



**Unity University**  
**Faculty of Engineering and Technology**  
**Department of Computer Science and MIS**

**Job Posting and Searching Platform**

**Final Project Document**

**Submitted to department of Computer Science in partial fulfillment of the  
requirements for the degree of Bachelor of Science in Computer Science**

**Developed by:**

- Dagmawi Alem – UU87403R
- Meba Mekuanent – UU87687R
- Melat Girmay – UU87459R
- Tihitina Chane – UU88457R

**Advisor:**

**Mr. Mezgebe Manaye**

**July 17, 2024**

**Addis Ababa, Ethiopia**

**Acknowledgment**

We would like to express our deepest gratitude to everyone who helped us complete this project. A special thanks to our project supervisor, Mr. Mezgebe Manaye, for his guidance, encouragement, and valuable feedback. Your support and expertise have been essential in making this project successful.

We are also very grateful to our instructors and the Department of Computer Science at Unity University for providing us with the basic knowledge and skills needed for this project. The supportive academic environment and resources have been crucial for our learning and development.

## **Table of Contents**

### **1. Introduction**

- 1.1 Background Information
- 1.2 Statement of the Problem
- 1.3 Objectives
  - 1.3.1 General Objective
  - 1.3.2 Specific Objectives
- 1.4 Scope of the project
- 1.5 Tools and Methodologies
  - 1.5.1 Data Collection Methodologies
  - 1.5.2 System Development Methodology
  - 1.5.3 Development Tools
- 1.6 Beneficiaries
- 1.7 Schedule

### **2. Project Management**

- 2.1 Introduction
- 2.2 Project Planning - Work Breakdown Structure (WBS)
- 2.2 Resource Planning
  - 2.2.1 Human Resource Planning
  - 2.2.2 Material/Equipment Planning
- 2.3 Resource Planning
  - 2.3.1 Human Resource Financial Plan
  - 2.3.2 Material/Equipment Financial Plan
- 2.4 Select Project Model
  - 2.4.1 Human Resource Financial Plan

- 2.4.2 Material / Equipment Financial Plan
- 2.5 Team Organization
- 2.6 Process Model
  - 2.6.1 Requirements Gathering and Analysis
  - 2.6.2 System Architecture and Design
  - 2.6.3 Development and Implementation
  - 2.6.4 Testing and Quality Assurance
  - 2.6.5 Deployment and Release
  - 2.6.6 Maintenance and Support
  - 2.6.7 Project Management and Documentation
- 2.7 Risk MMM Plan
  - 2.7.1 Risk Items Table
  - 2.7.2 RMMM Plan

### **3. System Analysis**

- 3.1 Introduction
- 3.2 Current System Overview
  - 3.2.1 Job Seeker Experience
  - 3.2.2 Employer Perspective
  - 3.2.3 Transition from Education to Employment
  - 3.2.4 Technological Limitations
- 3.3 Proposed System Overview
  - 3.3.1 Functional Requirements
  - 3.3.2 Non-Functional Requirements
- 3.4 System Models - Requirement Determination
  - 3.4.1 Essential Use Case Modeling
    - 3.4.1.1 Use Case Diagram
    - 3.4.1.2 Use Case Documentation
  - 3.4.2 Essential UI Prototype
  - 3.4.3 User Interface Flow Diagram
  - 3.4.4 Supplementary Specifications
    - 3.4.4.1 Business Rules
    - 3.4.4.2 Constraints
    - 3.4.4.3 Change Case
- 3.5 System Models – Analysis
  - 3.5.1 System Use Case Modeling
    - 3.5.1.1 Use Case Diagram
    - 3.5.1.2 Use Case Documentation

- 3.5.2 Sequence Diagram
- 3.5.3 Activity Diagram

#### **4. System Design**

- 4.1 Introduction
- 4.2 Design Goals
- 4.3 Design Trade offs
- 4.4 Subsystem Decomposition
- 4.5 Design Phase Models
  - 4.5.1 Class Modeling
  - 4.5.2 Persistent Model
    - 4.5.2.1 Mapping Class Diagram to Relation
    - 4.5.2.2 Normalization
  - 4.5.3 User Interface Design
  - 4.5.4 Deployment Diagram
  - 4.5.5 Network Design

#### **5. Implementation**

- 5.1 Introduction
  - 5.1.1 Database Implementation
    - 5.1.1.1 Database Design
    - 5.1.1.2 Data Handling and Storage
    - 5.1.1.3 Database Operations
    - 5.1.1.4 Data Security and Access Control
  - 5.1.2 Interface and Business Logic Implementation
    - 5.1.2.1 Frontend Interface Design
    - 5.1.2.2 User Interaction and Experience
- 5.2 Sample Code

#### **6. Evaluation**

- 6.1 Conclusion
- 6.2 Recommendation

## **Chapter One - Introduction**

### **1.1 Background Information**

In the dynamic landscape of education and employment, the need for an efficient and accessible Job Posting and Searching Platform has become increasingly evident. As educational institutions strive to bridge the gap between academia and the professional world, students and job seekers face the challenge of navigating a complex and competitive job market.

The Job Posting and Searching Platform addresses this challenge by providing a dedicated space for employers and job seekers to connect seamlessly. Recognizing the diverse needs of both parties, the platform caters to a wide range of job listings, encompassing internships, part-time positions, and full-time employment opportunities. This initiative is poised to serve as a valuable resource for students and professionals, offering a unified and user-friendly interface to streamline the job search and recruitment processes.

Over the years, the dynamics of job markets have evolved significantly. The traditional methods of job hunting and recruitment are being reshaped by technological advancements, creating opportunities for more efficient and inclusive systems. The Job Posting and Searching Platform is positioned at the intersection of these changes, leveraging technology to enhance the overall experience for both employers and job seekers.

While the platform extends its reach beyond the university community, it is anchored in a university-centric approach. Recognizing the unique needs and challenges faced by students transitioning from education to employment, the platform aims to simplify the recruitment process, creating a bridge that facilitates a smoother transition from the academic realm to the workforce.

Beyond being a mere job portal, the platform is designed to be a catalyst for career development. By providing a centralized hub for job opportunities, the platform equips job seekers with the tools to explore positions that align with their skills and career goals.

In essence, the Job Posting and Searching Platform is not just a technological solution but a strategic response to the evolving demands of the contemporary job market. This chapter sets the stage for an in-depth exploration of the platform's development, functionality, and the impact it aims to make on the education-to employment journey.

## **1.2 Statement of the Problem**

Despite the abundance of job opportunities in today's rapidly evolving job market, the process of connecting job seekers with suitable employers remains a challenge. Various pain points exist within the current employment ecosystem, necessitating the development of a dedicated Job Posting and Searching Platform. The following key issues underscore the need for such a platform:

- ◆ **Fragmented Job Search Ecosystem:** The existing job search landscape is often fragmented, with job seekers navigating multiple platforms, websites, and networks to find suitable opportunities. This fragmentation not only consumes valuable time and effort but also hinders the efficiency of the job search process for both employers and prospective employees.
- ◆ **Inefficiencies in Recruitment Processes:** Employers, on the other hand, encounter challenges in efficiently reaching potential candidates. Traditional recruitment methods may lack the precision and speed required to match qualified individuals with available positions. This inefficiency can result in prolonged hiring processes and missed opportunities for both employers and job seekers.
- ◆ **Lack of a Seamless Transition from Education to Employment:** The transition from education to employment represents a critical phase in an individual's career journey. However, this transition is often marred by uncertainties, as students grapple with finding suitable opportunities and employers struggle to identify talent that aligns with their needs. The absence of a dedicated platform exacerbates these challenges, hindering a seamless progression from academia to the workforce.
- ◆ **Technological Gaps in Existing Solutions:** Current job platforms may not fully leverage the potential of emerging technologies to enhance the user experience for both job seekers and employers. This technological gap limits the effectiveness of these platforms in delivering tailored and user-centric solutions, impeding the optimization of the job search and recruitment processes.

Considering these challenges, the development of a comprehensive Job Posting and Searching Platform becomes imperative. This platform aims to address these issues, providing a unified and technologically advanced solution that facilitates a more efficient and inclusive job market for all stakeholders. The subsequent chapters delve into the specific objectives, scope, and methodologies employed in creating this transformative platform.

## **1.3 Objectives**

### **1.3.1 General Objective**

The overarching goal of the Job Posting and Searching Platform is to establish a robust and user-centric online ecosystem that seamlessly connects job seekers with employers, fostering efficient and inclusive employment opportunities. By leveraging technology and innovation, the platform aims to transform the traditional job search and recruitment processes, contributing to a smoother transition from education to the workforce.

### **1.3.2 Specific Objectives**

- ◆ **Create a User-Friendly Platform:** Develop an intuitive and accessible user interface that caters to the diverse needs of both job seekers and employers, ensuring a positive and efficient user experience.
- ◆ **Diversify Job Listings:** Enable employers to post a comprehensive range of job listings, including internships, part-time positions, and full-time employment opportunities, to cater to the varied needs of the workforce.
- ◆ **Implement Advanced Search and Matching Algorithms:** Integrate cuttingedge search and matching algorithms to enhance the precision and relevance of job recommendations, ensuring that job seekers are connected with opportunities that align with their skills and aspirations.
- ◆ **Extend Reach Beyond the University Community:** Establish partnerships and outreach strategies to expand the platform's user base beyond the university community, fostering connections between a broader range of job seekers and employers in the wider job market.
- ◆ **Streamline Recruitment Processes:** Simplify and expedite the recruitment processes for employers, providing tools that facilitate efficient candidate screening, communication, and selection.

- ◆ **Incorporate Career Development Resources:** Integrate features and resources that go beyond job listings, offering career development tools, guidance, and resources to empower job seekers in advancing their professional trajectories.
- ◆ **Ensure Data Security and Privacy:** Implement robust security measures to safeguard user data and privacy, instilling confidence in both job seekers and employers to actively engage with the platform.
- ◆ **Promote Inclusivity and Diversity:** Foster an inclusive environment by promoting equal opportunities for individuals from diverse backgrounds, ensuring that the platform contributes to building a more equitable job market.
- ◆ **Continuous Improvement and Adaptation:** Establish mechanisms for continuous improvement, incorporating user feedback and staying abreast of technological advancements to adapt the platform to evolving needs and trends in the job market.

These specific objectives collectively contribute to the attainment of the general objective, setting the foundation for a Job Posting and Searching Platform that not only meets the immediate needs of job seekers and employers but also anticipates and adapts to the changing dynamics of the employment landscape. The subsequent sections of this document delve into the scope, tools, methodologies, beneficiaries, and schedule that will guide the successful development and implementation of this transformative platform.

## **1.4 Scope of the Project**

The scope of the Job Posting and Searching Platform encompasses the development and implementation of a comprehensive online ecosystem that facilitates the connection between job seekers and employers. This project aims to address the challenges inherent in the current job search and recruitment processes, providing a solution that transcends the limitations of existing platforms. The scope is defined across several dimensions:

- ◆ **User Demographics:** The platform primarily targets students, recent graduates, and professionals seeking employment opportunities. While the initial focus is on the university community, the platform's design allows for scalability, enabling outreach to a broader audience beyond the academic realm.
- ◆ **Job Types:** The platform accommodates a diverse range of job types, including internships, part-time positions, and full-time employment opportunities. By offering this variety, the platform caters to the multifaceted needs of job seekers at different stages of their careers.



- ◆ **Industry Verticals:** The platform is designed to be industry-agnostic, facilitating job postings across various sectors such as technology, finance, healthcare, and more. This approach ensures that job seekers from different fields can access relevant opportunities, contributing to the platform's inclusivity.
- ◆ **Features and Functionalities:** The platform incorporates features that go beyond basic job listings. Advanced search and matching algorithms, career development resources, and streamlined recruitment processes are integral components. The scope includes the integration of tools that enhance the overall user experience, providing value-added services for both job seekers and employers.
- ◆ **Technological Infrastructure:** The project involves the development of a scalable and secure technological infrastructure to support the platform's functionality. The use of modern technologies and frameworks will be employed to ensure a robust and responsive system capable of handling the platform's growth.
- ◆ **Exclusions:** The scope of the project explicitly excludes the provision of direct employment services. The platform serves as a facilitator, connecting job seekers with employers, but does not engage in the actual hiring process. Additionally, the project does not encompass specific job training programs.
- ◆ **Future Expansion:** While the initial focus is on creating a strong foundation, the project acknowledges the potential for future expansion. The platform's architecture and design will allow for the incorporation of additional features, services, and partnerships to adapt to evolving trends in the job market.

In summary, the scope of the Job Posting and Searching Platform is comprehensive, covering a wide range of users, job types, industries, and geographies. The intention is to create a versatile and adaptive platform that addresses the current challenges in the job market while remaining poised for future growth and enhancement. The subsequent chapters will delve into the tools, methodologies, beneficiaries, and schedule that will guide the successful execution of this transformative project.

## **1.5 Tools and Methodologies**

### **1.5.1 Data Collection Methodologies**

The data collection methodologies for the Job Posting and Searching Platform project will be systematically executed to ensure the acquisition of relevant and accurate information. The following approaches will be employed:

- ◆ **Analysis of Existing Platforms:** Evaluate and analyze existing job posting and searching platforms to identify strengths, weaknesses, opportunities, and threats. This analysis will inform the development of features that address current gaps in the market.

### **1.5.2 System Development Methodology**

The system development methodology chosen for this project is the Agile methodology. Agile offers a flexible and iterative approach, allowing for incremental development and adaptation to changing requirements. The development process will be divided into sprints, each focusing on specific features and functionalities. Key elements of the Agile methodology include:

- ◆ **Iterative Development:** Embrace an iterative development approach, allowing for continuous feedback from stakeholders. This ensures that the platform evolves based on real-time user input and market demands.
- ◆ **Cross-Functional Teams:** Form cross-functional development teams with expertise in areas such as front-end and back-end development, database management, user experience (UX) design, and quality assurance. This facilitates a holistic approach to system development.
- ◆ **Regular Sprint Reviews:** Conduct regular sprint reviews to assess progress, gather feedback, and make necessary adjustments. This ensures that the development process remains aligned with the project objectives and timelines.
- ◆ **Continuous Integration and Testing:** Implement continuous integration and testing practices to maintain the stability and reliability of the platform throughout the development process. Automated testing tools will be employed to streamline the testing phase.
- ◆ **Adaptability to Change:** Emphasize adaptability to changing requirements and market dynamics. Agile allows for the incorporation of new features or modifications based on user feedback and emerging trends in the job market.

### 1.5.3 Development Tools

The development tools carefully selected for the implementation of the Job Posting and Searching Platform are tailored for efficiency, seamless collaboration, and alignment with the chosen Waterfall methodology. These tools are chosen to ensure a well-structured and systematic development process. The primary development tools include:

#### Programming Languages:

- Frontend:
  - **HTML, CSS, and JavaScript:** Utilized for creating a dynamic and responsive user interface, ensuring an engaging and user-friendly experience.
  - **React:** Employed for building the frontend with reusable components, providing a seamless and efficient user interface.
- Backend:
  - **Node.js with Express Framework:** Leveraged for efficient server-side development. Node.js offers asynchronous event-driven JavaScript runtime, and Express provides a robust framework for building scalable web applications.

#### Database Management:

- MySQL:
  - Utilized for its reliability and well-established relational database management capabilities. MySQL ensures data consistency and integrity throughout the development lifecycle.

#### Version Control:

- GitLab:
  - Employed for version control and collaborative development. GitLab provides a comprehensive set of features for code management, collaboration, and tracking changes.

## 1.6 Beneficiaries

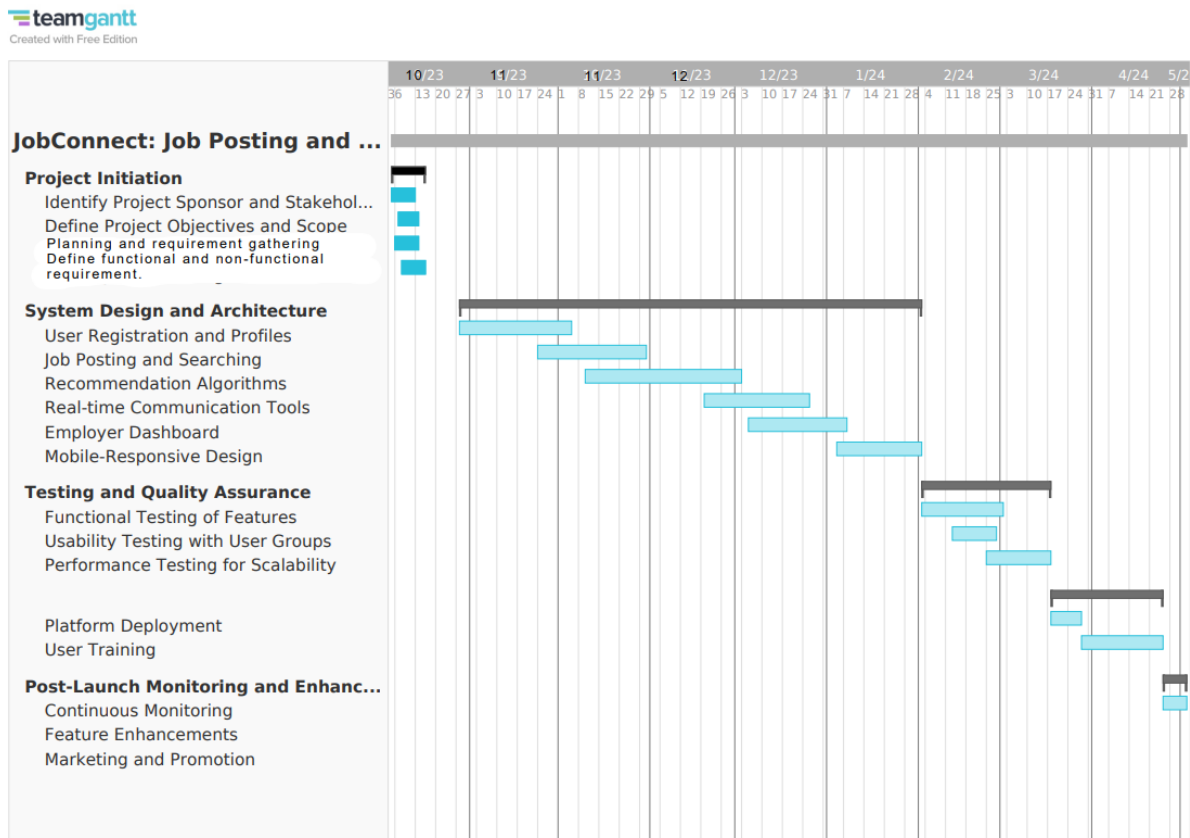
The Job Posting and Searching Platform is designed to cater to a diverse set of beneficiaries, encompassing various stakeholders in the education-to-employment ecosystem. The primary beneficiaries include:

- Job Seekers:
  - Students and Recent Graduates: Access to a centralized platform for discovering a wide range of job opportunities, including internships, part-time positions, and full-time employment. The platform provides tools for enhancing their job search, connecting with potential employers, and advancing their career development.
- Employers:
  - Companies and Organizations: Gain access to a pool of qualified and diverse candidates. The platform streamlines the recruitment process, allowing employers to post job listings, connect with potential hires, and efficiently manage the hiring workflow. It provides a platform for fostering talent acquisition and building a strong workforce.
- Universities and Educational Institutions:
  - Career Services and Counselors: Benefit from a resource that facilitates the transition from education to employment for their students. The platform serves as a valuable tool for career counselors to guide students in exploring job opportunities aligned with their skills and aspirations.
- Recruitment Agencies:
  - Agencies and Headhunters: Find a centralized source for identifying potential candidates. The platform streamlines the candidate search process, enabling recruitment agencies to connect with individuals seeking employment opportunities.
- Industry-Specific Associations:
  - Professional and Industry Groups: Access a platform that facilitates networking and career development within specific industries. Industry associations can leverage the platform to promote opportunities, share insights, and contribute to the professional growth of their members.
- Government Employment Services:

- Public Employment Services: Utilize the platform as a supplementary resource for connecting job seekers with employment opportunities. Government agencies can collaborate with the platform to enhance the reach and efficiency of their job placement services.
- Individual Professionals:
  - Freelancers and Part-Time Job Seekers: Explore a variety of opportunities beyond traditional full-time positions. The platform accommodates individuals seeking freelance work or part-time engagements, providing a diverse range of employment options.
- Career Development Organizations:
  - Nonprofit Organizations and Career Development Centers: Collaborate with the platform to support individuals in their career development journey. The platform can serve as a tool for nonprofits and career development centers to extend their reach and impact.

The Job Posting and Searching Platform is envisioned as a collaborative ecosystem that benefits a wide array of stakeholders by fostering efficient connections, simplifying processes, and contributing to the overall development of careers and workforce capabilities.

## 1.7 Schedule



## Chapter Two - Project Management

### 2.1 Introduction

This chapter serves as a guide to the project management intricacies associated with the creation and execution of the Job Posting and Searching Platform. It navigates through crucial facets such as project planning, resource allocation, financial considerations, team structuring, process modeling, and risk management. These elements collectively contribute to the efficient development and successful implementation of the platform.

### 2.2 Project Planning – WBS

Level	Code	Activities	Duration (Days)	Dependency	Status

1	1	Project Initiation	15	-	Completed
2	1.1	Form Project Team	1	-	Completed
2	1.2	Define Project Objectives	2	1.1	Completed
2	1.3	Define Project Scope	3	1.2	Completed
2	1.4	Gather Project Requirements	5	1.1, 1.2	Completed
2	1.5	Develop Project Schedule	2	1.4	Completed
2	1.6	Identify Project Stakeholders	2	1.4	Completed
1	2	Project Management	15	-	In Progress
2	2.1	Develop Work Breakdown Structure (WBS)	1	1.6	In Progress
2	2.2	Resource Planning	5	2.1	In Progress
3	2.2.1	Human Resource Planning	2	2.1	In Progress

3	2.2.2	Material/Equipment Planning	3	2.1	In Progress
2	2.3	Financial Planning	5	2.2	In Progress
3	2.3.1	Human Resource Financial Plan	2	2.2.1	In Progress
3	2.3.2	Material/Equipment Financial Plan	3	2.2.2	In Progress
2	2.4	Select Project Model	1	2.2	In Progress
2	2.5	Organize Project Team	1	2.2	In Progress
2	2.6	Risk Analysis	2	2.2, 2.3	In Progress
3	2.6.1	Identify Risks	1	2.2, 2.3	In Progress
3	2.6.2	Develop Risk Mitigation and Management (RMMM)	1	2.2, 2.3	In Progress
1	3	System Analysis	45	-	In Progress



2	3.1	Analyze Current System	7	-	In Progress
2	3.2	Define Proposed System	8	3.1	In Progress
3	3.2.1	Define Functional Requirements	4	3.2	In Progress
3	3.2.2	Define Non-Functional Requirements	4	3.2	In Progress
2	3.3	Requirement Determination	20	3.2	In Progress
3	3.3.1	Design Use Case Model	3	3.3	In Progress
3	3.3.2	Design UI Prototype	7	3.3.1	In Progress
3	3.3.3	User Interface Flow Diagram	3	3.3.1	In Progress
3	3.3.4	Define Specifications, Business Rules, Constraints, and Change	7	3.3	In Progress

		Control Mechanisms			
2	3.4	System Model Analysis	10	3.3	In Progress
3	3.4.1	Develop Sequence Diagram	5	3.4	In Progress
3	3.4.2	Develop Activity Diagram	5	3.4	In Progress
1	4	System Design	30	-	In Progress
2	4.1	Define Design Goals	5	-	In Progress
2	4.2	Subsystem Decomposition	5	4.1	In Progress
2	4.3	Design Phase Model	20	4.2	In Progress
3	4.3.1	Develop Class Diagram	3	4.3	In Progress
3	4.3.2	Develop Database Design including Normalization	7	4.3	In Progress

3	4.3.3	Develop Deployment Diagram	5	4.3	In Progress
3	4.3.4	Develop Network Design	5	4.3	In Progress
1	5	Implementati on and Testing	80	-	In Progress
2	5.1	Coding	60	-	In Progress
3	5.2	Final Software Design	10	5.1	In Progress
2	5.3	Testing	10	5.2	In Progress
1	6	Evaluation	15	-	In Progress
2	6.1	Conclusion	7	-	In Progress

## 2.3 Resource Planning

Resource planning is a critical component of project management essential for identifying, allocating, and effectively managing the resources required to ensure the successful execution of the Job Posting and Searching Platform project.

### 2.3.1 Human Resource Planning

Identifying Project Team Roles and Responsibilities:

Team Member	Role	Responsibilities
<b>Project Manager (Melat Girmay)</b>	Oversee the entire project, ensuring timely completion within scope.	Develop and manage detailed project schedules and work plans.

<b>Backend Developer (Dagmawi Alem)</b>	Responsible for server-side development and integration.	Develop and maintain server-side logic to support the platform.
<b>Database Administrator (DBA) (Thitina Chane)</b>	Manage the database infrastructure and performance.	Design, implement, and maintain the database schema ensuring security and availability.
<b>UI/UX Designer (Meba Mekuanent)</b>	Design intuitive user interfaces and enhance user experience.	Ensure user-centric design that meets project requirements and enhances usability.
<b>Frontend Developer (Melat Girmay)</b>	Focus on creating responsive and dynamic user interfaces.	Develop and maintain the client-side of the platform using HTML, CSS, JavaScript, and React.

This structured approach ensures clarity in roles and responsibilities, facilitating smooth coordination and collaboration among team members throughout the project lifecycle.

### 2.3.2 Material / Equipment Planning

<b>Requirement Category</b>	<b>Item</b>	<b>Specifications/Purpose</b>
<b>Hardware Requirements</b>	<b>Networking Equipment</b>	Mobile data routers, Wi-Fi access points.
		Facilitate secure communication between servers and user devices.
	<b>Workstations</b>	High-performance desktops or laptops with a minimum of 8GB RAM and modern processors.
		Provide individual computing environments for developers and designers to contribute effectively.
<b>Software Requirements</b>	<b>Integrated Development</b>	Support collaborative and efficient coding practices for developers.

	<b>Environment (IDE) - VS Code</b>	
	<b>Database Management System (DBMS) - MySQL Workbench</b>	Manage, design, and interact with the MySQL database seamlessly.
<b>Office Supplies</b>	<b>Notebooks, Papers, and Pens</b>	Facilitate collaborative discussions, brainstorming sessions, and task planning.
	<b>Notebooks and Stationery</b>	Support individual note-taking, idea generation, and documentation during meetings and development phases.

## 2.4 Financial Planning

In this document, we're going to look at the different expenses we expect for the project. By understanding and planning for these costs, we can make sure that the project goes smoothly without any financial surprises. We'll go through different categories, like development, operational needs, marketing efforts, and other unexpected expenses, to come up with a total estimate of how much the project might cost.

### 2.4.1 Human Resource Financial Plan

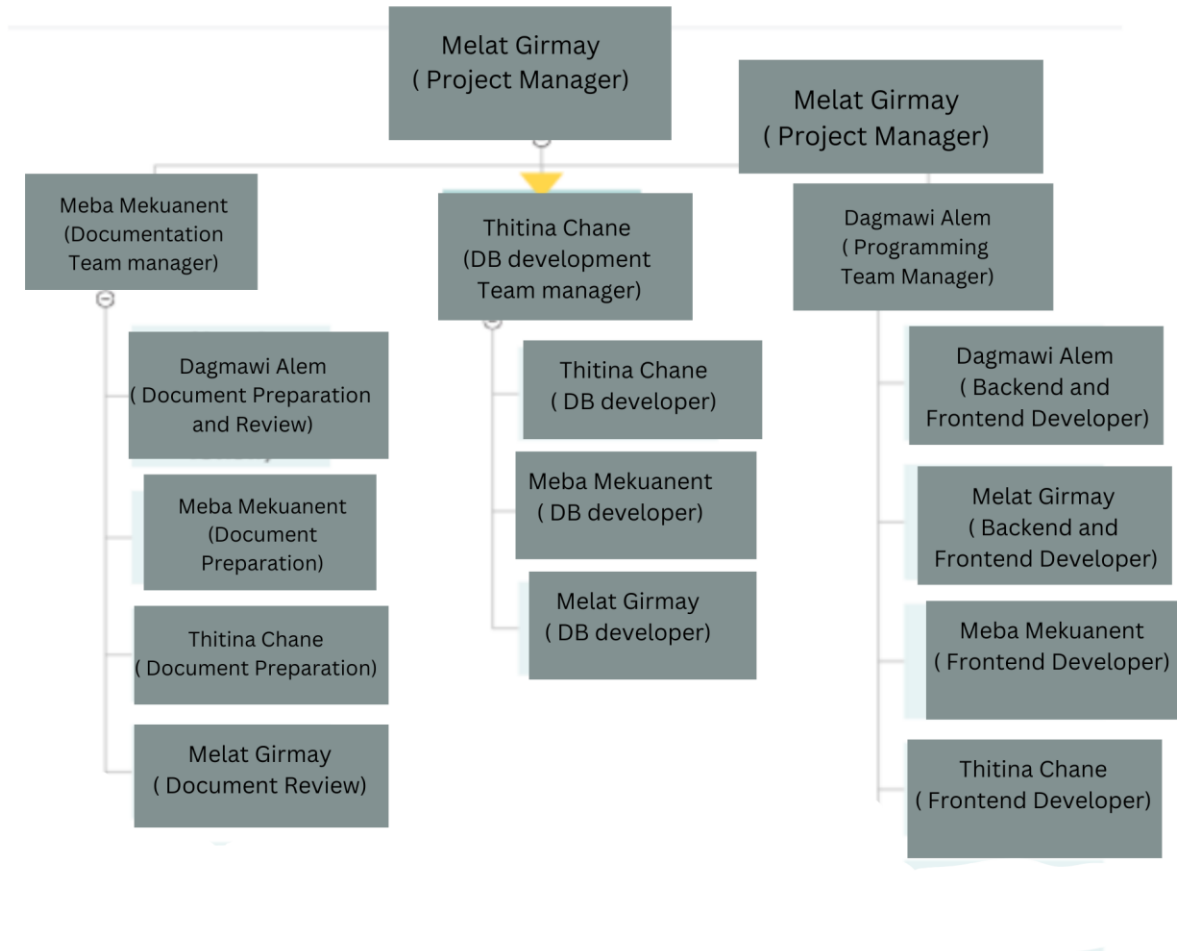
The people working on this project won't be paid because they're doing it as part of their graduation project. This is a way for them to gain experience and learn while working on something valuable.

### 2.4.2 Material / Equipment Financial Plan

<b>Section</b>	<b>Item</b>	<b>Purpose</b>	<b>Estimated Cost (Birr)</b>
<b>2.4.2 Material/ Equipment Financial Plan</b>	<b>Mobile Data Costs</b>	Ensure continuous online accessibility of the project.	500

	<b>Operational Costs</b>		
	Printing of Documentation	Produce necessary printed documents for project documentation and presentation.	500
<b>2.4.3 Project Budget</b>	<b>Contingency Fund</b>	Set aside as a backup fund for unexpected expenses.	500
<b>Total Estimated Cost</b>			<b>1500</b>

## 2.5 Team Organization



### Acquiring Project Team Members and Assigning Resources:

- **Resource Identification:**
  - o Identified Need: Developers with proficiency in web development languages, designers with UI/UX expertise, testers skilled in quality assurance methodologies.
- **Resource Acquisition:**
  - o Selection Process: Candidates selected based on qualifications, experience, and alignment with project goals.

- **Resource Allocation:**

- o Assignment: Developers assigned to coding tasks, designers collaborating with developers for design implementation, testers responsible for validation.

## **Developing and Managing the Project Team:**

- **Team-Building Activities:**

Initiatives: Conducted team-building workshops and icebreaker sessions to promote camaraderie and collaboration.

- **Training and Skill Development:**

- o Training Provided: Agile methodologies, UI/UX design principles, and automated testing tools to enhance team members' capabilities.

- **Performance Evaluation:**

- o Frequency: Conducted quarterly performance evaluations, providing constructive feedback and recognizing outstanding contributions.

- **Feedback and Recognition:**

- o Regular Recognition: Provided regular feedback on tasks and acknowledged team members' achievements to boost morale.

- **Conflict Resolution:**

- o Process: Implemented a conflict resolution process to address disagreements promptly and ensure a positive work environment.

- **Empowerment:**

- o Strategy: Empowered team members to make decisions within their roles, enhancing their sense of ownership and accountability.

## **Communication and Collaboration:**

- **Open Communication:**

- o Policy: Maintained an open-door policy and encouraged team members to share ideas, concerns, and suggestions openly.



- **Collaboration Tools:**

- o Utilized: Project management software (e.g., Trello, Slack) for task tracking, communication, and collaboration.

### **Maintaining Work-Life Balance:**

- **Balanced Workload:**

Prevention: Ensured a balanced workload for team members to prevent burnout, promoting a healthy work-life balance and maintaining productivity.

### **Continuous Improvement:**

- **Learning Culture:**

- o Promotion: Encouraged continuous learning through online courses and workshops, promoting a culture of personal and professional growth.

- **Adaptation to Change:**

- o Flexibility: Adapted team roles as the project evolved to address changing requirements and new challenges.

## **2.6 Process Model**

The process model for the development of the Job Posting and Searching Platform is structured to ensure systematic progress and successful project delivery.

### **2.6.1 Requirements Gathering and Analysis**

- **Objective:** Define project goals, scope, and requirements to guide development.
- **Activities:**
  - Conduct stakeholder interviews and workshops to understand user needs.
  - Document functional and non-functional requirements.
  - Analyze feasibility and constraints.

## 2.6.2 System Architecture and Design

- **Objective:** Design a robust and scalable architecture for the platform.
- **Activities:**
  - Develop system architecture, including database design and integration points.
  - Create wireframes, UI/UX designs, and prototypes to visualize user interactions.
  - Define data models, API specifications, and security protocols.

## 2.6.3 Development and Implementation

- **Objective:** Build and integrate platform components according to design specifications.
- **Activities:**
  - Implement backend logic using Node.js and Express for server-side operations.
  - Develop frontend components using React, ensuring responsive and intuitive user interfaces.
  - Integrate third-party services and APIs for enhanced functionality (e.g., payment gateways, job search APIs).

## 2.6.4 Testing and Quality Assurance

- **Objective:** Ensure the platform meets quality standards and functional requirements.
- **Activities:**
  - Conduct unit testing, integration testing, and system testing to validate functionality.
  - Perform performance testing to optimize speed and responsiveness.
  - Implement security testing to identify and mitigate vulnerabilities.

## 2.6.5 Deployment and Release

- **Objective:** Prepare the platform for deployment to production environments.
- **Activities:**
  - Set up hosting infrastructure and configure deployment pipelines.
  - Deploy application components using containerization (e.g., Docker) for scalability.
  - Conduct user acceptance testing (UAT) and finalize feature readiness.

## 2.6.6 Maintenance and Support

- **Objective:** Ensure ongoing operation, maintenance, and enhancement of the platform.
- **Activities:**
  - Monitor platform performance and user feedback.
  - Implement bug fixes, updates, and security patches regularly.
  - Provide user support and training documentation.
  - Plan for future enhancements and scalability improvements based on usage metrics and feedback.

## 2.6.7 Project Management and Documentation

- **Objective:** Manage project progress, resources, and documentation throughout the lifecycle.
- **Activities:**
  - Track project milestones, timelines, and resource allocation.
  - Maintain comprehensive project documentation (e.g., design documents, user manuals).
  - Conduct regular project reviews and retrospectives to optimize processes.

## 2.7 Risk MMM Plan

The Risk MMM (Monitoring, Mitigation, and Management) Plan involves identifying potential risks, implementing strategies to mitigate these risks, and establishing a system for ongoing risk management.

### 2.7.1 Risk Items Table

Risk No.	Risks	Probability	Impact
R1	Poor maintenance	Medium	Marginal
R2	Time management	High	Critical
R3	Lack of computer skills	Medium	Negligible
R4	Cyber attacks	High	Catastrophic

R5	Quality issues	Low	Marginal
R6	Misinformation	Medium	Critical
R7	Accuracy	Low	Critical

## 2.7.2 RMMM Plan

### Risk Number - R1

<b>Description</b>	<b>The platform might experience poor maintenance due to budget mishandling.</b>
Mitigation	Regularly review the project's budget and business case with stakeholders.
Monitoring	Monitor team accountability and adherence to budget constraints.
Management	Promptly address budgetary issues through proactive management actions.

### Risk Number - R2

<b>Description</b>	<b>Challenges with time management could lead to project delays.</b>
Mitigation	Regularly assess project timelines and adjust schedules as necessary.
Monitoring	Monitor work hours to prevent burnout and ensure workload distribution.
Management	Implement efficiency measures to streamline processes and meet deadlines.

### Risk Number - R3

<b>Description</b>	<b>Potential issues may arise from team members lacking necessary computer skills.</b>
Mitigation	Provide training and development opportunities for team members.

Monitoring	Assess team competency through periodic evaluations and training sessions.
Management	Support team members with resources and mentorship to enhance skills.

#### **Risk Number - R4**

<b>Description</b>	<b>The platform is vulnerable to cyber attacks due to its informational nature.</b>
Mitigation	Strengthen system security with robust firewall and antivirus measures.
Monitoring	Conduct regular security audits and monitor system vulnerabilities.
Management	Implement an incident response plan to swiftly address security breaches.

#### **Risk Number - R5**

<b>Description</b>	<b>Quality issues may arise, potentially affecting the platform's standards.</b>
Mitigation	Adhere strictly to development procedures and guidelines.
Monitoring	Regularly review and update system to meet industry standards.
Management	Continuously evaluate and improve platform quality assurance processes.

#### **Risk Number - R6**

<b>Description</b>	<b>Misinformation could lead to critical errors, particularly in medical information.</b>
Mitigation	Ensure accuracy and reliability of data through stringent validation processes.
Monitoring	Increase oversight on data access and usage to prevent errors.
Management	Implement strict data segregation policies to safeguard sensitive information.

## Risk Number - R7

Desription	Technical glitches and autocorrect issues may impact system accuracy.
Mitigation	Implement checks before data entry to mitigate errors. Ensure system stability through regular maintenance.
Monitoring	Monitor system performance for potential errors and delays.
Management	Promptly address technical issues to maintain system integrity and reliability.

This MMM plan addresses specific risks relevant to your job posting and searching platform project, outlining mitigation strategies, monitoring approaches, and management actions tailored to each identified risk.

## Chapter Three - System Analysis

### 3.1 Introduction

System analysis forms the backbone of our project's development lifecycle, crucial for designing a robust and user-centric job posting and searching platform. This chapter initiates a detailed exploration into the functionality, structure, and requirements of our system, aiming to translate user needs into actionable technical specifications.

In today's dynamic digital landscape, the need for an efficient job marketplace is paramount. Our platform seeks to revolutionize the way job seekers connect with opportunities and employers find talent. To achieve this, a meticulous system analysis is imperative.

### 3.2 Current System Overview

The existing job search ecosystem is characterized by a fragmented landscape, wherein job seekers find themselves navigating a multitude of platforms, while employers grapple with

inefficiencies in connecting with potential candidates. The contemporary recruitment paradigm is beset with challenges, ranging from the lack of precision in traditional hiring methods to protracted recruitment processes and overlooked opportunities.

### **3.2.1 Job Seeker Experience**

For job seekers, the current system poses a considerable challenge as they are required to engage with disparate platforms, each offering a limited view of the available job market. Navigating through various websites and applications not only consumes valuable time but also introduces a significant degree of complexity and frustration. The lack of a unified platform results in a disjointed experience, hindering users from efficiently discovering and applying for relevant job opportunities.

### **3.2.2 Employer Perspective**

From the standpoint of employers, the prevailing system presents obstacles in reaching potential candidates with precision and speed. Traditional recruitment methods, characterized by their generic nature, contribute to prolonged hiring processes and, consequently, increased time-to-fill positions. The inefficiency of these methods often results in missed opportunities, as employers struggle to identify suitable candidates amidst the vast pool of applicants.

### **3.2.3 Transition from Education to Employment**

One notable gap in the current system is the inadequate support for individuals transitioning from education to employment. The absence of a dedicated platform catering to this crucial phase leaves aspiring professionals facing uncertainties and challenges in navigating the job market. Bridging this gap is imperative to ensure a smoother transition for fresh graduates and those entering the workforce for the first time.

### **3.2.4 Technological Limitations**

Compounding these issues are technological gaps in existing solutions, which impede the effectiveness of job platforms in delivering tailored and user-centric experiences. Outdated or insufficiently advanced technologies restrict the personalization and customization features that are essential for meeting the diverse needs of both job seekers and employers. As a result, the

current system falls short in providing solutions that are adaptive to the evolving dynamics of the job market.

In summary, the current job posting and searching platform is characterized by fragmentation, inefficiency, and a lack of dedicated support for specific user groups. Addressing these shortcomings is imperative to lay the groundwork for the development of a more cohesive, efficient, and user-friendly platform. The subsequent sections of this chapter will delve into the proposed system, outlining the strategies and features aimed at overcoming the challenges inherent in the current landscape.

### **3.3 Proposed System Overview**

The proposed Job Posting and Searching Platform is positioned as a transformative solution to address the shortcomings of the current system. It aims to create a unified, user-centric online ecosystem that seamlessly connects job seekers with employers. This section provides an overview of the functional and non-functional requirements of the proposed system.

#### **3.3.1 Functional Requirements**

The functional requirements include:

##### **1. User Registration and Authentication:**

- **Requirement:** Users (job seekers and employers) must securely register for accounts.
- **Details:** Capture essential information such as name, email, password, and user type during registration.
- **Authentication:** Implement secure login functionality with options for password recovery and multi-factor authentication.

##### **2. Profile Management:**

- **Requirement:** Users should be able to create, update, and manage their profiles.
- **Details:** Job seekers can add personal information, work experience, education, skills, and upload resumes.



- **Employers:** Ability to manage company profiles, add job listings, and view applications.

### 3. Job Search and Filtering:

- **Requirement:** Job seekers need to search for job listings based on various criteria.
- **Details:** Implement filters such as job title, location, industry, salary range, and experience level.
- **Advanced Search:** Provide advanced options for detailed filtering based on skills, company size, benefits, etc.

### 4. Job Posting and Management:

- **Requirement:** Employers must be able to post job openings on the platform.
- **Details:** Include fields for job title, description, responsibilities, requirements, location, salary, and application deadline in job posting forms.
- **Management:** Employers should have the ability to edit, deactivate, or delete job postings, as well as view applications received.

### 5. Saved Jobs:

- **Requirement:** Job seekers should be able to save jobs for later viewing.
- **Details:** Provide functionality for users to bookmark interesting job postings.

### 6. Application Tracking:

- **Requirement:** Job seekers need to track the status of their job applications.
- **Details:** Include features for applicants to see which jobs they've applied to, application status updates.

### 7. Career Development Resources:

- **Requirement:** Provide additional resources beyond job listings, including articles for professional development.
- **Details:** Offer a selection of career advice, industry insights, and skill-building resources to support users in advancing their careers.

### 3.3.2 Non-Functional Requirements

The non-functional requirements for the job posting and searching platform encompass:

- **Usability:**
  - The platform must feature an intuitive and user-friendly interface designed for seamless navigation and accessibility by both job seekers and employers.
- **Performance:**
  - Ensure the platform maintains optimal responsiveness and performance under varying user loads, including peak times and increased data processing demands.
- **Scalability:**
  - Design the system to scale efficiently to accommodate a growing user base, increasing job listings, and expanding data volumes without compromising performance or user experience.
- **Security:**
  - Implement robust security measures to protect user data, including encryption for sensitive information, secure authentication mechanisms, and regular security audits to mitigate risks of unauthorized access or data breaches.
- **Compatibility:**
  - Ensure compatibility across multiple devices (desktops, tablets, smartphones) and various web browsers to provide a consistent and responsive user experience across different platforms.
- **Adaptability:**
  - The platform should be adaptable to technological advancements and future market trends, with the flexibility to integrate new features, updates, and enhancements seamlessly.

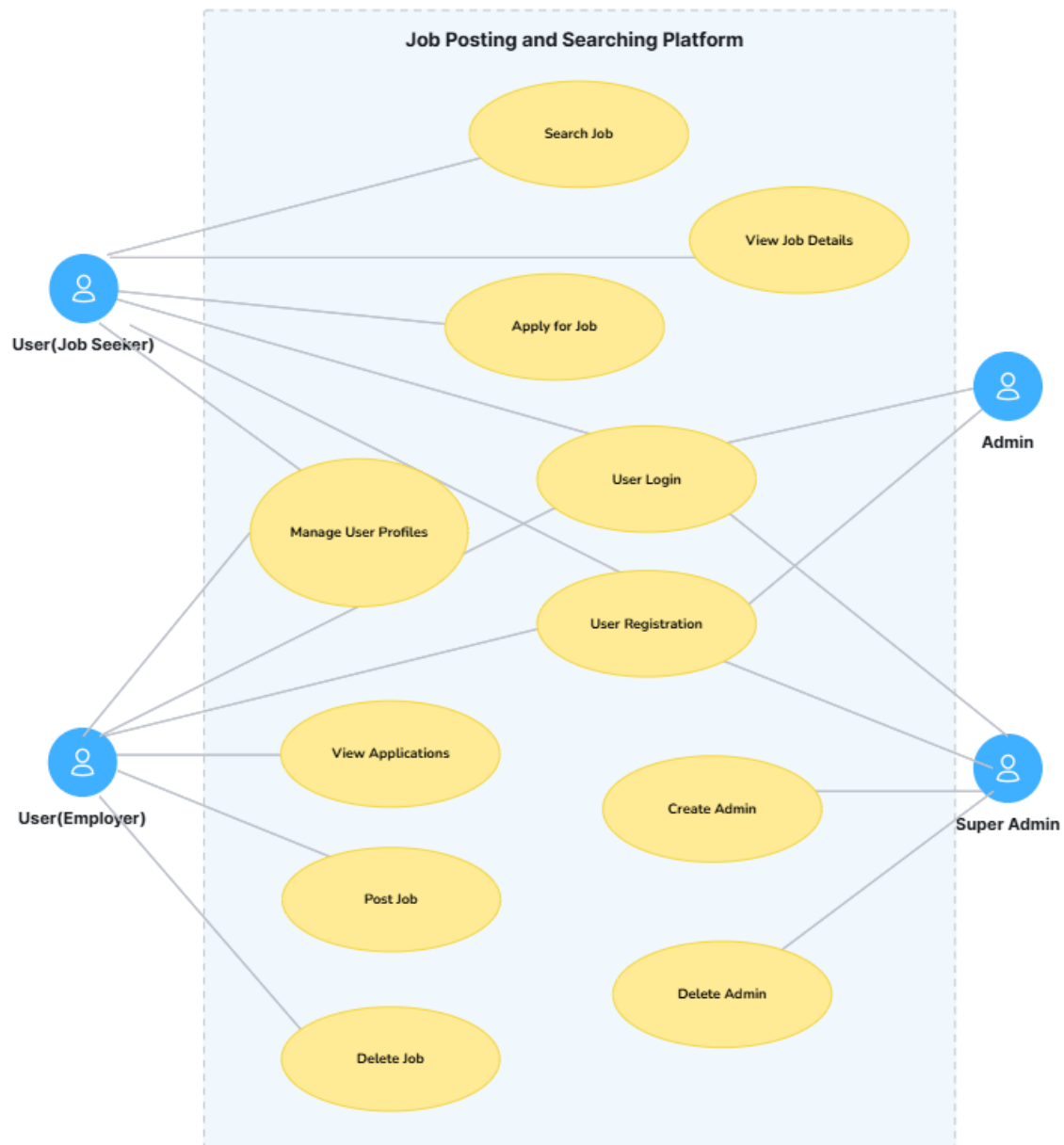
### 3.4 System Models - Requirement Determination

In the phase of Requirement Determination, the job posting and searching platform focuses on defining comprehensive system requirements based on user needs and business objectives. This phase encompasses several critical activities:

- **User Requirements Elicitation:**
  - Conduct interviews, surveys, and workshops with stakeholders, including job seekers and employers, to gather detailed insights into their needs, preferences, and expectations from the platform.
- **Functional Requirements Specification:**
  - Define clear and specific functional requirements that outline the core capabilities and features of the platform, such as job posting, search, application management, and user profile management.
- **Non-Functional Requirements Specification:**
  - Document non-functional requirements related to usability, performance, scalability, security, compatibility, and adaptability, ensuring the platform meets high standards of user experience and operational excellence.
- **Use Case Modeling:**
  - Develop use case diagrams and scenarios to illustrate how users interact with the platform, identifying key functionalities and user roles to streamline processes and enhance usability.
- **UI Prototype Design:**
  - Create user interface prototypes to visualize the platform's layout, navigation flow, and interaction design, ensuring alignment with user expectations and usability guidelines.
- **Business Rules and Constraints Definition:**
  - Establish business rules and constraints that govern the platform's behavior, ensuring consistency and adherence to industry regulations and user expectations.

### 3.4.1 Essential Use Case Modeling

#### 3.4.1.1 Use Case Diagram



#### 3.4.1.2 Use Case Documentation

## Use Case Identification

Identifying the key activities performed on the job posting and searching platform is crucial for system analysis. The following use cases have been identified based on the system requirements and functionalities:

### Actors:

- Super Admin
- Admin
- Job Seeker
- Employer

### Use Cases:

- **User Authentication (Login)**
- **Activity Log**
- **Search**
- **Create Admin**
- **Delete Admin**
- **Post Job Listing**
- **Delete Job Listing**
- **Apply for Job**
- **View Applications**
- **View Job Details**
- **Manage User Profile**

## Use Case Descriptions

### Use Case ID: UC001 - User Registration

Field	Description
Use Case ID	UC001
Use Case Name	User Registration

Description	Users register to create an account and access the platform's features and functionalities.
Actor	Super Admin, Admin, Job Seeker, Employer
Pre-Condition	None
Post-Condition	User successfully registers and gains access to their account dashboard.
Basic Course of Action	1. User opens the platform and navigates to the registration page. 2. User enters their registration details including first name, middle name (optional), last name, email, phone number, password, confirm password, and gender. 3. The system validates the entered information: - Ensures passwords match. - Validates email format. - Validates password complexity (minimum 8 characters, mix of letters, numbers, and special characters). 4. Upon successful validation, the system sends the registration data to the server for processing. 5. The server creates a new user account and responds with a success message. 6. User receives a confirmation message indicating successful registration and is redirected to the login page.
Alternative Course of Action	- If passwords do not match, the system displays an error message and prompts the user to re-enter passwords. - If the email format is invalid or does not meet required criteria, the system displays an error message. - If password complexity requirements are not met, the system displays an error message prompting the user to meet the criteria.

#### Use Case ID: UC002 - User Authentication (Login)

Field	Description
Use Case ID	UC002
Use Case Name	User Authentication (Login)
Description	Users authenticate themselves to access the platform's features and functionalities.
Actor	Super Admin, Admin, Job Seeker, Employer
Pre-Condition	None

Post-Condition	User gains access to their account dashboard.
Basic Course of Action	1. User opens the platform and navigates to the login page. 2. User enters their credentials (email and password). 3. The system verifies the credentials. 4. Upon successful verification, the user gains access to their respective dashboard.
Alternative Course of Action	If the user forgets their password: 1. User clicks on the "Forgot Password" link. 2. The system sends a password reset link or verification code to the user's registered email. 3. User follows the instructions to reset their password.

### Use Case ID: UC003 - Search

Field	Description
Use Case ID	UC003
Use Case Name	Search
Description	Users search for jobs or candidates based on specific criteria.
Actor	Job Seeker, Employer
Pre-Condition	User must be logged in.
Post-Condition	Search results matching the criteria are displayed.
Basic Course of Action	1. User navigates to the search page. 2. User enters keywords, filters, or criteria for the search. 3. The system retrieves and displays relevant job listings or candidate profiles.

**Use Case ID: UC004 - Create Admin**

<b>Field</b>	<b>Description</b>
Use Case ID	UC004
Use Case Name	Create Admin
Description	Super Admin creates a new admin account for platform management.
Actor	Super Admin
Pre-Condition	Super Admin must be logged in.
Post-Condition	New admin account is created and granted administrative privileges.
Basic Course of Action	1. Super Admin accesses the admin management section. 2. Super Admin selects the option to create a new admin. 3. System displays the admin registration form. 4. Super Admin fills out the form with required details (name, email, role). 5. Super Admin submits the form to create the new admin account.

**Use Case ID: UC005 - Delete Admin**

<b>Field</b>	<b>Description</b>
Use Case ID	UC005
Use Case Name	Delete Admin



Descri ption	Super Admin removes an existing admin account from the platform.
Actor	Super Admin
Pre- Condi tion	Super Admin must be logged in.
Post- Condi tion	Admin account is successfully removed from the system.
Basic Cours e of Actio n	1. Super Admin accesses the admin management section. 2. Super Admin searches for the admin by ID or name. 3. System displays the admin details. 4. Super Admin selects the option to delete the admin account. 5. System confirms the deletion action.

#### **Use Case ID: UC006 - Post Job Listing**

<b>Field</b>	<b>Description</b>
Use Case ID	UC006
Use Case Name	Post Job Listing
Descri ption	Employer posts a new job listing on the platform.
Actor	Employer
Pre- Condi tion	Employer must be logged in.

Post-Condition	Job listing is successfully posted and visible to job seekers.
Basic Course of Action	1. Employer accesses the job posting section. 2. Employer selects the option to post a new job listing. 3. System displays the job listing form. 4. Employer fills out the job details (title, description, requirements, etc.). 5. Employer submits the job listing for posting.

#### Use Case ID: UC007 - Delete Job Listing

Field	Description
Use Case ID	UC007
Use Case Name	Delete Job Listing
Description	Employer removes an existing job listing from the platform.
Actor	Employer
Pre-Condition	Employer must be logged in.
Post-Condition	Job listing is successfully removed from the platform.
Basic Course of Action	1. Employer accesses the job management section. 2. Employer selects the job listing to be deleted. 3. System displays the job details for confirmation. 4. Employer confirms the deletion action.

#### Use Case ID: UC008 - Apply for Job

Field	Description
Use Case ID	UC008
Use Case Name	Apply for Job
Description	Job Seeker applies for a job listing posted on the platform.
Actor	Job Seeker
Pre-Condition	Job Seeker must be logged in.

Post-Condition	Application is successfully submitted to the employer.
Basic Course of Action	1. Job Seeker searches for a suitable job listing. 2. Job Seeker selects the job listing of interest. 3. System displays the job details and application form. 4. Job Seeker fills out the application form with required details (resume, cover letter, etc.). 5. Job Seeker submits the application.

#### Use Case ID: UC009 - View Applications

Field	Description
Use Case ID	UC009
Use Case Name	View Applications
Description	Employer views applications received for a specific job listing.
Actor	Employer
Pre-Condition	Employer must be logged in.
Post-Condition	Employer reviews job applications for the job listing.
Basic Course of Action	1. Employer accesses the job management section. 2. Employer selects the job listing to view applications. 3. System displays the list of applications received. 4. Employer reviews each application in detail.

#### Use Case ID: UC010 - View Job Details

--	--

Use Case ID	UC010
Use Case Name	View Job Details
Description	Job Seeker views detailed information about a specific job listing.
Actor	Job Seeker
Pre-Condition	Job Seeker must be logged in.
Post-Condition	Job details are displayed to the job seeker.
Basic Course of Action	1. Job Seeker searches for a specific job listing. 2. Job Seeker selects the job listing to view details. 3. System displays comprehensive job details (title, description, requirements, etc.).

#### Use Case ID: UC011 - Manage User Profile

Use Case ID	UC011
Use Case Name	Manage User Profile
Description	Users manage and update their profile information on the platform.
Actor	Job Seeker, Employer

<b>Pre-Condition</b>	User must be logged in.
<b>Post-Condition</b>	Profile information is successfully updated.
<b>Basic Course of Action</b>	1. User navigates to their profile settings. 2. User selects the option to edit profile information. 3. System displays the current profile details. 4. User modifies the profile information (contact details, skills, etc.). 5. User saves the updated profile.

### **3.4.2 Essential UI Prototype**

# Welcome to Job Platform.

Ullamco non occaecat amet aliquip mollit laborum duis dolor  
exercitation reprehenderit consequat magna consectetur anim. Mollit  
laboris consequat culpa consectetur ullamco eiusmod anim sunt  
reprehenderit reprehenderit cupidatat ipsum irure eu irure. Nulla proide

[Log in](#)

## Registration

First Name

Middle Name

Last Name

Email

Phone Number

Password

Confirm Password

Gender

☐ Male ☐ Female

Already have an account? [Sign in](#)

# Login

Email

Password

Login

Home

Job Search

My Jobs

Logout

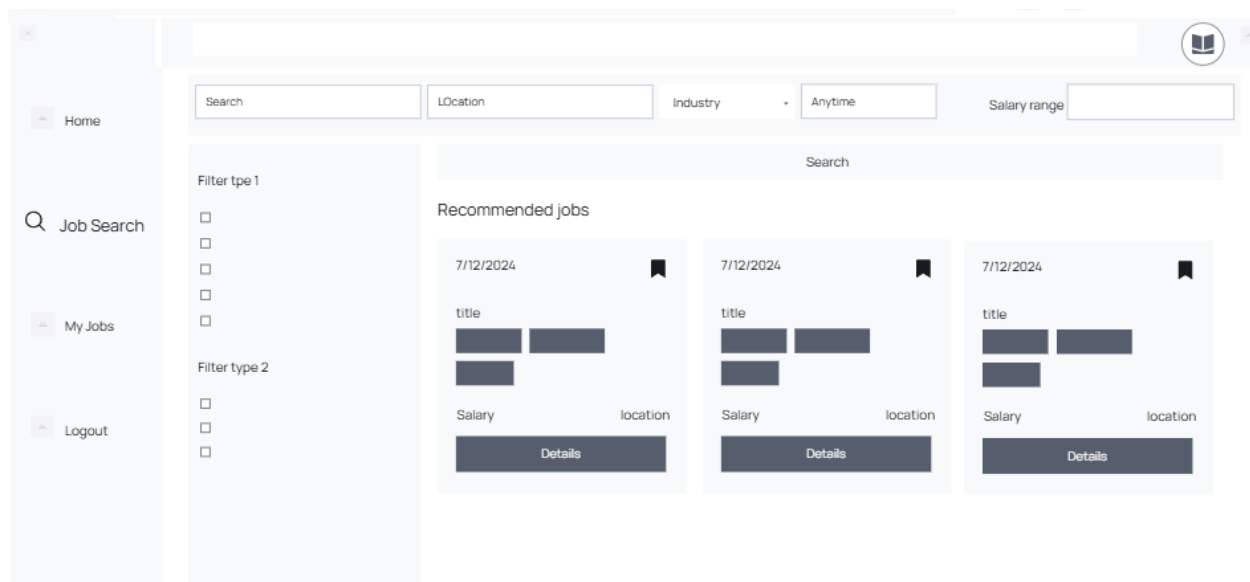
Welcome, User

User Information

Update Employment Status

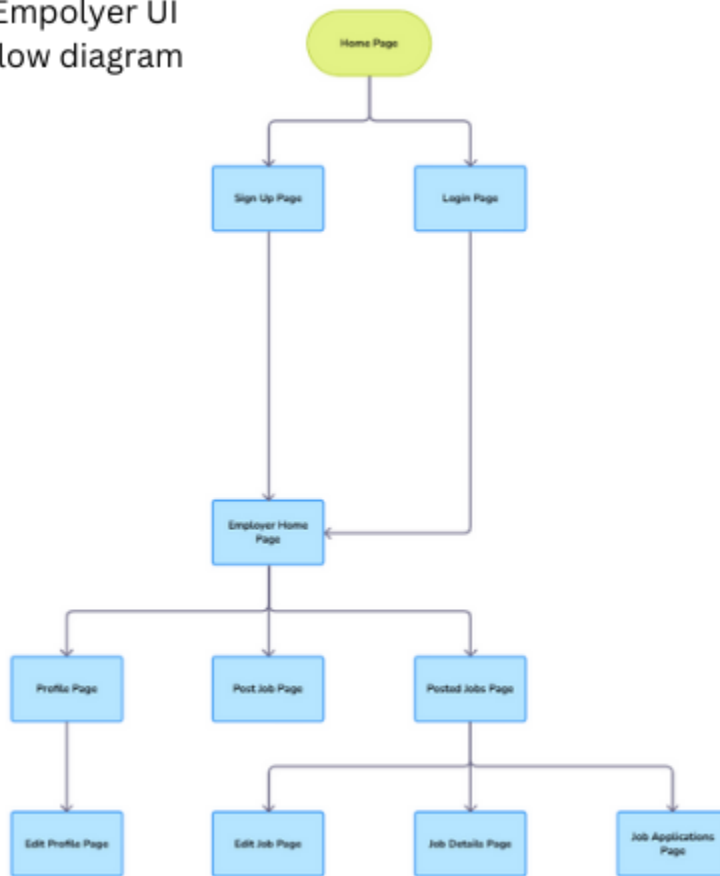
Are you hiring? Post a Job

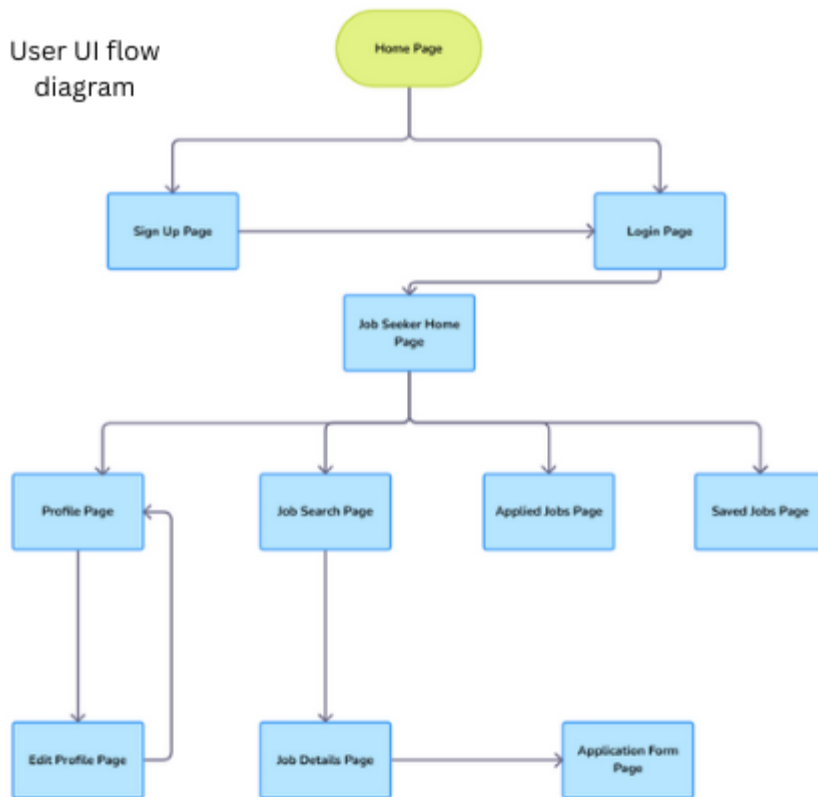




### 3.4.3 User Interface Flow Diagram

## Empolyer UI flow diagram





### 3.4.4 Supplementary Specifications

#### 3.4.4.1 Business Rules

Business rules serve as essential guidelines and constraints that govern how the job posting and searching platform operates. These rules are derived from industry standards, legal requirements, and specific operational needs identified during system analysis. They ensure consistency, compliance, and efficiency in platform operations. Below are the key business rules identified for the project:

Rule Category	Rule	Rationale
User Authentication and	Only authenticated users can access their respective roles (e.g., job	Ensures security and privacy by restricting unauthorized access to sensitive data and functionalities.

<b>Access Control</b>	seekers, employers, administrators).	
<b>Data Integrity and Accuracy</b>	All user-provided data must be validated for accuracy and completeness before being stored in the database.	Maintains data reliability and integrity, preventing errors and inconsistencies that could affect platform usability.
<b>Job Posting Standards</b>	Job postings must include mandatory fields such as job title, location, salary range, and job description.	Ensures clarity and completeness of job listings, providing accurate information to job seekers.
<b>Realistic Job Descriptions</b>	Job listings must accurately represent the responsibilities and expectations of the position.	Avoids misunderstandings between employers and job seekers, fostering a transparent and honest job market environment.

#### 3.4.4.2 Constraints

Constraints are limitations or conditions that restrict the design, implementation, or operation of the job posting and searching platform. These constraints are identified based on technical, regulatory, or organizational factors that impact the project. Here are the key constraints identified for the platform:

<b>Constraint Category</b>	<b>Constraint</b>	<b>Rationale</b>
<b>Technological Constraints</b>	The platform must be developed using React for the frontend and Node.js for the backend.	Chosen technologies align with project requirements for scalability, performance, and developer expertise.
<b>Time Constraints</b>	Project milestones and deliverables must be completed within the agreed timeline of X months.	Adheres to project schedule, meets stakeholder expectations, and ensures timely platform deployment.

<b>Compatibility Constraints</b>	The platform must be compatible with modern web browsers (Chrome, Firefox, Safari) and devices (desktops, tablets, mobile phones).	Provides a consistent user experience across different devices and browsers, maximizing accessibility for all users.
<b>Scalability Constraints</b>	Design architecture to support a scalable user base and increasing job postings without significant performance degradation.	Accommodates future growth and user demand, ensuring long-term platform sustainability.

These constraints outline the boundaries and requirements that must be considered throughout the development and deployment of the job posting and searching platform.

### 3.4.4.3 Change Case

Change cases define scenarios where modifications or updates may be required throughout the development and deployment phases of the job posting and searching platform. These cases ensure adaptability to evolving requirements and improvements in system functionality. Key change cases identified include:

<b>Change Category</b>	<b>Change Case</b>	<b>Rationale</b>
<b>User Interface Enhancements</b>	Iterative improvements based on user feedback to enhance usability and navigation.	Ensures the platform remains intuitive and user-friendly, enhancing overall user experience.
<b>Feature Expansion</b>	Addition of new features such as advanced search filters or integration with external APIs.	Enhances platform functionality, catering to diverse user needs and market demands.
<b>Performance Optimization</b>	Continuous optimization to improve platform responsiveness and reduce loading times.	Enhances user satisfaction and supports scalability as user base and data load increase.

<b>Technological Upgrades</b>	Adoption of new technologies or frameworks to leverage advancements in web development.	Improves platform efficiency, scalability, and compatibility with modern browsers and devices.
-------------------------------	---	--

Change cases serve as proactive measures to anticipate and address potential modifications or updates throughout the lifecycle of the job posting and searching platform.

### **3.5 System Models - Analysis**

#### **3.5.1 System Use Case Modeling**

##### **3.5.1.1 Use Case Diagram**



### 3.5.1.2 Use Case Documentation

Use Case ID: UC001

Use Case ID	UC001
Use Case Name	User Registration
Description	System for registering new users to access the job posting and searching platform.

Primary Actor	Job Seeker, Employer
Importance Level	High
Use Case Type	Detailed
Stakeholders & Interests	- Job Seeker: Create an account to apply for jobs, manage profile, and track applications. - Employer: Register to post jobs, manage job listings, and review applicant profiles.
Trigger	User navigates to the platform's registration page.
Trigger Type	External
Relationships	- Association: Users (Job Seeker, Employer) interact with the registration system to create new accounts. - Includes: Password strength validation, email format validation. - Extend: None - Generalization: None
Pre-Condition	None
Post-Condition	User successfully registered and gains access to their account dashboard.
Basic Course of Action	1. User accesses the platform's registration page. 2. User fills in registration details including first name, middle name (optional), last name, email, phone number, password, confirm password, and selects gender. 3. System validates the entered information: - Ensures passwords match. - Validates email format. - Validates password complexity (minimum 8 characters, mix of letters, numbers, and special characters). 4. Upon successful validation, system sends registration data to the server for processing. 5. Server creates a new user account and responds with a success message. 6. User receives a confirmation message indicating successful registration and is redirected to the login page.
Alternative Course of Action	- Passwords Do Not Match: System displays an error message and prompts the user to re-enter passwords. - Invalid Email Format: System displays an error message indicating incorrect email format. - Password Complexity Not Met: System displays an error message prompting the user to meet password complexity criteria.



**Use Case ID: UC002**

<b>Use Case Name</b>	<b>User Authentication (Login)</b>
Descript ion	System for authenticating users to access the job posting and searching platform.
Primary Actor	Job Seeker, Employer
Importa nce Level	High
Use Case Type	Detailed
Stakeho lders & Interest s	- Job Seeker: Access job listings and manage applications.- Employer: Post jobs and review applications.
Trigger	User navigates to the platform's login page.
Trigger Type	External
Relation ships	- Association: Users (Job Seeker, Employer) interact with the login system for access control. - Includes: Remember Me and Forgot Password functionalities. - Extend: None - Generalization: None
Pre- Condi ti on	None
Post- Condi ti on	User successfully logged into their respective dashboard.

Basic Course of Action	1. User accesses the platform's login page. 2. User enters their credentials (email and password). 3. System validates the entered credentials. 4. Upon successful validation, user gains access to their dashboard tailored for Job Seeker or Employer functionalities.
Alternative Course of Action	- Forgot Password: - User clicks on the "Forgot Password" link. - System prompts user to enter their email. - System sends a password reset link or code to the user's email. - User follows the link/code to reset their password.

### Use Case ID: UC003

<b>Use Case Name</b>	<b>Search</b>
Description	Users search for jobs or candidates based on specific criteria.
Primary Actor	Job Seeker, Employer
Importance Level	High
Use Case Type	Detailed
Stakeholders & Interests	- Job Seeker: Find suitable job listings. - Employer: Search for potential candidates.
Pre-Condition	User must be logged in.

Post-Condition	Search results matching the criteria are displayed.
Basic Course of Action	1. User navigates to the search page. 2. User enters keywords, filters, or criteria for the search. 3. The system retrieves and displays relevant job listings or candidate profiles.

#### Use Case ID: UC004

Use Case Name	<b>Create Admin</b>
Description	Super Admin creates a new admin account for platform management.
Primary Actor	Super Admin
Importance Level	High
Use Case Type	Detailed
Stakeholders & Interests	- Super Admin: Manage administrative access.

Pre-Condition	Super Admin must be logged in.
Post-Condition	New admin account is created and granted administrative privileges.
Basic Course of Action	1. Super Admin accesses the admin management section. 2. Super Admin selects the option to create a new admin. 3. System displays the admin registration form. 4. Super Admin fills out the form with required details (name, email, role). 5. Super Admin submits the form to create the new admin account.

#### Use Case ID: UC005

Use Case Name	<b>Delete Admin</b>
Description	Super Admin removes an existing admin account from the platform.
Primary Actor	Super Admin
Importance Level	High
Use Case Type	Detailed
Stakeholders &	- Super Admin: Maintain platform security by managing admin accounts.

Interests	
Pre-Condition	Super Admin must be logged in.
Post-Condition	Admin account is successfully removed from the system.
Basic Course of Action	1. Super Admin accesses the admin management section. 2. Super Admin searches for the admin by ID or name. 3. System displays the admin details. 4. Super Admin selects the option to delete the admin account. 5. System confirms the deletion action.

#### Use Case ID: UC006

<b>Use Case Name</b>	<b>Post Job Listing</b>
Description	Employer posts a new job listing on the platform.
Primary Actor	Employer
Importance Level	High
Use Case Type	Detailed
Stakeholders	- Employer: Advertise job openings to potential candidates.

& Interests	
Pre- Condition	Employer must be logged in.
Post- Condition	Job listing is successfully posted and visible to job seekers.
Basic Course of Action	1. Employer accesses the job posting section. 2. Employer selects the option to post a new job listing. 3. System displays the job listing form. 4. Employer fills out the job details (title, description, requirements, etc.). 5. Employer submits the job listing for posting.

#### Use Case ID: UC007

<b>Use Case Name</b>	<b>Delete Job Listing</b>
Description	Employer removes an existing job listing from the platform.
Primary Actor	Employer
Importance Level	High
Use Case Type	Detailed
Stakeholders &	- Employer: Manage and remove outdated job listings.

Interests	
Pre-Condition	Employer must be logged in.
Post-Condition	Job listing is successfully removed from the platform.
Basic Course of Action	1. Employer accesses the job management section. 2. Employer selects the job listing to be deleted. 3. System displays the job details for confirmation. 4. Employer confirms the deletion action.

#### Use Case ID: UC008

<b>Use Case Name</b>	<b>Apply for Job</b>
Description	Job Seeker applies for a job listing posted on the platform.
Primary Actor	Job Seeker
Importance Level	High
Use Case Type	Detailed
Stakeholders	- Job Seeker: Submit applications for desired job positions.

& Intere sts	
Pre- Condi tion	Job Seeker must be logged in.
Post- Condi tion	Application is successfully submitted to the employer.
Basic Cours e of Actio n	1. Job Seeker searches for a suitable job listing. 2. Job Seeker selects the job listing of interest. 3. System displays the job details and application form. 4. Job Seeker fills out the application form with required details (resume, cover letter, etc.). 5. Job Seeker submits the application.

#### Use Case ID: UC009

<b>Use Case Name</b>	<b>View Applications</b>
Descrip tion	Employer views applications received for a specific job listing.
Primary Actor	Employer
Importa nce Level	High
Use Case Type	Detailed



Stakeholders & Interests	- Employer: Review job applications to assess candidates.
Pre-Condition	Employer must be logged in.
Post-Condition	Employer reviews job applications for the job listing.
Basic Course of Action	1. Employer accesses the job management section. 2. Employer selects the job listing to view applications. 3. System displays the list of applications received. 4. Employer reviews each application in detail.

#### Use Case ID: UC010

<b>Use Case Name</b>	<b>View Job Details</b>
Description	Job Seeker views detailed information about a specific job listing.
Primary Actor	Job Seeker
Importance Level	High
Use Case Type	Detailed

Stakeholders & Interests	- Job Seeker: Obtain comprehensive details about job openings.
Pre-Condition	Job Seeker must be logged in.
Post-Condition	Job details are displayed to the job seeker.
Basic Course of Action	1. Job Seeker searches for a specific job listing. 2. Job Seeker selects the job listing to view details. 3. System displays comprehensive job details (title, description, requirements, etc.).

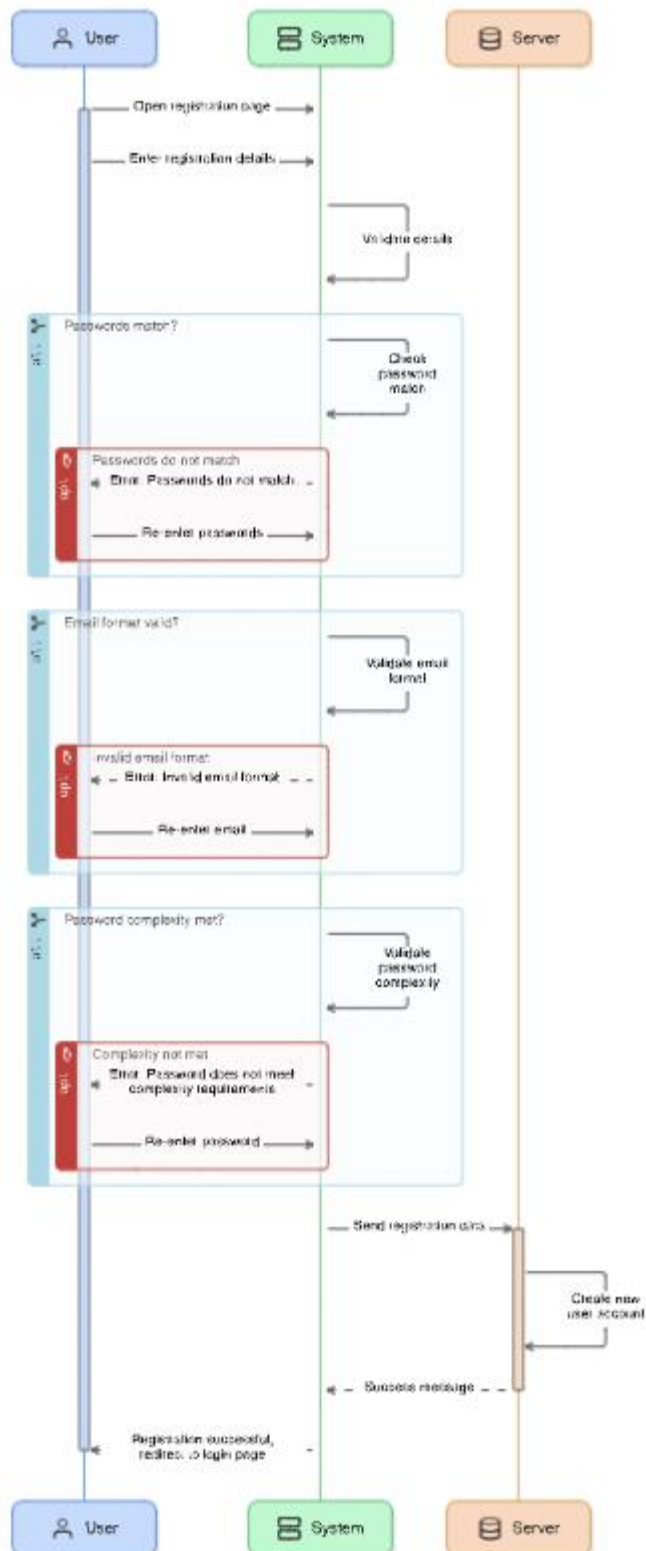
#### Use Case ID: UC011

<b>Use Case Name</b>	<b>Manage User Profile</b>
Description	Users manage and update their profile information on the platform.
Primary Actor	Job Seeker, Employer
Importance Level	Medium
Use Case Type	Detailed

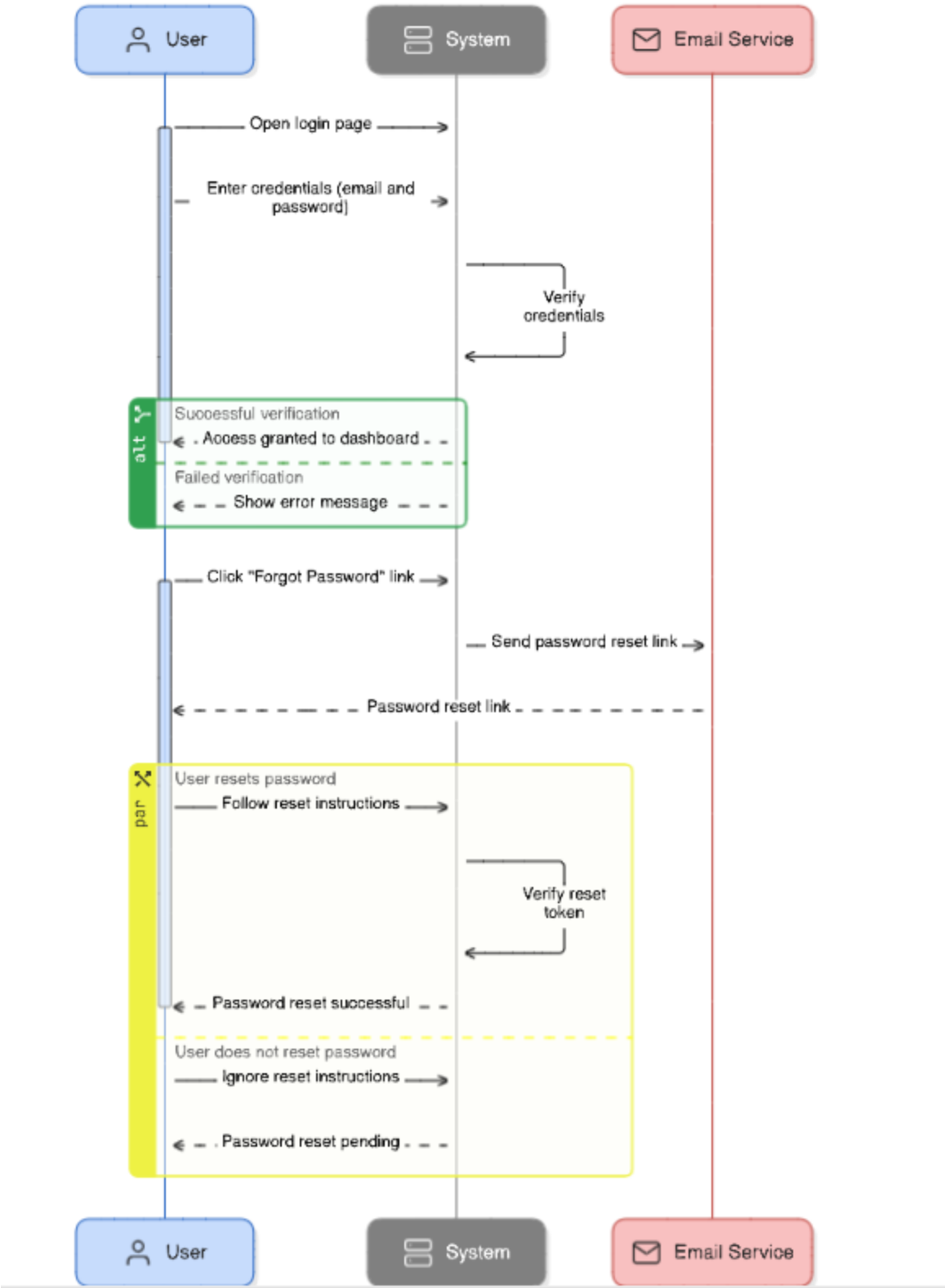
Stakeholders & Interests	<ul style="list-style-type: none"> <li>- Job Seeker: Update personal and professional details for improved job applications.</li> <li>- Employer: Maintain accurate contact and company information.</li> </ul>
Pre-Condition	User must be logged in.
Post-Condition	Profile information is successfully updated.
Basic Course of Action	1. User navigates to their profile settings. 2. User selects the option to edit profile information. 3. System displays the current profile details. 4. User modifies the profile information (contact details, skills, etc.). 5. User saves the updated profile.

### 3.5.2 Sequence Diagram

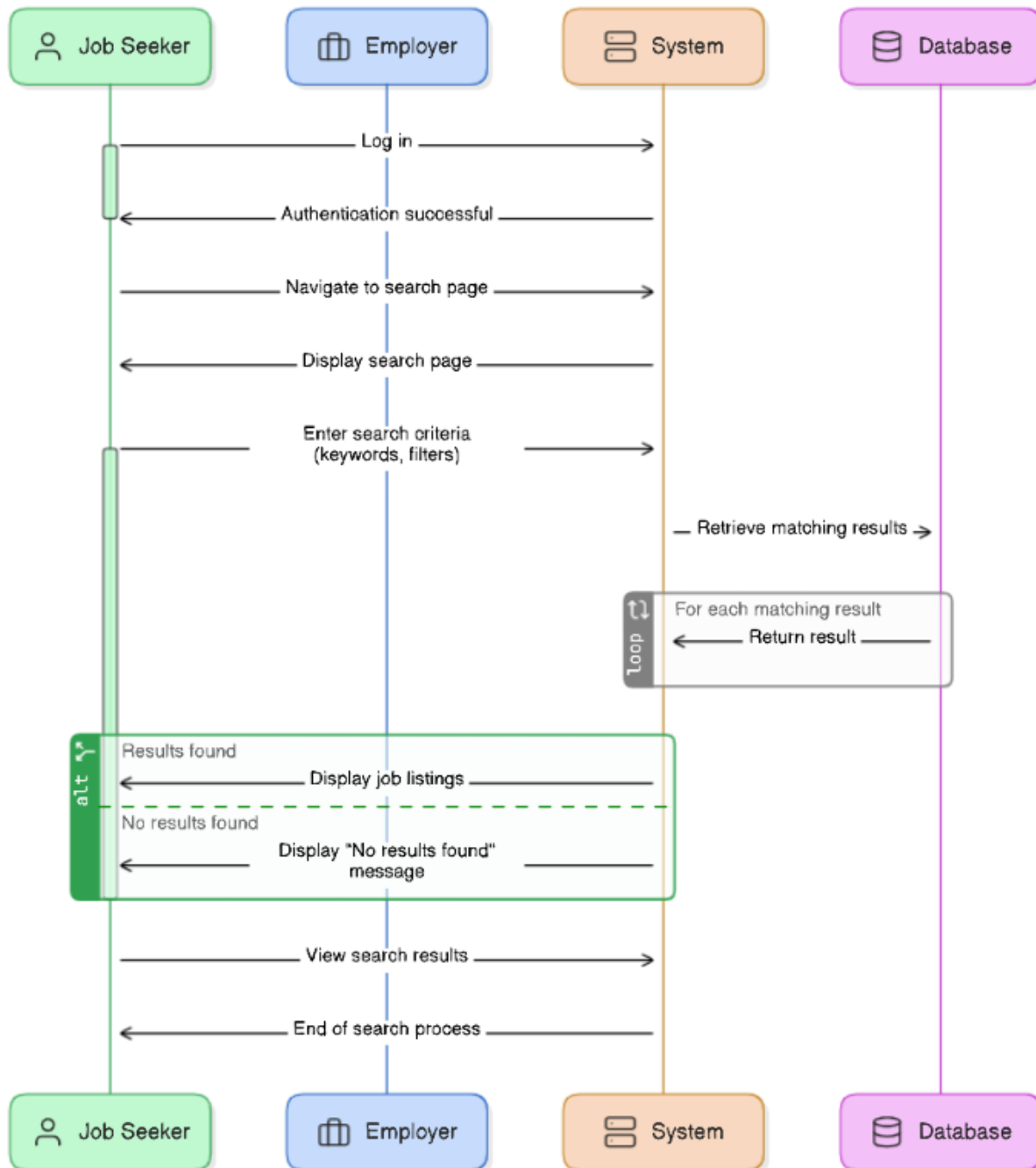
## User Registration



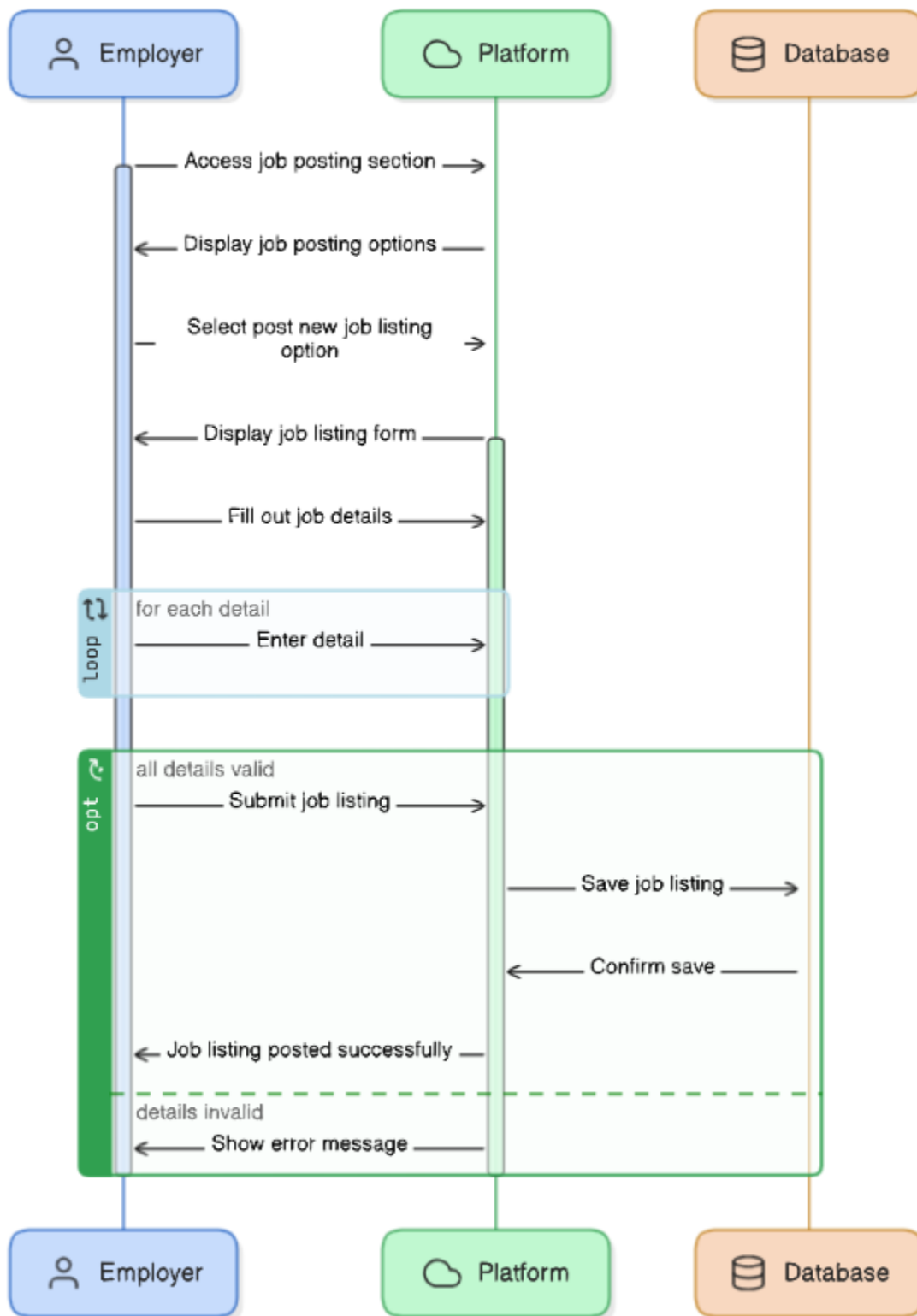
# User Authentication (Login)



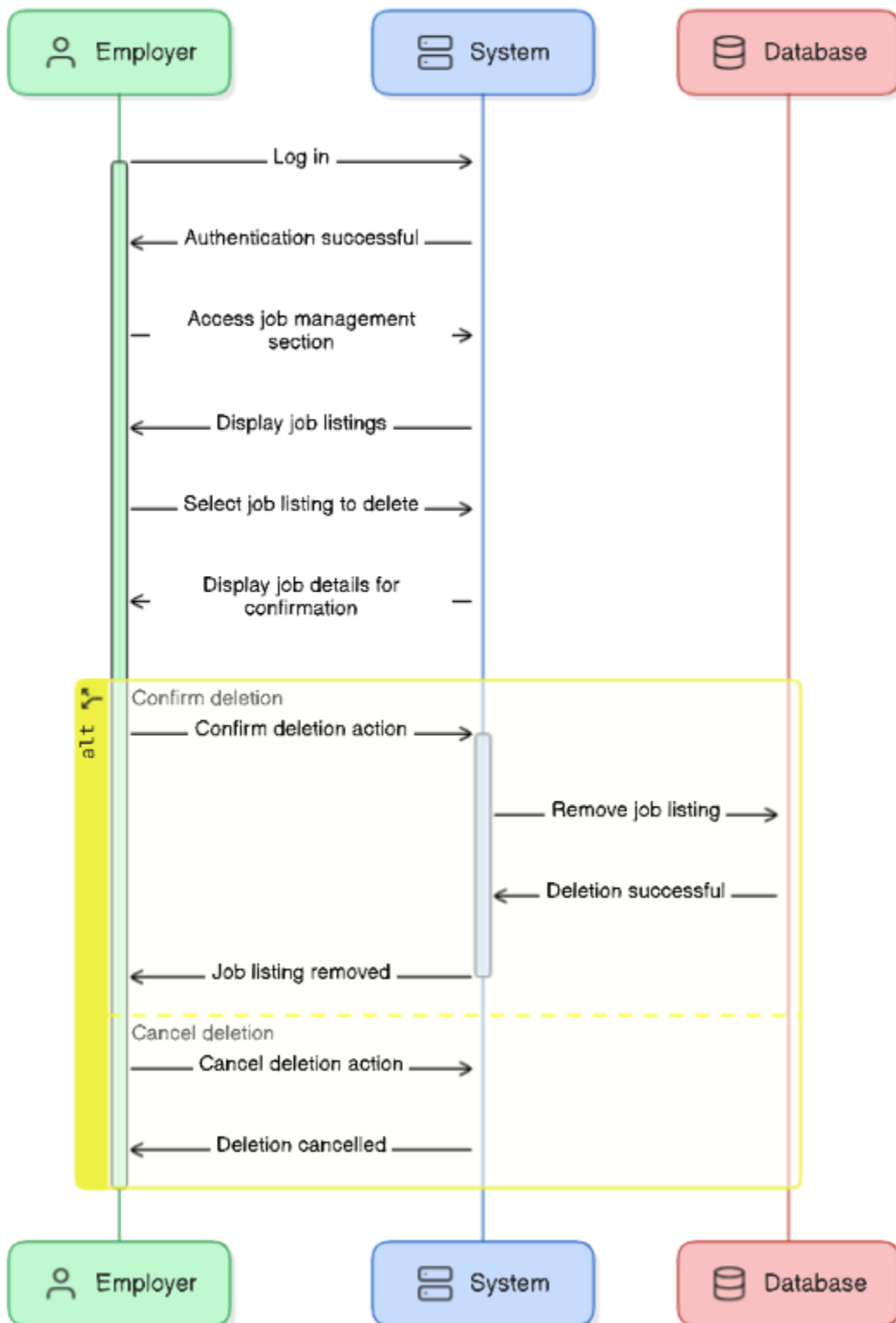
## Search Job



## Post Job Listing

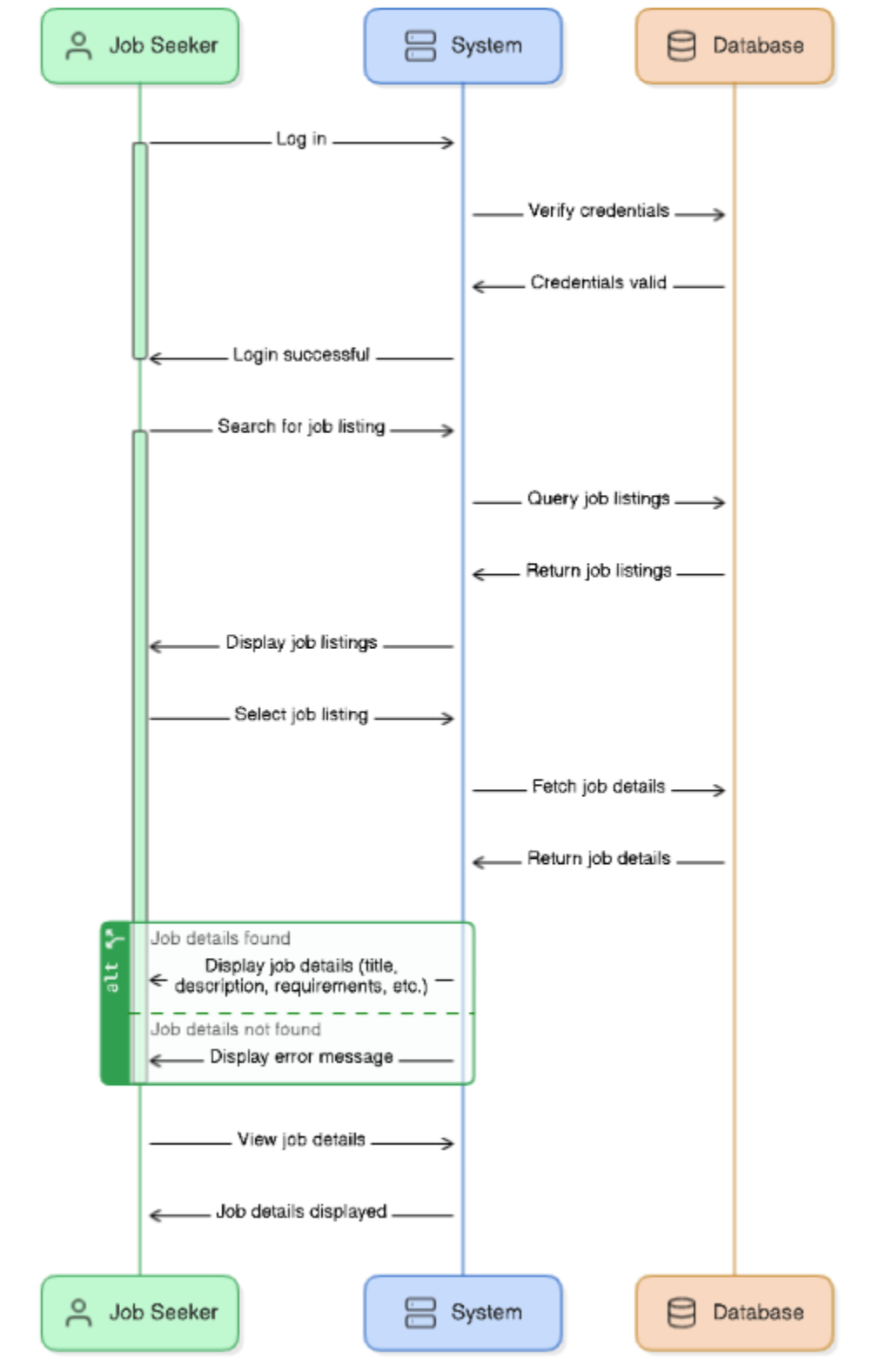


## Delete Job Listing

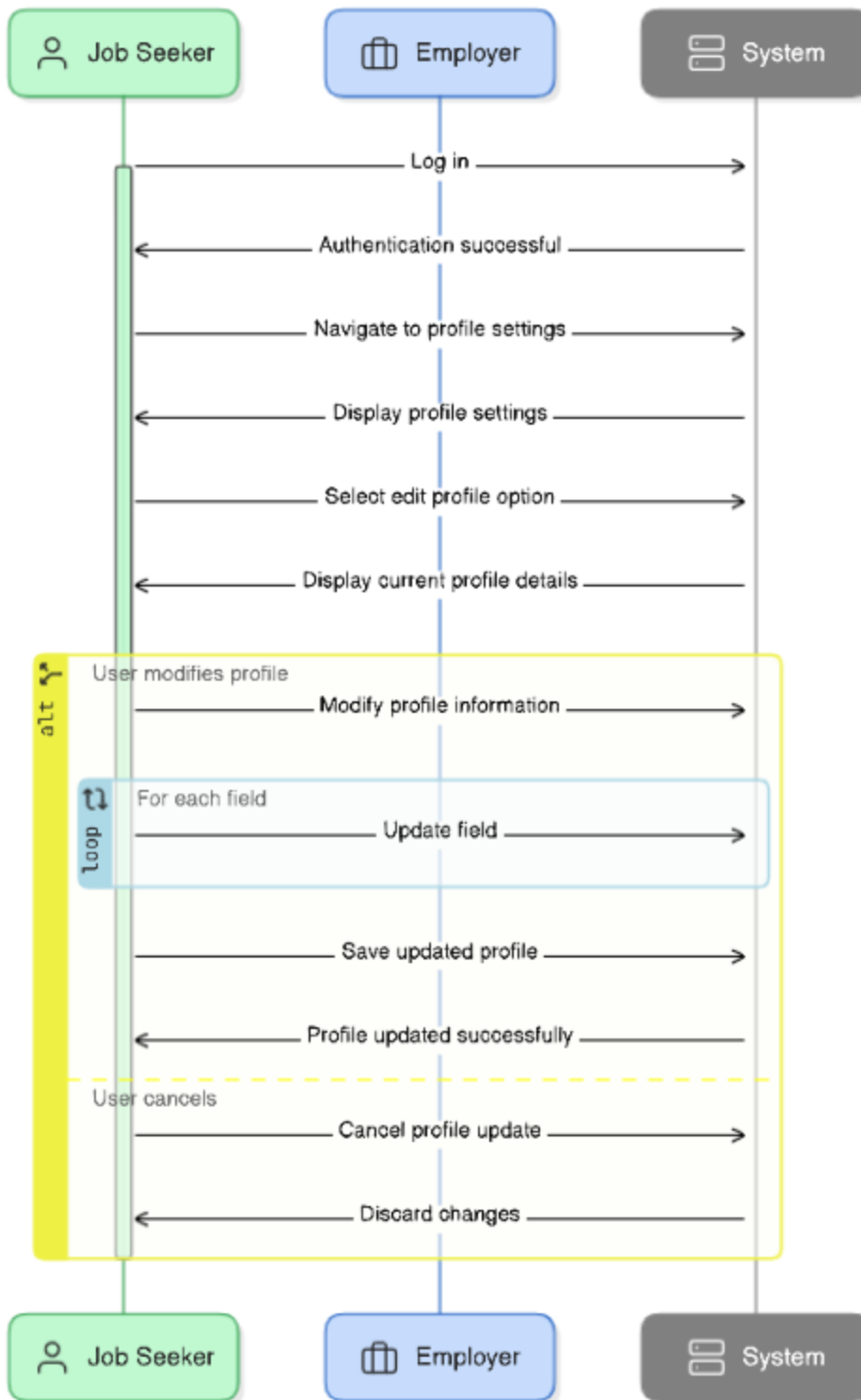




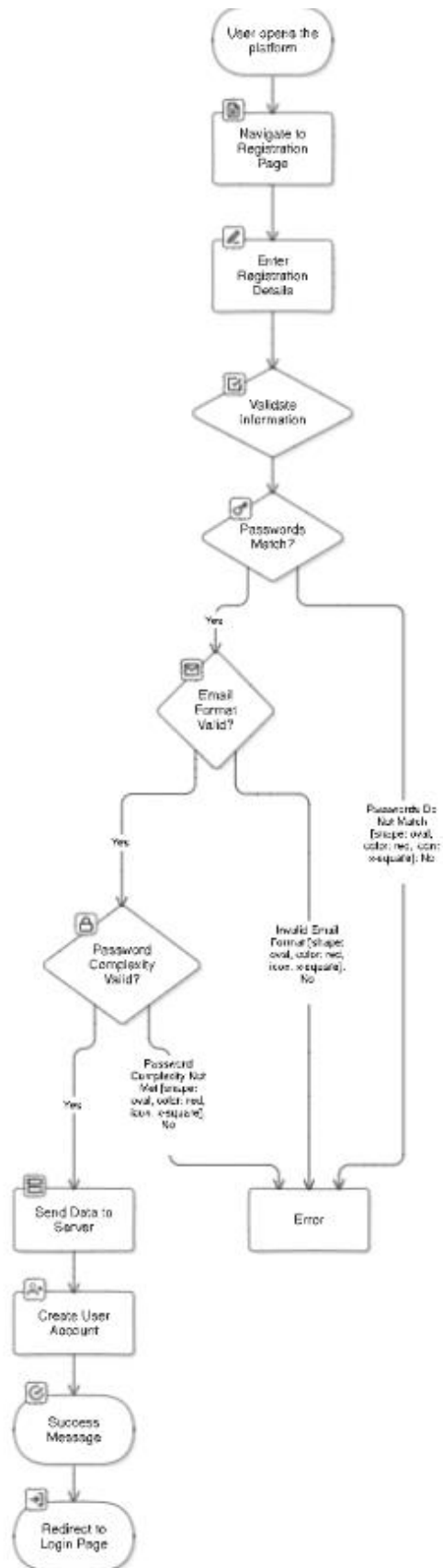
## View Job Details



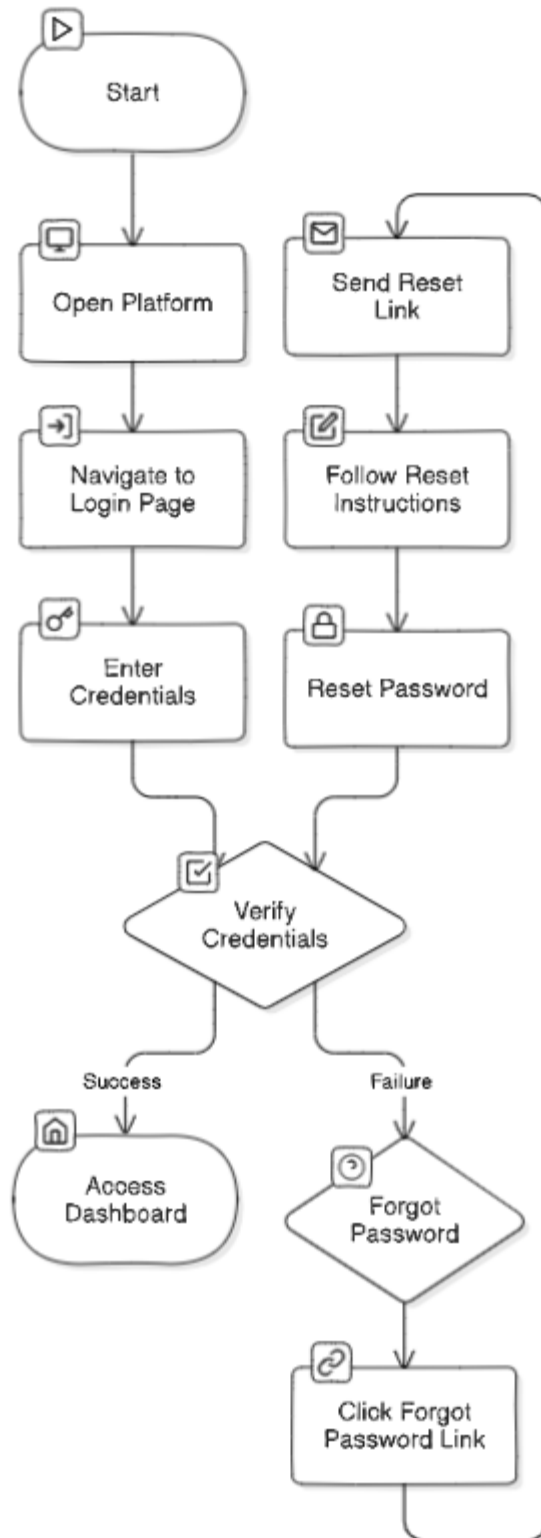
## Manage User Profile



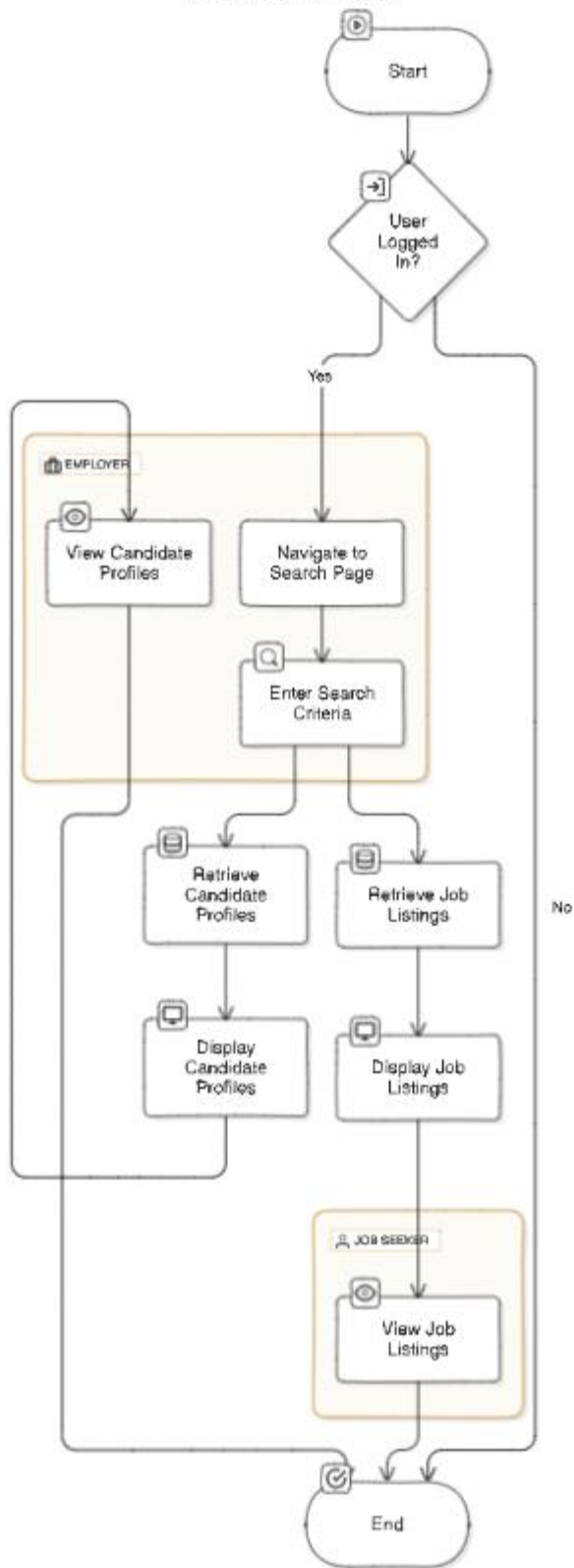
### 3.5.3 Activity Diagram



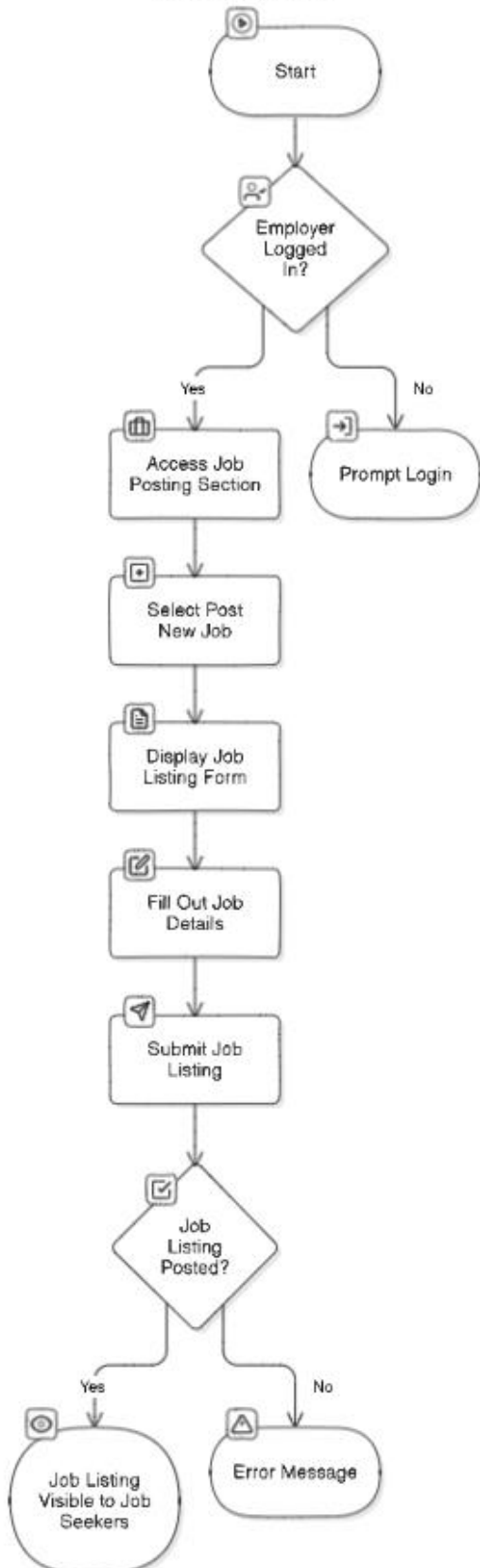
## User Authentication (Login)



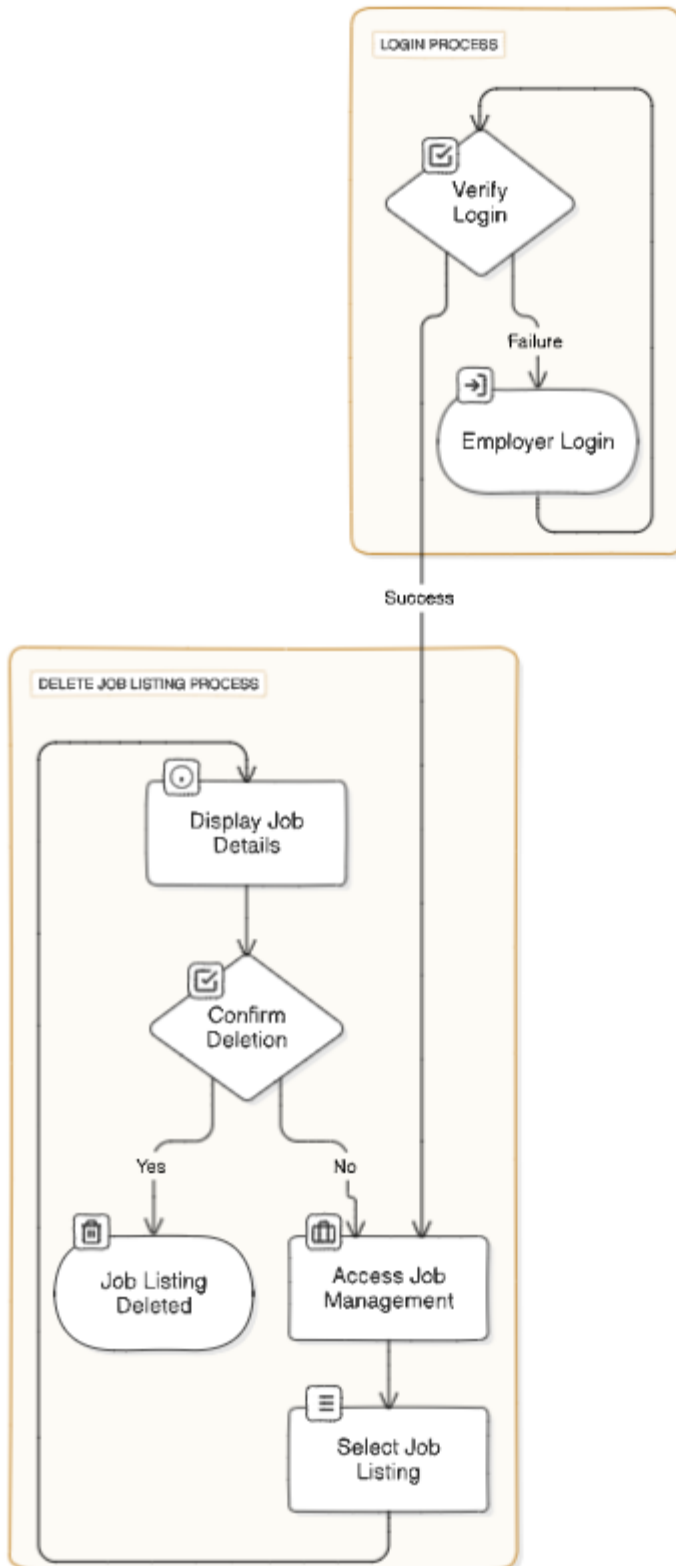
## Job Search Process



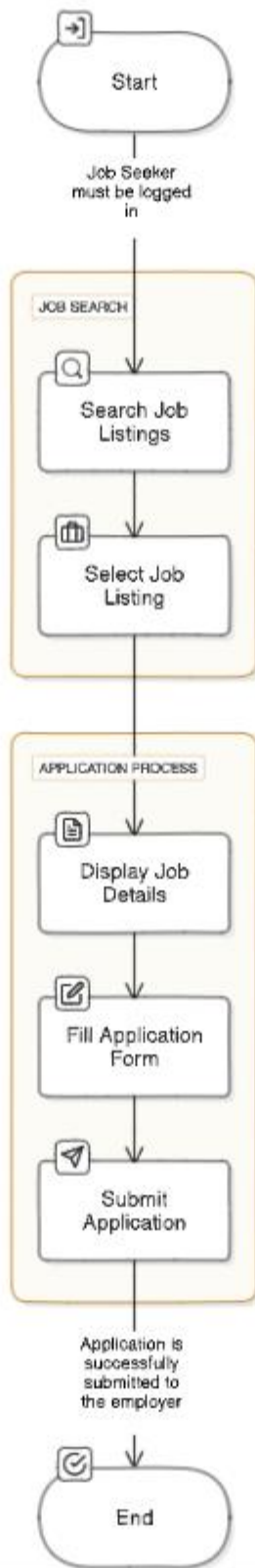
## Post Job Listing



## Delete Job Listing

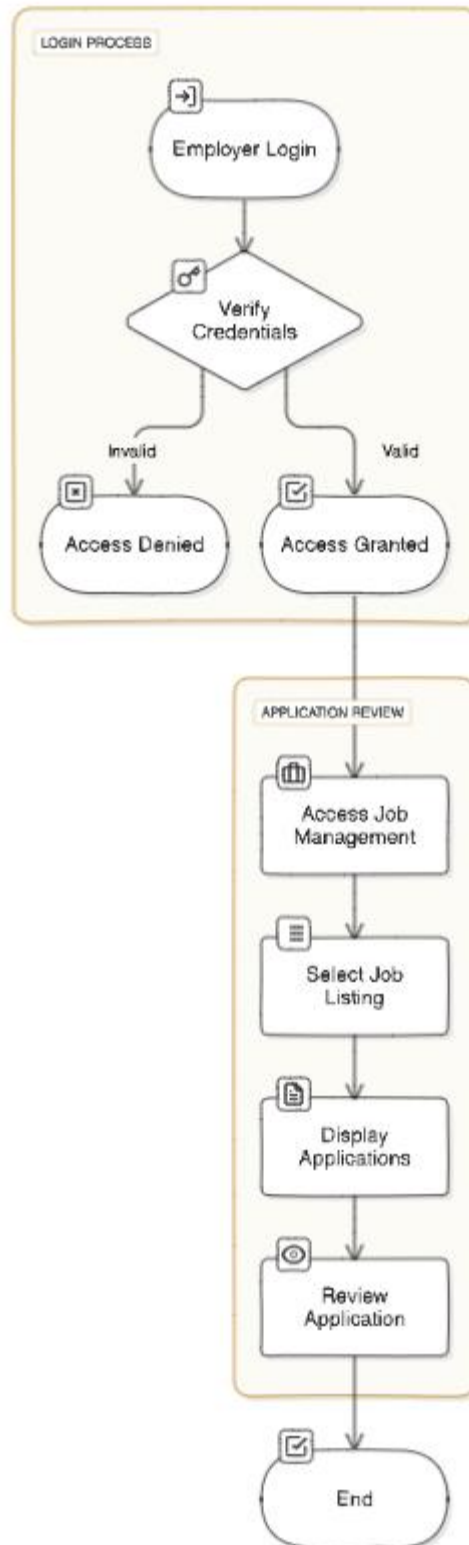


## Apply for Job

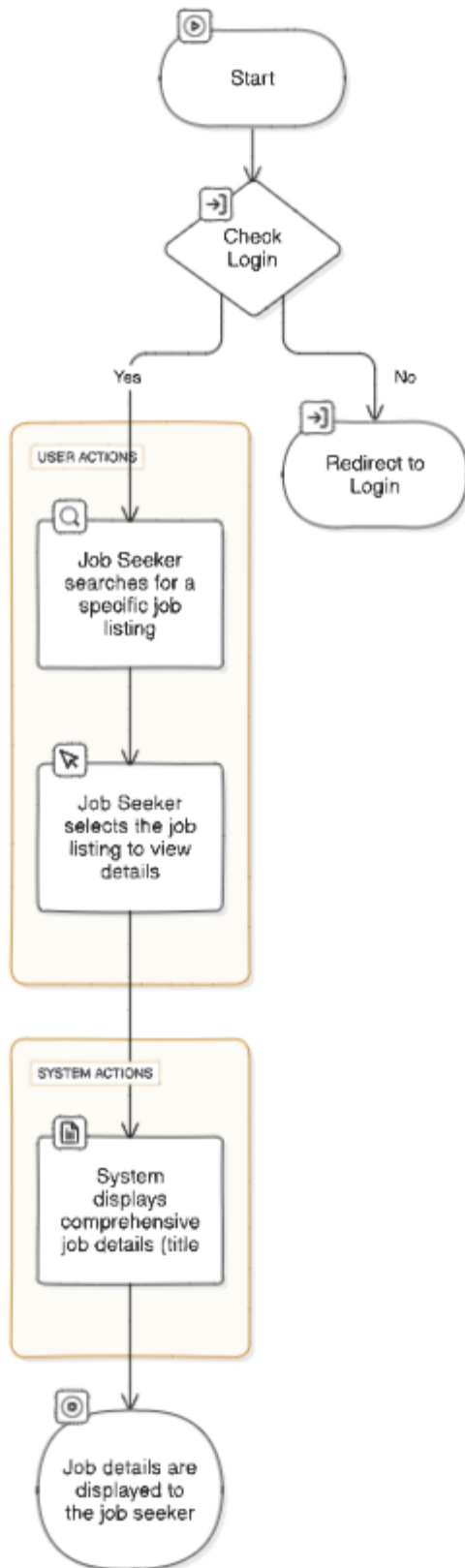




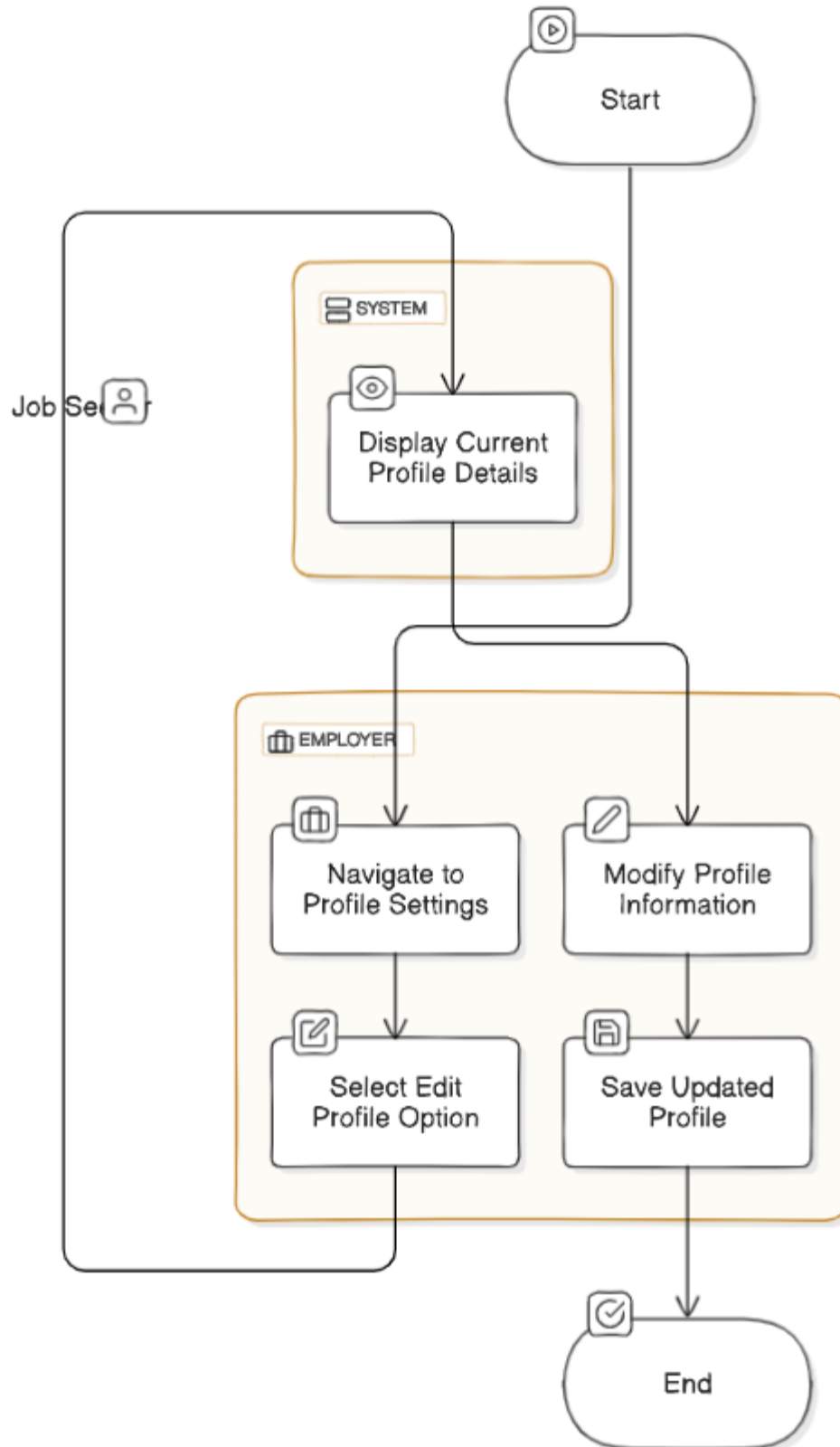
## View Applications



## View Job Details



## Manage User Profile



This system analysis lays the groundwork for the subsequent design and development phases, providing a comprehensive understanding of the platform's requirements, functionalities, and user interactions. The detailed models and documentation serve as a roadmap for creating a robust and user-centric Job Posting and Searching Platform.

## **Chapter Four – System Design**

### **4.1 Introduction**

The system design phase is pivotal in transforming requirements gathered during the previous phases into a concrete and functional architecture. This chapter outlines the design considerations, architectural decisions, and methodologies adopted to ensure the robustness, scalability, and usability of the job posting and searching platform.

### **4.2 Design Goals**

The design of the job posting and searching platform is driven by several key goals aimed at ensuring its effectiveness, usability, and security. These design goals are structured to align with the platform's requirements and user expectations. The primary design goals include:

#### **1. Scalability:**

- **Objective:** Design the platform to handle increasing numbers of users, job listings, and applications without compromising performance.
- **Implementation:** Utilize scalable architectures and technologies such as cloud services and microservices.

#### **2. Usability:**

- **Objective:** Provide an intuitive and user-friendly interface for job seekers, employers, and administrators.
- **Implementation:** Conduct user experience (UX) research and implement responsive design principles to enhance accessibility across devices.

#### **3. Security:**

- **Objective:** Safeguard user data, prevent unauthorized access, and ensure secure transactions.

- **Implementation:** Implement encryption protocols, secure authentication mechanisms, and regular security audits.
4. **Performance:**
- **Objective:** Optimize system response times and minimize latency for seamless user interactions.
  - **Implementation:** Employ caching mechanisms, database optimization techniques, and efficient frontend design practices.
5. **Modularity:**
- **Objective:** Design the system in modular components to facilitate maintainability and extensibility.
  - **Implementation:** Use component-based architectures (e.g., microservices, modular frontend components) and adhere to SOLID principles.
6. **Integration:**
- **Objective:** Ensure smooth integration between frontend and backend components.
  - **Implementation:** Define clear API contracts, use standardized communication protocols (e.g., RESTful APIs), and conduct thorough integration testing.
7. **Reliability:**
- **Objective:** Minimize downtime and ensure high availability of the platform.
  - **Implementation:** Implement fault-tolerant strategies, automated recovery mechanisms, and proactive monitoring tools.
8. **Adaptability:**
- **Objective:** Enable easy adaptation to evolving business requirements and technological advancements.
  - **Implementation:** Adopt agile development methodologies, conduct regular updates, and maintain documentation for future enhancements.

### 4.3 Design Trade offs

Designing the job posting and searching platform involves making strategic decisions that often involve trade-offs between conflicting factors. These trade-offs are crucial in balancing competing priorities and optimizing the platform's overall performance, usability, and cost-effectiveness. The key design trade-offs considered include:

1. **Scalability vs. Cost:**

- **Trade-off:** Choosing between scalable cloud solutions that may incur higher costs versus on-premises infrastructure that offers more control but may limit scalability.
- **Decision:** Opting for a hybrid approach to leverage cost-effective cloud services for scalability while maintaining critical components on-premises.

2. **Usability vs. Security:**

- **Trade-off:** Enhancing user experience with streamlined processes versus implementing stringent security measures that may introduce friction.
- **Decision:** Prioritizing user-friendly interfaces while integrating robust security protocols such as two-factor authentication and data encryption.

3. **Performance vs. Complexity:**

- **Trade-off:** Optimizing system performance with complex algorithms and extensive data processing versus maintaining simplicity for ease of maintenance.
- **Decision:** Employing efficient algorithms and caching mechanisms to enhance performance without overly complicating the system architecture.

4. **Modularity vs. Integration Effort:**

- **Trade-off:** Designing modular components for flexibility and easy maintenance versus investing effort in ensuring seamless integration between modules.
- **Decision:** Adopting a microservices architecture to promote modularity while establishing clear API contracts and standardized communication protocols for integration.

5. **Reliability vs. Cost:**

- **Trade-off:** Ensuring high system reliability with redundant components and failover mechanisms versus managing operational costs.
- **Decision:** Implementing automated monitoring tools and fault-tolerant strategies to enhance reliability while balancing cost-effectiveness.

6. **Adaptability vs. Development Time:**

- **Trade-off:** Building adaptable systems that can easily accommodate future changes versus meeting strict project deadlines.
- **Decision:** Embracing agile methodologies and iterative development cycles to continuously adapt and refine the platform based on evolving requirements.

7. **Privacy vs. Data Accessibility:**

- **Trade-off:** Safeguarding user privacy and compliance with data protection regulations versus ensuring convenient access to relevant user data.
- **Decision:** Implementing robust privacy controls and data anonymization techniques while providing secure yet accessible data access to authorized users.

These design trade-offs are carefully evaluated and balanced throughout the system design process to align with the platform's overarching goals and stakeholders' needs.

## 4.4 Subsystem Decomposition

In designing the job posting and searching platform, subsystem decomposition breaks down the system into manageable components, each responsible for specific functionalities. This approach enhances modularity, scalability, and maintainability while fostering clear separation of concerns. The subsystems identified for the platform include:

### 9. User Management Subsystem:

- Responsible for user authentication, authorization, and profile management.
- Components: Authentication service, authorization middleware, profile management module.

### 10. Job Management Subsystem:

- Manages job listings, applications, and interactions between job seekers and employers.
- Components: Job listing service, application management module, communication interface.

### 11. Search and Recommendation Subsystem:

- Facilitates job search functionalities based on user preferences and provides job recommendations.
- Components: Search engine, recommendation algorithm, filtering and sorting utilities.

### 12. Administrative Subsystem:

- Supports administrative tasks such as user management, analytics, and system monitoring.
- Components: Admin dashboard, analytics engine, monitoring and logging services.

### **13. Integration Subsystem:**

- Ensures seamless integration with external services, APIs, and data sources.
- Components: API gateway, data integration layer, third-party service connectors.

### **14. Security Subsystem:**

- Implements security measures including access control, data encryption, and threat detection.
- Components: Access control module, encryption service, security monitoring tools.

### **15. User Interface Subsystem:**

- Manages the presentation layer, user interactions, and responsive design across platforms.
- Components: Web interface, mobile application UI, responsive design framework.

Each subsystem encapsulates specific functionalities and interacts with others through well-defined interfaces and protocols.

## **4.5 Design Phase Models**

### **4.5.1 Class Modeling**

Class modeling in the context of the job posting and searching platform involves defining the essential entities and their relationships to capture the system's structure and behavior. This section focuses on identifying classes, attributes, methods, and their associations to facilitate a clear understanding of the system's design.

User

- id: string
- firstName: string
- middleName: string
- lastName: string
- email: string



- phoneNumber: string
- location: string
- password: string
- gender: string
- getAppliedJobs(): JobPosting[]

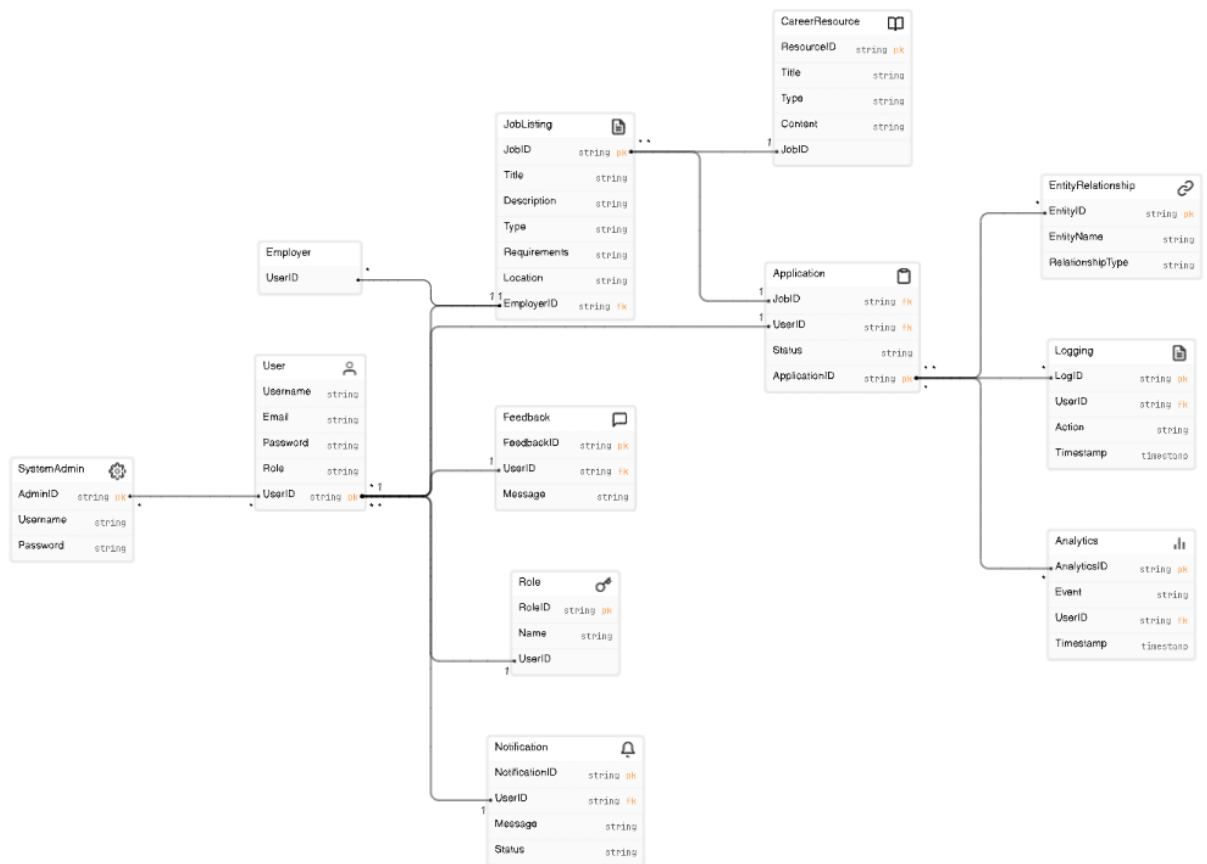
#### UserDetails

- userId: string
- education: Education[]
- experience: Experience[]
- skills: string[]
- languages: string[]
- certifications: Certification[]
- honorsAndAwards: string[]
- updateProfile(): void

#### Job

- id: string
- title: string
- location: string
- type: string
- level: string
- salary: number

- description: string
- responsibilities: string
- requirements: string
- remoteOption: string
- userId: string

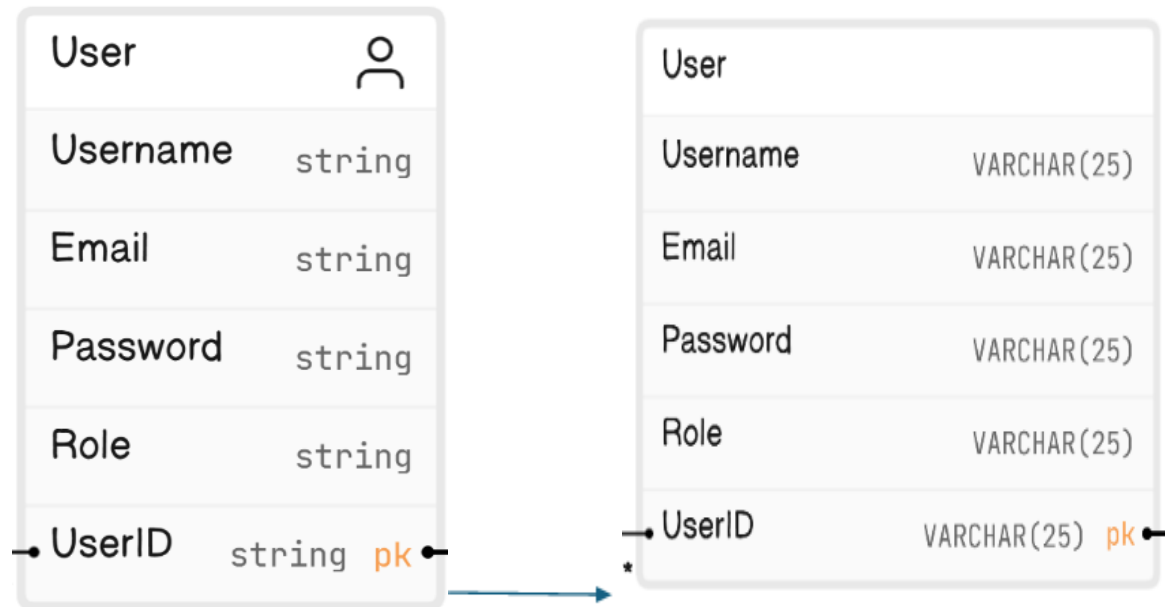


## 4.5.2 Persistent Model

### 4.5.2.1 Mapping Class Diagram to Relation

Mapping the class diagram to relational tables involves translating each class and its attributes into corresponding tables and columns in a relational database schema. Here's how we can map the classes from the provided class diagram to relational tables:

## Relational Schema Mapping:



### 4.5.2.2 Normalization

#### First Normal Form (1NF)

Table Name	Attributes
User	id (PK), firstName, middleName, lastName, email, phoneNumber, location, password, gender
Education	id (PK), institution, degree, fieldOfStudy, startYear, endYear, userId (FK to User)
Experience	id (PK), company, position, description, startYear, endYear, userId (FK to User)
Certification	id (PK), name, issuingOrganization, issueDate, expirationDate, userId (FK to User)

JobPosting	id (PK), title, location, type, level, salary, description, responsibilities, requirements, remoteOption, userId (FK to User)
SavedJob	id (PK), userId (FK to User), jobId (FK to JobPosting), savedAt
AppliedJob	id (PK), userId (FK to User), jobId (FK to JobPosting), appliedAt

All attributes have atomic values.

## Second Normal Form (2NF)

Table Name	Attributes
User	id (PK), firstName, middleName, lastName, email, phoneNumber, location, password, gender
Education	id (PK), institution, degree, fieldOfStudy, startYear, endYear, userId (FK to User)
Experience	id (PK), company, position, description, startYear, endYear, userId (FK to User)
Certification	id (PK), name, issuingOrganization, issueDate, expirationDate, userId (FK to User)
JobPosting	id (PK), title, location, type, level, salary, description, responsibilities, requirements, remoteOption, userId (FK to User)
SavedJob	id (PK), userId (FK to User), jobId (FK to JobPosting), savedAt
AppliedJob	id (PK), userId (FK to User), jobId (FK to JobPosting), appliedAt

In this case, all entities are already in 2NF since each non-key attribute is fully dependent on the primary key.

### Third Normal Form (3NF)

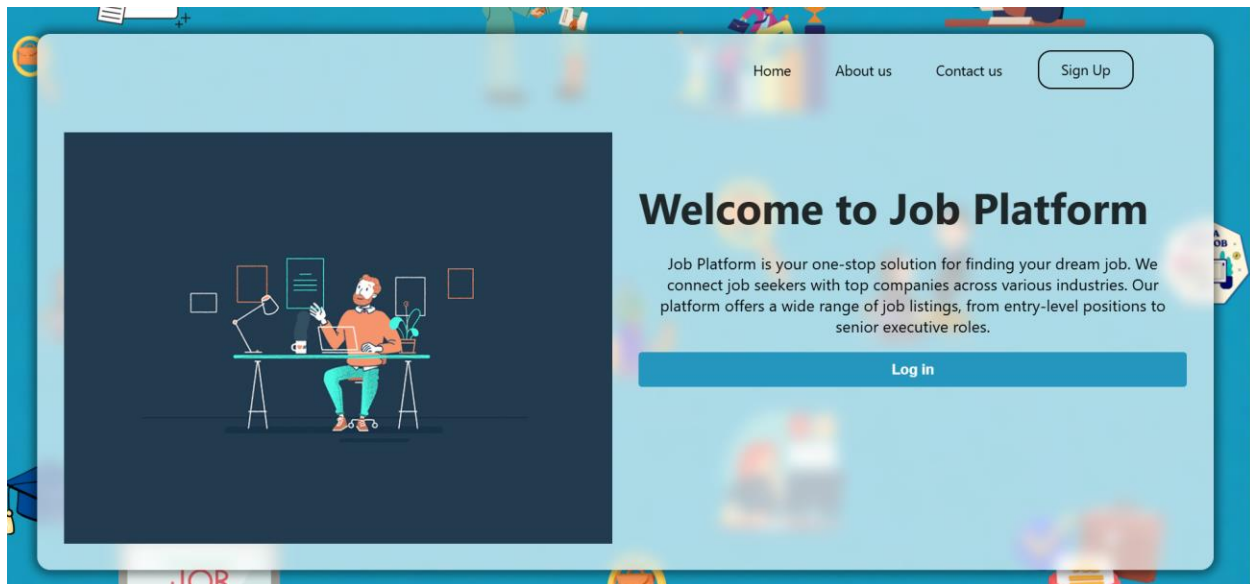
Table Name	Attributes
User	id (PK), firstName, middleName, lastName, email, phoneNumber, location, password, gender
Education	id (PK), institution, degree, fieldOfStudy, startYear, endYear, userId (FK to User)
Experience	id (PK), company, position, description, startYear, endYear, userId (FK to User)
Certification	id (PK), name, issuingOrganization, issueDate, expirationDate, userId (FK to User)
JobPosting	id (PK), title, location, type, level, salary, description, responsibilities, requirements, remoteOption, userId (FK to User)
Saved Job	id (PK), userId (FK to User), jobId (FK to JobPosting), savedAt
AppliedJob	id (PK), userId (FK to User), jobId (FK to JobPosting), appliedAt

Again, the entities are in 3NF since there are no transitive dependencies.

#### Explanation:

- **First Normal Form (1NF):** Ensures atomicity of all attributes in each table.
- **Second Normal Form (2NF):** All attributes are dependent only on the primary key (no partial dependencies).
- **Third Normal Form (3NF):** Eliminates transitive dependencies by ensuring that all attributes are functionally dependent on the primary key and not on other non-key attributes.

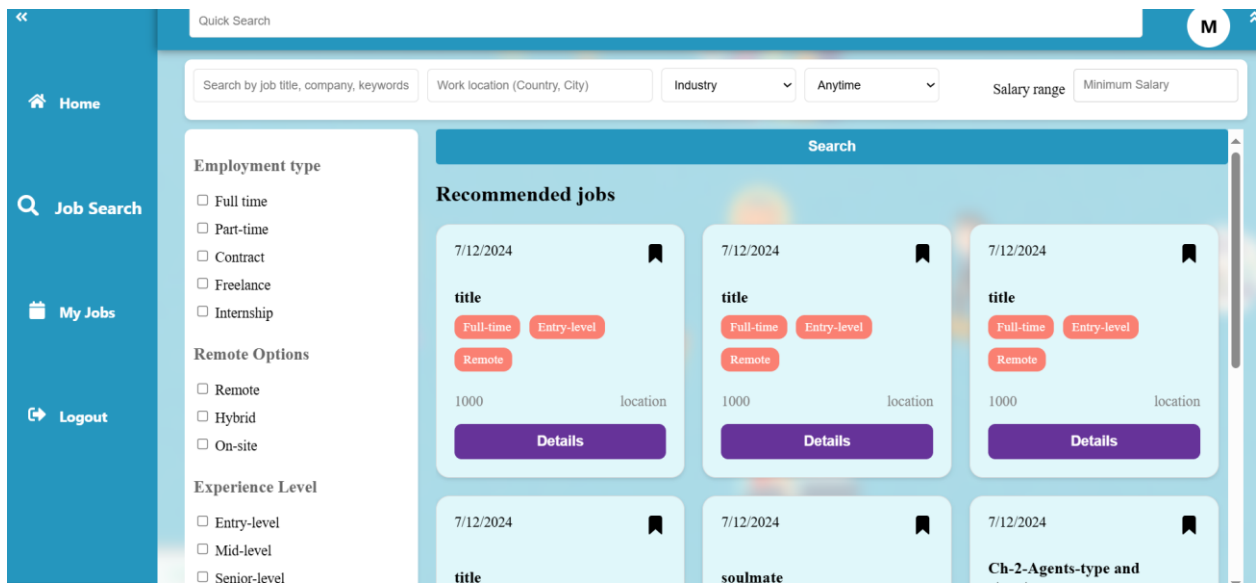
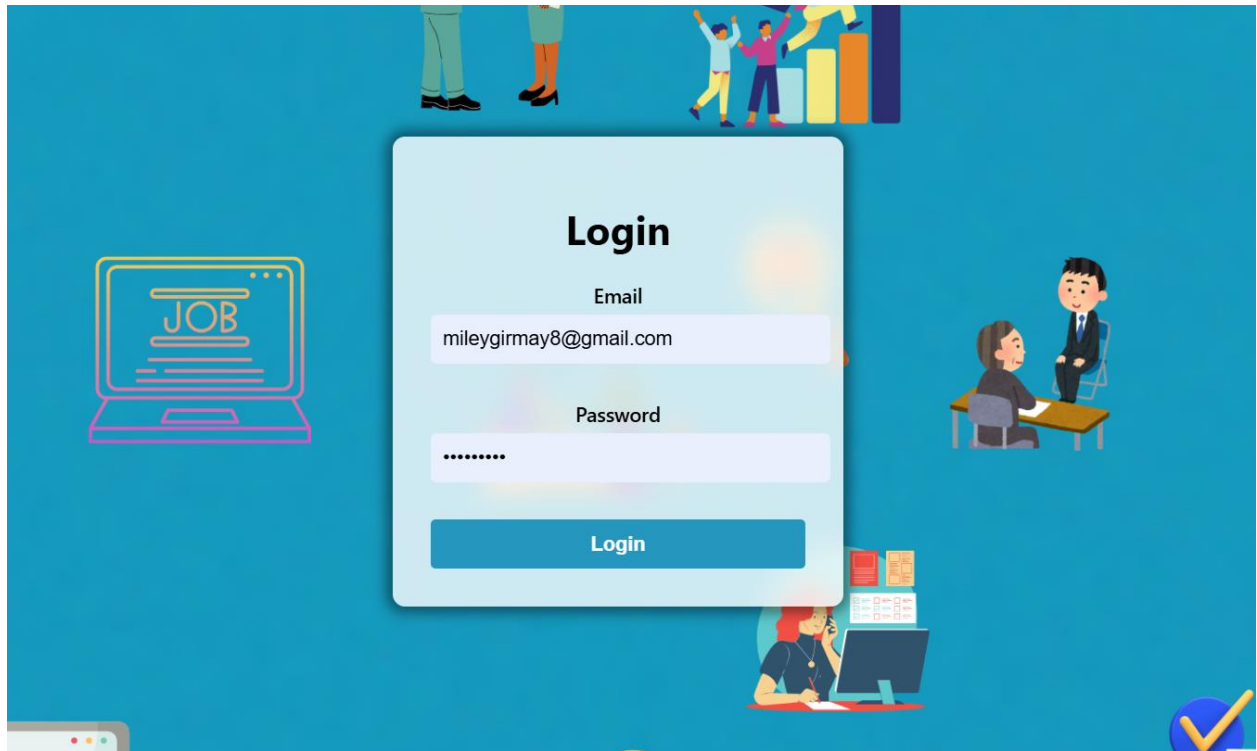
### **4.5.3 User Interface Design**



The registration form is titled "Registration" and is located on the right side of the page. It includes a large illustration on the left showing a person at a desk. The form fields are as follows:

- First Name**: Input field with placeholder "First Name"
- Middle Name**: Input field with placeholder "Middle Name"
- Last Name**: Input field with placeholder "Last Name"
- Email**: Input field with placeholder "Email"
- Phone Number**: Input field with placeholder "Phone number"
- Password**: Input field with placeholder "Enter password"
- Confirm Password**: Input field with placeholder "Confirm password"
- Gender**: Radio buttons for ☐ Male and ☐ Female

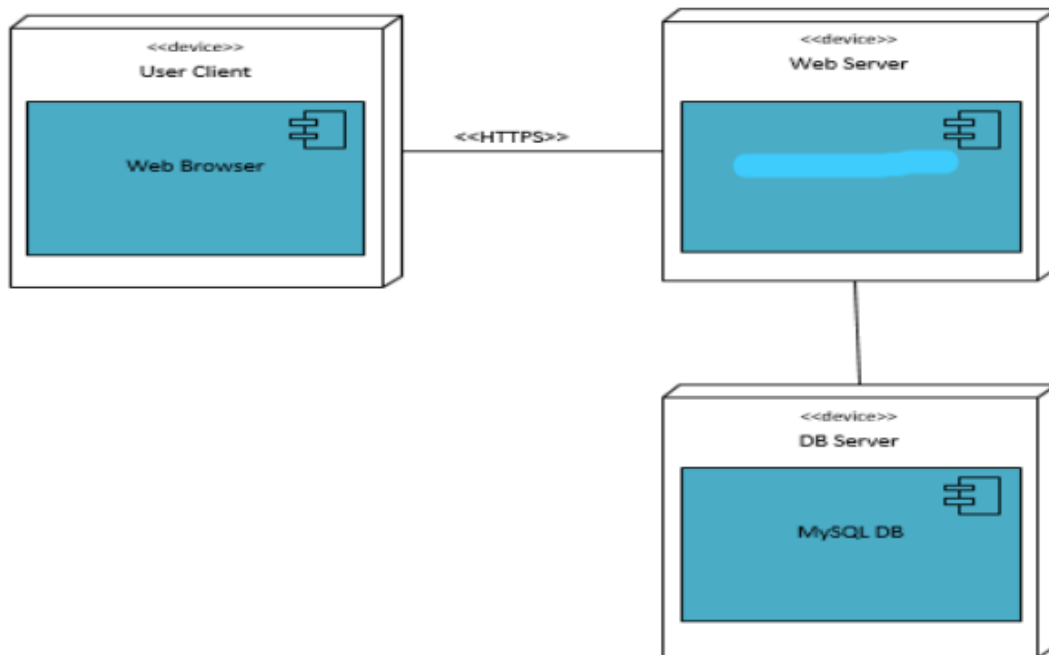
Below the form fields is a blue "Sign up" button. At the bottom of the form, there is a link: "Already have an account? [Sign in](#)".





#### 4.5.4 Deployment Diagram

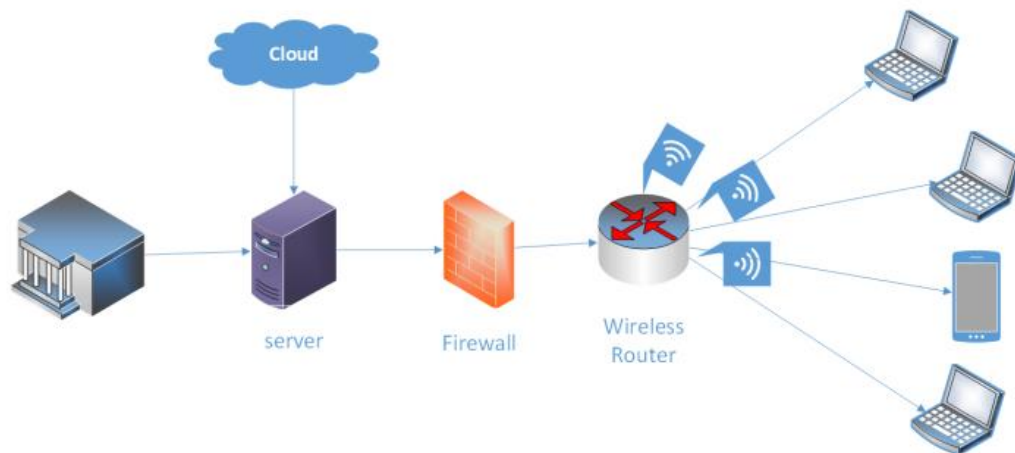
A Deployment Diagram for your job posting and searching platform project should illustrate how software components and hardware resources are deployed across the system. Below is a complete deployment diagram for the Job Posting and Searching Platform:



### 4.5.5 Network Design

Below is a basic outline of the network design for the Job Posting and Searching Platform:

- **Internet Connection:** Ensure reliable high-speed internet connectivity for user access.
- **Router:** Connects the internal network to the internet.
- **Firewall:** Protects against unauthorized access and threats from the internet.
- **Web Server:** Helps enhance security and optimize performance.



## Chapter Five – Implementation

### 5.1 Introduction

In this chapter, we will discuss the implementation details of the Job Posting and Searching Platform. This includes the description of the system architecture, key components, and sample code snippets to demonstrate how various functionalities are achieved. The implementation is based on the design outlined in previous chapters and utilizes modern web development technologies.

#### 5.1.1 Database Implementation

### 5.1.1.1 Database Design

- **Database Selection:** MySQL was chosen as the database management system (DBMS) due to its robust support for relational data structures, transaction management capabilities, and scalability options suitable for web applications.
- **Schema Design:** The database schema is designed to efficiently manage user profiles, job postings, applications, and related data while ensuring data integrity and performance.

### Tables and Relationships

- **User Table:**
    - **Fields:** user\_id (Primary Key), username, password, email, created\_at.
    - **Description:** Stores user credentials, role information, and timestamps for creation and updates.
  - **Job Postings Table:**
    - **Fields:** job\_id (Primary Key), employer\_id (Foreign Key), title, description, location, salary, created\_at.
    - **Description:** Contains details of job postings created by employers, linked to the respective employer's profile.
  - **Applications Table:**
    - **Fields:** application\_id (Primary Key), job\_id (Foreign Key), applicant\_id (Foreign Key), status, date\_applied, created\_at.
    - **Description:** Stores applications submitted by job seekers for specific job postings, tracking application
- |  |    |
|--|----|
| ○ 2.5 Team Organization .....                      | 23 |
| ○ 2.6.1 Requirements Gathering and Analysis.....   | 25 |
| ○ 2.6.2 System Architecture and Design .....       | 26 |
| ○ 2.6.3 Development and Implementation.....        | 26 |
| ○ 2.6.4 Testing and Quality Assurance.....         | 26 |
| ○ 2.6.5 Deployment and Release .....               | 26 |
| ○ 2.6.6 Maintenance and Support .....              | 27 |
| ○ 2.6.7 Project Management and Documentation ..... | 27 |

○ Risk Number - R1 .....	28
○ Risk Number - R2 .....	28
○ Risk Number - R3 .....	28
○ Risk Number - R4 .....	29
○ Risk Number - R5 .....	29
○ Risk Number - R6 .....	29
○ Risk Number - R7 .....	30
○ 1. User Registration and Authentication:.....	32
○ 2. Profile Management: .....	32
○ 3. Job Search and Filtering: .....	33
○ 4. Job Posting and Management: .....	33
○ 5. Saved Jobs:.....	33
○ 6. Application Tracking:.....	33
○ 7. Career Development Resources:.....	33
○ Use Case Identification .....	37
○ Use Case Descriptions.....	37
○ Use Case ID: UC001 - User Registration .....	37
○ Use Case ID: UC002 - User Authentication (Login).....	38
○ Use Case ID: UC003 - Search .....	39
○ Use Case ID: UC004 - Create Admin.....	40
○ Use Case ID: UC005 - Delete Admin.....	40
○ Use Case ID: UC006 - Post Job Listing .....	41
○ Use Case ID: UC007 - Delete Job Listing .....	42
○ Use Case ID: UC008 - Apply for Job .....	42
○ Use Case ID: UC009 - View Applications .....	43
○ Use Case ID: UC010 - View Job Details .....	43
○ Use Case ID: UC011 - Manage User Profile .....	44
○ Use Case ID: UC001 .....	55
○ Use Case ID: UC002 .....	57

○ Use Case ID: UC003 .....	58
○ Use Case ID: UC004 .....	59
○ Use Case ID: UC005 .....	60
○ Use Case ID: UC006 .....	61
○ Use Case ID: UC007 .....	62
○ Use Case ID: UC008 .....	63
○ Use Case ID: UC009 .....	64
○ Use Case ID: UC010 .....	65
○ Use Case ID: UC011 .....	66
○ Relational Schema Mapping: .....	91
○ First Normal Form (1NF) .....	91
○ Second Normal Form (2NF) .....	92
○ Third Normal Form (3NF) .....	93
○ Explanation: .....	93
○ 5.1.1.1 Database Design .....	99
○ Tables and Relationships .....	99
○ 5.1.1.2 Data Handling and Storage .....	102
○ 5.1.1.3 Database Operations .....	102
○ 5.1.2 Interface and Business Logic Implementation .....	102
○ 5.1.2.1 Frontend Interface Design .....	102
○ 5.1.2.2 User Interaction and Experience .....	103
○ 5.2 Sample Code .....	104
○ Chapter Six - Conclusion and Recommendation .....	106
○ 6.1 Conclusion .....	106
○ 6.2 Recommendation .....	106
○ status and timestamps.	

### 5.1.1.2 Data Handling and Storage

- **Data Types:** Utilizes appropriate data types such as VARCHAR, INT, DATE, and TEXT based on the nature of data stored in each field.
- **Normalization:** Ensures data is organized into normalized tables to minimize redundancy and maintain data consistency through 1NF, 2NF, and 3NF.
- **Indexes:** Implements indexes on primary keys (user\_id, job\_id, application\_id) and foreign keys (employer\_id, applicant\_id) to optimize query performance.

### 5.1.1.3 Database Operations

- **CRUD Operations:** Examples of CRUD operations include:
  - **Create:** INSERT INTO users (username, password, email, role) VALUES ('exampleuser', 'hashedpassword', '[example@email.com](mailto:example@email.com)', 'jobseeker');
  - **Read:** SELECT \* FROM job\_postings WHERE location = 'New York';
  - **Update:** UPDATE applications SET status = 'shortlisted' WHERE application\_id = 123;
  - **Delete:** DELETE FROM users WHERE user\_id = 456;

### 5.1.1.4 Data Security and Access Control

- **Encryption:** Implements bcrypt hashing for passwords stored in the users table to enhance data security against unauthorized access.

## 5.1.2 Interface and Business Logic Implementation

### 5.1.2.1 Frontend Interface Design

- **User Interface (UI) Components:**
  - **React Components:** The frontend is built using React, organized into components such as HomePage, JobSearchPage, UserDashboard, JobPostingForm, etc. Each component encapsulates specific functionality and UI elements related to its purpose.
  - **Component Libraries:** Utilizes Material-UI for its robust UI component library, ensuring consistent design and user experience throughout the platform.

- **Responsive Design:** Ensures responsiveness across devices with CSS media queries and flexbox/grid layouts. The UI adapts seamlessly to different screen sizes to enhance accessibility and usability.

#### 5.1.2.2 User Interaction and Experience

- **Navigation and Routing:** Implements React Router for client-side routing. Navigation between pages (e.g., from homepage to job search) is handled through declarative routing components (<Link>, <Route>) to maintain a single-page application (SPA) experience.
- **Forms and Input Validation:** Implements controlled forms in React. Utilizes form validation libraries for validation rules (e.g., required fields, email format) to ensure data integrity before submission.

## 5.2 Sample Code



```
server > JS server.js > app.get('/api/user/email') callback > db.query() callback
// Endpoint to handle user login
app.post('/api/login', (req, res) => {
  const { email, password } = req.body;

  const query = 'SELECT * FROM users WHERE email = ?';
  db.query(query, [email], async (err, results) => {
    if (err) {
      console.error('Error querying the database:', err);
      return res.status(500).json({ message: 'An error occurred. Please try again.' });
    }

    if (results.length === 0) {
      return res.status(401).json({ message: 'Invalid email or password' });
    }

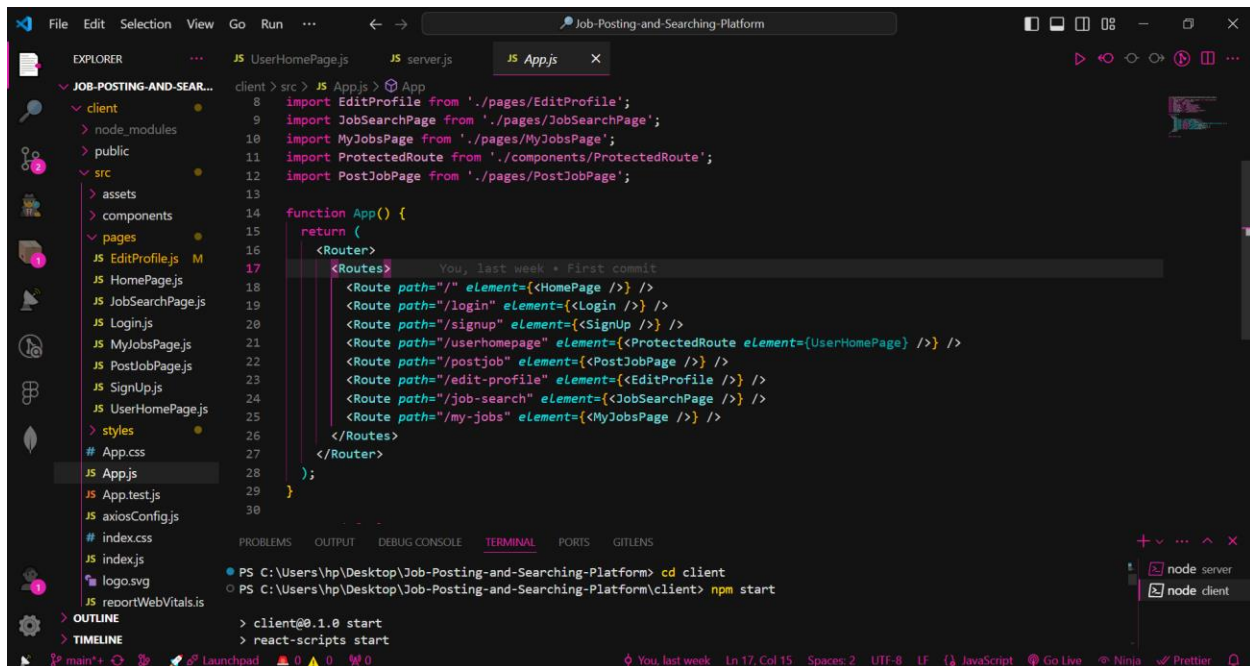
    const user = results[0];
    // Compare the provided password with the stored hashed password
    const match = await bcrypt.compare(password, user.password);
    if (match) {
      req.session.userId = user.id;
      return res.status(200).json({ message: 'Login successful', user });
    } else {
      return res.status(401).json({ message: 'Invalid email or password' });
    }
  });
});
```

PS C:\Users\hp\Desktop\Job-Posting-and-Searching-Platform> cd client  
PS C:\Users\hp\Desktop\Job-Posting-and-Searching-Platform\client> npm start

```
client > src > pages > JS HomePage.js > HomePage
You, 2 days ago | 1 author (You)
///C:\Users\hp\Desktop\Job-Posting-and-Searching-Platform\client\src\pages\HomePage.js
import React from 'react';
import { Link } from 'react-router-dom';
import '../styles/HomePage.css';
import yourGif from '../assets/Desk.gif';

function HomePage() {
  return (
    <div className="homepage">
      <nav className="navbar">
        <div className="nav-links">
          <Link to="/">Home</Link>
          <Link to="/about">About us</Link>
          <Link to="/contact">Contact us</Link>
        </div>
        <div className="signup-button">
          <Link to="/signup">Sign Up</Link>
        </div>
      </nav>
      <div className="container">
        <div className="image">
          <img src={yourGif} alt="Job Posting Platform" />
        </div>
      </div>
    </div>
  );
}
```

PS C:\Users\hp\Desktop\Job-Posting-and-Searching-Platform> cd client  
PS C:\Users\hp\Desktop\Job-Posting-and-Searching-Platform\client> npm start



## Chapter Six - Conclusion and Recommendation

### 6.1 Conclusion

In conclusion, the job posting and searching platform project has effectively addressed the key challenges and requirements identified during the system analysis phase. Through strategic design decisions and careful subsystem decomposition, the platform provides a robust, scalable, and user-friendly solution for both job seekers and employers. The implementation of advanced features such as dynamic form entries, resume uploads, and advanced search filters ensures a comprehensive and efficient user experience. Additionally, the focus on security measures, including data encryption and user privacy, aligns the platform with industry standards and regulations, providing a secure environment for users.

### 6.2 Recommendation

Based on the development and evaluation of the job posting and searching platform, several recommendations can be made for future improvements and enhancements:

16. **Enhanced User Analytics:** Implement more detailed user analytics to better understand user behavior and preferences, allowing for more tailored job recommendations and improved user engagement.
17. **Mobile Application Development:** Develop a dedicated mobile application to complement the web platform, ensuring accessibility and convenience for users on the go.
18. **AI and Machine Learning Integration:** Incorporate AI and machine learning algorithms to enhance the recommendation engine, making job searches more relevant and personalized.
19. **Partnerships with Educational Institutions:** Establish partnerships with educational institutions to provide career resources and support for students transitioning to the workforce.
20. **Continuous Security Enhancements:** Regularly update and enhance security protocols to protect user data and maintain trust in the platform.
21. **User Feedback Mechanisms:** Implement mechanisms for collecting and analyzing user feedback to continuously improve the platform based on user needs and suggestions.

These recommendations aim to further refine the platform, ensuring it remains a leading solution in the job market ecosystem.