HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEMS

ETHIOPIAN JOB SEARCH PLATFORM

By:

Group members	ID Number
1. ABDREZAK ZEYNU	1930/14
2. ABDULMALIK MUZE…	1970/14
3. ADNANNUREDIN	2067/14
4. EZEDINHUSSEN	2911/14
5. EMRAN GERO…	2680/14

Feb 2025

Haramaya University, Ethiopia

HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEMS

ETHIOPIAN JOB SEARCH PLATFORM

A Project Submitted to the Department of Information Systems of Haramaya University in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Information Systems.

By:

Group members	ID Number
1. ABDREZAK ZEYNU	1930/14
2. ABDULMALIK MUZE…	1970/14
3. ADNANNUREDIN	2067/14
4. EZEDINHUSSEN	2911/14
5. EMRAN GERO…	2680/14

Advisor: Mr. Tilahun M.

Co-advisor: Mr. Milkesa A.

Feb 2025

Haramaya University, Ethiopia

HARAMAYA UNIVERSITY

HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEMS

CERTIFICATE OF APPROVAL

We here by certify that we have read and evaluated this Industrial Project I entitled **"ETHIOPIAN JOB SEARCH PLATFORM"** prepared under our guidance by the following students**.** We recommend that it be submitted as fulfilling the Industrial Project I requirement

Name ID

1. ABDREZAK ZEYNU.....1930/14
2. ABDULMALIK MUZE….....1970/14
3. ADNANNUREDIN 2067/14
4. EZEDINHUSSEN.....2911/14
5. EMRAN GERO… 2680/14

Name and Signature of members of Examining Board

Name	Title	Signature	Date
	Advisor		
	Examiner 1		
	Examiner 2		
	Examiner 3		

DECLARATION

we, the undersigned, declare that this project is our original work and has not been presented for a degree in any other university, and that all sources of materials for the industrial project I have been duly acknowledged.

Abdulmalik Muze

Feb 2025

This project has been submitted for examination with my approval as an advisor.

Advisor: Mr. Tilahun M.

Feb 2025

Acknowledgment

First and foremost, we would like to thank our Lord, the Almighty, for helping us accomplish this project. We would also like to express our deepest gratitude to our advisor, Mr. Tilahun M., for his invaluable guidance, continuous support, and constructive feedback throughout the course of this project. His expertise and insights have been instrumental in shaping the "Ethiopian Job Search" platform and ensuring its successful development.

We are grateful to the Department of Information Systems at Haramaya University for providing us with the opportunity and resources to undertake this project. Their encouragement and support have been pivotal in fostering our academic growth.

Lastly, we would like to extend our heartfelt thanks to our families, friends, and peers for their unwavering support and encouragement during this challenging yet rewarding journey. Their belief in our abilities has motivated us to persevere and achieve our goals.

Thank you all for your contributions and support.

Table of Contents

List of Tables

List of Figures

Acronyms

API – Application Programming Interface

CI/CD – Continuous Integration / Continuous Deployment

CRUD – Create, Read, Update, Delete

DBMS – Database Management System

ERD – Entity-Relationship Diagram

HTTP – Hypertext Transfer Protocol

HTTPS – Hypertext Transfer Protocol Secure

IDE – Integrated Development Environment

JSE – Job Search Engine

JSON – JavaScript Object Notation

JWT – JSON Web Token

Management System

MVC – Model-View-Controller

Protocol/Internet Protocol

RDBMS – Relational Database

SQL – Structured Query Language

TCP/IP – Transmission Control

UI – User Interface

UML – Unified Modeling Language

UX – User Experience

Executive Summary (Abstract)

web-based recruitment platform designed to streamline the hiring process by leveraging AI-driven job matching, real-time notifications, and data analytics. The platform aims to bridge the gap between job seekers and employers by providing a seamless and efficient job search experience.

One of the system's core features is its AI-driven job matching algorithm, which analyzes job seekers' skills, experience, and preferences to recommend the most relevant job opportunities. This enhances job search efficiency and increases the likelihood of successful employment. Additionally, an application tracking system allows job seekers to monitor their application status while enabling employers to manage candidates efficiently, reducing hiring time and improving the overall recruitment process.

To provide real-time insights, the system includes an analytics dashboard that gives employers a comprehensive view of job post performance, applicant engagement, and hiring trends. The integration of real-time notifications ensures that job seekers and employers receive instant updates on job postings, application status changes, and system alerts, keeping users informed at all times.

An essential feature of the platform is its automated report generation, which provides insights on job trends, user activity, and hiring metrics without requiring manual data compilation. This helps organizations make data-driven hiring decisions and better understand employment patterns.

To guarantee scalability, performance, and security, the system is developed using React.js for the frontend, Node.js for the backend, and MySQL for data storage. AI-powered recommendations utilize machine learning algorithms, while WebSockets enable real-time updates, ensuring a responsive and interactive user experience.

By enhancing job search accessibility, transparency, and efficiency, the Ethiopian Job Search System modernizes the employment landscape, making recruitment more intelligent and data-driven. The platform contributes to the digital transformation of Ethiopia's job market, benefiting both job seekers and employers.

CHAPTER ONE

INTRODUCTION

1.1 Introduction/Background

In today's competitive job market, finding the right job or hiring the right candidate is a challenging task. Traditional recruitment methods, such as newspaper job listings, physical CV submissions, and word-of-mouth referrals, are often inefficient, time-consuming, and lack transparency. To address this issue, our organization aims to develop the Ethiopian Job Search Platform.

The 'Ethiopian Job Search' platform is owned and operated by ‘Ethio Hire-hub’ Organization, a private recruitment company focused on connecting job seekers and employers in Ethiopia. It is web-based system that connects job seekers with employers efficiently. This platform will serve as an intelligent job marketplace, enabling employers to post jobs, shortlist candidates, and track applications, while job seekers can search for jobs, apply, and receive AI-driven recommendations.

Unlike existing job portals, our system will integrate AI-powered job matching to recommend the best jobs to seekers, an application tracking system that allows job seekers and employers to monitor hiring progress, and an analytics dashboard for employers to track job post performance. Additionally, it will provide real-time notifications for job seekers and employers and include automated report generation to offer insights into job market trends and hiring metrics. The system will streamline the recruitment process, reducing hiring time and increasing the efficiency of our private recruitment company. With modern web technologies like React.js, Node.js, and MySQL, the platform will be scalable, secure, and user-friendly, making it a valuable tool for job seekers, employers, and recruitment agencies.

1.2 Motivation of the Project

The Ethiopian job market faces inefficiencies due to reliance on outdated recruitment methods like newspapers and notice boards, making job searching and hiring slow and ineffective. Job seekers struggle to find relevant opportunities, while employers face difficulties in identifying qualified candidates. The lack of a centralized platform leads to fragmented job listings and increased costs for both parties. A digital solution is necessary to modernize the process, making job searching more accessible and hiring more efficient. [1]

The Ethiopian Job Search System aims to bridge this gap by providing a centralized, AI-driven platform that enhances accessibility, efficiency, and transparency. It automates job matching, application tracking, and real-

time notifications, ensuring users stay updated on relevant opportunities. Employers benefit from analytics dashboards and automated reports, enabling data-driven decision-making. By leveraging modern technology, this project supports Ethiopia's digital transformation and promotes a fair and structured job market.

1.3 Statement of the Problem

The job search and recruitment process in Ethiopia is inefficient due to scattered job listings, lack of application tracking, and outdated hiring methods. Job seekers struggle to find suitable opportunities, while employers face difficulties in identifying qualified candidates efficiently. Existing platforms lack AI-powered job matching, real-time notifications, and proper analytics, making recruitment slow and ineffective. Additionally, there is no automated system to generate reports on hiring trends, limiting data-driven decision-making. Without a modern, intelligent recruitment system, both job seekers and employers continue to face challenges.

1.4 Objectives of the Project

1.4.1 General Objective

To develop a web-based job search platform that connects job seekers and employers, streamlining the recruitment process and enhancing efficiency, accessibility, and user satisfaction.

1.4.2 Specific Objectives

- To Develop an AI-driven job matching algorithm to recommend relevant jobs to job seekers based on their skills, experience, and preferences.
- To implement an application tracking system that allows job seekers to monitor their application status and employers to manage candidates efficiently.
- To Design an analytics dashboard to provide employers with real-time insights into job applicant engagement, and hiring trends.
- To Enable real-time notifications for job seekers and employers to receive instant updates about job postings, application status changes, and system alerts.
- To Ensure automated report generation by allowing the system to generate insights on job trends, user activity, and hiring metrics without manual intervention.
- To Develop a scalable and secure system architecture that ensures smooth performance, data security, and user-friendly interactions.

1.5 Feasibility Study

1.5.1 Technical Feasibility

The system will be developed using modern web technologies, including React.js for the frontend, Node.js for the backend, and MySQL for structured data storage. AI-powered job matching will use machine learning algorithms, and WebSockets will enable real-time notifications. Cloud-based hosting will ensure scalability and high availability, making the platform technically viable.

1.5.2 Financial Feasibility

As a private recruitment company, the platform's cost will be managed through affordable cloud hosting, open-source technologies, and revenue generation from job posting fees and premium employer subscriptions. The return on investment (ROI) is high, as the system reduces manual hiring efforts and improves recruitment efficiency.

1.5.3 Operational Feasibility

The platform is designed to be user-friendly, requiring minimal training for job seekers and employers. It integrates automated job matching, application tracking, and report generation, ensuring smooth operations. Admins will monitor security and fraud prevention, making the system reliable and easy to manage.

1.5.4 Schedule Feasibility

A Gantt chart has been developed to outline the timeline for the project's development phases. The chart below illustrates the estimated schedule for each phase:

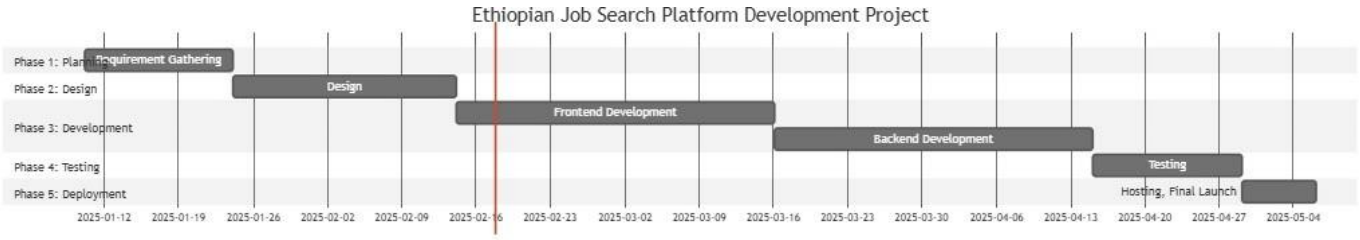

Figure 1 Gantt chart

1.6 Methodology

The Ethiopian Job Search Platform follows the Agile Software Development Methodology, ensuring flexibility, iterative progress, and continuous user feedback. Agile allows for incremental development, enabling the system to be built in small, manageable phases while incorporating improvements at each stage. [2]

1. Information Gathering & Requirement Analysis
    - Conducted surveys and interviews with job seekers, employers, and recruitment agencies.
    - Analyzed existing job search platforms to identify limitations and areas for improvement.
    - Competitive Analysis: A review of similar Ethiopian and international job search platforms will identify gaps and opportunities to differentiate the system.
2. System Design & Modeling
    - Created UML diagrams (Use Case, Class, Activity, and Sequence diagrams) for system visualization.
    - Designed the database structure (ERD, table normalization) for efficient data management.
    - Developed UI/UX wireframes for user-friendly navigation.
    - Real-Time Notifications: WebSocket will be used for real-time updates, allowing job seekers and employers to receive instant notifications for job postings, application statuses, and system alerts.
    - Version Control: Git will be used for tracking changes in the codebase, enabling collaboration among team members and maintaining version integrity.
3. Development Phase
    - Frontend: React.js for an interactive user interface.
    - Backend: Node.js with Express.js for handling system logic and API requests.
    - Database: MySQL for structured data storage and retrieval.
    - AI Integration: Machine learning algorithms for job recommendations.
    - WebSockets: Real-time notifications for job updates and application tracking.
    - Agile Methodology: The project will follow an Agile development approach, emphasizing iterative development and regular feedback loops. This will ensure flexibility and responsiveness to changing requirements throughout the development process.

4. Testing & Deployment
    - Performed unit testing, integration testing, and user acceptance testing (UAT).
    - Hosted the system on a cloud server to ensure scalability and high availability.
5. Maintenance & Future Enhancements
    - Regular updates based on user feedback.
    - Potential future improvements like mobile app development and AI-driven interview scheduling.

1.7 Scope and Delimitation

1.7.1 Scope

1. User Roles:

- Job Seekers: Can create a profile, browse available jobs, and apply for jobs.
- Employers: Can post jobs, view applications, and manage job postings.
- Admins: Can manage users, view job postings, track applications, and generate analytics reports.

2. Key Features:

- AI-driven Job Matching Algorithm: An AI-powered feature that matches job seekers with relevant job postings based on their profile, skills, education, and experience.
- Application Tracking System: A system that allows users to track the status of their job applications (e.g., Applied → Interview → Hired).
- Analytics Dashboard: A dashboard designed for admins to view metrics like the number of job postings, number of applications per job, and overall platform activity.
- Real-time Notifications: Web-based notifications that inform job seekers and employers about new job postings, application status updates, and other relevant events in real-time.

3. Technologies:

- Frontend: React for building the user interface.
- Backend: Node.js to handle server-side logic and data processing.
- Database: MySQL or MongoDB to store user and job-related data.
- AI/ML: A simple, initial AI-driven matching algorithm based on job seeker profiles and job descriptions.

4. Geographical Scope:

- The platform will primarily focus on job seekers and employers located in Ethiopia, targeting universities, recruitment agencies, and local companies.

5. Project Duration:

- The project will be developed in phases, with an initial prototype including basic functionalities and later phases adding AI features and real-time notifications.

1.7.2 Delimitation

The project has the following Delimitations:

1. **Geographical Focus**
    - The platform targets Ethiopian users and is not intended for international markets during its initial implementation.

6. **Mobile Application**
    - The current scope is limited to a web-based platform and does not include the development of a mobile application.

7. **Third-Party Integration**
    - Integration with external job boards, social media, or payment systems will not be included in the initial phase.

8. **Offline Features:**
    - The platform will rely on internet connectivity for real-time notifications, application tracking, and analytics. Offline functionality is not included in the initial release.

1.8 Limitations

The "Ethiopian Job Search" platform is designed to address several inefficiencies in the job search and recruitment process in Ethiopia. However, the system has certain limitations due to resource, technical, and operational constraints. These limitations include:

1. **Limited Data at Launch:**

Challenge: Since the platform will start from scratch without any pre-existing data on job seekers or job postings, the AI-driven job matching algorithm may initially have limited accuracy.

Impact: The algorithm may need time to improve as more data is collected, potentially affecting the quality of job matches for users in the early stages.

2. **Scalability Concerns:**

Challenge: As the number of users grows, performance may degrade without adequate infrastructure. The system might face issues handling a large number of job postings or applications.

Impact: There may be slower response times or data processing delays as the platform scales.

3. **Dependence on User Input:**

Challenge: The effectiveness of the AI-driven job matching algorithm relies heavily on the completeness and accuracy of user profiles (e.g., job seekers must provide detailed skills, education, and experience information).

Impact: If users don't fully update their profiles, it may reduce the precision of job recommendations and the overall success of the job matching system.

1.9 Significance of the Project

The "Ethiopian Job Search" platform is a significant contribution to the Ethiopian job market, addressing key challenges faced by job seekers and employers. Its development brings numerous benefits that enhance accessibility, efficiency, and transparency in the recruitment process.

For Job Seekers:

1**. Improved Accessibility:**

- Enables users to search for job opportunities from anywhere with internet access, reducing the need for physical travel to agencies or offices.

2. **Efficiency in Job Applications:**

- Streamlines the job application process, allowing job seekers to apply directly through the platform, saving time and effort.

3. **Comprehensive Job Search:**

- Offers job seekers a centralized platform to find opportunities across multiple industries and locations, increasing their chances of finding suitable employment.

4. **Enhanced User Experience:**
    - Provides intuitive tools for creating profiles, uploading resumes, and receiving notifications, simplifying the overall job search process.

For Employers:

1. Streamlined Recruitment:
    - Helps employers post job openings and manage applications efficiently, reducing the time and resources spent on hiring processes.
2. Access to a Larger Talent Pool:
    - Expands employers' reach, enabling them to connect with qualified candidates from diverse locations and backgrounds.
3. Simplified Management:
    - Allows employers to organize and track applications in one place, enhancing decision-making and reducing administrative overhead.

For the Ethiopian Job Market:

1. **Modernization of Recruitment:**
    - Transforms the traditional, paper-based job search and recruitment methods into a digital, web based process, aligning with global best practices.
2. **Increased Employment Opportunities:**
    - By connecting job seekers and employers more effectively, the platform can contribute to higher employment rates and economic growth.
3. **Data-Driven Insights:**
    - Enables administrators to generate analytics and reports, providing insights into labor market trends and helping policymakers make informed decisions.

Academic Contribution:

1. Learning Opportunity:
    - Serves as a practical application of theoretical knowledge, allowing the project team to gain hands on experience in system development and Agile methodologies.
    - Foundation for Future Work:

      Establishes a basis for further research and development, encouraging the integration of advanced features such as AI-based job matching and mobile application support. This platform is expected to have a transformative impact on the Ethiopian job market by creating a more connected, efficient, and accessible system for both job seekers and employers. It also highlights the importance of leveraging

technology to address real-world challenges, paving the way for innovation in similar domains.

1.10 Organization of the Document

The document is structured into three chapters. Chapter One: Introduction covers the project's background, motivation, problem statement, objectives (general and specific), feasibility study (technical, financial, and operational), methodology, scope and delimitation, limitations, significance, and organization of the document. Chapter Two: User Requirements and Analysis focuses on existing systems and their limitations, the proposed system, functional and non-functional requirements, use case modeling, user interface prototyping, and activity and sequence diagrams. Chapter Three: System Design details the layered architecture, class modeling, user interface design, database design (including ERDs and table normalization), performance evaluation, and various diagrams like state chart, component, and deployment diagrams. The document concludes with references in IEEE style and appendices for supplementary materials.

CHAPTER TWO

USER REQUIREMENTS AND ANALYSIS

2.1 Overview of Existing Systems

In Ethiopia, existing job search and recruitment platforms primarily operate through websites, social media, and recruitment agencies. Popular platforms like EthioJobs, EmployEthiopia, and GeezJobs allow employers to post job listings while job seekers browse and apply. These platforms provide basic filtering options based on industry, location, and experience but lack advanced AI-driven job matching features. Many companies still rely on manual recruitment through email submissions and walk-in applications. Additionally, social media platforms like Telegram and LinkedIn play a significant role in job postings, where employers share vacancies in groups and channels. However, these systems lack real-time application tracking, analytics dashboards, and AI-driven job recommendations, making recruitment less efficient. The existing platforms are mostly centralized, with limited interaction between employers and job seekers, and do not offer real-time notifications or automated job matching based on skills and preferences.

2.2 Overview of Proposed System

The proposed system, "Ethiopian Job Search," is a web-based recruitment platform designed to improve job search efficiency and hiring processes by integrating AI-driven job matching, real-time notifications, an application tracking system, and an analytics dashboard. Unlike existing Ethiopian job portals that primarily display job listings, this system will use smart automation to enhance user experience. Job seekers will be able to create detailed profiles, receive personalized job recommendations, and track their applications, while employers can post jobs, manage applications, and engage with candidates efficiently. The platform aims to provide a seamless connection between job seekers and employers while ensuring real-time updates and insights for administrators.

Main Features and Their Roles Are:

1. AI-Driven Job Matching Algorithm
   - Automatically recommends jobs to seekers based on their skills, education, experience, and preferences.
   - Uses keyword-based matching initially, with potential for machine learning improvements over time.
2. Application Tracking System
   - Allows job seekers to track the status of their applications with updates like "Applied," "Reviewed," "Interview," and "Hired."
   - Helps employers manage applications efficiently by categorizing and updating applicant status.

3. Real-Time Notification System

- Sends instant web notifications to job seekers when new jobs are posted or when their application status changes.
- Notifies employers when they receive new applications or candidate updates.

4. Analytics Dashboard for Admins

- Displays key metrics like the number of job postings, applications per job, and overall platform activity.
- Helps administrators monitor user engagement, employer activity, and job market trends.

Technology Stack:

- Frontend: React for a modern, responsive user interface.
- Backend: Node.js for server-side processing.
- Database: MySQL or MongoDB for storing user profiles, job postings, and application records.
- AI/ML Frameworks: Initially rule-based, with potential for advanced machine learning models later.

With these features, the Ethiopian Job Search platform will provide a more efficient**,** user-friendly**,** and intelligent job search experience, addressing the challenges in Ethiopia's current recruitment process.

2.3 Functional Requirements

The functional requirements describe the specific functionalities that the system must support. These features are essential to meet the objectives of the platform and provide value to users (job seekers, employers, and admins).

1. User Authentication and Authorization

- Job Seeker's: Ability to create, edit, and delete profiles, including personal information, skills, education, work experience, and preferences.
- Employers: Ability to register, create employer profiles, and manage job postings.
- Admins: Admins will have access to manage users, view job postings, and generate reports.
- Login/Logout: All users must have login functionality with secure authentication (e.g., email/password or OAuth for social media integration).

2. Job Seeker Features

- Profile Management: Job seekers can input and update personal details, professional skills, education, and experience.

- Job Search: Job seekers can search for jobs based on keywords, skills, location, industry, and other filters.
- Job Matching: Based on their profile, job seekers will receive AI-driven job recommendations.
- Application Process: Job seekers can apply for jobs by submitting applications directly through the platform.
- Application Tracking: Job seekers can view the status of their applications (e.g., "Applied," "Reviewed," "Interview," "Hired").

3. Employer Features

- Job Posting: Employers can create and manage job postings, including job descriptions, required skills, qualifications, location, and deadlines.
- Application Management: Employers can view and manage applications, including shortlisting candidates, sending interview invitations, and tracking the progress of applicants.
- Job Posting Analytics: Employers can see the number of applications each job posting receives and view application status.

4. Admin Features

- User Management: Admins can manage both job seekers and employer accounts (activate, deactivate, or delete accounts).
- Analytics Dashboard: Admins will have access to key metrics such as job posting counts, applications per job, and user activity. They can generate reports and insights.

5. Real-Time Notifications

- Job Seekers: Job seekers will receive real-time notifications about new job postings, application status updates (e.g., "Under Review," "Interview Scheduled"), and employer responses.
- Employers: Employers will be notified when they receive new job applications, when a job seeker updates their application status, or when their job posting is about to expire.

6. AI-Driven Job Matching

Job Recommendations: The platform will suggest jobs to job seekers based on the profile information they provide (skills, experience, location, etc.).

Improvement Over Time: The matching algorithm will improve as more data is collected about user preferences and job market trends.

7. Data Security and Privacy

Encryption: All sensitive user information (e.g., passwords, personal details) will be encrypted.

Access Control: Different roles (job seekers, employers, admins) will have restricted access based on their level of authorization.

8. Application Tracking System

Job Seekers: They can see real-time updates on the status of each job application they submit (e.g., "Applied," "Under Review," "Interview," "Hired").

Employers: Employers can update application statuses, move candidates through stages (e.g., screening, interview), and make decisions on hiring.

These functional requirements will help guide the development of the system, ensuring that each feature serves its intended purpose for users while aligning with the overall objectives of the "Ethiopian Job Search" platform.

2.4 Supplementary Specification

2.4.1 Business Rules

1. Registration:
    - All users (job seekers and employers) must create accounts before accessing the platform's features.

2.Job Posting Policies:
    - Employers are required to provide complete and accurate job details, including job title, qualifications, location, and deadlines.
    - Expired job postings will be automatically archived.

3.Application Limits:
    - Job seekers can only apply to open job postings that meet the deadline.

4.Administrative Oversight:
    - Administrators reserve the right to suspend or deactivate accounts that violate platform policies, such as posting inappropriate content or submitting fraudulent applications.

2.4.2 Non-Functional Requirements

1. Performance:
    - The system should handle up to 500 concurrent users without performance degradation.
    - Notifications should be delivered within 2 seconds of triggering an event.

2. Scalability**:**
    - The platform must accommodate a growing user base by supporting additional features, users, and database capacity as needed.
3. Security:
    - User data, including resumes and job postings, must be encrypted during storage and transmission.
    - Authentication will use secure methods, such as password hashing and role-based access control.
4. Usability:
    - The user interface should be intuitive and accessible to users with basic computer literacy.
    - The analytics dashboard must provide clear and interactive data visualizations for administrators.
5. Availability:
    - The platform should maintain an uptime of at least 99.5%, ensuring accessibility to users at all times.

2.5 Use Case Modeling

Use Case Diagram is a graphical representation of a system's behavior. It is used to view how the system is used by different actors from an outsider perspective.

The Ethiopian Job Search platform includes both essential and system-level use cases to capture interactions between users (job seekers, employers, and administrators) and the system. These use cases describe the primary functions the system must perform to meet its objectives.

2.5.1Essential Use Case Modeling

An essential use case is a high-level representation of the required behavior of a system, independent of any particular implementation or design decisions. It is not specific to an existing or proposed system but rather focuses on the fundamental functionality and goals of the system from a user's perspective. It describes the interactions between the actors and the system to achieve a specific goal or task.

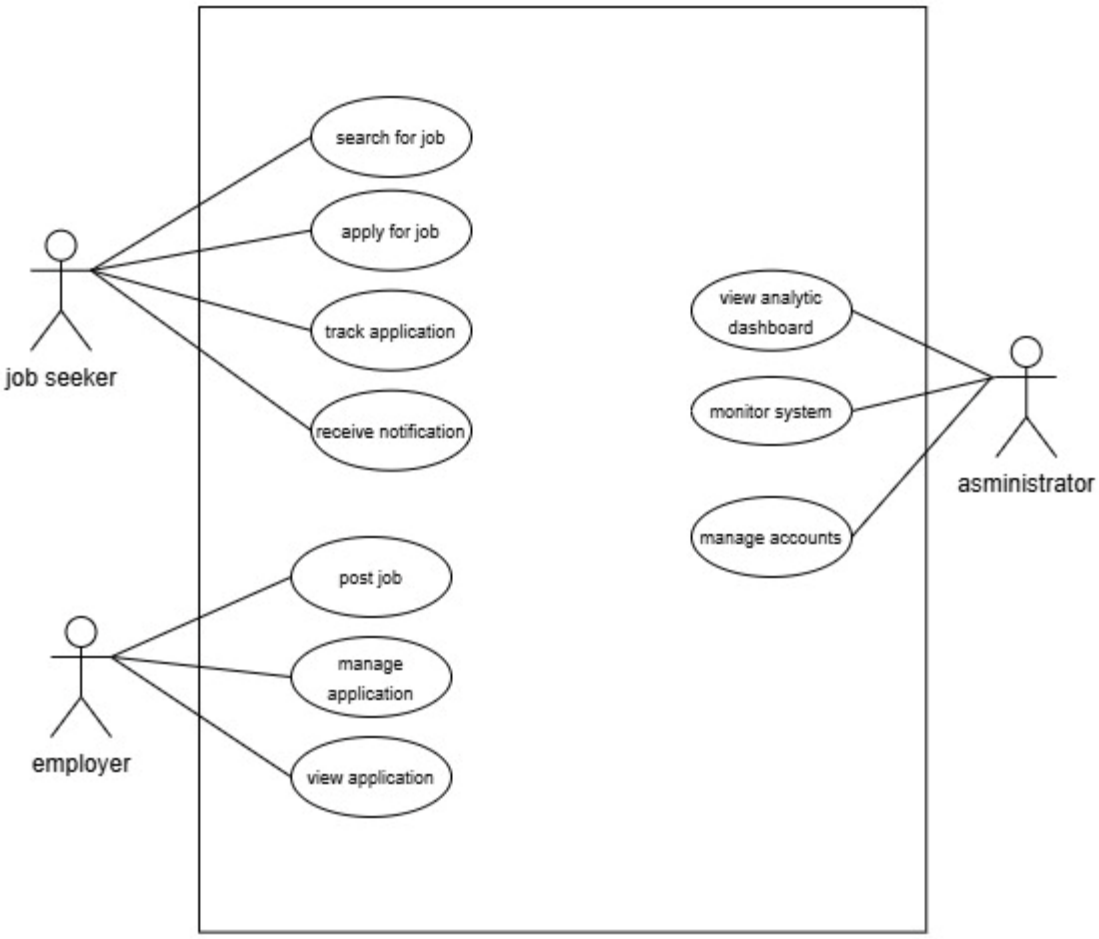

Figure 2 Essential Use case

2.5.1 System Use Case Modeling

System use case modeling is also another technique used to model the behavior of the system like essential use case through the use of use case diagrams. System use case modeling involves identifying the actors and use cases that are required to support the system's functionality, and defining the relationships between them. But system use cases provide a detailed description of the functionality of the system, including the flow of events and the specific interactions between the actors and the system. [3]

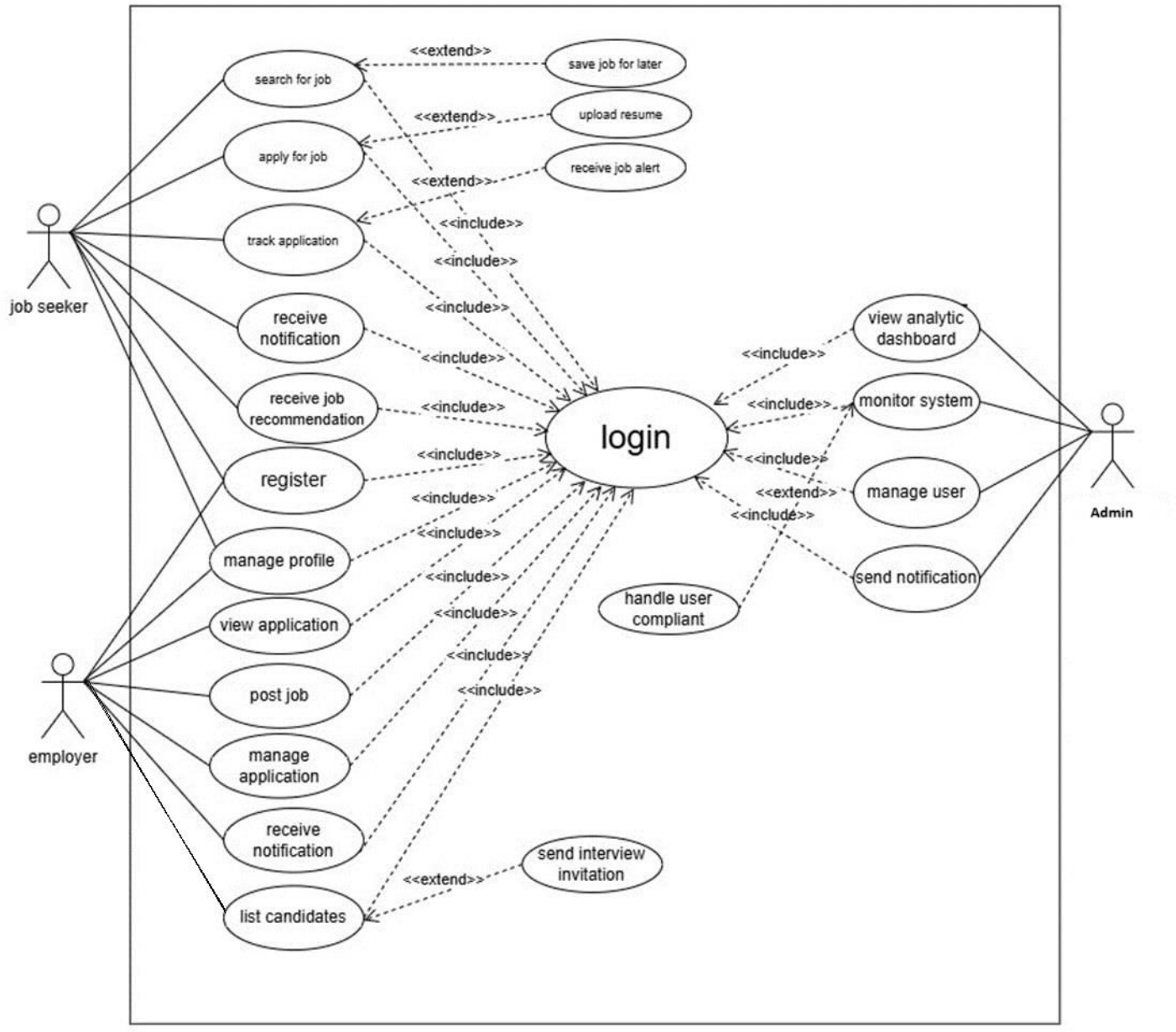

Figure 3 System Use Case

2.5.2 System Use Case Description

A use case description outlines the interactions and actions taken by users and the system to achieve specific goals. It provides a clear understanding of how a system or application is intended to be used and helps guide development by outlining the main steps, inputs, outputs, and potential exceptions. Use case descriptions serve as valuable documentation, ensuring that the system meets the needs and expectations of its users. The following are the use case descriptions for Ethio job search platform. [4]

Table 1System Use Case description for register

Use Case Name	Register
Identifier	UC01
Includes	
Actors	Job Seeker, Employer
Description	The user wants to create a new account on the platform
Overview	The Job Seeker or the Employer registers his/her personal information

Pre-condition	The user accesses the registration page of the platform
Post-condition	The user's account is successfully created and can be used to access the platform using the provided credentials.

Normal Flow of Action

Actor Action	System Action
Step 1: The user selects the "Sign Up" option on the user interface. Step 3: The user fills in the required information in the registration form. Step 4: The user submits the registration form. Step 7: The user receives a confirmation message that the account creation was successful.	Step 2: The system presents a registration form with fields to enter necessary information based on the user type. Step 5: The system validates the entered information, ensuring that the email is unique and the password meets the required criteria. Step 6: The system creates a new account for the User and assigns a unique User ID. Step 8: The use case ends.

Alternative Course of Action

A3.1: The user entered an email that is already associated with an existing account. A3.2: The user enters password that does not meet the required criteria.	A4.1: The system displays an error message, informing user to choose a different email. A4.2: The system prompts user to enter a valid password and provides guidance on the requirements. A4: The system returns to step 3. A5: The Use case ends.

Table 2 System Use Case description for login

Use Case Name	Login
Identifier	UC02
Includes	
Actors	Job Seeker, Employer, Administrator
Description	The user logs into the platform using their credentials.
Overview	The actors enter the required information and the system will authenticate
Pre –condition	Open the application The actors must have an existing account
Post-condition	The user successfully logs in and accesses their dashboard.

Normal Flow of Action	
Actor Action	System Action
Step 1: The user selects the "Login" option Click Login Button	Step 2: The system displays the login form.
Step 3: The user enters their username and password.	Step 4: The system verifies the entered credentials. Password
	Step 5: If credentials are valid, the system grants access to the user's dashboard.
Step 6: The user is redirected to their respective dashboard.	Step 7: The use case ends.
Alternative Course of Action	
A3.1: The user enters incorrect credentials.	A4.1: The system displays an error message.
3.2: The user forgets their password.	A4.2: The system provides a password recovery option.
A4: The system returns to Step 2.	A5: The use case ends.

Table 3 System Use Case description for search job

Use Case Name	Search Job
Identifier	UC03
Actor	Job Seeker
Description	The job seeker searches for job opportunities based on various criteria.
Pre-condition	The user must be logged in.
Post-condition	The system displays relevant job listings.
Normal Flow of Action	
Actor Action	System Action
Step 1: The job seeker selects the "Search Jobs" option. Step 3: The user enters keywords, location, or filters. Click Search Button Step 6: The job seeker views the results.	Step 2: The system presents a search bar and filters. Step 4: The system retrieves matching job listings. Step 5: The system displays the job listings. Step 7: The use case ends.
Alternative Course of Action	
A3.1: The user enters invalid search criteria. A3.2: No jobs match the criteria A4: The system returns to Step 2	A4.1: The system displays an error message. A4.2: The system suggests alternative search options. A5: The use case ends.

Table 4 System Use Case description for Applying for a Jobs

Use Case Name	Apply for Jobs
Identifier	UC04
Includes	Upload resume
Actor	Job Seeker
Description	The job seeker applies for a job by submitting their resume.
Pre-condition	The user must be logged in. There should exist some new job offers
Post-condition	The job application is successfully submitted.

Normal Flow of Action	
Actor Action	**System Action**
Step 1: The user selects a job posting.	
	Step 2: The system displays job details.
Step 3: The user clicks "Apply Now".	
	Step 4: The system prompts the user to upload a resume or select a saved one.
Step 5: The user selects and submits their resume.	
	Step 6: The system validates the resume format.
	Step 7: The system submits the application to the employer.
Step 8: The job seeker receives a confirmation.	
	Step 9: The use case ends.
Alternative Course of Action	
A3.1: The user has not uploaded a resume.	A4.1: The system prompts the user to upload a resume.
A3.2: The uploaded resume is in an invalid format.	A4.2: The system displays an error and requests a valid format.
A4: The system returns to Step 3.	A5: The use case ends

Table 5 System Use Case description for post jobs

Use Case Name	Post job
Identifier	UC05
Includes	Login
Actor	Employer
Description	The employer posts a new job listing.
Pre-condition	The employer must be logged in.
Post-condition	The job posting is published.
Normal Flow of action	
Actor Action	System Action
Step 1: The employer selects "Post a Job".	Step 2: The system displays a job posting form.
Step 3: The employer fills in job details.	Step 4: The system validates the input.
Step 5: The employer submits the job posting.	Step 6: The system saves and publishes the job.
Step 7: The employer receives a confirmation.	Step 8: The use case ends.
Alternative Course of Action	
A3.1: The employer leaves required fields empty.	A4.1: The system prompts the employer to complete the missing fields.
A3.2: The employer enters invalid salary details	A4.2: The system requests valid salary information.
A4: The system returns to Step 3	A5: The use case ends.

Table 6 System Use Case Description for receive notification

Use Case Name	Receive notification
Identifier	UC07
Includes	
Actor	Job Seeker, Employer
Description	The user receives real-time notifications about job postings, applications, or system updates.
Pre-condition	The job seeker must be logged in.
Post-condition	The notification is displayed to the user.
Normal Flow of action	
Actor Action	System Action
Step 4: The user views the notification.	Step 1: The system detects a relevant event (new job posting, application update, etc.) Step 2: The system generates a notification. Step 3: The system delivers the notification to the user's dashboard or sends an email alert. Step 5: The use case ends.
Alternative Course of Action	
A4.1: The user has disabled notifications	A5.1: The system does not send notifications.

Table 7 System Use Case Description for upload resume

Use Case Name	Upload resume
Identifier	UC06
Includes	
Actor	Job Seeker
Description	The job seeker uploads their resume to the platform for job applications.
Pre-condition	The job seeker must be logged in.
Post-condition	The resume is successfully uploaded and stored in the system.
Normal Flow of action	
Actor Action	System Action
Step 1: The job seeker navigates to the "Upload Resume" section.	Step 2: The system displays the resume upload interface.
Step 3: The job seeker selects a resume file from their device.	Step 4: The system validates the file format (PDF, DOC, etc.)
Step 5: The job seeker clicks the "Upload" button.	Step 6: The system stores the resume and links it to the user's profile.
Step 7: The job seeker receives a confirmation message.	Step 8: The use case ends.
Alternative Course of Action	
A3.1: The user selects an unsupported file format. A3.2: The uploaded file exceeds the allowed size limit. A4: The system returns to Step 2	A4.1: The system displays an error message and requests a valid format. A4.2: The system prompts the user to upload a smaller file. A5: The use case ends.

Table 8 System Use Case description for managing an Account

Use Case Name	View application	
Identifier	UC10	
Includes		
Actor	Job Seeker	
Description	The job seeker views their submitted job applications.	
Pre-condition	The job seeker must be logged in.	
Post-condition	The user sees a list of their job applications.	
Normal flow of Action		
Actor Action		System Action
Step 1: The job seeker selects "My Applications" Step 4: The user reviews the application details		Step 2: The system retrieves submitted job applications. Step 3: The system displays the applications with status (Pending, Reviewed, Accepted, Rejected). Step 5: The use case ends.

Table 9 use case for Track application

Use Case Name	Track application	
Identifier	UC09	
Includes		
Actor	Job seeker	
Description	The job seeker tracks the status of their job applications.	
Pre-condition	The user must have submitted a job application.	
Post-condition	The user sees the latest application status.	
Normal flow action		
Actor Action		System Action
Step 1. The job seeker selects an application to track. 3.Click on a specific Employer Step 4: The user reviews the status and any employer feedback.		Step 2: The system retrieves the current status of the application Step 3: The system displays the status (Under Review, Interview Scheduled, Hired, Rejected). Step 5: The use case ends

Table 10 System Use Case description for manage application

Use Case Name	Manage application	
Identifier	UC10	
Includes		
Actor	Employer	
Description	The employer manages job applications submitted by job seekers.	
Pre-condition	The employer must have job posting and received application	
Post-condition	The application status is updated	
Normal flow of Action		
Actor Action		System Action

Step 1: The employer navigates to "Manage Applications". Step 3: The employer reviews applications. Step 5: The employer updates application status (Accepted, Rejected, Interview Scheduled).	Step 2: The system retrieves submitted applications. Step 4: The system displays application details. Step 6: The system notifies the job seeker of the status change. Step 7.Use Case Ends

Table 11 System Use Case description for view analytic dashboard

Use Case Name	View Analytic dashboard	
Identifier	UC11	
Includes		
Actor	Administrator	
Description	The administrator views job platform analytics.	
Pre-condition	The admin must be logged in.	
Post-condition	he analytics data is displayed.	
Normal flow of Action		
Actor Action		System Action
Step 1: The admin selects "Analytics Dashboard" Step 4: The admin reviews the dashboard.		Step 2: The system retrieves and processes job platform data. Step 3: The system displays analytics (Job postings, applications, user activity). Step 5: The use case ends.

Table 12 System Use Case description for view notification

Use Case Name	Monitor system
Identifier	UC12
Includes	
Actor	Administrator
Description	The administrator monitors the system's overall performance and security
Pre-condition	The admin must be logged in.
Post-condition	The admin receives system performance updates
Normal flow of Action	
Actor Action	System Action
Step 1: The admin navigates to "System Monitoring".	Step 2: The system retrieves logs and performance metrics.
	Step 3: The system displays data (server uptime, error logs, user activity).
Step 4: The admin reviews the system status.	Step 5: The use case ends.
Alternative Course of Action	
3a.the system generates and displays “no new notification found” message	

2.6 User Interface Prototyping

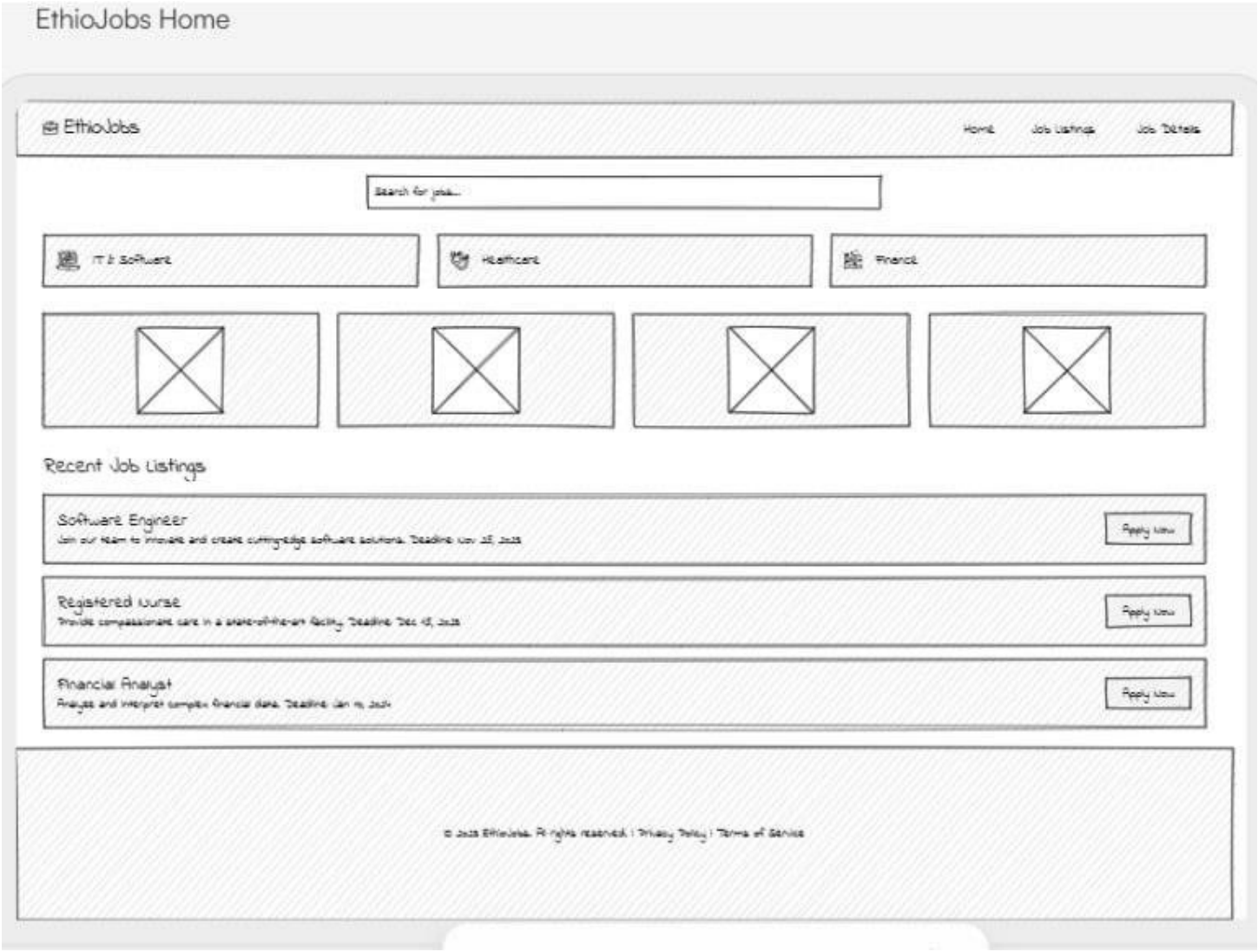

Figure 4 User Interface Prototyping

2.7 Activity and Sequence Diagrams

2.7.1 Activity Diagrams

Activity diagrams represent workflows and show the step-by-step flow of actions in the system.

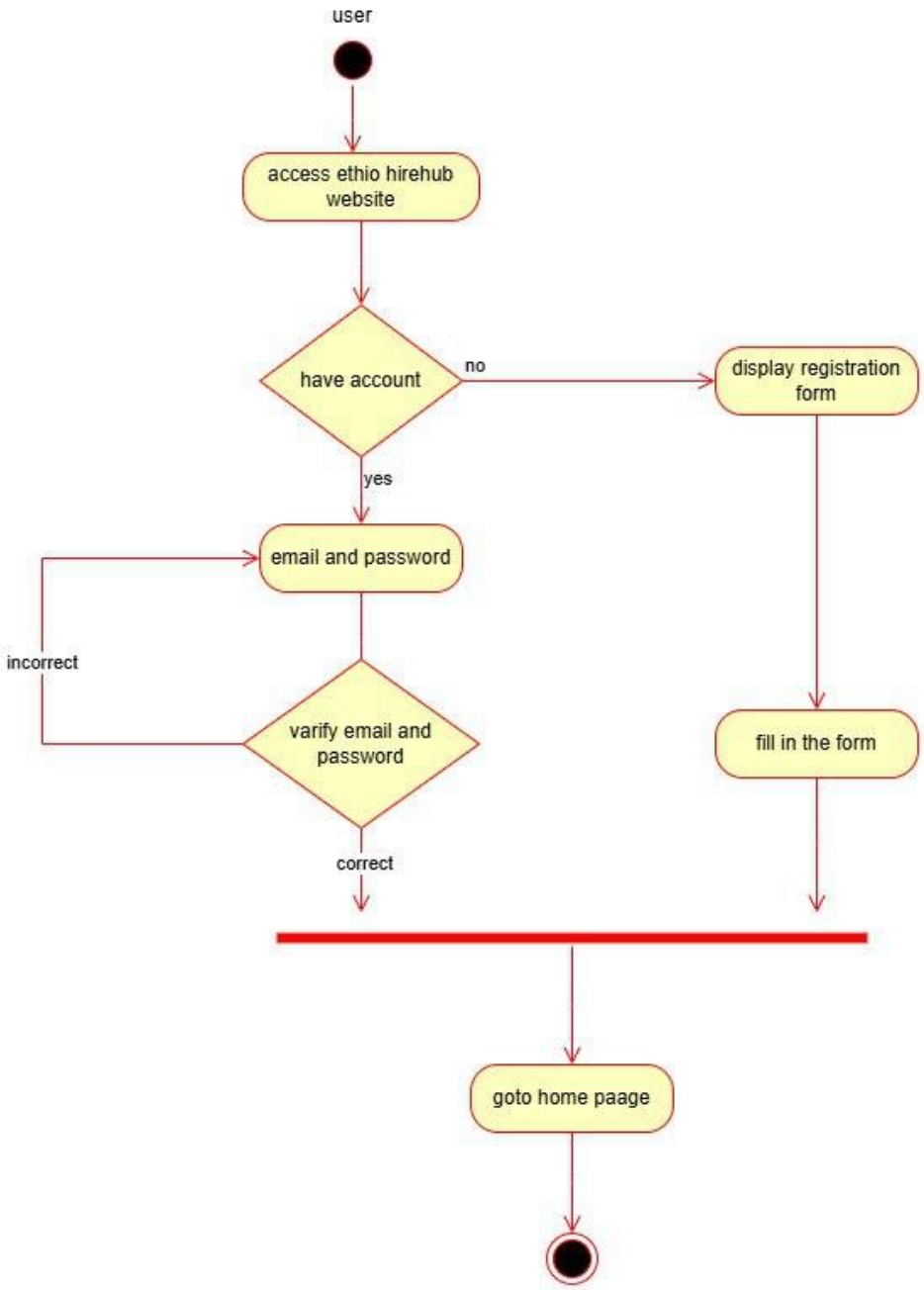

Figure 4 Activity Diagram for login and registration

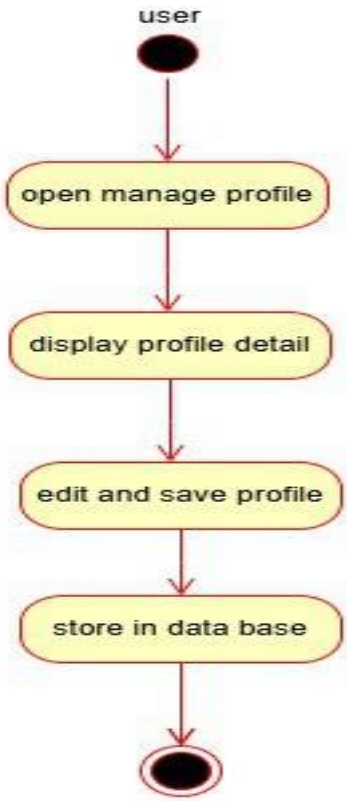

Figure 5 activity diagram for manage profile

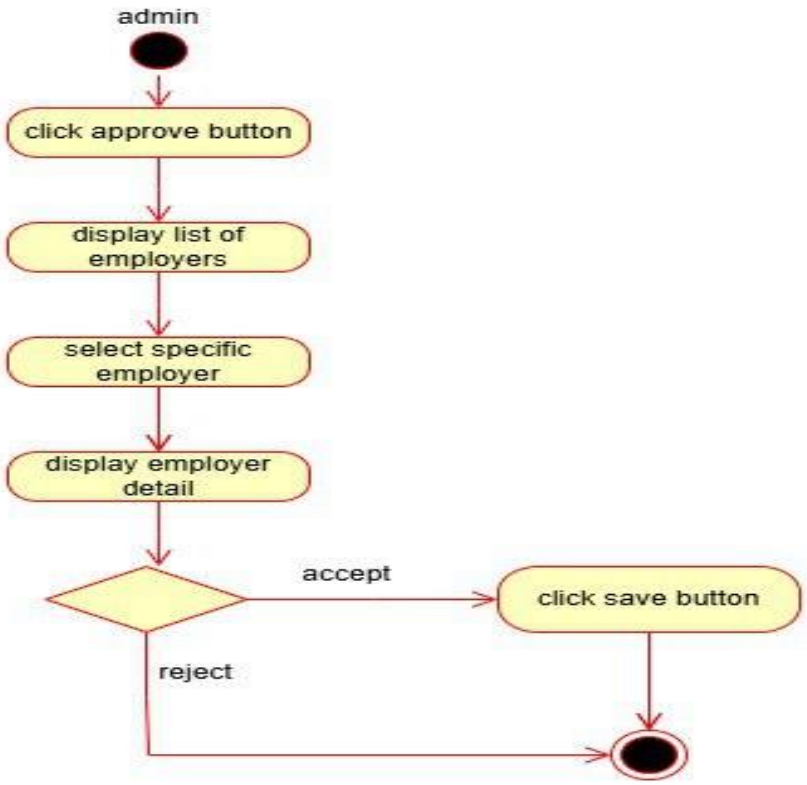

Figure 6 Activity Diagram for Approve Employer

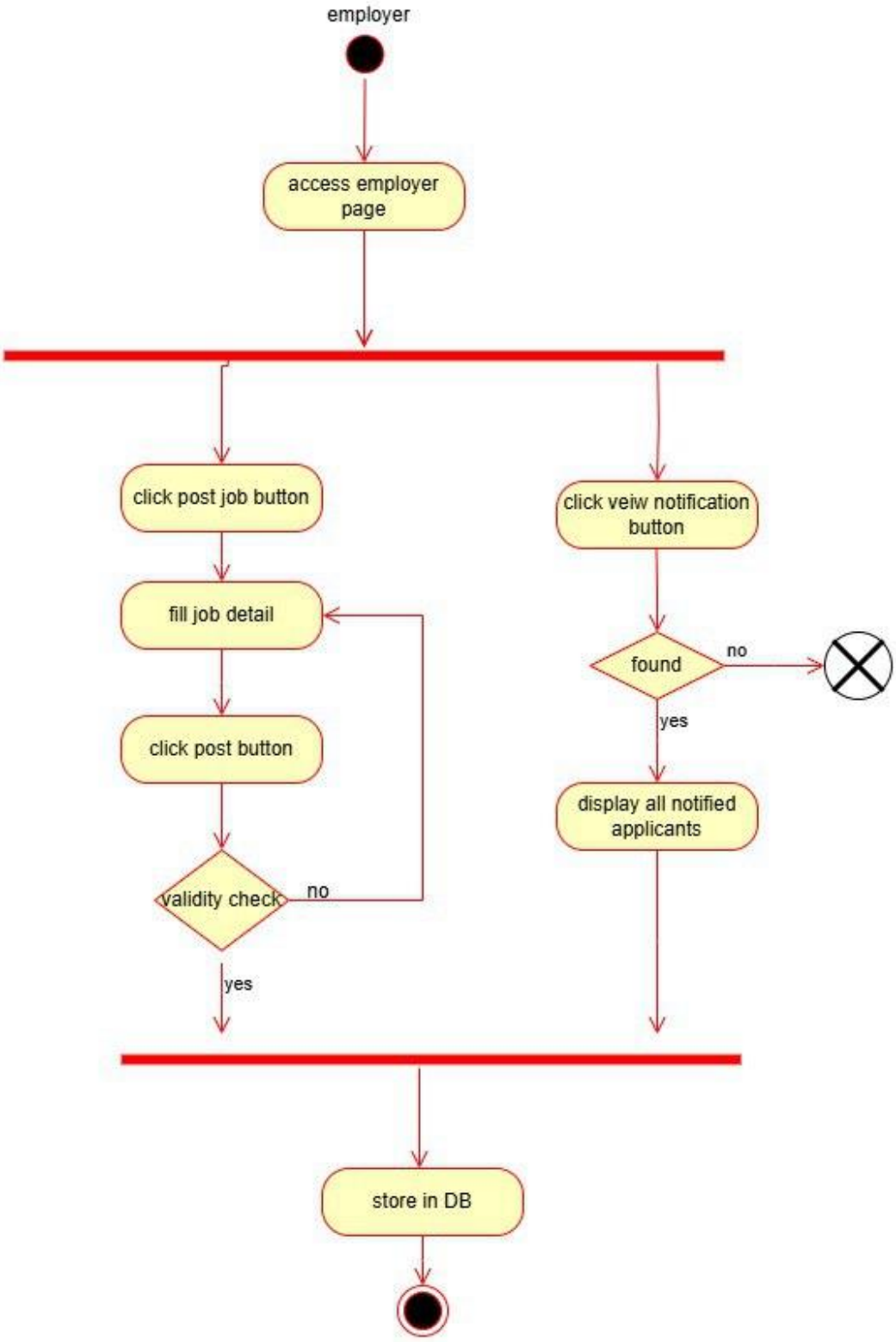

Figure 7 activity diagram for post job and view notification

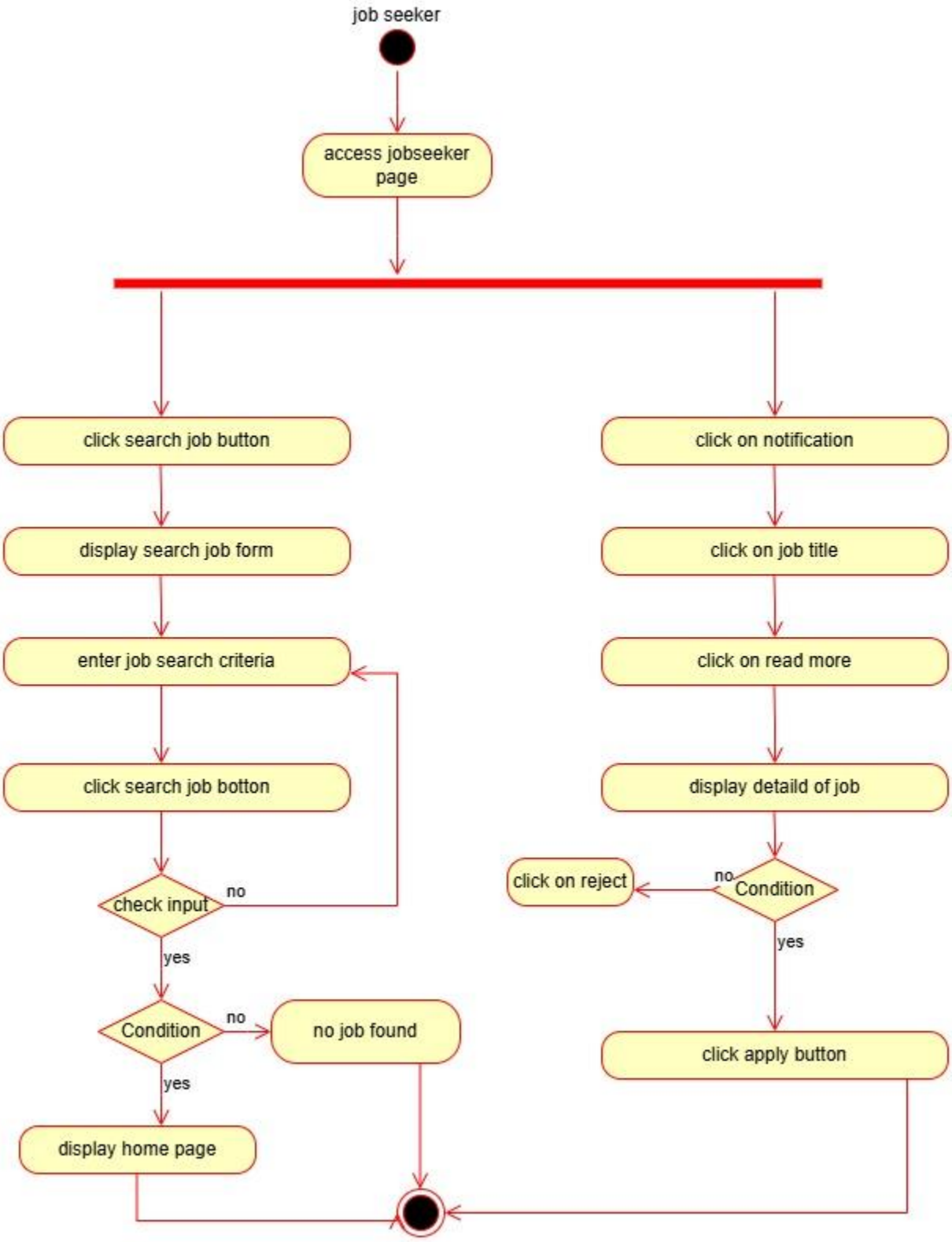

Figure 8 Activity Diagram for search job and view notification

2.7.2 Sequence Diagrams

Sequence diagrams represent the interaction between objects (users and system components) over time.

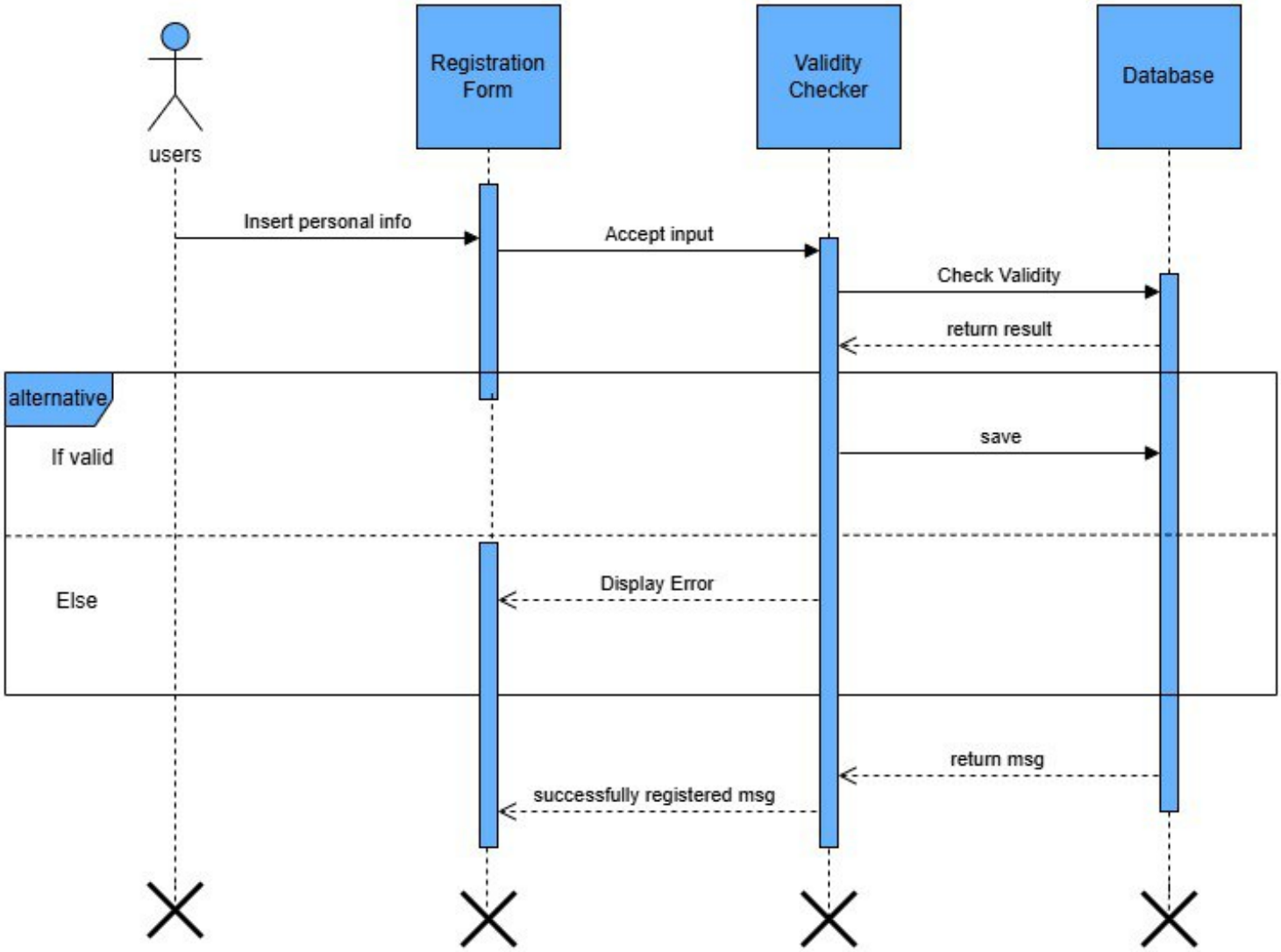

Figure 9 Sequence Diagram for Register

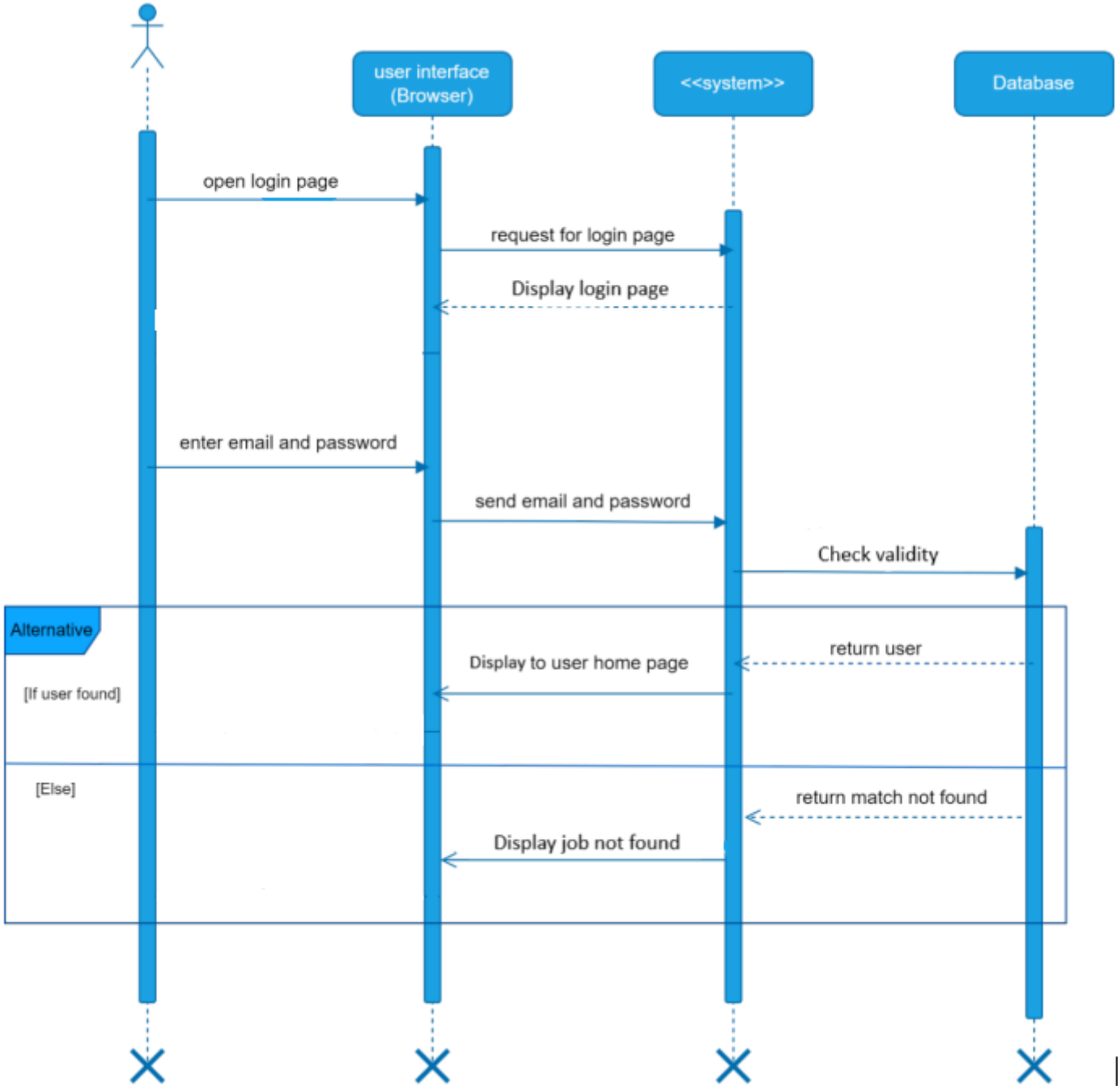

Figure 10 Sequence Diagram for login

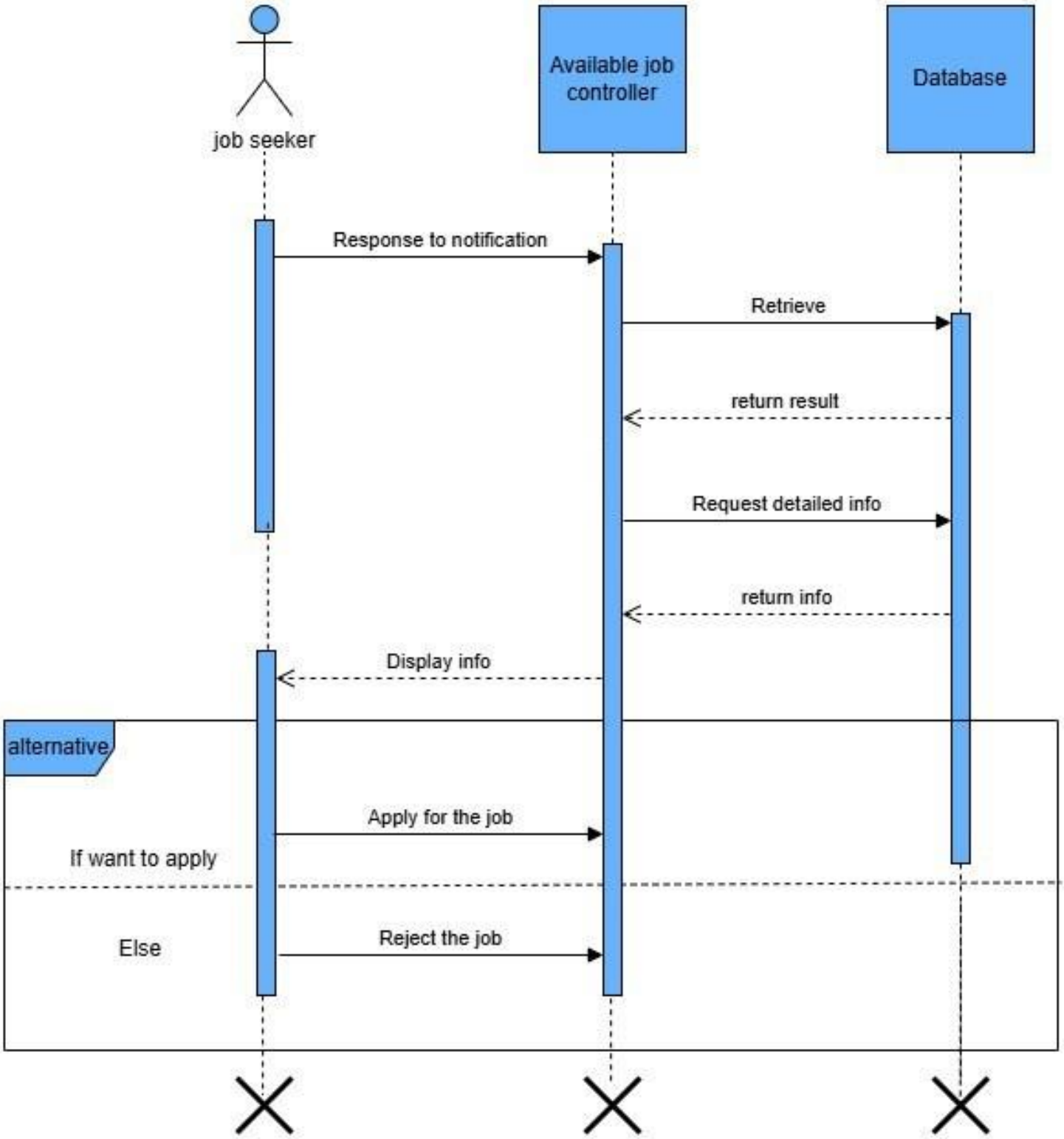

Figure 11 Sequence Diagram for apply with notification

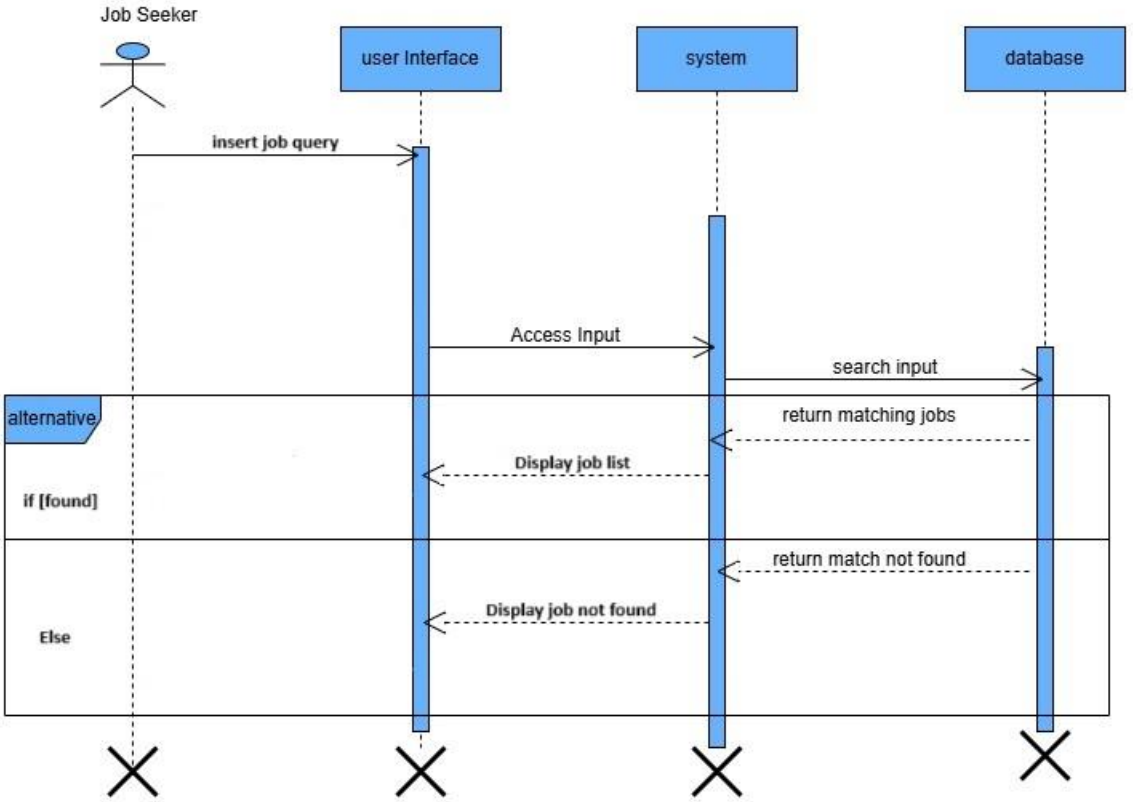

Figure 12 Sequence Diagram for job search

CHAPTER THREE

SYSTEM DESIGN

3.1 Layered Architecture

The Ethiopian Job Search platform is designed using a layered architecture, which enhances modularity, scalability, and maintainability, following established software engineering principles [3]. This architecture consists of four distinct layers, each responsible for specific tasks:

1. User Interface (UI) Layer:
   This is the front-end of the system, where users interact with the platform. It includes the home page, job search interface, job seeker dashboard, employer dashboard, and admin panel.
2. Controller/Process Layer:
   This layer acts as the middleware, processing user requests, validating input data, and communicating with the business logic layer. Key operations include job search filtering, job application submission, and employer job posting management.
3. Business/Domain Layer:
   This is the core logic layer, handling system rules, job matching algorithms, application tracking, and analytics. It ensures that job seekers receive relevant job recommendations, employers can manage their postings, and administrators can monitor platform activities.
4. Persistence Layer:
   The database management layer handles data storage and retrieval. It stores information on users, jobs, applications, notifications, and analytics data.

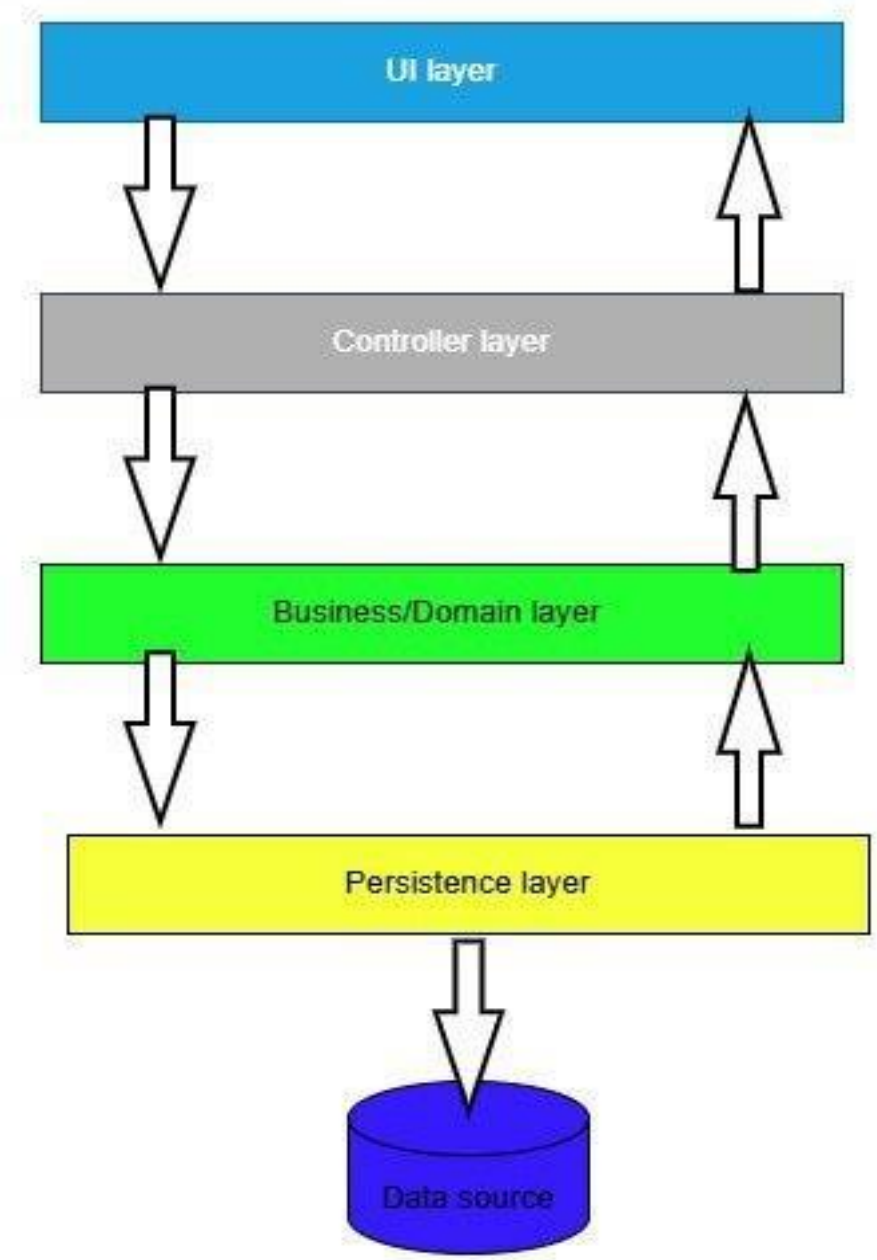

Figure 13 Class type architecture

3.2 Class Modeling

The Ethiopian Job Search Platform is designed using Object-Oriented Programming (OOP) principles to ensure modularity, reusability, and maintainability**.** The system employs inheritance, association, and aggregation**,** while also introducing interfaces and attributes to define functionalities**.**

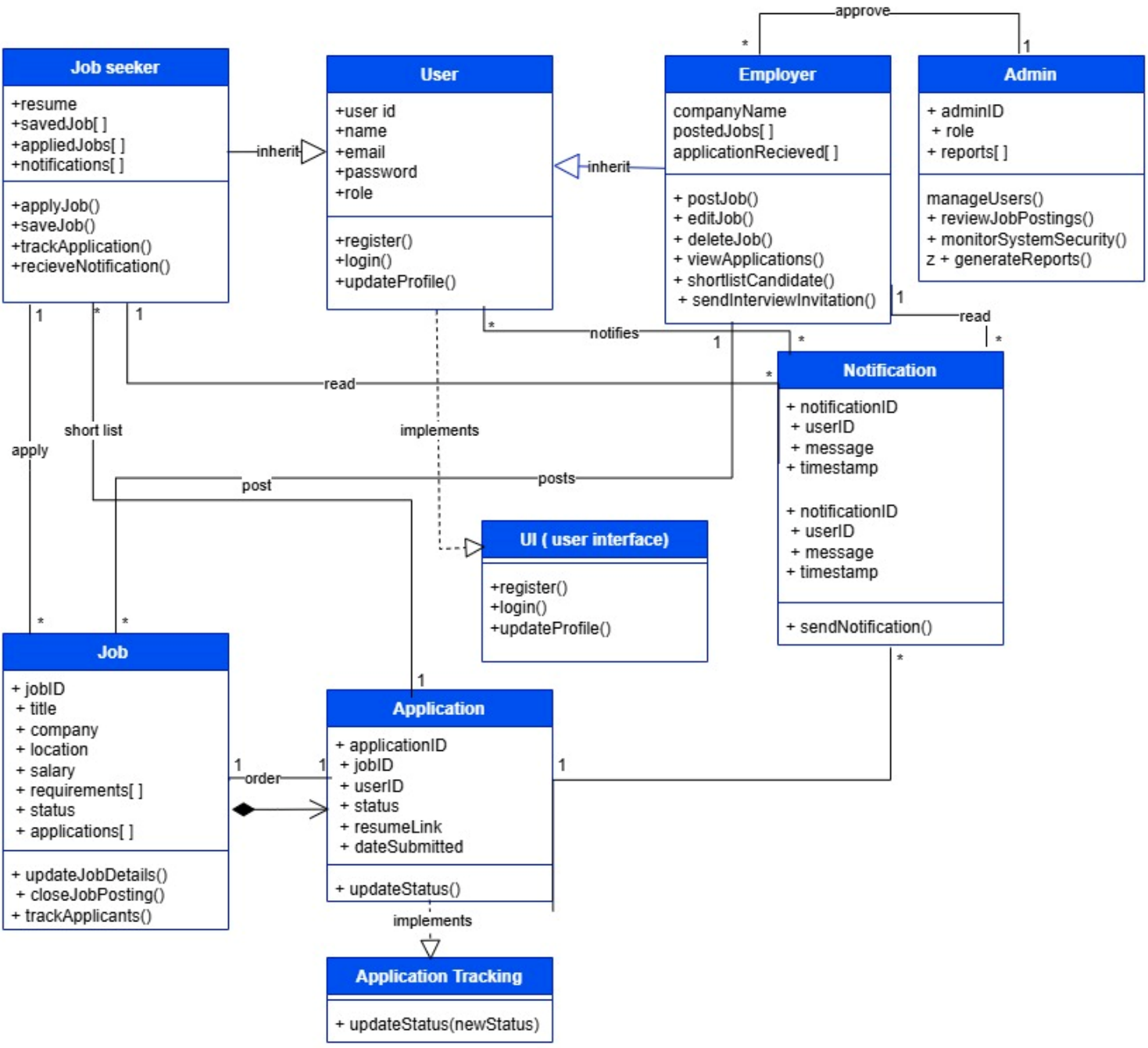

Figure 14 Class Diagram

3.3 User Interface Design

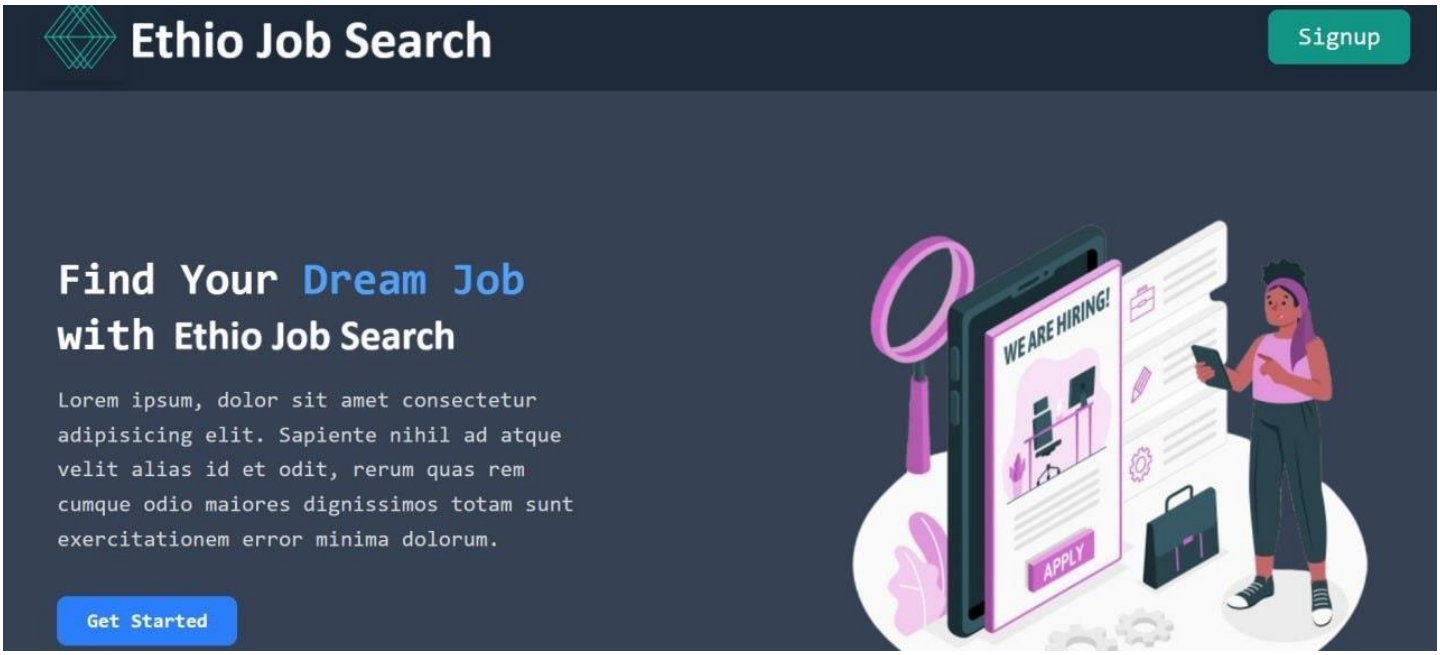

Figure 30 UI design

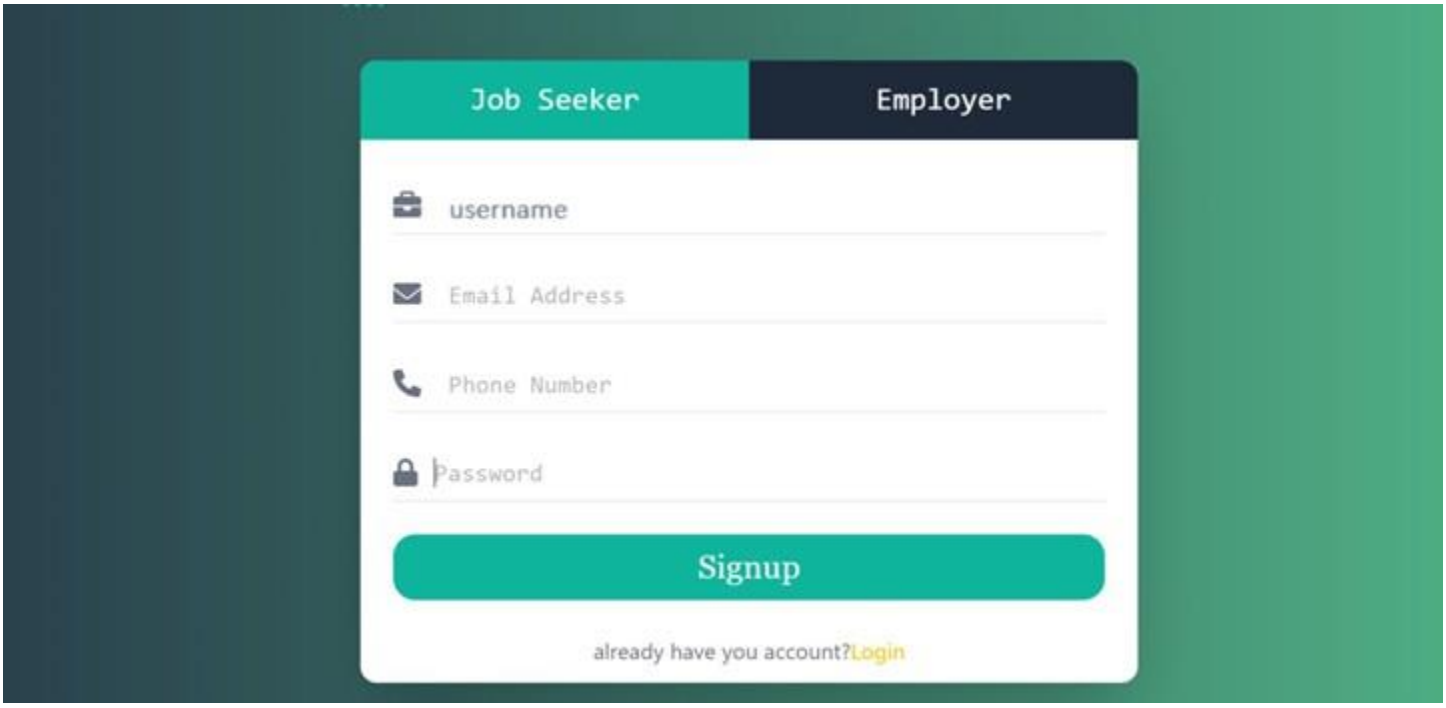

Figure 15 UI Sign up page

3.4 Database Design

The Ethiopian Job Search Platform requires a well-structured MySQL database to ensure efficient data storage, retrieval, and security while supporting scalability and high performance**.**

3.4.1 Entity-Relationship Diagram (ERD)

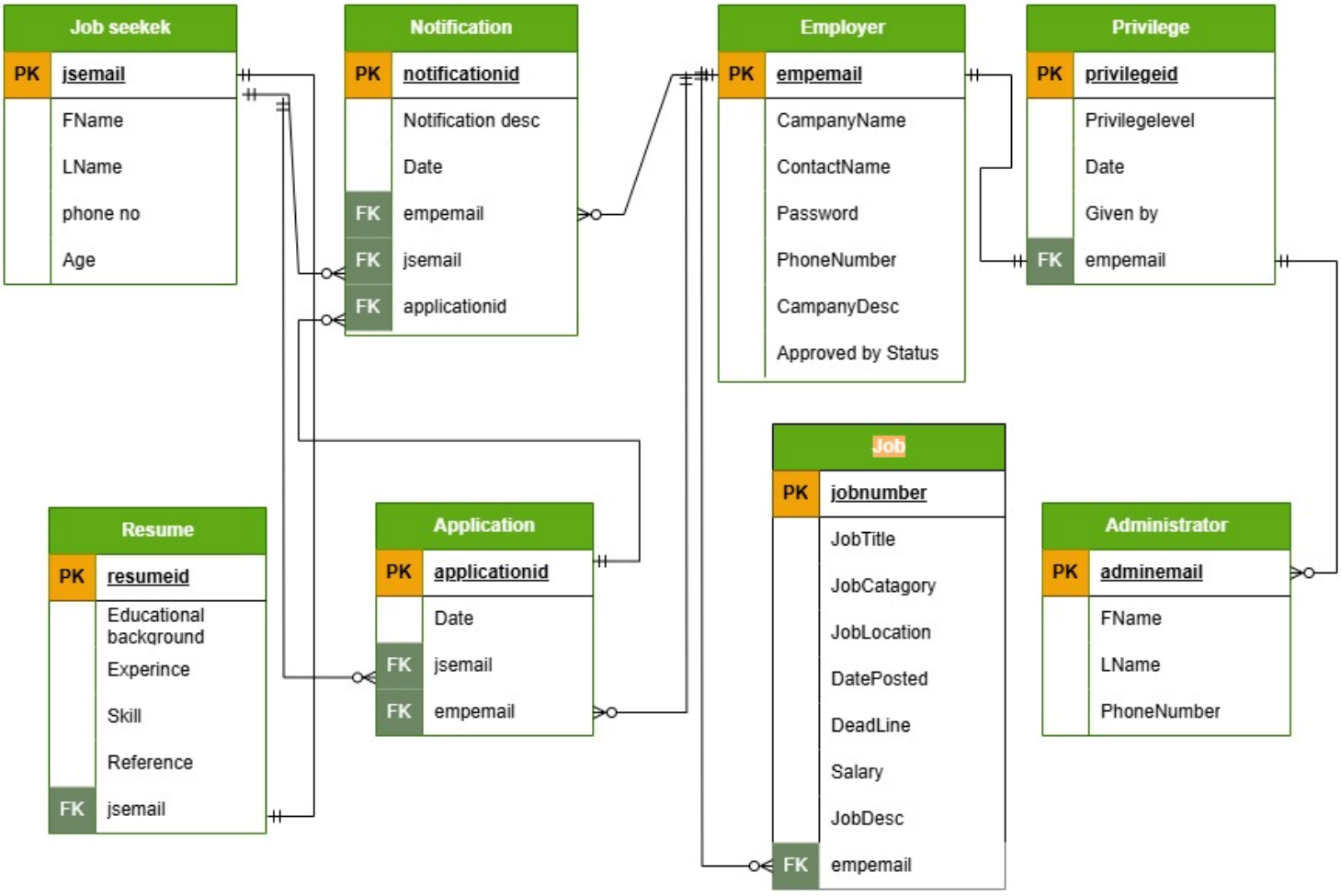

Figure 16 Entity Relationship Diagram

3.4.2 Table Normalization

To eliminate data redundancy and ensure data integrity, the database follows 3rd Normal Form (3NF) normalization principles:

1st Normal Form (1NF) - Remove Repeating Groups:

Ensure each column contains **atomic** values, and each row is unique.

JobID	**JobTitle**	**EmployerName**	**JobType**	**Applicants**

101	Software Engineer	ABC Corp	Full-Time	Ahmed, Abebe, Usul
102	Data Analyst	XyZ Ltd	Part-Time	Birhan, Murad

Problem: Applicants should not be stored as a repeating group.

2nd Normal Form (2NF) - Remove Partial Dependencies:

Every non-key column should be fully dependent on the primary key.

Splitting into two tables:

Jobs Table:

JobID	**JobTitle**	**EmployerID**	**JobType**
101	**Software Engineer**	**1**	Full-Time
102	**Data Analyst**	**2**	Part-Time

Employers Table:

EmployerID	**EmployerName**
1	ABC Corp
2	XYZ Ltd

Applicants Table:

ApplicantID	**JobID**	**ApplicantName**
201	101	Ahmed
202	101	Abebe
203	102	Birhan

3rd Normal Form (3NF) - Remove Transitive Dependencies: [2]

No column should depend on a non-primary key column.

If we have:

JobID	**JobTitle**	**EmployerID**	**EmployerName**

Problem: EmployerName depends on EmployerID, not JobID.

Solution: Move EmployerName to the Employers table.

Final Optimized Tables:

Jobs Table

JobID	JobTitle	EmployerID	***JobType***
101	*Software Engineer*	*1*	*Full-Time*
102	*Data Analyst*	*2*	*Part-Time*

Employers Table

EmployerID	**EmployerName**
1	ABC Corp
2	XYZ Ltd

Applicants Table

ApplicantID	**JobID**	**ApplicantName**
201	101	Ahmed
202	101	Abebe
203	102	Birhan

Users Table

UserID	**Username**	**Password**	**UserType (Job Seeker / Employer)**

Conclusion

- **1NF:** Removed repeating groups.
- **2NF:** Separated data into meaningful tables.
- **3NF:** Ensured no transitive dependencies.

3.4.3 Persistence diagram

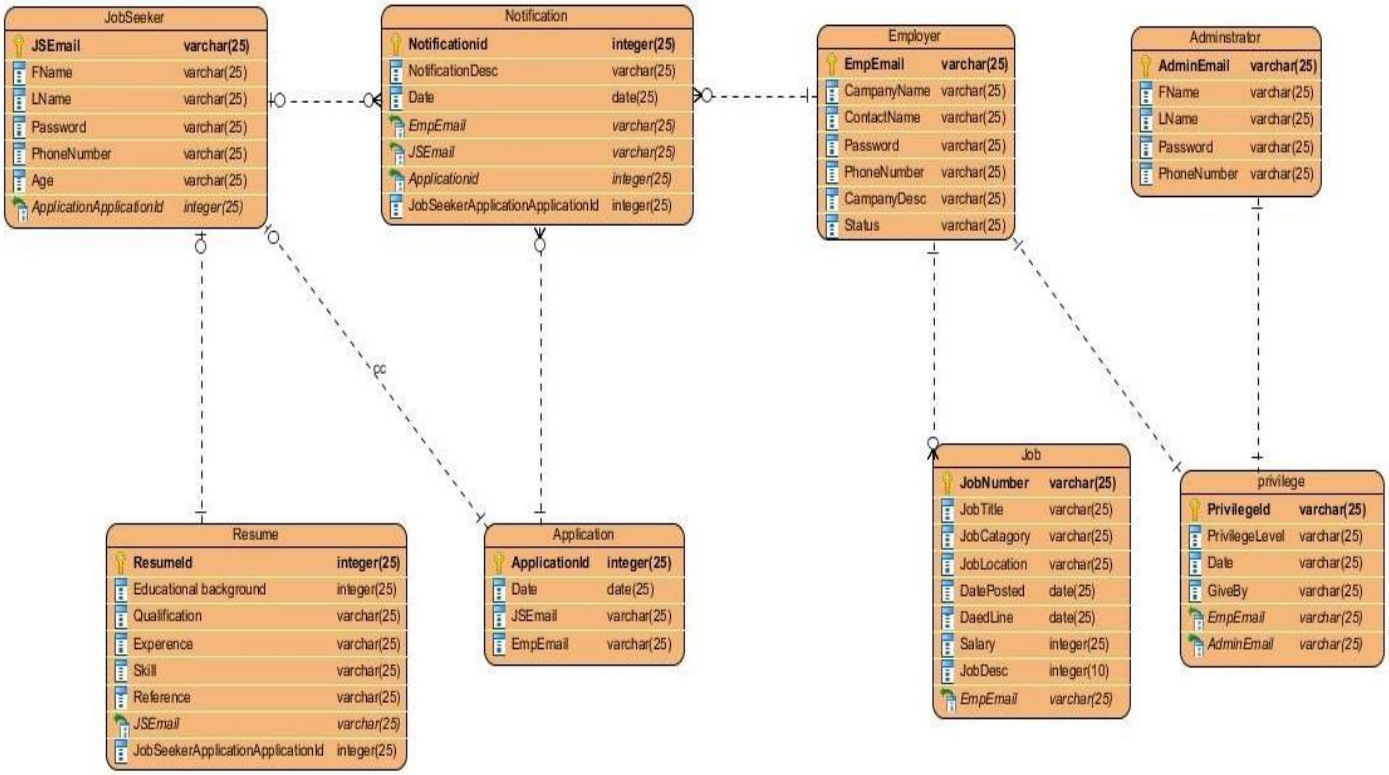

Figure 17 Persistence Diagram

3.5 Performance Evaluation

Performance evaluation is crucial for ensuring that the Ethiopian Job Search platform operates efficiently and meets user expectations. The evaluation involves assessing system response time, scalability, reliability, and resource utilization. Below are the key aspects of performance evaluation:

3.5.1 Response Time

Response time measures how quickly the system processes user requests. The platform should ensure:

- Fast page load times (<2 seconds for job searches and application tracking).
- Quick response for database queries, especially when filtering jobs.
- Optimized API calls to reduce delay in fetching job listings and notifications.
- Optimized indexing and caching strategies can significantly reduce the response time for job search queries in large-scale employment platforms. [4]

3.5.2 Scalability

Scalability ensures the system can handle a growing number of users and job postings without performance degradation.

- Using MySQL indexing to speed up query execution.
- Implementing caching mechanisms for frequently accessed data (e.g., job listings).
- Deploying load balancers if traffic increases significantly.

3.5.3 Reliability

Reliability ensures the system runs smoothly without unexpected crashes.

- Regular database backups to prevent data loss.
- Error-handling mechanisms for failed job postings or applications.
- Redundant server setup to avoid downtime.

3.5.4 Security & Data Integrity

The platform must protect user data and prevent unauthorized access. [3]

- Secure authentication (hashed passwords, role-based access control).
- Preventing SQL injection and XSS attacks.
- Encryption for sensitive data like employer details.

3.6 State Chart Modelling

The state chart diagram shows the change of an object through time from one state to the other state. State chart modelling is used to show the sequence of states that an object goes through, the events that cause the transition from one state to the other and the actions that result from a state change.

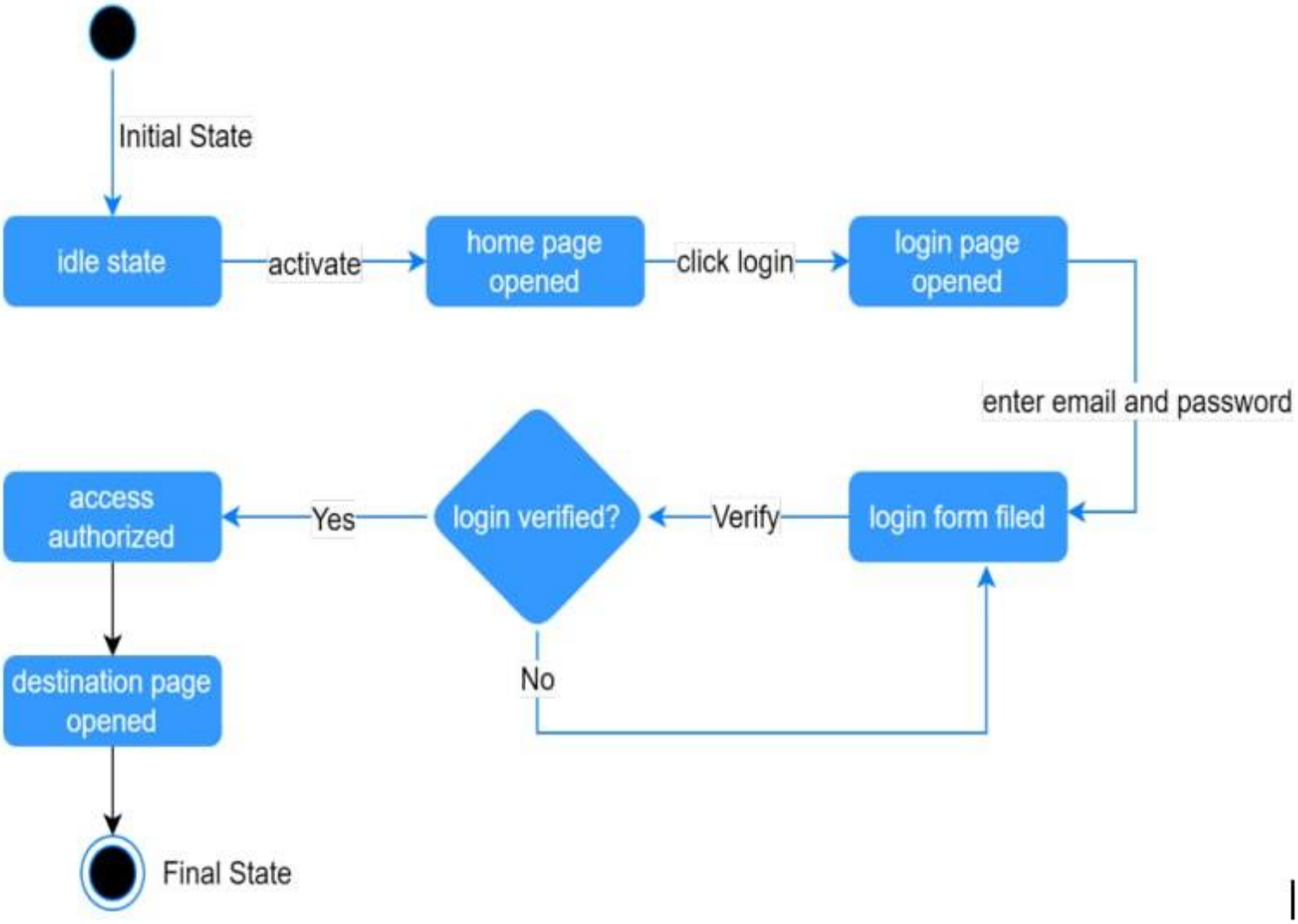

Figure 18 state chart modelling for login

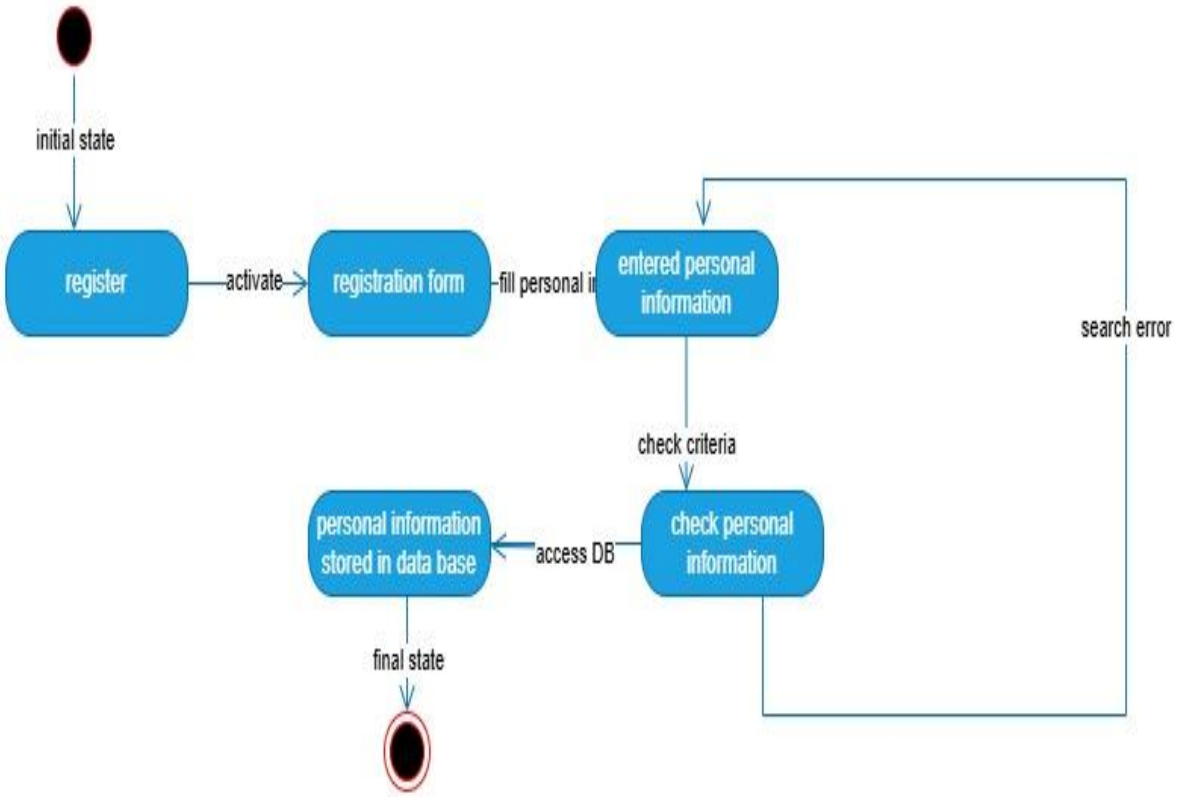

Figure 19 state chart modelling for register

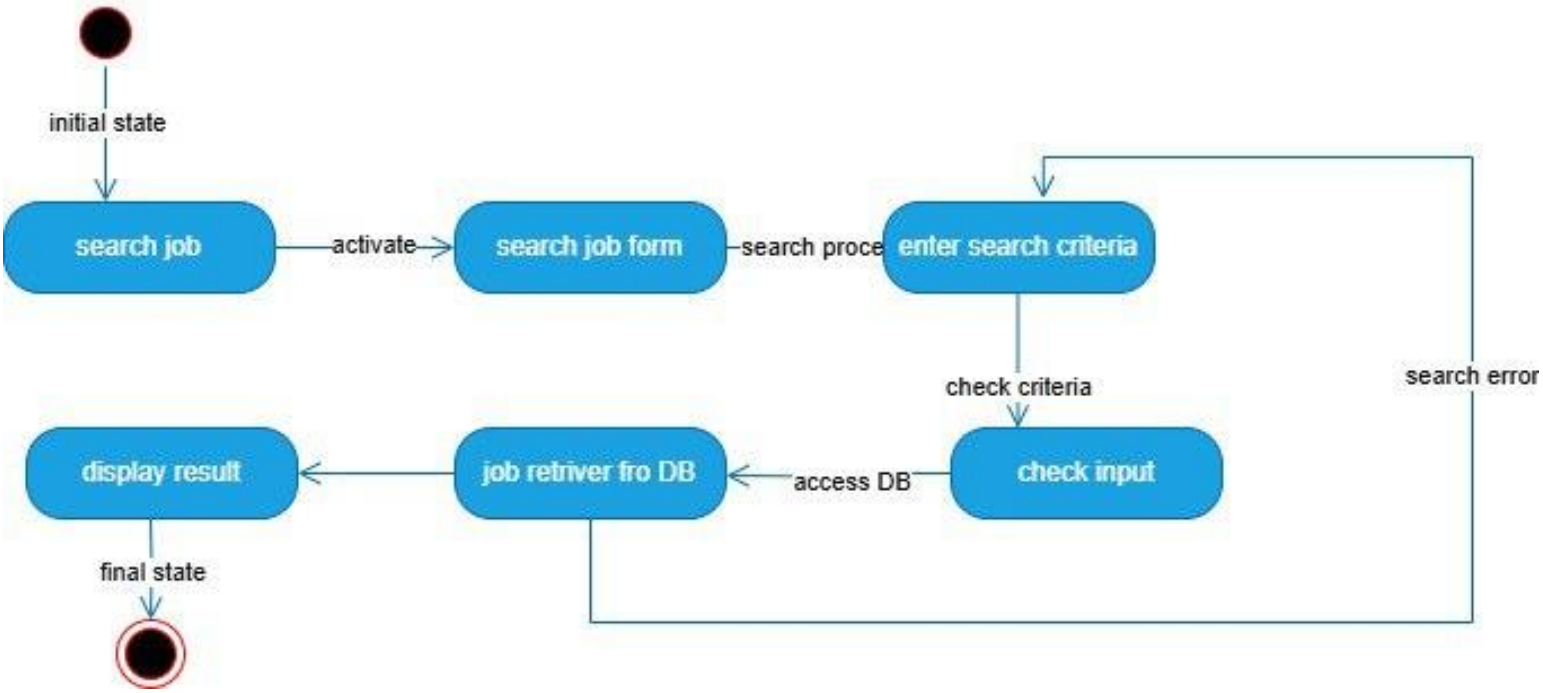

Figure 20 state chart modelling for job search

3.7 Component Diagram

The Component Diagram shows how different parts of the system interact with each other.

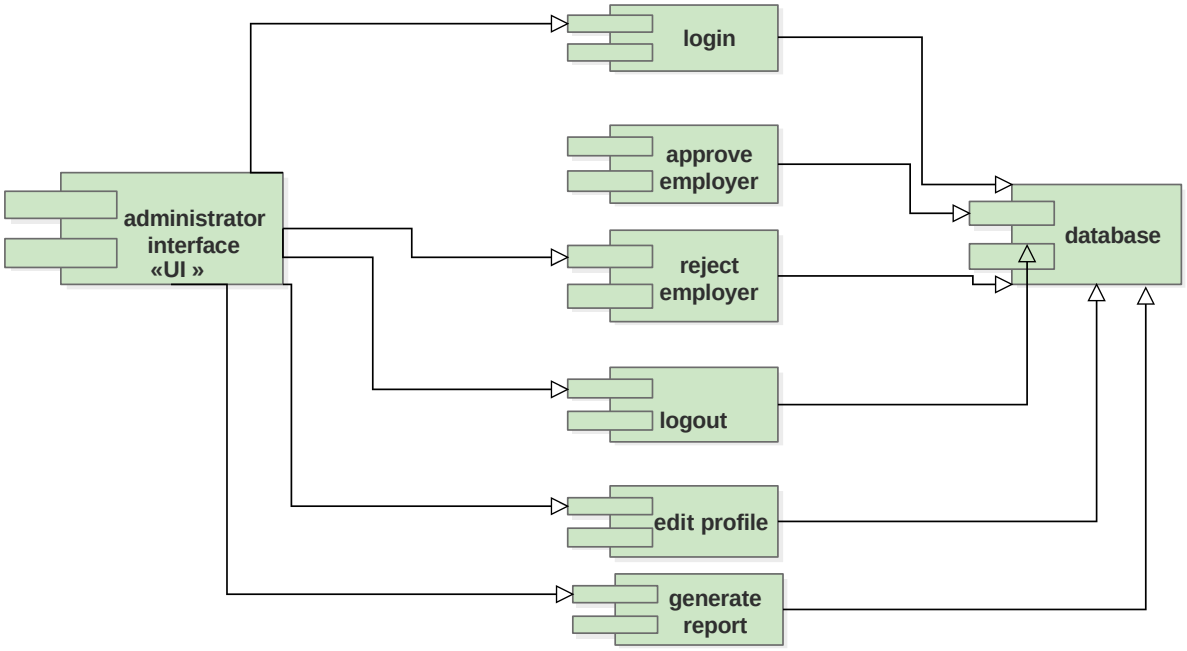

Figure 21component modelling diagram for administrator functionalities

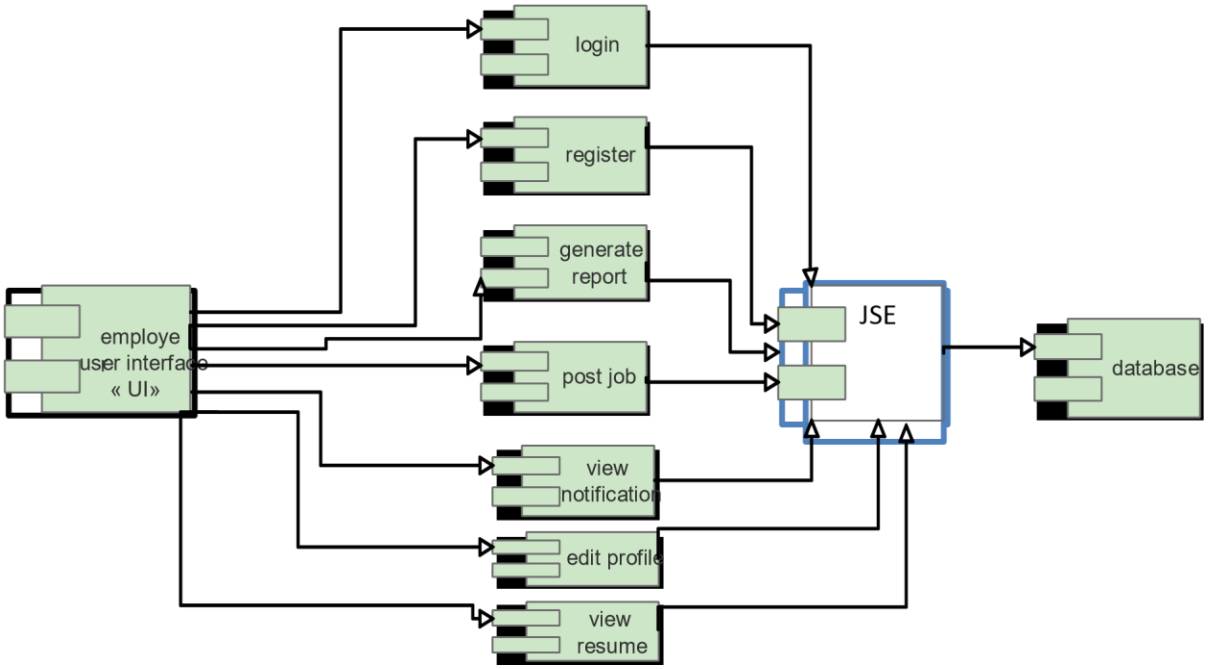

Figure 22 component modelling diagram for employer functionalities

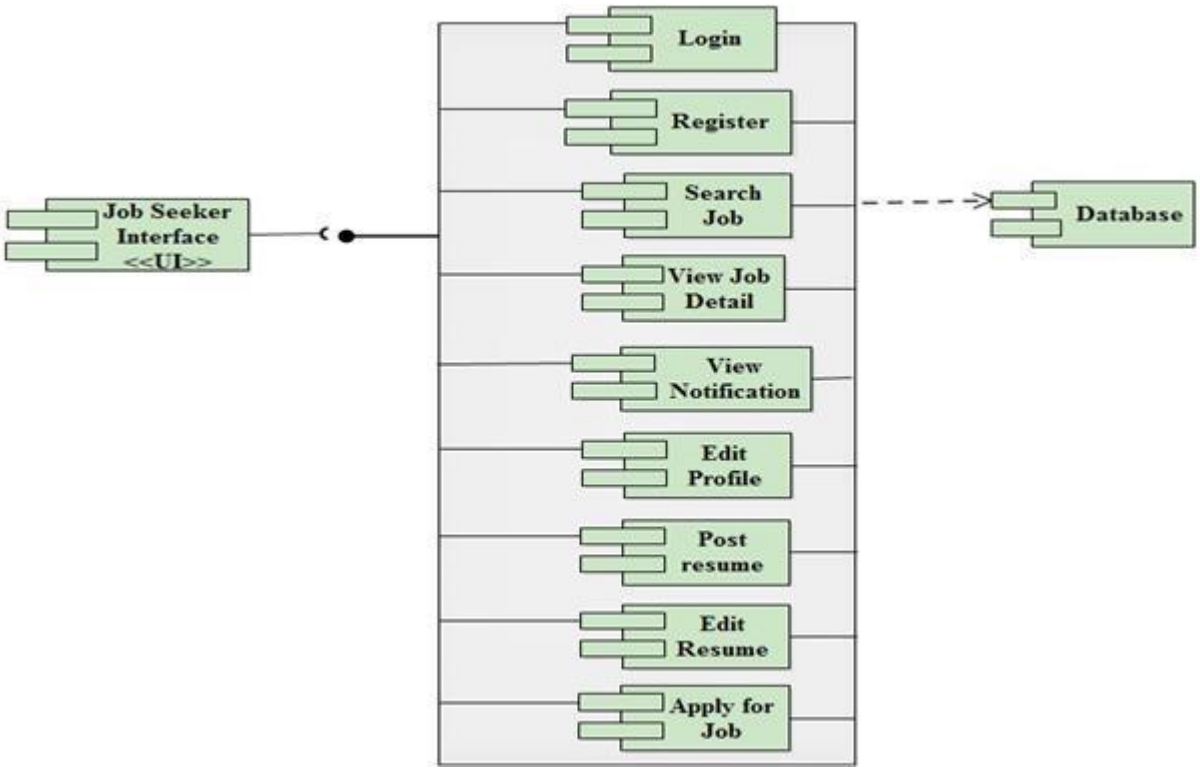

Figure 23 component modelling diagram for job seeker functionalities

3.8 deployment diagrams

Deployment modelling is used to show the hardware of the system, the software that is installed in the hardware and also shows how the software and the hardware components work together.

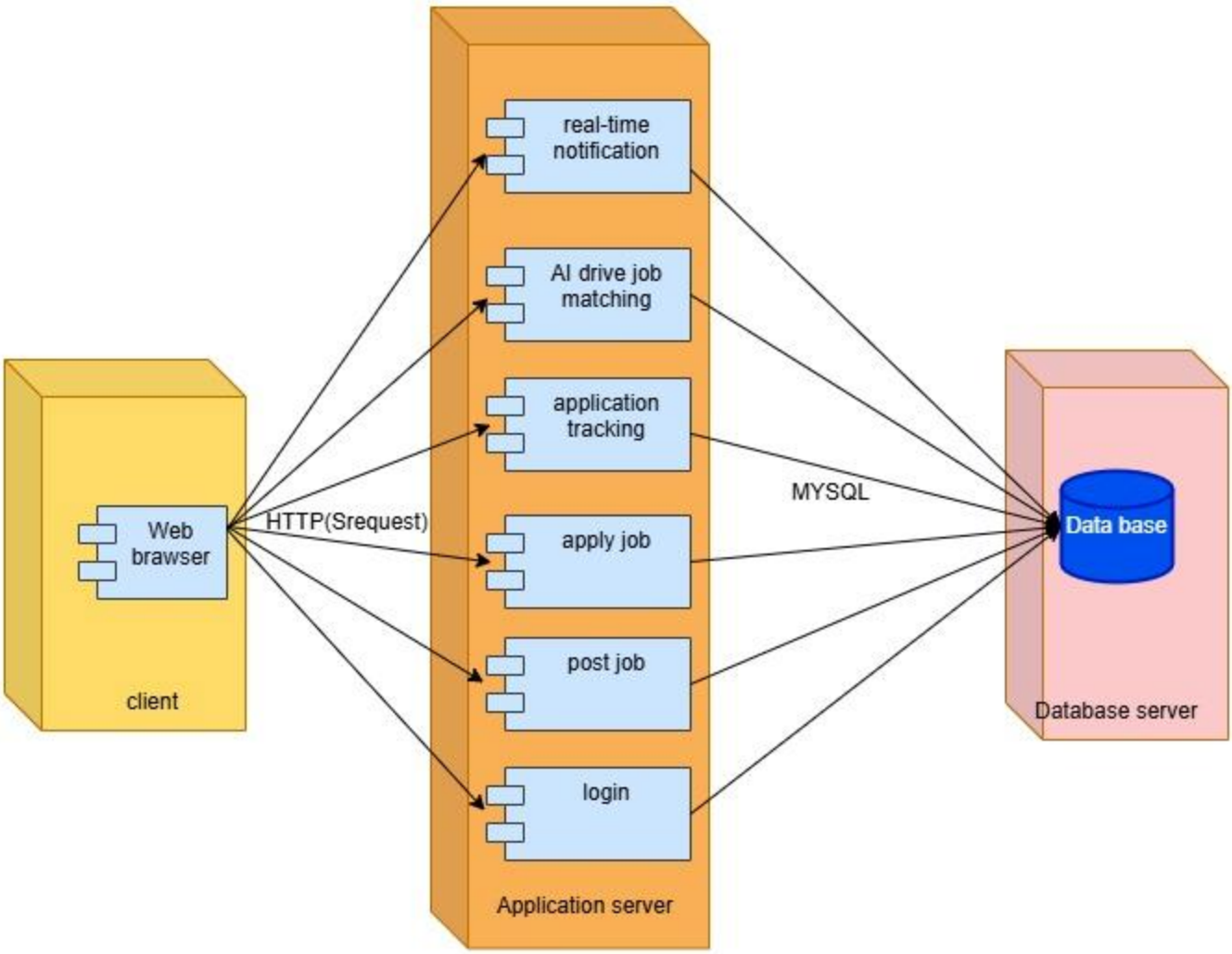

Figure 24 deployment modelling of JSE system

References

References

[1] T. A. a. B. Tesfaye, "Challenges and Opportunities in Ethiopian Job Market Digitalization," *Journal of African Digital Transformation,* vol. 15, pp. 100-115, 2023.

[2] K. B. e. al, "Manifesto for Agile Software Development," *Agile Alliance,* p. Available: https://www.agilealliance.org, 2001.

[3] ,. H. W. a. D. T. :. L. 5. e. H. N. A. Dennis, An Object-Oriented Approach with UML, USA: Wiley, 2015.

[4] A. Cockburn, Writing Effective Use Cases, Boston, MA, USA: Addison-Wesley, 2001, 2001.

[5] ,. 1. e. I. Sommerville, Software Engineering, Pearson, 2015.

Appendix: Glossary of Terms

- Agile Methodology – A software development approach that emphasizes iterative development, flexibility, and collaboration.
- API (Application Programming Interface) – A set of functions that allow different software applications to communicate with each other.
- Authentication – The process of verifying a user's identity before granting access to a system.
- Authorization – The process of granting or denying user permissions to access system features.
- Backend – The part of a web application that handles data processing, database interactions, and server-side logic.
- CRUD (Create, Read, Update, Delete) – The four fundamental operations performed on database records.
- DBMS (Database Management System) – A software system used to store, retrieve, and manage data in a structured format.
- Deployment – The process of making a software application available for use.
- ERD (Entity-Relationship Diagram) – A visual representation of how different entities (e.g., users, jobs, applications) relate to each other in a database.
- Frontend – The part of a web application that users interact with, usually designed using HTML, CSS, and JavaScript.
- HTTP (HyperText Transfer Protocol) – The protocol used for communication between web browsers and servers.
- HTTPS (HyperText Transfer Protocol Secure) – A secure version of HTTP that encrypts data to protect user information.
- Job Matching Algorithm – A set of rules used to recommend job listings based on a job seeker's profile and preferences.
- Load Balancer – A system that distributes network traffic across multiple servers to improve performance and reliability.
- MySQL – A popular open-source relational database management system used to store structured data.
- Normalization – The process of organizing a database to reduce redundancy and improve efficiency.
- NoSQL – A type of database that handles unstructured or semi-structured data, often used for scalability.
- RBAC (Role-Based Access Control) – A security model that restricts system access based on a user's role.

- Real-Time Notifications – Instant alerts sent to users when specific events occur, such as new job postings or application status updates.
- Scalability – The ability of a system to handle increasing amounts of work by adding resources efficiently.
- SQL (Structured Query Language) – A programming language used for managing and querying databases.
- UML (Unified Modeling Language) – A standardized way to visualize system architecture using diagrams like class diagrams, use case diagrams, and sequence diagrams.
- UI (User Interface) – The visual and interactive elements of an application that users interact with.
- UX (User Experience) – The overall experience a user has when interacting with a system, focusing on ease of use and satisfaction.
- WebSocket – A communication protocol that enables real-time, two-way interaction between users and the server.