**EpoLink Phase 3: Software Design and Modeling**

Course: Software Engineering

Submission Date: April 23, 2025

Group Name: EpoLink Team

Team Members:

- Hristina Stefani – Project Coordinator and Backend Developer  
- Emelia Mazniku – Database & Cloud Management  
- Alesia Selenica – AI & Data Integration  
- Xhosela Rraboshta – UI/UX Design  
- Uendi Zuna – Frontend Developer  
- All Members – Quality & Assurance Testing

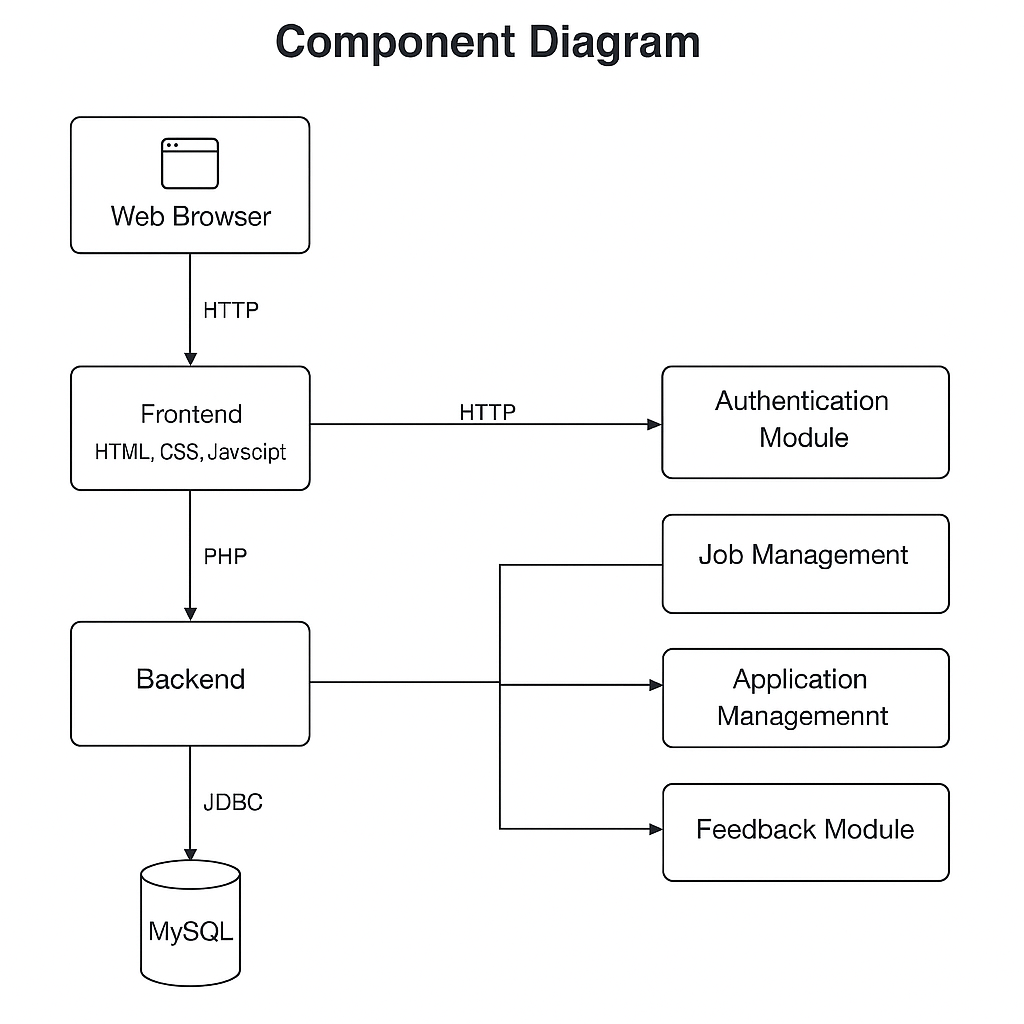
# 1. Software Architecture

## System Architecture

EpoLink is a modular web platform designed for students to connect socially and academically. The architecture includes a frontend built with HTML, CSS, and JavaScript, a PHP backend to handle business logic, a MySQL database, and Python-based AI module. Apache is used for deployment. The architecture ensures clean separation of concerns.

## Component Diagram

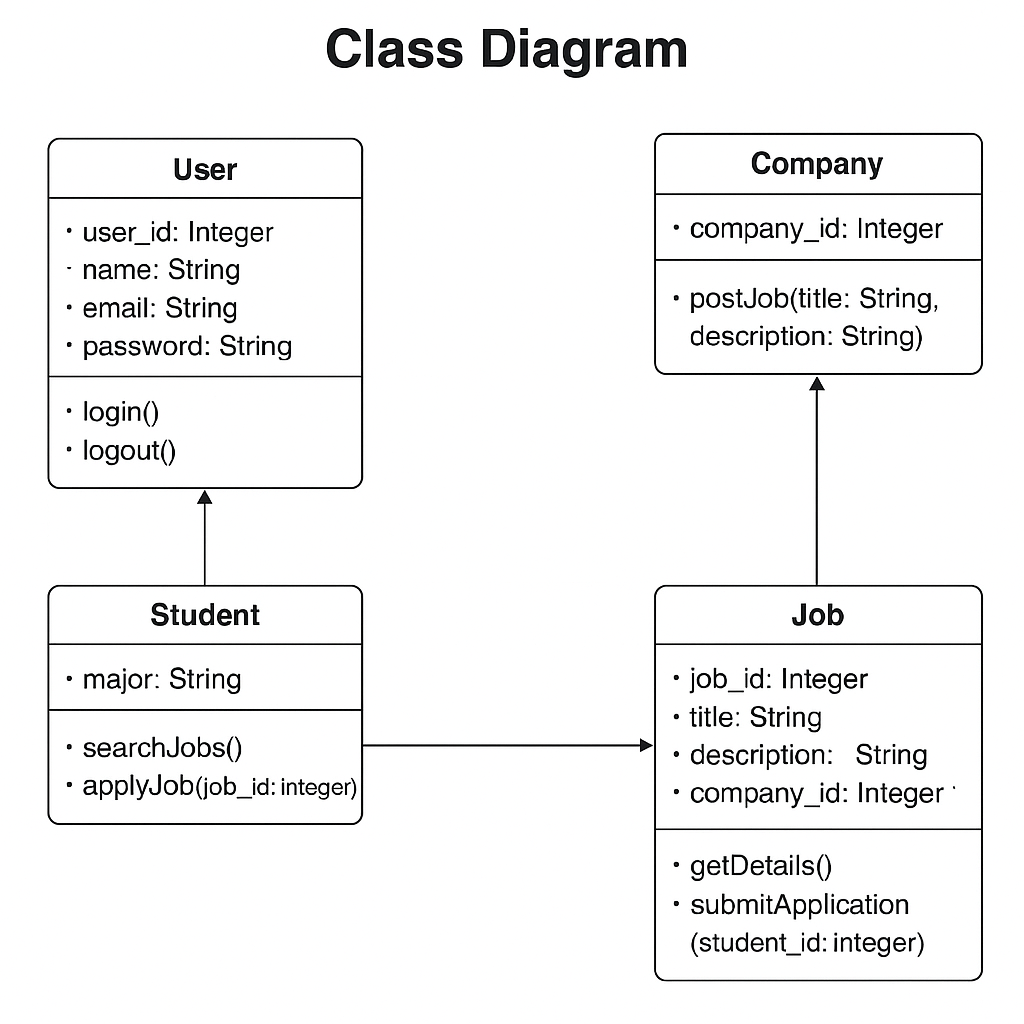
The following diagram shows the interaction between frontend, backend, database, and major functional modules.



# 2. Detailed Design

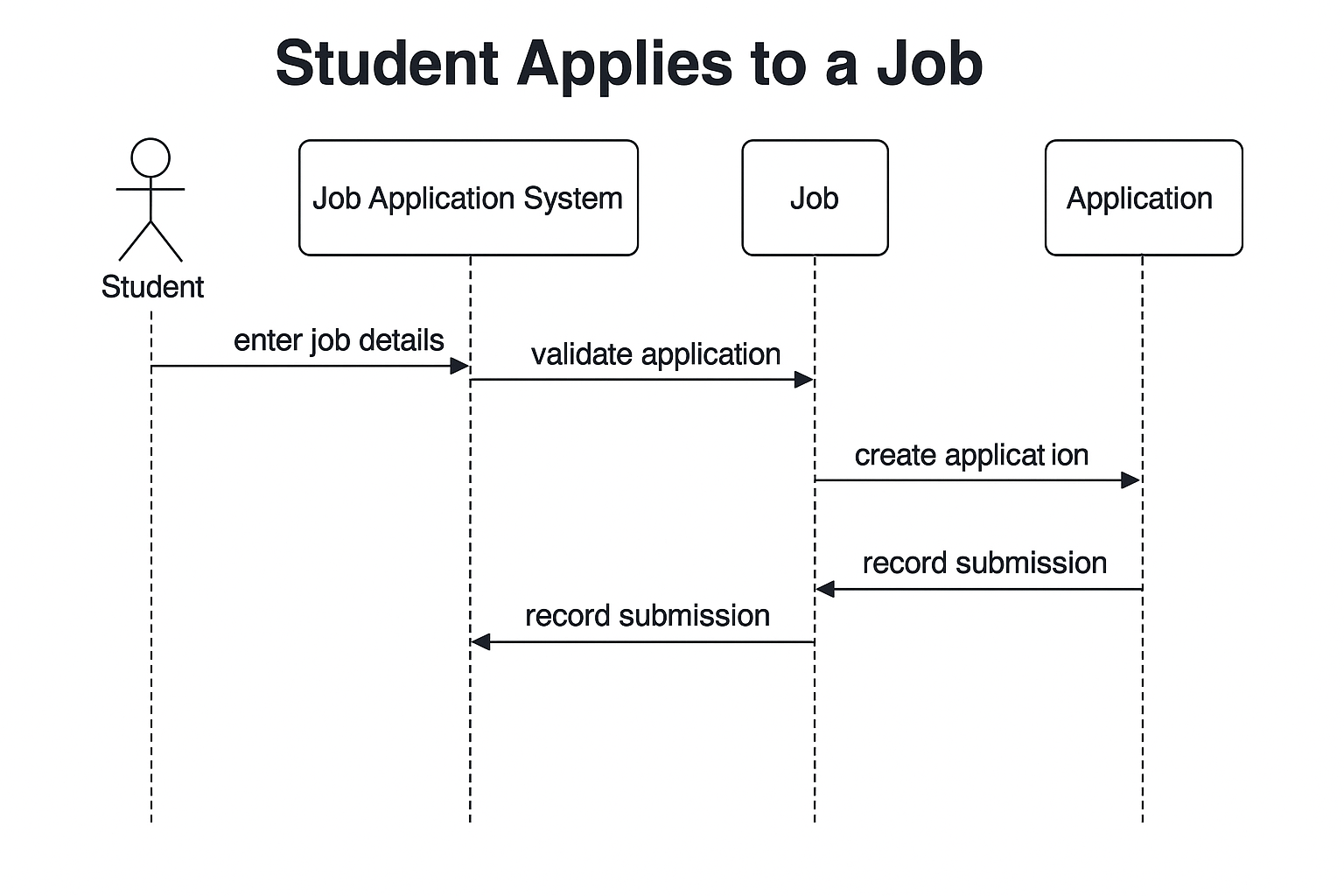
## Class Diagram

The class diagram shows the relationships and operations of key entities like Student, Company, Job, and Application.



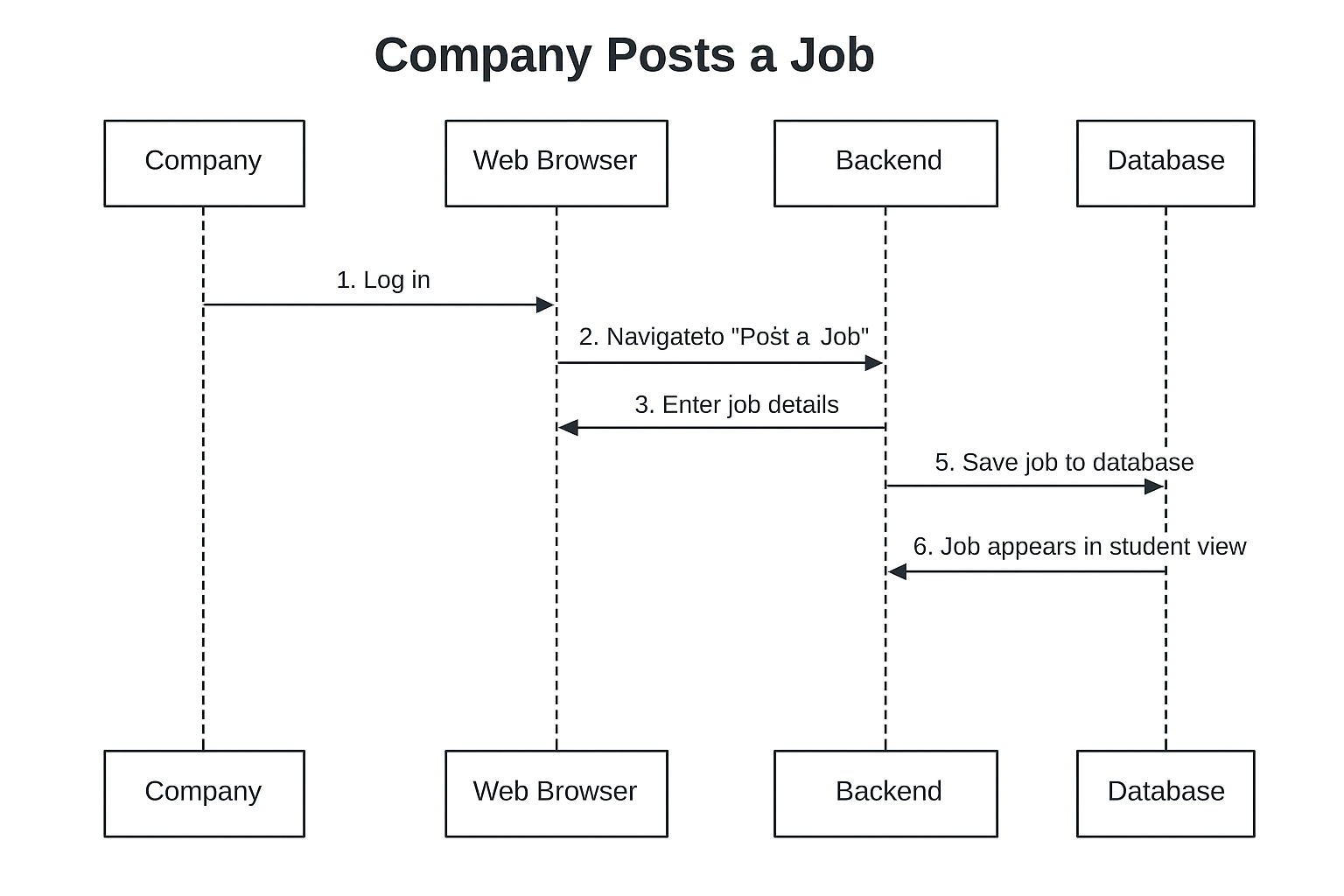
## Sequence Diagram 1: Student Applies to a Job

This sequence diagram models the flow of actions when a student applies to a job via EpoLink.



## Sequence Diagram 2: Company Posts a Job

This diagram shows the process a company follows to post a job on the platform.



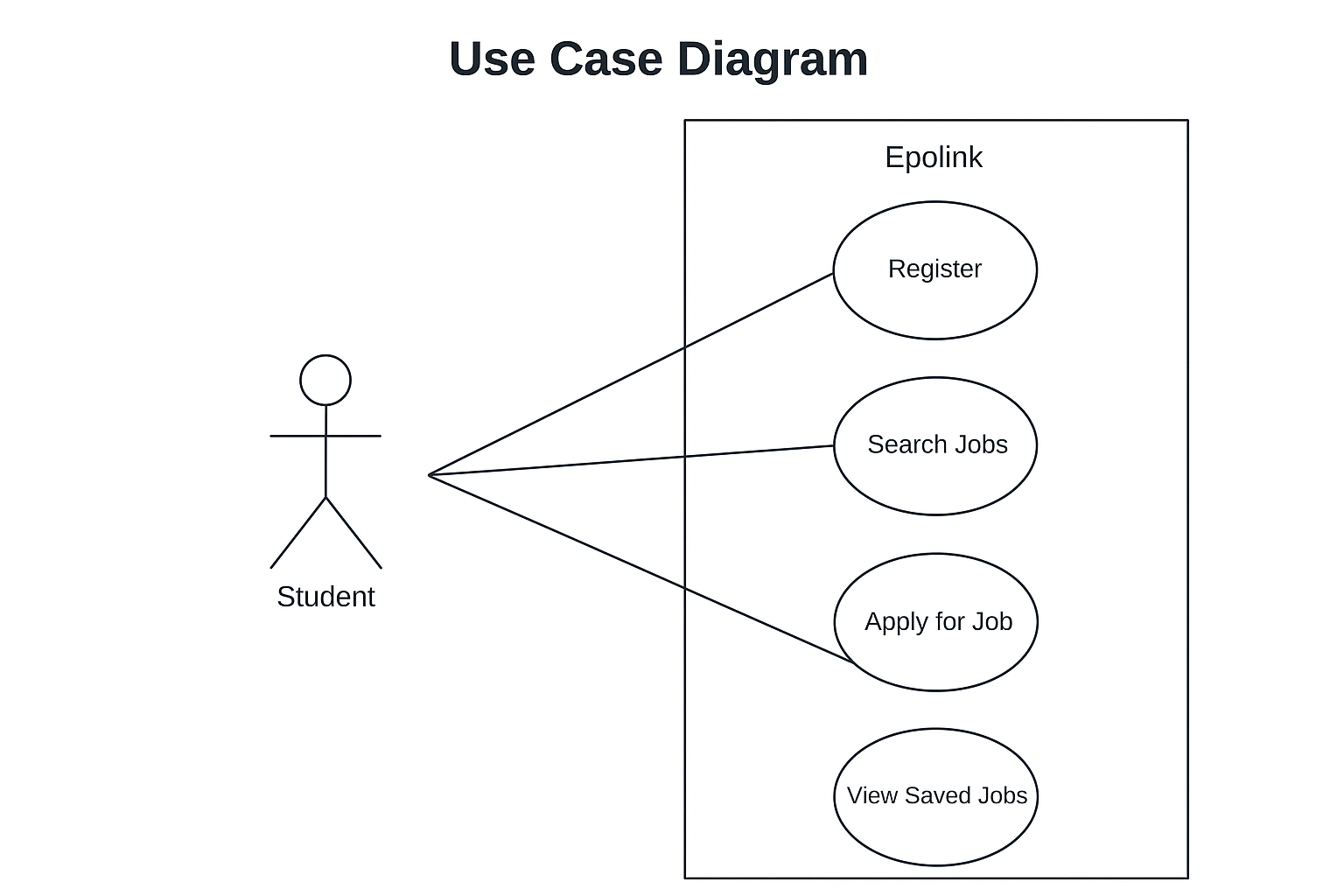
# Database Design

EpoLink uses a MySQL relational schema with the following key tables:  
- Users: Stores user data and roles (student, alumni, admin)  
- Forums, Events, Messages, Study Buddies, Lost & Found  
These tables support the platform's features and interactions.

# 3. Modeling

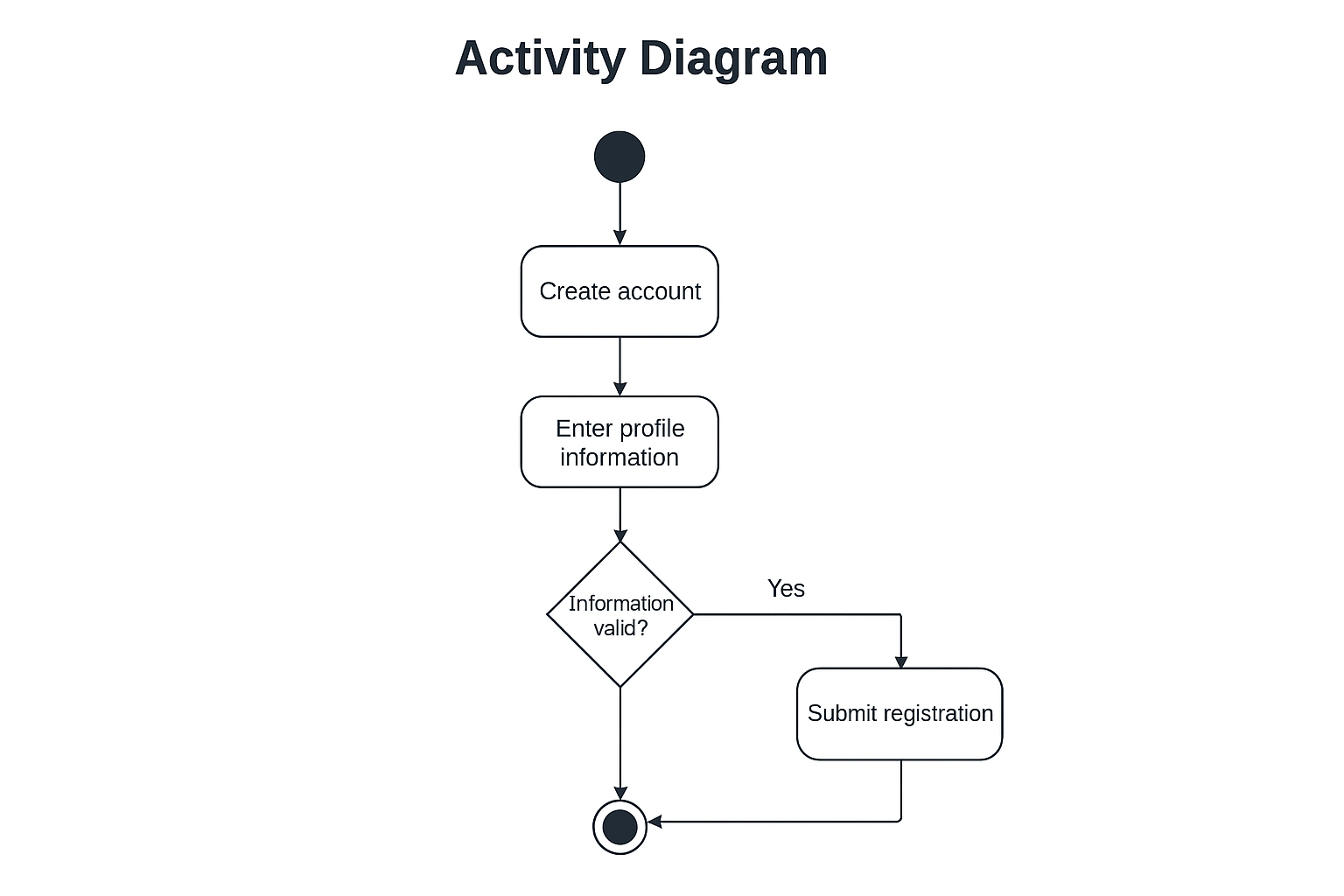
## Use Case Diagram

This use case diagram represents student interactions with EpoLink.



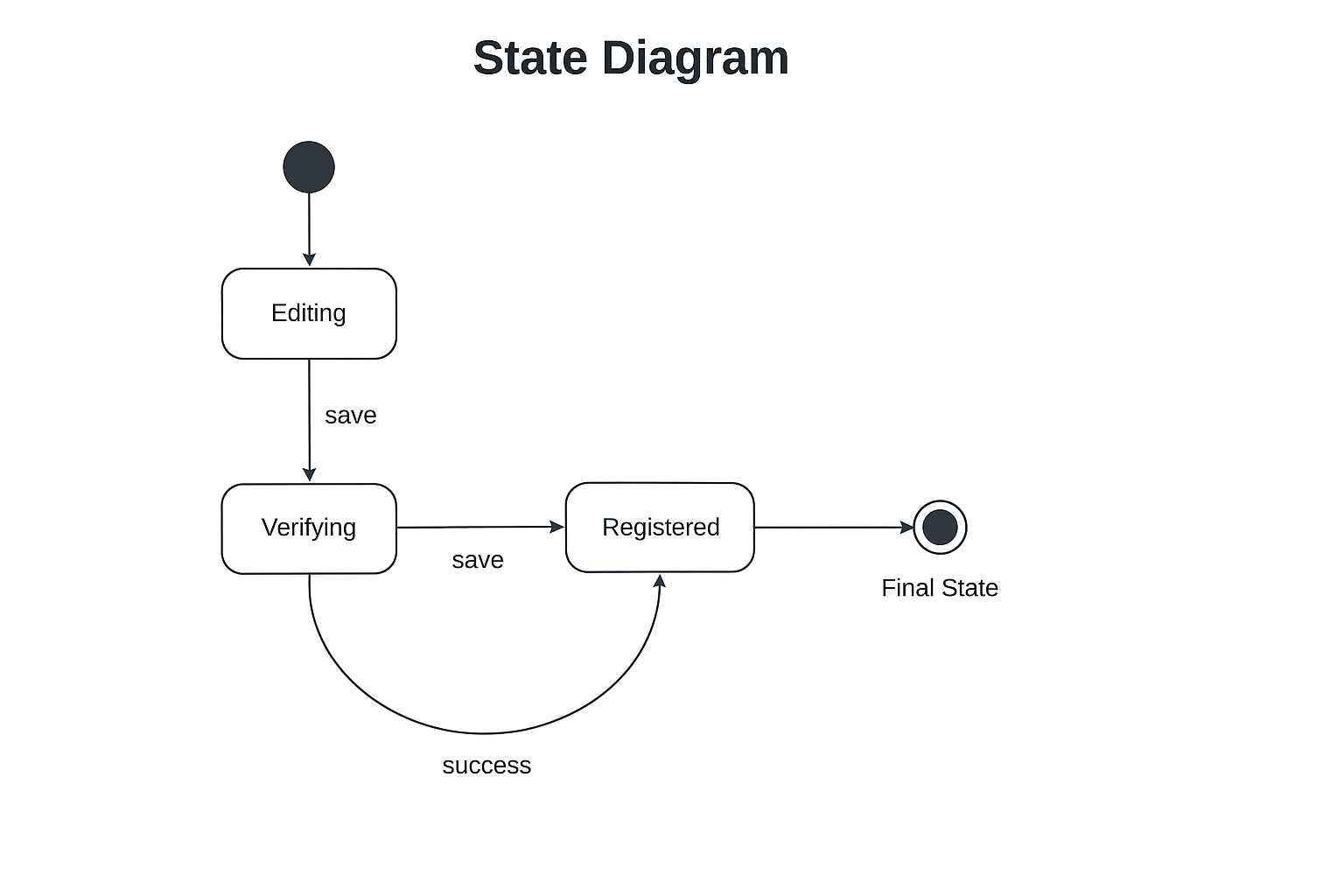
## Activity Diagram

This diagram shows the flow of registering on the platform.



## State Diagram

The state diagram below shows how a registration process transitions from editing to registration.



# 4. Additional Notes

All diagrams were created in Lucidchart-style format and all content aligns with requirements from Phases I and II. Agile methodology was followed throughout development. The diagrams and documentation ensure clear understanding of design, behavior, and data management of EpoLink.