Notification System: Airtasker gives notifications to users in different events, including the receipt of job offers, application status changing, and message receipt, that helps users to stay engaged from the moment of posting a job until it is completed.

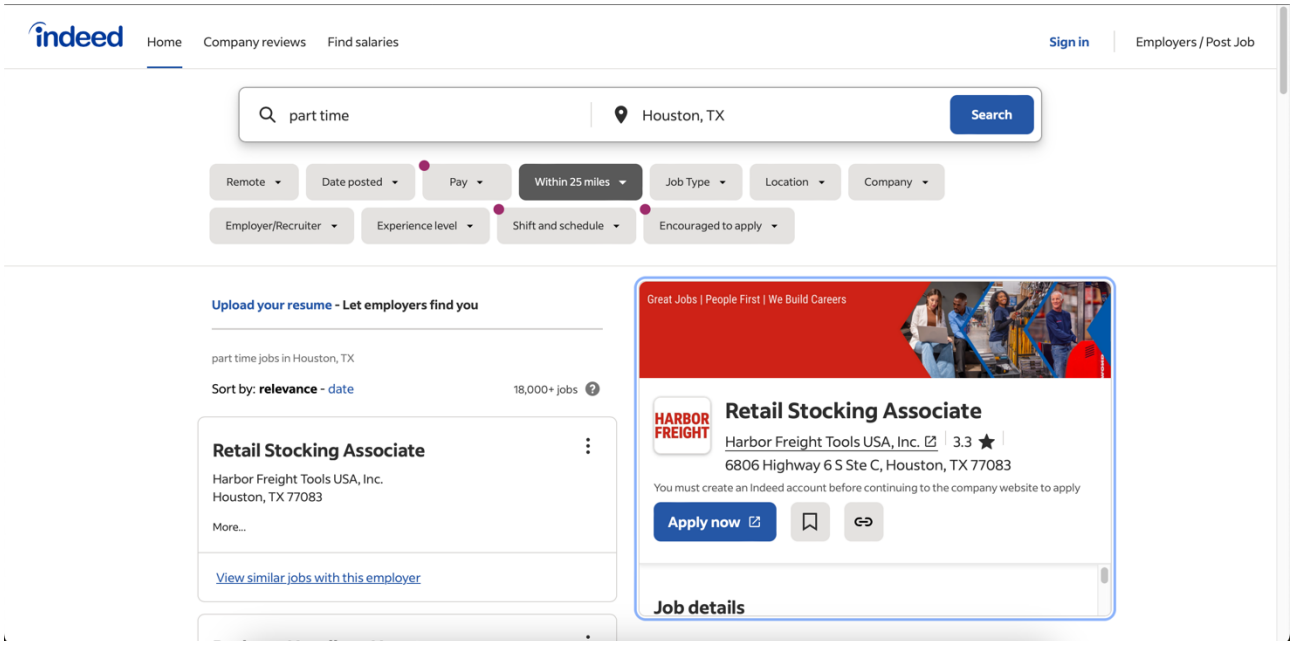
(RC10) Security: Airtasker has a privacy policy according to which they collect and manage user data. This policy complies with the laws and regulations of countries that Airtasker operates in. In addition, it adopts sophisticated security measures for user authentication and data encryption.

All in all, Airtasker provides an intuitive and visually appealing user interface, with advanced search capabilities and sophisticated job categorizing features. However, the ability to post jobs without user accounts could increase the risk of fradulent job postings. Also, the ability to view offers made by other prospects on jobs could be viewed as exposing confidential information on users.

2.4.3 Indeed

(RC3) Job search: Users can search for jobs in Indeed using keywords, job titles and locations. It provides more than 10 market specific search parameters for search results optimization. In addition, users can sort job search results based on relevance and date posted which makes job searching a seamless experience for job seekers.

Figure 2.4 Indeed Search Functionality



(RC8) Rating Workers: Rating is mutual in Indeed. Both employers and job seekers can be rated in Indeed. When a worker completes a job, employers can rate workers based on their job performance, empowering prospective employers to choose reliable workers. Workers can also rate employees based on the working conditions provided and the monetary and non-monetary rewards made.

2.5 Conclusions and Comparisons of Systems

This subsection provides a comparison of the systems with respect to each review criteria, assessing their pros and cons, highlighting the best features to include in Gig Work Marketplace application, and an evaluation of which system is the best overall.

The table below summarises the above comparison by scoring each system against the review criteria (score is given out of 5).

Table 2.1 Comparison of Systems Review Criteria

Review Criteria	GigSmart	Airtasker	Indeed	Taskrabbit	Upwork
(RC1) Job posting and Management	3	4	4	3	3
(RC2) Categorizing tasks	4	4	4	3	3
(RC3) Job search	3	4	4	3	2

(RC4) Job picking	3	3	4	3	3
(RC5) User profiles	3	4	4	4	3
(RC6) Communication Tools	3	4	4	3	3
(RC7) User Interface	3	4	4	3	3
(RC8) Rating Workers	4	4	3	4	3
(RC9) Notification System	2	3	3	2	2
(RC10) Security	3	3	4	3	3
Total Score out of 50	31	37	35	31	28

- Job posting: Both GigSmart and Indeed require employers to create employer accounts to post jobs, while Airtasker allows job posting without business accounts.
- Categorizing tasks: GigSmart provides a clean and organized interface to view the available job categories.
- Job search: While Airtasker requires creating a user account for browse jobs, Indeed allows job searching without creating a user account.
- Task picking: Indeed provides a user friendly option to apply directly for the position by logging in from the user account.
- User profiles: The sign up process in Airtasker is easy to follow with a limited number of manual steps.
- Communication Tools: Airtasker allows employers and workers to maintain 1-1 communication via private messages.

Finalized list of features

Based on the above comparison of similar applications, Airtasker is the best system overall. Inspired by the best features of all the above systems, users of Gig Work Marketplace platform would benefit from the following features.

1) Job Posting: Employers can post jobs quickly without the hassle of creating a user account. The risk of fradulent job postings will be minimized by enabling workers and employers to communicate with each other via the in- app messaging system.

2) Categorizing tasks: Tasks in the platform are categorized based on the nature of the service to enable employers to easily post their jobs and workers to efficiently find suitable opportunities. The interface should be visually appealing and easy to navigate.
3) Job search: Gig workers can search for jobs using keywords, job titles and locations without creating a user account.
4) Task picking: Gig workers can pick tasks that match their skills, experience, and availability by logging in from a user account.
5) User Profiles: Both employers and gig workers can create user accounts, where worker profiles include details on skills, ratings, and completion rates to help employers make informed decisions.
6) Communication Tools: Both employers and workers can maintain clear and efficient communication with an in-app messaging system.

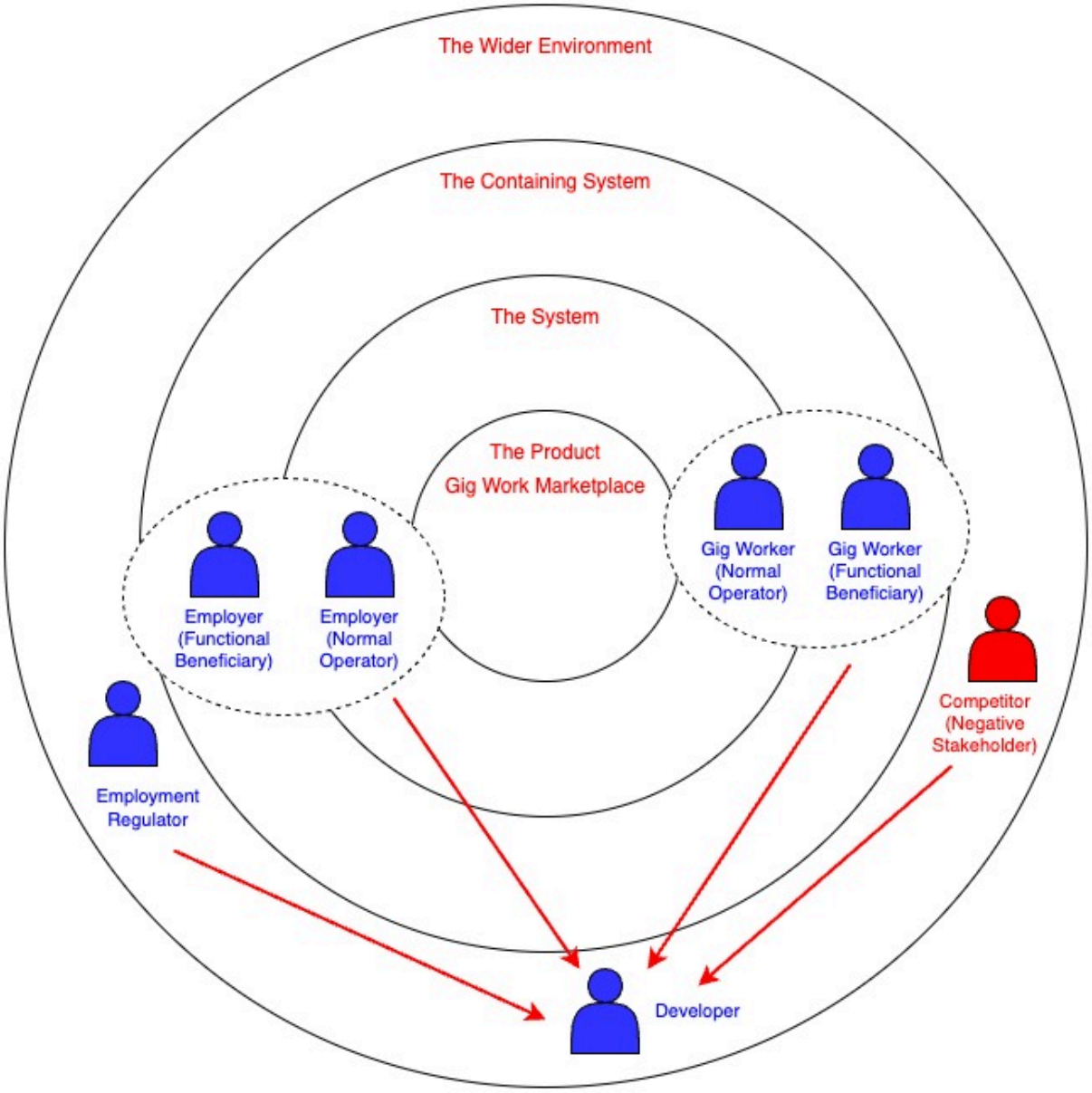
3 Chapter 3 System Requirements

3.1 Introduction

This chapter provides an overview of the stakeholders who has an interest in the Gig Work Marketplace web application, the external systems that interact with the web application, the information flow between the stakeholders/external entities and system, and how the system requirements that fulfils the needs of the identified stakeholders can be classified into functional, non-functional and user interface categories.

3.2 System Stakeholder

Figure 3.1 Onion Model for Stakeholder Analysis



The stakeholders mentioned above are involved in the development and use of this gig work marketplace web application. Their roles and responsibilities towards the project are as follows.

1) Project Manager
Plan, execute and monitor the entire project, managing resources needed, ensuring the project stays within budget, meets milestones, and generates the deliverables at the expected quality. The project manager is responsible for setting the scope of the project, managing risks and maintaining communication with other stakeholders.

2) Business Analyst
Probe the user requirements to identify the real problem, suggest alternative solutions to address the user's problem, select the most feasible, desirable and viable solution, explain how the solution can be supported by the system, and design the detailed workflow of the solution.

3) Developer
The developer investigates the technical feasibility of the suggested solution, evaluate different approaches to implement the solution, write and maintain code to develop the frontend and backend of the system, conduct unit testing and integration testing, and fix defects identified during development.

4) Tester
Finalize the scope of testing and writing test cases to cover that scope. Execute different testing methodologies, including blackbox testing, whitebox testing and exploratory testing. Report and monitor defects using test management tools and documenting the test results. The tester is responsible for the evaluation of whether the system fulfilling the agreed project aims and objectives.

5) Gig Worker
Gig worker is an end user of this system who benefits from the services provided. They can use this digital platform to find work that meets their skills, preferences, budget and availability. They are responsible for the completion of tasks they agree to perform via the web application, in the expected quality within the given timeframe.

6) Employer
Employer is also an end user of this digital platform who uses this system to post work that they need to get done, accept applications from interested workers, review and select applicants, and compensate the workers for successful job completion. They will rate workers based on the services provided and help other users to find reliable workers. Employers are responsible to comply with labour laws of the region they operate in and provide the resources needed to carry out jobs.

7) Competitor
Competitors run similar systems that connects talent with opportunities. They are responsible for complying with fair business and marketing practices, maintain suitable standards in their applications and continuously improve the quality of their businesses for the progress of the industry.

8) Employment Regulator

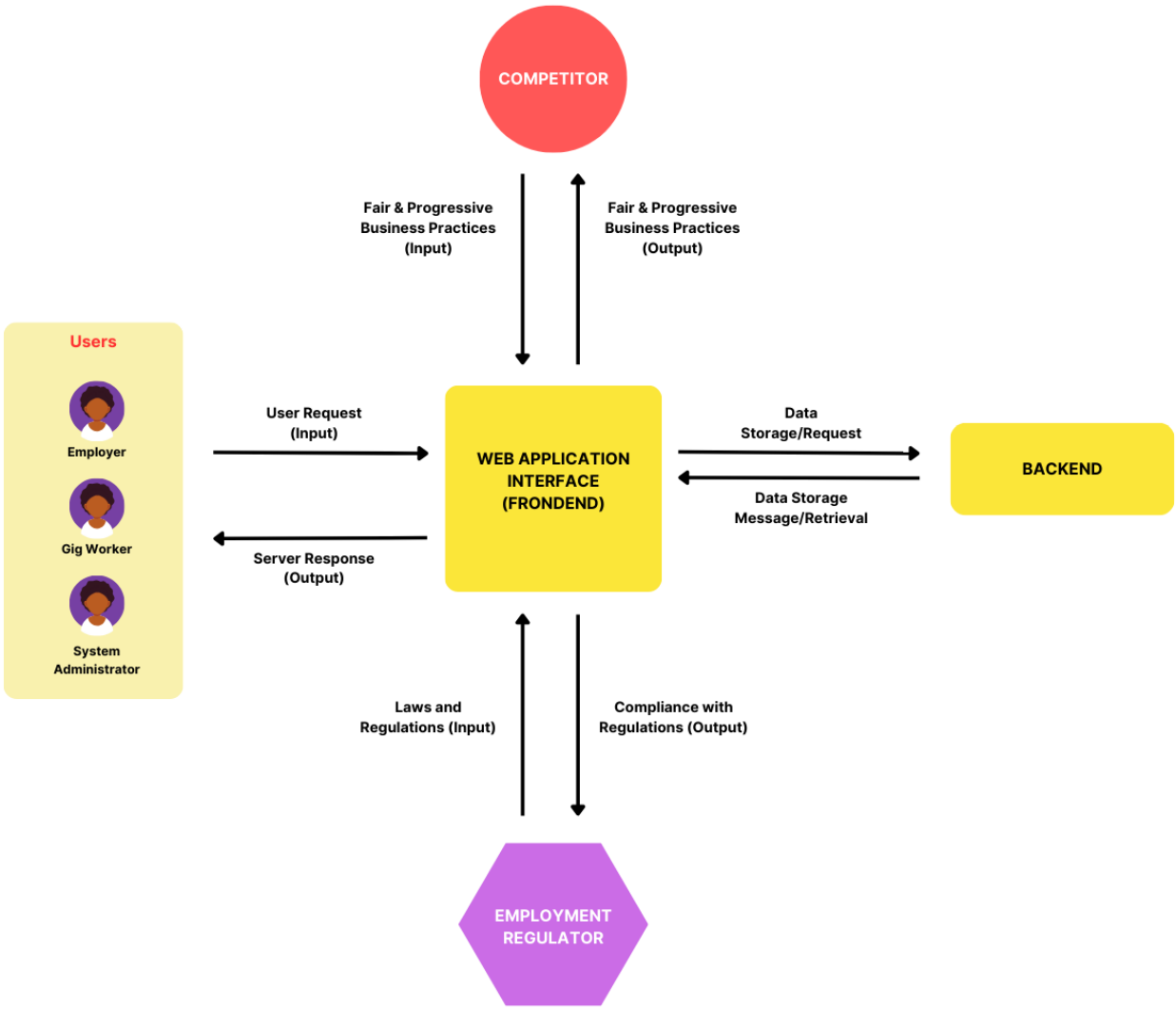
Regulators establish laws and regulations for the businesses operating in the industry to comply with. They review the industry trends and identify areas that can cause violations of basic standards, impose laws to prevent the violations from happenning and take action against parties who break the regulations. They are responsible to ensure that different stakeholders maintain proper conduct in the industry.

3.3 System Context

This subsection describes the system context outlining the information flows between the system and external entities and stakeholders that interact with it.

The context diagram given below illustrates the information flow between the application and the external entities and stakeholders.

Figure 3.2 System Context Diagram



An overview of how each of these external entities engage with the system is given below.

1) Gig Worker

These are end users that search for jobs in different categories, select suitable opportunities, create user accounts and apply through the web application. They also engage with employers who post jobs in the system through the messaging system.

Information flow

- **Input**: Queries for search, user account information, job application information, messages to other users.
- **Output**: Search results, toast messages, notifications, messages from other users.

2) Employer

Employers are end users that create user accounts, post jobs, review applications and select applicants. They engage with prospective or current workers through the messaging system.

Information flow

- **Input**: User account information, job details.
- **Output**: Toast messages, notifications, job applications, messages from other users.

3) System administrator

System administrators organize information in the system. They create different job categories for the employers to post their jobs and for the workers to narrow down their search to. System administrators monitor the industry trends and user behaviour to determine what job categories represent the different opportunities that arise in the job market.

Information flow

- **Input:** Job category information.
- **Output:** Toast messages.

4) Competitors

Competitors engage in businesses similar to the services provided by Gig Work Marketplace application.

Information flow

- **Input:** Fair and progressive business practices.
- **Output:** Fair and progressive business practices.

5) Employment Regulator

Regulators establish laws and regulations for the businesses operating in the industry to comply with.

Information flow

- **Input:** Employment laws and regulations, data protection laws.
- **Output:** Compliance with laws and regulations.

6) Database server
Database connects with the frontend application interface to process and manipulate data.

Information flow
- **Input:** Data storage and manipulation requests.
- **Output:** Data retrieval and request execution messages.

3.4 System Requirements

3.4.1 Functional Requirements

3.4.1.1 Essential Functional Requirements

(FR1) User Registration and Authentication
- Employers and workers can create accounts.
- Secure login and password maintenance.

(FR2) Profile Management
- Employers can create and manage company or individual profiles.
- Workers can create and update personal profiles, including skills and experience.

(FR3) Job Posting and Management
- Employers can post jobs with details such as job description, requirements, and budget.
- Employers can manage and edit job postings.

(FR4) Job Search and Application
- Gig workers can search for jobs based on various filters (location, pay, job type).
- Workers can apply for jobs directly through the platform.

(FR5) Communication Tools
- In-app messaging between employers and workers.
- Notifications for job postings, applications, and messages.

(FR6) Application Tracking
- Employers can track applications and manage the hiring process.
- Workers can track the status of their applications.

(FR7) Ratings and Reviews
- Employers can rate and review workers.

3.4.1.2 Desirable Functional Requirements

(FR8) Advanced Search Filters
- More detailed filters for job searches (e.g., remote/on-site, date posted).

(FR9) Calendar Integration
- Integration with calendar apps for scheduling interviews and job start dates.

3.4.2 Non-Functional Requirements

3.4.2.1 Desirable Non-Functional Requirements

(NFR1) Performance
- Fast load times and responsive design.
- Scalability to handle a large number of users.

(NFR2) Security
- Data encryption for sensitive user information.
- Regular security audits and updates.

3.4.3 User Interface Requirements

3.4.3.1 Essential User Interface Requirements

(UIR1) Usability
- Organized, visually appealing, and easy to navigate.

3.4.3.2 Desirable User Interface Requirements

(UIR2) Usability
- Intuitive and user-friendly interface.

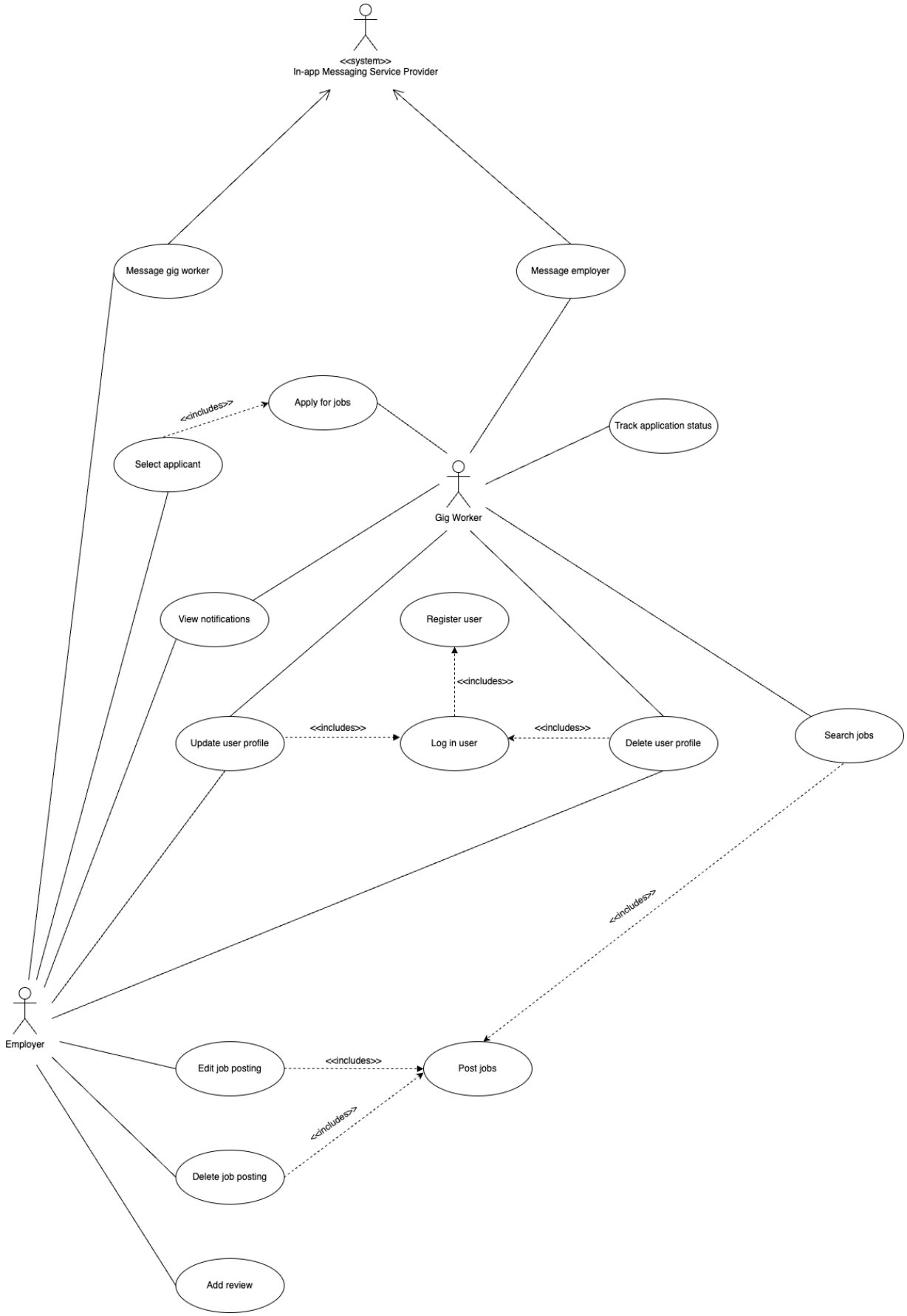
4 Chapter 4 System Design

4.1 Introduction

This chapter includes the design of the Gig Work Marketplace system, including use case diagrams, use case descriptions, sequence diagrams of the use cases, class diagram of the system, and entity relationship diagram. It provides the wireframes that illustrates the design screens of the interface followed by screen graphs that shows links between the design screens.

4.2 Use Case Diagram

Figure 4.1 Use Case Diagram



4.3 Use Case Descriptions

This subsection provides use case descriptions for each use case identified in the use case diagram of the Gig Work Marketplace system.

4.3.1 Register user

Description: This describes the process an employer or a gig worker follows to get registered in the system to perform other tasks.

Actors: Employers and gig workers.

Prerequisites: The actors must have an email address.

Main Flow:
- The user navigates to the ‘Home’ page.
- The user clicks the ‘Sign Up’ button on the menu.
- The system navigates the user to the ‘Sign Up’ page.
- The user enters the email and the password.
- The system validates the details and save them in the database.
- The system displays a toast message to the user.

Alternative flows:
- If the email address is not in the required format, an error is given.
- If the email address is already registered, a message is given to the user.
- If the password is strong enough, a warning is given.

4.3.2 Post job

Description: This use case outlines the process that an employer follows to post a job opportunity in the system.

Actors: Employer.

Prerequisites: Employers must have user accounts to post jobs.

Main Flow:
- The user logs in to the system.
- The user clicks the ‘Post Job’ button on the menu.
- The system navigates the user to ‘Post Job’ page.
- The user enters the job details and click the ‘Post Job’ button.
- The system validates the details and save them in the database.
- The system displays a toast message to the user.

Alternative flows:
- If the job details are not in the required format, an error is given.
- If the user has not logged in, the user is directed to the ‘Log In’ page. Once the user logs in, the user is directed to the ‘Post Job’ page.

4.3.3 Search jobs

Description: This describes the process gig workers use to search for jobs in the web application.

Actors: Gig Worker.

Prerequisites: Not Applicable.

Main Flow:
- The user navigates to the 'Home' page.
- The user clicks the 'Browse Jobs' button on the menu.
- The system navigates the user to the 'Browse Jobs' page.
- The user enters the search criteria.
- The system validates the details and search for jobs that match the entered keywords in the database.
- The system displays the search results to the user.

Alternative flows:
- If there are no jobs that match the entered keywords, a message is returned.

4.3.4 Apply for jobs

Description: This use case describes how users apply for jobs that fulfil their requirements.

Actors: Gig Worker.

Prerequisites: Gig worker needs to have a user profile and have chosen a job to apply for.

Main Flow:
- The user is in the job search results page.
- The user clicks the 'Apply Now' button on the selected job.
- The system saves the user's profile details and session details in the database.
- The system displays a toast message to the user.

Alternative flows:
- If the user has not logged in, the user is directed to the 'Log In' page. Once the user logs in, the user is directed to the search results page.

4.3.5 Message employer

Description: This process is used to send a message to an employer who has posted a job in the system.

Actors: Gig Workers.

Prerequisites: Gig worker should be logged into the system and have chosen a job.

Main Flow:
- The user selects a job posting.

- The user clicks the message icon on the selected job.
- The system directs the user to the ‘Message’ page.
- The user enters the message and clicks ‘Send’.
- The system sends the message to the employer who posted the job.

Alternative flows:

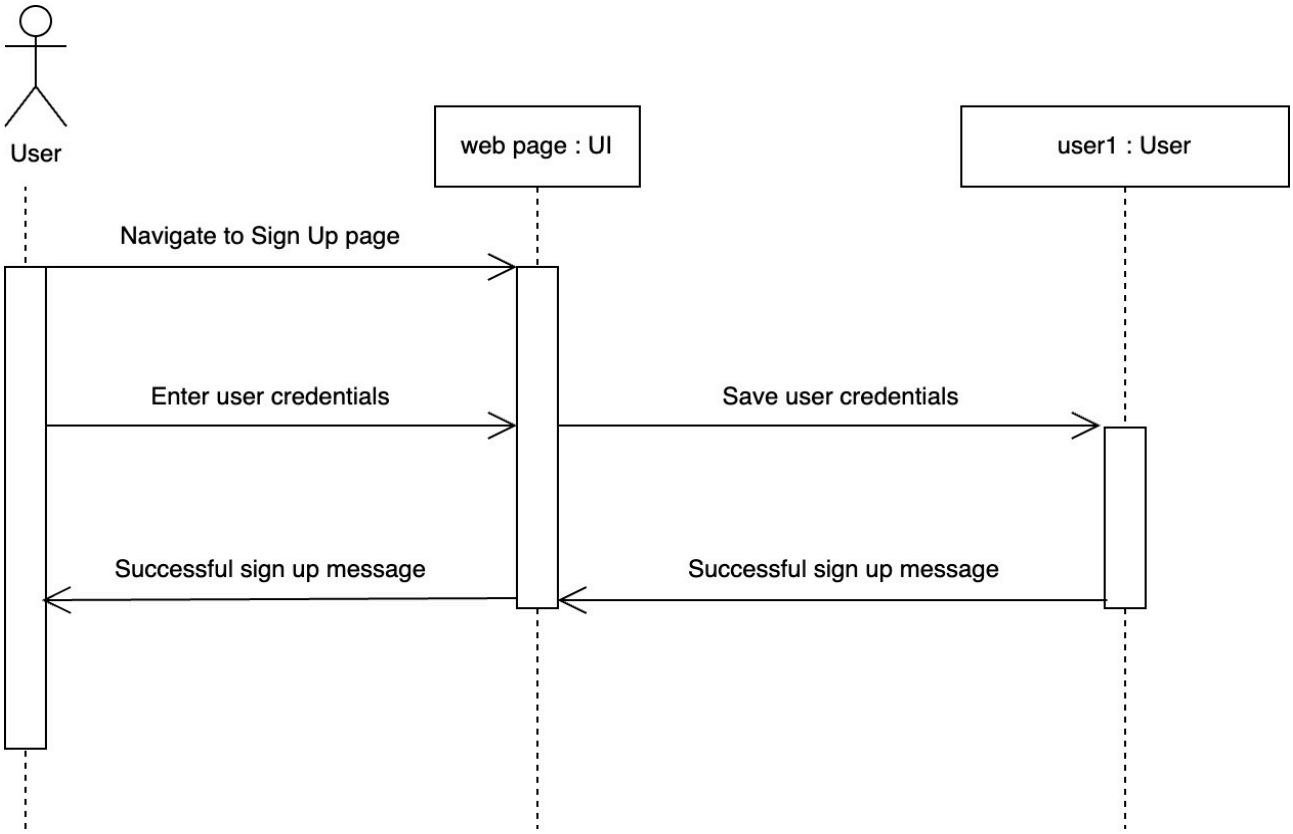
- If the user has not logged in, the user is directed to the ‘Log In’ page. Once the user logs in, the user is directed to the search results page to select a job.

4.4 Sequence Diagrams

This subsection provides sequence diagrams for each use case identified in the system.

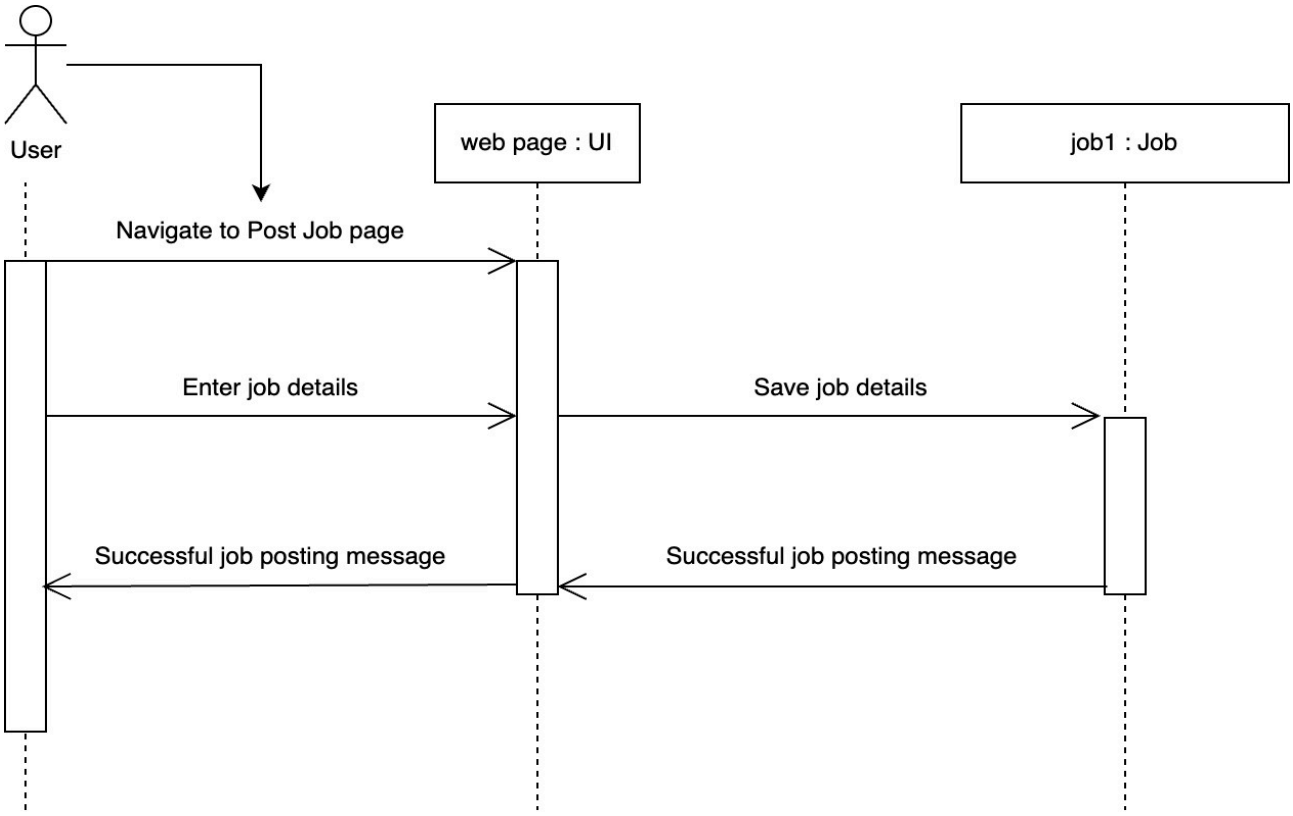
4.4.1 Register user

Figure 4.2 Sequence Diagram for Register User



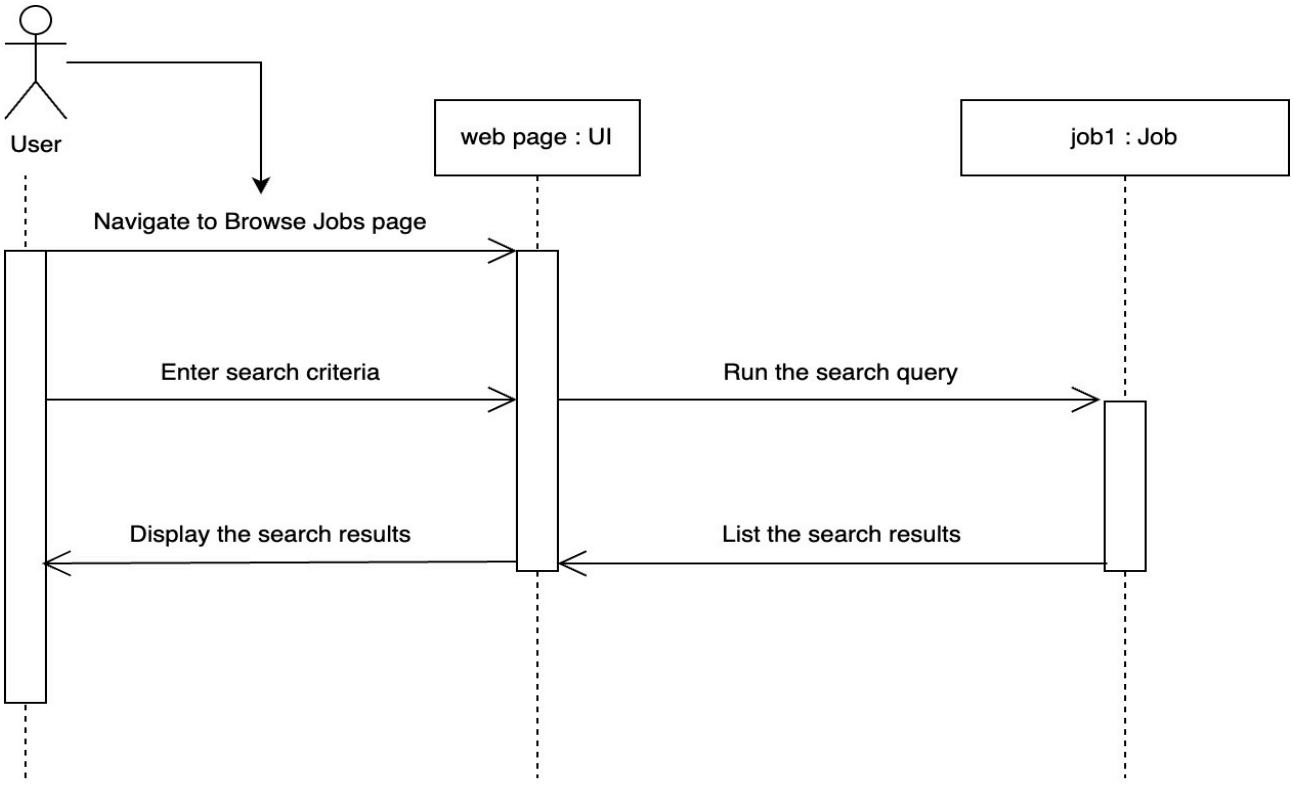
4.4.2 Post job

Figure 4.3 Sequence Diagram for Post Job



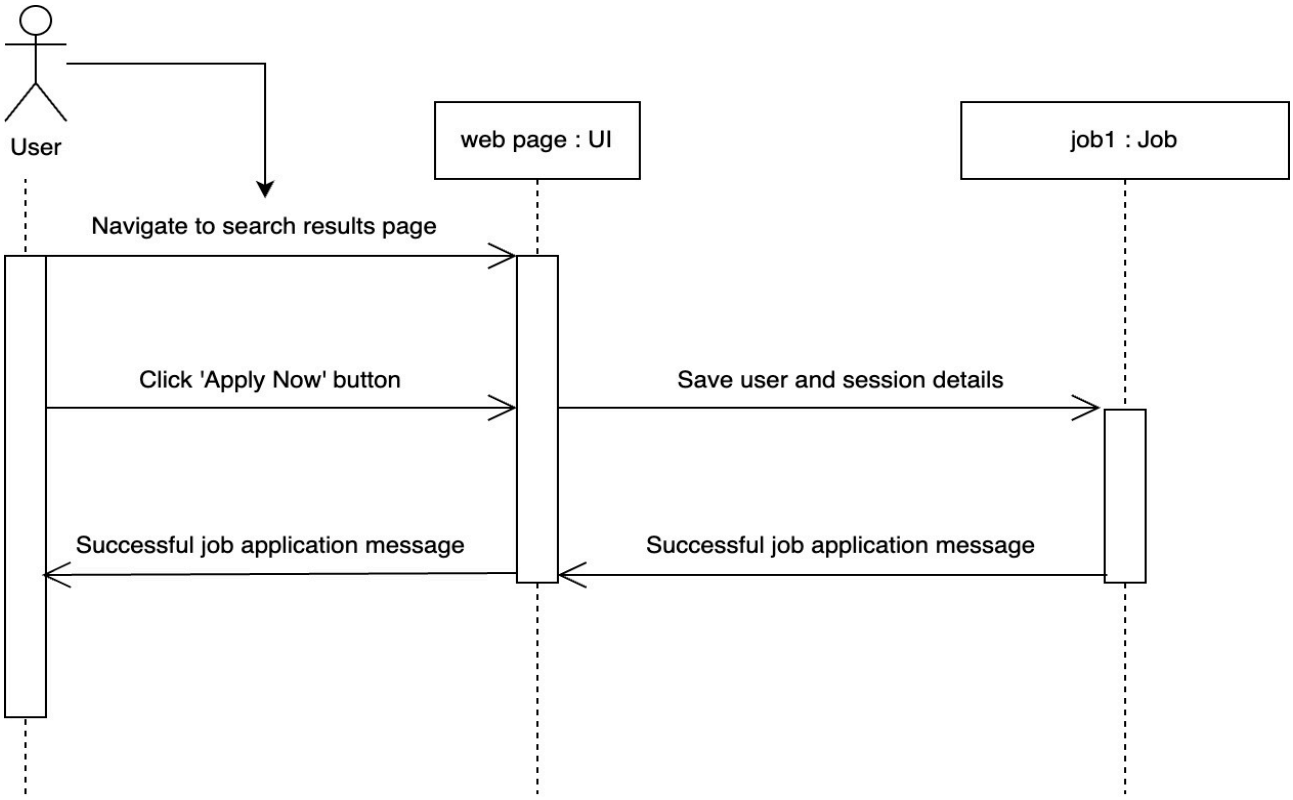
4.4.3 Search jobs

Figure 4.4 Sequence Diagram for Search Jobs



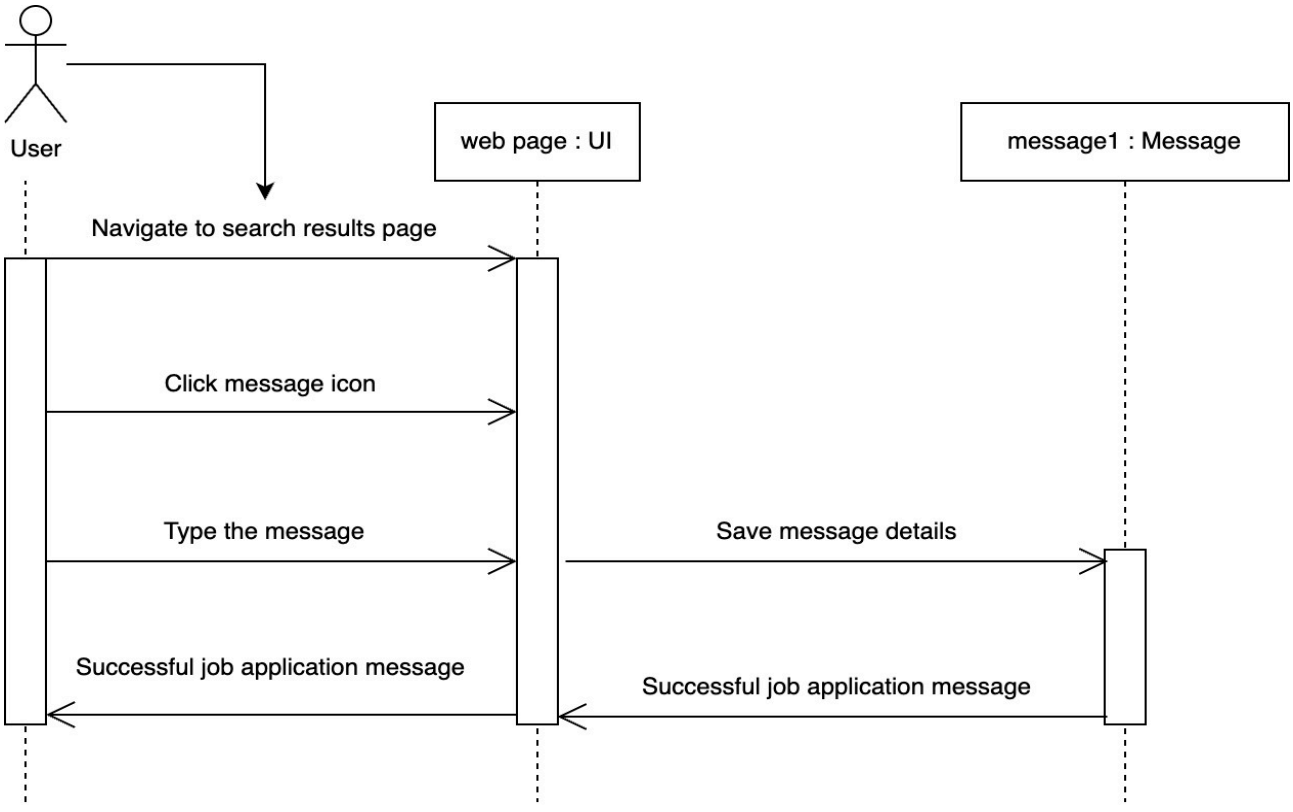
4.4.4 Apply for jobs

Figure 4.5 Sequence Diagram for Apply for Jobs



4.4.5 Message employer

Figure 4.6 Sequence Diagram for Message Employer



4.5 Class Diagram

The class diagram of this web application is given in Figure 4. There are several association relationships among classes in this system, which are listed below.

- Employer creates Job Posting.
- Job Posting belongs to Job Category.
- Gig Worker creates Job Application.
- Gig Worker applies for Job Posting.
- User sends Message.
- User receives Message.
- User receives Notification.
- Employer gives Rating.

The aggregation relationships among the classes in this system are as follows.

- User has a User Account.
- Notification has a message.

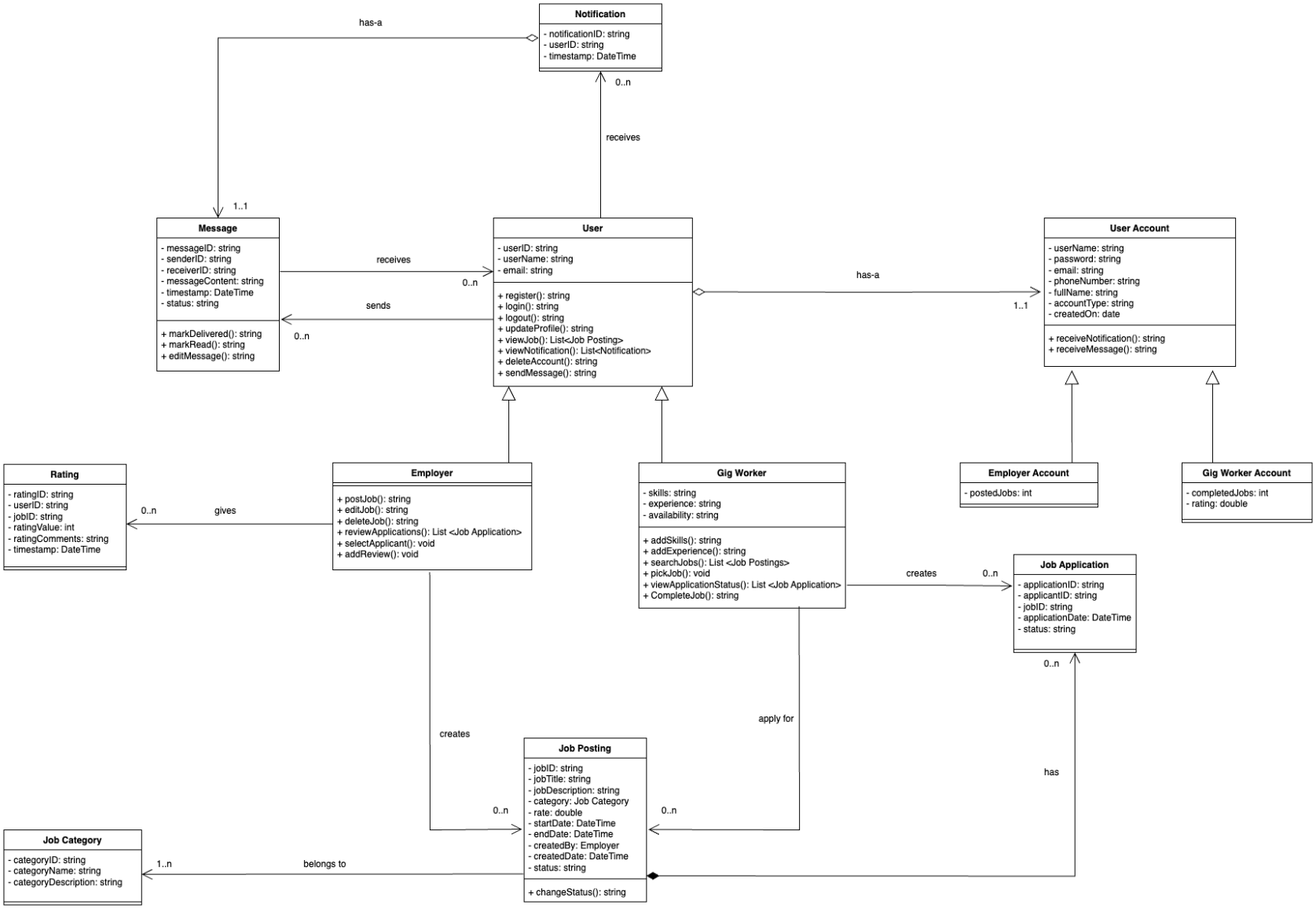
Composition relationships exist between two classes in this system, which is given below.

- Job Application has a Job Posting.

The generalization relationships between classes in this web application are listed below.

- Employer is a User.
- Gig Worker is a User.
- Employer Account is a User Account.
- Gig Worker Account is a User Account

Figure 4.7 Class Diagram



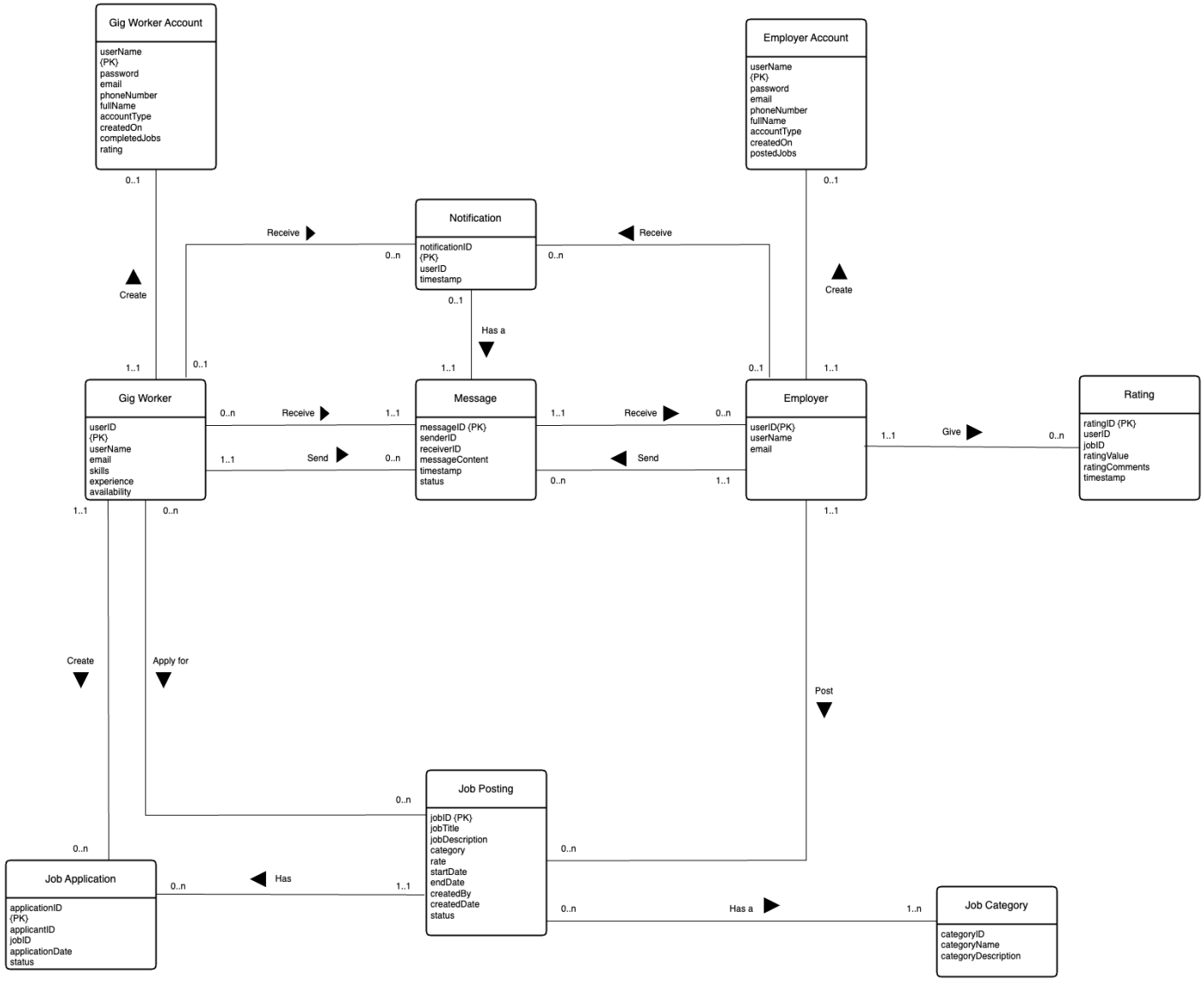
4.6 Entity-Relation Diagram

The entity-relationship diagram of this system given in Figure 5. The main relationships in the database of this web application are,

- Gig Worker create Gig Worker Account.
- Employer create Employer Account.
- Employer post Job Posting.
- Gig Worker create Job Application.
- Gig Worker apply for Job Posting.
- Job Posting has Job Application.
- Job Posting has a Job Category.
- Gig Worker receive Message.
- Gig Worker send Message.
- Employer receive Message.
- Employer send Message.
- Notification has a Message.
- Gig Worker receive Notification.
- Employer receive Notification.
- Employer give Rating.

Cardinality and participation of each relationship, and primary keys of each entityt are represented in the diagram.

Figure 4.8 Entity-Relation Diagram

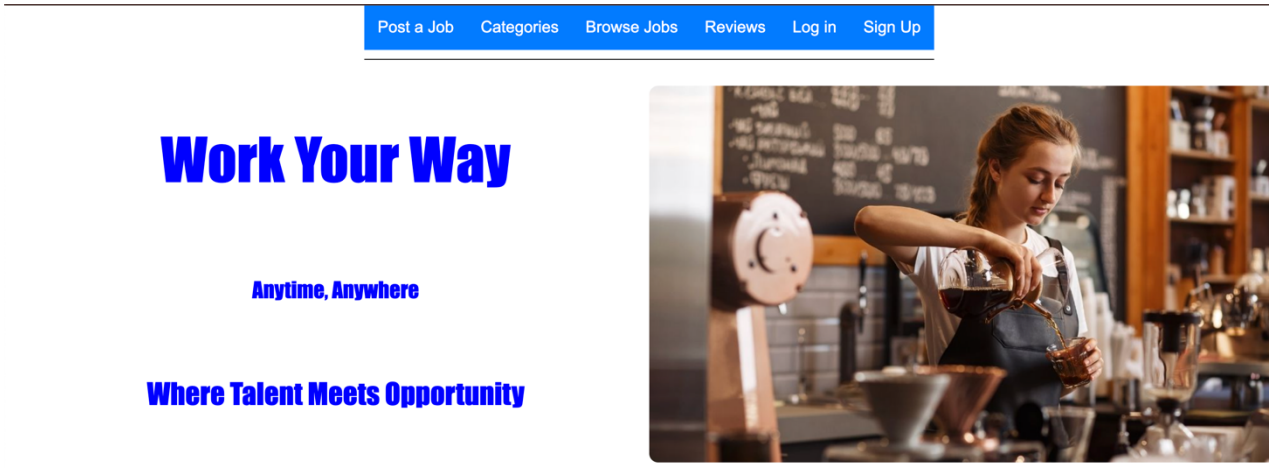


4.7 GUI design

4.7.1 Wire frames

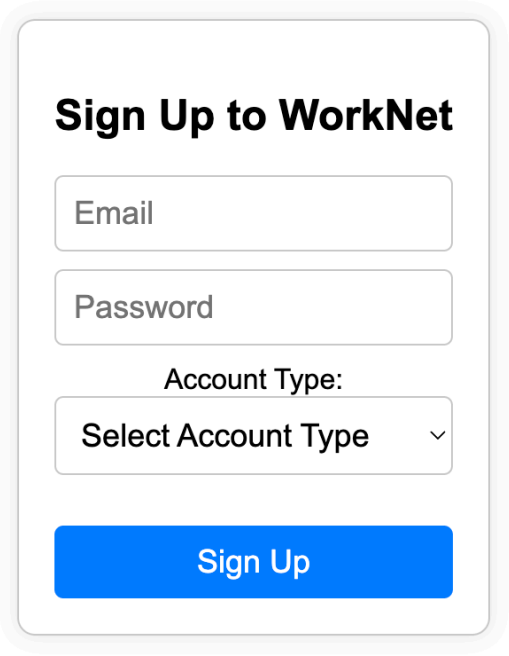
Home page

Figure 4.9 Wire Frame of Home Page



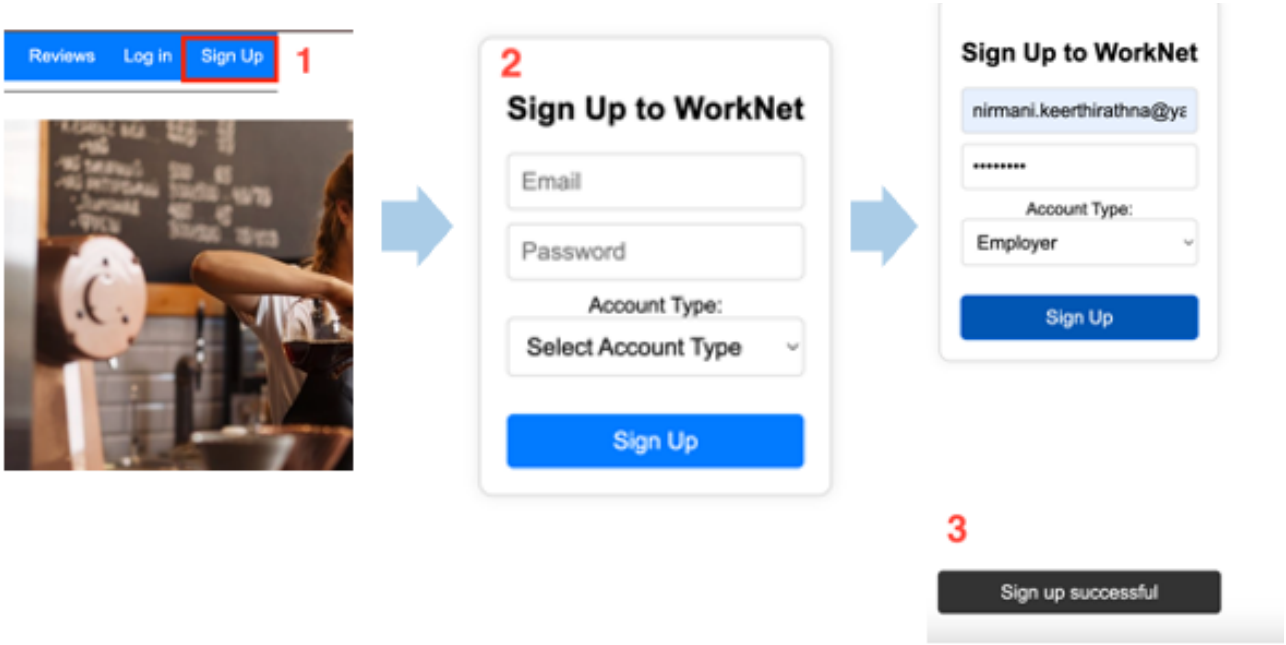
Use case 1: Register User

Figure 4.10 Wire Frame of Sign Up User Functionality



4.7.2 Screen graphs

Figure 4.11 Screen Graph of Sign Up User Functionality



5 Chapter 5 Implementation

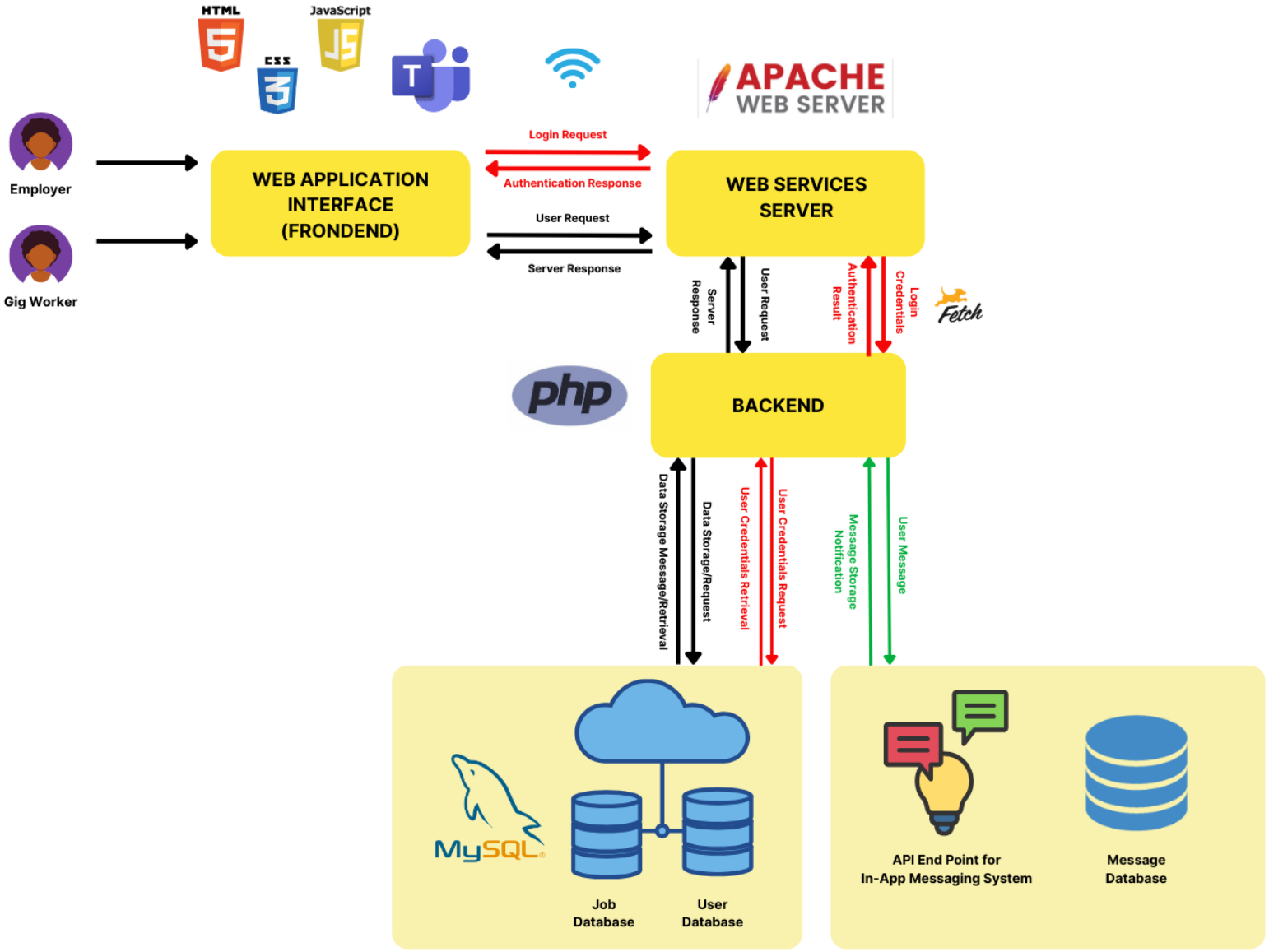
5.1 Introduction

This section describes the implementation of this web application, illustrating the high-level components of the software and links between them, reviewing the technologies used in implementation, and discussing the issues encountered and how they were resolved. It details the classes that were created during implementation, their purpose and how they were used in building this system.

5.2 System Architecture

Figure 5.1 illustrates the high-level components of this system with the links in between them.

Figure 5.1 System Architecture Diagram



5.3 Review of Technologies

There are various software and hardware that can be used to develop this gig work marketplace web application.

The available web development tools are as follows.

- **Hardware**: MacBooks with macOS, Windows laptops and Linux laptops support the efficient development of stable, user friendly web applications.
- **Frontend development:** React.js, Angular, HTML are popular and comprehensive frontend development technologies that can be used to build dynamic web applications.
- **Backend development:** Node.js, PHP and Ruby on Rails are server-side scripting tools known for their compatibility with javascript and scalability.
- **Database management**: MYSQL, PostgreSQL and MongoDB are popular and robust open-source relational database management systems that are flexible and scalable.
- **Text editor**: Sublime Text, Visual Studio Code and Atom are open-source, user-friendly text editors that offer extensive features and stability, to write and edit code.
- **Development environment:** WampServer, MAMP, Laragon and XAMPP are local web development environments that supports multiple scripting and programming languages.

A text editor is needed to create web pages of a web application (MDN Web Docs, no date). There are built-in text editors in every operating system. MacOS has TextEdit that can be used to write HTML code (Apple Inc., no date).

Out of the available alternatives the below technologies are selected for this web application.

- **Hardware:** MacBook Pro M2 with macOS Sonoma 14.5 is used for the development of this application due to its stability, efficiency and high performance in fulfilling the requirements of web development. Additionally, purchasing new hardware is not cost-friendly as the available MacBook fulfills the demands of web application development of the required scale.
- **Frontend development**: HTML is used to structure the web pages, defining their layout and content (AppMaster, 2023). CSS makes the web pages visually appealing and user-frinedly by styling the HTML elements (Abramowski, 2023). Javascript is used to add interactivity to the web pages using its features such as form validation, event handling etc (Joshi, 2023). These frontend development tools enabled creating a responsive and interactive web application. While React.js and Angular are sophisticated frontend development tools, they were not chosen due to their incompatibility with macOS Sonoma 14.5 version. Attempts to install both these frameworks resulted in configuration failures due to their component dependencies.
- **Backend development**: PHP is used for server-side scripting where the web pages interact with database to store and retrieve data, due to its compatibility with a wide range of database management systems and the availability of a rich source of libraries for web development. Node.js and Ruby on Rails did not make the cut due to the popularity of PHP and its compatibiliy with the chosen tools.

- **Database management**: MYSQL is used as the relational database management system to query and manipulate data using SQL due to its scalability and ease of use. MYSQL's seamless integration with PHP for data manipulation and retrieval made it the obvious choice.
- **Text editor**: To write and edit the code efficiently Visual Studio Code is used due to its compatibility with various programming languages, and its features such as code completion and syntax highlighting, which enhances productivity. Its well-established community support and extensions are preferred over the features provided by other alternatives.
- **Development environment**: XAMPP was used as the local server environment for development and testing of this web application due to its inclusion of the chosen frontend and backend development tools, and ease of installation and use, making it the convenient choice. WampServer was not chosen as it does not support macOS. Even though other local development environments are user-friendly and easy to configure, not all of them support the frontend and backend development tools chosen to develop this application.

5.4 Implementation Issues and Solutions

Significant implementation issues were encountered during the implementation. Investigations were conducted to identify solutions to those issues and feasible solutions were implemented. A list of those issues, investigations conducted, and solutions implemented are detailed below.

1) **Issue:** Choosing different frontend and backend development tools that are compatible with the available hardware.
   Investigation coverage: Different software and hardware that can be used to develop this gig work marketplace web application were reviewed, including frontend and backend development tools, database management systems, text editors and different development environments.
   Solution implemented: The tools described in section 5.3 were selected based on that review.

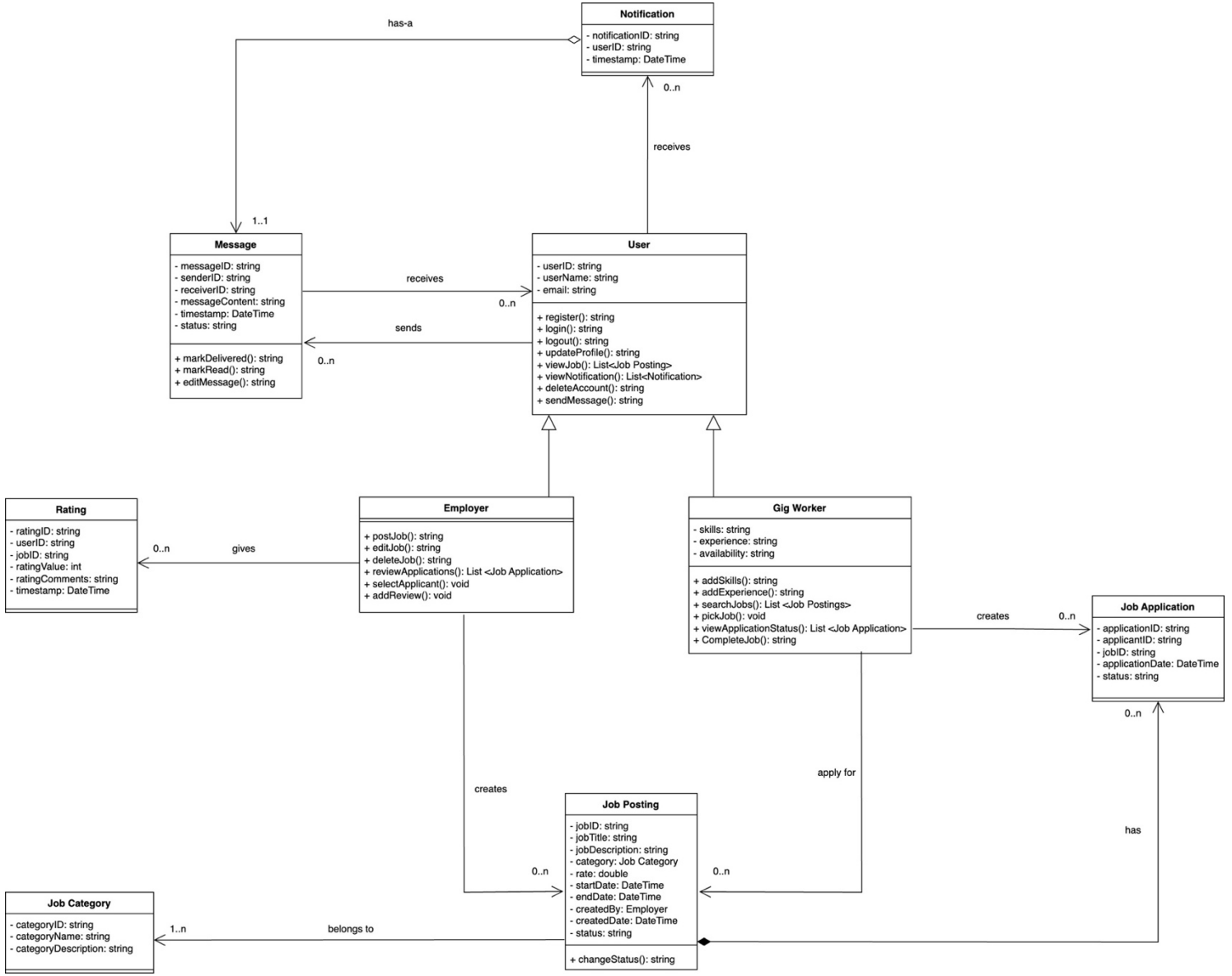
2) **Issue:** Protecting user data from unauthorized access became a significant issue as the system stores users' personal information and transmits that information in several functional flows.
   Investigation coverage: To protect users' data from authorized access, investigations were conducted around various data encryption methods and security protocols available for secure data transmission. Data encryption methods reviewed include SSL/TLS protocol that secure data transmitted between frontend and backend by encrypting the transmitted data, secure authentication mechanisms such as hashing passwords and two factor authentication, and AES encryption that secure data stored in databases by encrypting them.
   Solution implemented: Password hashing technique is implemented where passwords entered at signup and login are hashed when saving in the database.

3) **Issue**: Designing a responsive and user-friendly client interface that improves user experience became an important aspect of this project.
   Investigation coverage: Conducted research on creating visually appealing and easy to navigate interfaces.

Solution implemented: Implemented responsive design techniques that makes the application work on different screen sizes and provides a simple and clear navigation path to the user. Additionally, features provided CSS were experimented during development to identify features that provides a seamless user experience.

5.5 "Implementation" Class Diagram

Figure 5.2 Implemented Class Diagram



5.6 Overview of Software Components

An overview of the software components used, with their purposes, data structures and methods used, and pseudo code of algorithms are described below.

5.6.1 User Class

Represents users who obtains the services provided by the system, i.e. employers and gig workers. The system administrators are not represented by this class or any class in the class diagram.

Data structures and methods

- Email – the email address of the user who uses the system.
- Password – the password entered when signing up or logging into the system.
- Account Type – there are 2 types of user accounts in this system namely employer and worker.

Pseudo code of algorithms used

```
function signup (email, password, account type)
{
        hash_password = password_hash (password)
        save_values_to_database (email, hash_password, account type)

        if (account type = gig worker)
        {
                function signup_gigworker (skills, experience, availability)
                {
                        save_values_to_database (skills, experience, availability)
                }
        }

        return success
}
```

5.6.2 Gig Worker Class

Purpose: Represents gig workers who use the application to seek part-time work opportunities.

5.6.3 Employer Class

Purpose: Represents the employers who post jobs in the system and hire workers to get them done.

5.6.4 Job Posting Class

Purpose: Represents the jobs employers posted in the system, to which workers can apply.

5.6.5 Job Application Class

Purpose: Job applications are submitted by employees for the jobs that they prefer.

5.6.6 Message Class

Purpose: This class represents messages sent by workers to employers and messages sent by employers to workers.

5.7 Conclusions

During the implementation, the web application was structured in a manner to ensure smooth interaction between its components. Several technologies were reviewed, and the selected ones enhanced the user-friendliness, efficiency and robustness of the system. Implementation was interrupted at multiple occasions by several issues, and they were strategically handled to complete the development successfully. The classes determined during the design phase were subjected to changes during the implementation phase to fulfil the requirements of this web application. The choice of classes together with their data structures and methods resulted in a program that is scalable and efficient.

6 Chapter 6 Testing

6.1 Types of testing

Evaluating the quality of the web application and its components (International Software Testing Qualifications Board, no date) is important to make a judgement about the acceptability of the system and discover problems (Jorgensen, 2014). The following types of testing are conducted to ensure the quality of this Gig Work Marketplace web application.

- Black box testing examines the functionalities of applications without looking at the implementation specifics (Testsigma Engineering Team, no date). There are several types of black box testing:
  1) Functional testing to evaluate whether the web application satisfies the functional requirements outlined in the list of requirements (International Software Testing Qualifications Board, no date).
  2) Non-functional testing focuses on aspects such as scalability, performance, security and usability ensuring that applications fulfil security, performance and user experience considerations ((Testsigma Engineering Team, no date).
  3) Regression testing to identify the defects introduced in unchanged areas of the application (International Software Testing Qualifications Board, no date).
- White box testing evaluates the internal implementation of an application to ensure that internal paths of the code are structured correctly and optimized (Hamilton, 2024). There are 2 types of white box testing:
  1) Unit testing is done at each segment (unit) of code while the code is written, to ensure each segment of the code executes what it intends to do (Hamilton, 2024).
  2) Integration testing focuses on the interaction between frontend, backend and APIs (International Software Testing Qualifications Board, no date).
- Risk based testing are done by prioiritizing test scenarios with highest level of risk considering the tight timeline of the project (International Software Testing Qualifications Board, no date).
- Use case testing where testing is based on the execution of use cases outlined at the design phase of the project (International Software Testing Qualifications Board, no date).
- GUI testing is conducted to review the responsiveness and intuitivity of the graphical user interface (International Software Testing Qualifications Board, no date).
- End-to-end testing where busiess processes are executed from start to finish (International Software Testing Qualifications Board, no date).
- Exploratory testing at the end of the implementation based on the knowledge of the system and results of previous tests (International Software Testing Qualifications Board, no date).

6.2 Record of Black Box testing

Test cases derived from the essential functional requirements along with their black box test details are as follows.

Test case	Sign up to the application
Inputs provided (Data)	Email address, Password, Account Type, User Details
Test steps	1) Navigate to the 'Home' page. 2) Click the 'Sign Up' button on the menu. The system navigates to the 'Sign Up' page. 3) Enter the email, password, account type and user details. 4) Click 'Sign Up' button.
Output received (Result)	'Sign up successful' toast message is returned.

Test case	Log in to the application
Inputs provided (Data)	Email address, Password
Test steps	1) Navigate to the 'Home' page. 2) Click the 'Log in' button on the menu. The system navigates to the 'Log In' page. 3) Enter the email and the password. 4) Click 'Log In' button.
Output received (Result)	'Log in successful' toast message is returned.

Test case	Post a job
Inputs provided (Data)	Job details
Test steps	1) Navigate to the 'Home' page. 2) Log in to the system. 3) Click the 'Post Job' button on the menu. The system navigates the user to 'Post Job' page. 4) Enter the job details and click the 'Post Job' button.
Output received (Result)	'Job posted successfully' toast message is returned.

Test case	Search for a job
Inputs provided (Data)	Search keywords
Test steps	1) Navigates to the 'Home' page. 2) Click the 'Browse Jobs' button on the menu. The system navigates the user to the 'Browse Jobs' page. 3) Enter the search criteria.
Output received (Result)	Search results are displayed.

Test case	Apply for a job
Inputs provided (Data)	Not applicable
Test steps	1) Navigate to the 'Home' page. 2) Log in to the system. 3) Search for jobs. 4) Click the 'Apply Now' button on a selected job.
Output received (Result)	'Application submitted successfully' toast message is returned.

Test case	Message employer
Inputs provided (Data)	Not applicable
Test steps	1) Navigate to the 'Home' page. 2) Log in to the system. 3) Select a job posting. 4) Click the message icon on the selected job. The system directs the user to the 'Message' page. 5) Enters the message and click 'Send'.
Output received (Result)	'Message sent' toast message is returned.

6.3 Other Types of Testing Carried out

- Unit testing
- Use case testing
- GUI testing
- End-to-end testing
- White box testing
- Regression testing

6.4 Outstanding Bugs

This subsection describes missing functionalities, and discrepancies between requirements and implementation identified from black box and other types of testing. These defects are not corrected yet and require additional technical investigations and corrections. However, these defects are not flow stoppers, and hence, have a moderate impact on the system functionality. Addressing these issues will enhance the overall quality of the system and ensure its reliability.

Bug title	The status of the jobs are not transitioned based on the occurrence of the respective events.
Bug description	When a job is posted and visible to the users, the job status should be 'New'. This works as expected. However, the below statuses are not updated when the respective event occurs. • **Applications Received** - When applications are received and the employer can start reviewing, the status should be 'Applications Received'. • **Applicant Selected** - When an applicant is chosen for the job, the status should transition to 'Applicant Selected'. • **In Progress** - When the worker starts work, the job status should be 'In Progress'. • **Completed** - When the worker finishes the job, the status should change to 'Completed'.
Steps to reproduce	1) Log in from an employer's profile and post a job. Note that the status of the job is 'New'. 2) Log in from a worker's profile and apply for that job. Note that the job status does not change to 'Applications Received'. It remains in 'New' status. 3) Switch back to employer's profile and select the applicant. Note

	that the job status does not change to ‘Applications Selected. It remains in ‘New’ status. 4) When the worker starts or finishes the job, note that there is no command to mark the occurrence of those events to change statuses.
Bug severity	Employers get confused about the progress of their task.
Priority (out of 5)	3

Bug title	The status of the applications are not transitioned based on the occurrence of the respective events.
Bug description	When a job application is submitted, the application status should be ‘Submitted’. This works as expected. However, the below statuses are not updated when the respective event occurs. • **Under Review** - When applications are received and the employer starts reviewing, the status should be ‘Under Review’. • **Selected** - When an applicant is chosen for the job, the status should transition to ‘Selected’. • **Not Selected** - When the applicant is not chosen for the job, the job status should be ‘Not Selected. • **Finalized** - When the worker accepts the job offer, the status should change to ‘Finalized. • **Withdrawn** – When the applicant withdraws the application, the status should be ‘Withdrawn’.
Steps to reproduce	1) Log in from a worker’s profile and apply for a job. Note that the status of the application is ‘Submitted’. 2) Log in from a employer’s profile and pick that application for review. Note that the job status does not change to ‘Under Review’. It remains in ‘Submitted’ status. 3) When the employer selects an applicant, note that there is no command to mark the occurrence of that event (and the subsequent events) to change statuses.
Bug severity	Gig workers get confused about the progress of their job applications.
Priority (out of 5)	3

7 Chapter 7 Conclusions

7.1 Introduction and Summary

This chapter summarizes the project outcomes and key takeaways of the development process. The purpose of this project was to develop a web application that connects on-demand workers with employers who wants to get work done. This report detailed the highlights of each stage of the project, from requirement gathering, designing, implementation to testing. This chapter discusses how the project aims and objectives were achieved, to which extent the user requirements were fulfilled, and what areas can be considered for future developments.

7.2 Review of Project Aims and Objectives

This subsection describes how each project aim and objective was achieved by developing the Gig Work Marketplace web platform.

Aim 1 (PA1) – Design and develop a user friendly, simple and intuitive digital platform that enables seamless task outsourcing for employers.

The user interface of this web application is simple and easy to navigate, the UI elements are self explanatory and visually appealing, the layout is structured in accordance with internationally accepted UI designing guidelines. The application provides the flexibility to post jobs with detailed information, update the job postings as necessary and track the job status. Several existing systems were reviewed to gather requirements to design the user interface. Black box testing was conducted to ensure the user friendliness of the application.

The objectives for PA1 are as follows.

- (PO1.1) Implement a responsive user interface that allows employers to easily post jobs.
- (PO1.2) Develop a database to manage employer data and job postings securely and efficiently.

Aim 2 (PA2) – Design and develop a user friendly, simple and intuitive digital platform that facilitates efficient job seeking for part-time workers.

The search functionality is extensive and optimized to cover a wide range of search parameters. Gig workers can find jobs that match their criteria and apply for the job from the search results page itself, without navigating to other pages. The manual steps to execute a function is minimized to enhance the overall user experience.

The objectives for PA2 are as follows.

- (PO2.1) Implement a responsive user interface that allows part-time workers to easily search and apply for jobs.
- (PO2.2) Develop a database to manage employee data and applications securely and efficiently.

Aim 3 (PA3) - Enhance communication between employers and part-time workers to improve task outsourcing.

Gig workers can request more details on tasks, negotiate terms of employement and provide frequent updates to employers using the built-in messaging system in the application. Workers can also communicate with gig workers using this messaging system to offer gigs and provide additional details on jobs. Additionally, both employers and gig workers can receive real-time updates on jobs via the notification system implemented. These features enhance the overall work allocation process by avoiding the dependency and hassle of communication via other applications.

The objectives for PA3 are as follows.

- (PO3.1) Implement an in-app messaging system to facilitate seamless communication between employers and part-time workers.
- (PO3.2) Develop a notification system to notify users about application statuses and messages.

7.3 Review of Requirements

This subsection explains the requirements that were met at the completion of the project and what requirements were deprioritized to be considered at a later iteration of the software.

7.3.1 Functional Requirements

Fulfilled requirements - Essential

1) User Registration and Authentication
- Employers and workers can create accounts.
- Secure login and password maintenance.

2) Profile Management
- Employers can create and manage company or individual profiles.
- Workers can create and update personal profiles, including skills and experience.

3) Job Posting and Management
- Employers can post jobs with details such as job description, requirements, and budget.
- Employers can manage and edit job postings.

4) Job Search and Application
- Gig workers can search for jobs based on various filters (location, pay, job type).
- Workers can apply for jobs directly through the platform.

5) Communication Tools
- In-app messaging between employers and workers.
- Notifications for job postings, applications, and messages.

6) Ratings and Reviews
- Employers can rate and review workers.

Unfulfilled requirements - Essential

1) Application Tracking
- Employers can track applications and manage the hiring process.
- Workers can track the status of their applications.

Unfulfilled requirements - Desirable

1) Advanced Search Filters
- More detailed filters for job searches (e.g., remote/on-site, date posted).

2) Calendar Integration
- Integration with calendar apps for scheduling interviews and job start dates.

7.3.2 Non-Functional Requirements

Unfulfilled requirements - Desirable

1) Performance
- Fast load times and responsive design.
- Scalability to handle a large number of users.

2) Security
- Data encryption for sensitive user information.
- Regular security audits and updates.

7.3.3 User Interface Requirements

Fulfilled requirements - Essential

1) Usability
- Intuitive and user-friendly interface.

7.4 Further Work and Improvements

This subsection lists other areas that can be explored to further improve this Gig Work Marketplace web application. Improvements in these areas will enhance the usability of the system, its scalability and the overall performance.

- Advanced analytics to provide insights into job performance, completion, selection trends, system usage and the user activity.
- Customizable themes that allow users to change the web applications appearance for better user experience.
- Advanced security features to protect user data from unauthorized access and improve trust.
- Introducing a mobile app to provide better user experience and accessibility.

7.5 Conclusion

The purpose of building an application that connects part-time employees with employers was achieved by this development project. This project delivered a user-friendly web application that enables efficient job posting, comprehensive job search and application submitting, provides real-time updates on jobs to users and maintaining effective communication between users. There were several challenges during the execution of the project, which posed risks to the completion of the project. However, those challenges were successfully addressed, and risks were mitigated. One of the key takeaways is the need for continuous learning to ensure the project incorporates the most efficient technologies, that helps the project to stay on track. Additionally, the significance of closely monitoring project outcomes helped in delivering this project within the set timeline. All in all, this development project was a learning experience that highlighted the importance of strategic thinking, continuous commitment and resilience. The skills and experience gained from this project will be useful for future projects.

Bibliography

Abramowski, N. (2023). What is CSS? A Beginner's Guide. *CareerFoundry*. Available from https://careerfoundry.com/en/blog/web-%20%20development/what-is-css/ [Accessed 23 March 2025].

Airtasker (no date). About us. *Airtasker*. Available from https://www.airtasker.com/au/about/ [Accessed 30 March 2025].

AppMaster (2023). The Role of HTML in Website Development. *AppMaster*. Available from https://appmaster.io/blog/html-website-development [Accessed 23 March 2025].

Apple Inc. (no date). Work with HTML documents in TextEdit on Mac. *Apple Inc*. Available from https://support.apple.com/en-gb/guide/textedit/txted0b6cd61/1.20/mac/15.0 [Accessed 16 February 2025].

De Stefano, V. & Aloisi, A. (2018). *European legal framework for 'digital labour platforms'*. Luxembourg: Publications Office of the European Union. Available from https://www.researchgate.net/publication/330683831_European_legal_framework_for_digital_labour_platforms [Accessed 03 February 2025].

GigSmart (no date). About us. *GigSmart*. Available from https://gigsmart.com/about-us/?& [Accessed 01 April 2025].

Hamilton, T. (2024). White Box Testing – What is, Techniques, Example & Types. *Guru99*. Available from https://www.guru99.com/white-box-testing.html [Accessed 23 March 2025].

Indeed (no date). About Indeed. *Indeed*. Available from https://www.indeed.com/about [Accessed 01 April 2025].

International Software Testing Qualifications Board (no date). Glossary. *ISTQB.* Available from https://glossary.istqb.org/en_US/search?term=testing&exact_matches_first=true [Accessed 23 March 2025].

Jorgensen, P. C. (2014). *Software Testing: A Craftsman's Approach,* 4th ed. Boca Raton: Taylor & Francis Group. Available from https://malenezi.github.io/malenezi/SE401/Books/Software-Testing-A-Craftsman-s-Approach-Fourth-Edition-Paul-C-Jorgensen.pdf [Accessed 23 March 2025].

Joshi, M. (2023). A detailed guide on JavaScript Web Development. *BrowserStack*. Available from https://www.browserstack.com/guide/javascript-web-development [Accessed 23 March 2025].

MDN Web Docs (no date). What software do I need to build a website?. *MDN Web Docs*. Available from https://developer.mozilla.org/en-US/docs/Learn_web_development/Howto/Tools_and_setup/What_software_do_I_need [Accessed 16 February 2025].

Ray, B., Sengupta, A. and Varma, A. (2024). The gig verse: building a sustainable future. *International Journal of Organizational Analysis*, 32 (*10*), 2275-2298. Available from https://www.emerald.com/insight/content/doi/10.1108/ijoa-08-2023-3946/full/pdf?title=the-gig-verse-building-a-sustainable-future [Accessed 26 January 2025].

Taskrabbit (no date). About us. *Taskrabbit*. Available from https://www.taskrabbit.com/about [Accessed 30 March 2025].

Testsigma Engineering Team (no date). Black Box Testing : What it is, Types , Techniques & Examples. *Testsigma*. Available from https://testsigma.com/guides/black-box-testing/ [Accessed 23 March 2025].

Upwork (no date). About us: The world's work marketplace. *Upwork*. Available from https://www.upwork.com/about [Accessed 30 March 2025].