ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

InternConnect Internship Management System

FINAL YEAR PROJECT I

BY

NAME OF THE STUDENTS	**IDNO**
Addis Alemayehu	UGR/4143/15
Bersufikad Mihret	UGR/4785/15
Dawit Workiye	UGR/4403/15
Dejen Achenef	UGR/7100/15

Project Advisor: Mr. Samuel.G

Date: 27 January 2025

DECLARATION

This is to declare that this project work, which is done under the supervision of Mr. Samuel and has the title InternConnect Internship Management System**,** is the sole contribution of

Addis Alemayehu

Bersufikad Mihret

Dawit Workiye

Dejen Achenef

There hasn't been any unauthorized copying or pasting of project work, which is a red flag for plagiarism. Every mentioned passage has been used to support the thesis and has been appropriately cited. Should it be established that this declaration has been violated, we shall bear full responsibility and liability**.**

Date: January 27 2025

Group Members:

Full Name	**Signature**
Addis Alemayehu	_____
Bersufikad Mihret	_____
Dawit Workiye	_____
Dejen Achenef	_____

CERTIFICATE

I certify that this BSc final project report entitled by: InternConnect Internship Management System

1.Addis Alemayehu

2. Bersufikad Mihret

3. Dawit Workiye

4.Dejen Achenef

is approved by me for submission. I certify further that, to the best of my knowledge, the report represents work carried out by the students.

_____ _____

Date Name and Signature of Supervisor

ACKNOWLEDGMENTS

Above all, we give thanks to the Almighty God for granting us the strength, courage, and guidance to successfully complete this work. We are deeply grateful to our family and friends for their constant encouragement, understanding, and support throughout the process. We would also like to express our sincere appreciation to our advisor, Mrs. Samuel, for her valuable guidance, motivation, and leadership during the project. Our heartfelt thanks extend to all the teaching and non-teaching staff who contributed to the completion of this project, both directly and indirectly. We are truly thankful to everyone who supported us with their patience, encouragement, and assistance.

ABSTRACT

InternConnect is a comprehensive digital system designed to modernize and automate the entire internship workflow among students, companies, and universities. Traditional internship coordination often involves scattered communication, multiple email threads, paper-based documentation, and inconsistent approval timelines; InternConnect eliminates these inefficiencies by centralizing all core activities including posting opportunities, submitting applications, uploading required documents, tracking approvals, and monitoring placements into one structured system. The system provides a shared environment where companies can publish internship opportunities with clear requirements, students can access verified listings, and universities can easily monitor and validate internship engagements.

By digitizing every step, the system enhances transparency, accountability, and operational accuracy for all stakeholders involved. Built using structured design methodologies including UML use cases, activity diagrams, and sequence diagrams the system emphasizes usability, scalability, and compliance with academic and industry standards. The InternConnect System aims to improve placement outcomes, optimize administrative workflows, and generally raise the quality of the internship experience through facilitating communication, reducing manual burdens, and enabling proactive progress monitoring. This project underlines the transformative potential of modern web technologies, such as the MERN stack, in bridging gaps in educational and professional delivery, providing a scalable model for student-centered innovation.

Table Of Content

CHAPTER ONE

PROJECT PROPOSAL

1.1 Introduction

Professional development and hands-on experience are essential for students in Ethiopian higher education. While classroom instruction provides theoretical knowledge, preparing students for the workforce requires practical exposure through internships. The success of these programs depends on effective coordination between universities, students, and industry partners[1].

In Ethiopia, internship management in many universities still relies on manual processes. Communication often depends on emails, paper forms, and inconsistent approval timelines, leaving students uncertain about their application status and making it difficult for universities to monitor students' internship progress. These manual systems not only slow down placements but can also discourage Ethiopian companies from collaborating with universities due to the lack of streamlined and transparent workflows[2] .

Digital technology has the potential to improve internship administration in Ethiopian higher education. Web-based solutions allow institutions to transition from paper-based processes to interactive systems that provide real-time updates, secure access, and role-based functionalities. By leveraging the MERN stack (MongoDB, Express, React, Node.js), universities can implement scalable systems that enhance operational efficiency, reduce administrative delays, and improve communication between students and industry partners [3].

This project aims to develop **InternConnect**, a web-based system designed to serve Ethiopian universities and students. InternConnect will centralize core internship activities, including opportunity discovery, application tracking, and document verification, promoting transparency and reducing delays in the placement process. By connecting students, universities, and companies, the system seeks to enhance professional development opportunities and strengthen industry-academic collaboration within Ethiopia .

In conclusion, while internships are vital for the professional growth of Ethiopian students, traditional manual systems can act as obstacles. InternConnect provides a modern, digital approach that improves workflow efficiency, supports industry partnerships, and offers a scalable solution for Ethiopian universities to better manage student internships [4].

1.2 Statement of the Problem and Justification

Internship management in most universities and organizations still relies on manual and fragmented processes[5]. Students often search across multiple websites or physical notice boards for internship opportunities, which can result in missed or delayed applications. Companies must manually review inconsistent application formats, incomplete documents, and disorganized submissions. Universities lack a streamlined method to track which students have applied, where they have been placed, or whether required documents have been submitted[6].

This fragmented process creates several key challenges:

1. **Uniform Submission Formats**
   Students submit applications and documents in different formats, leading to errors, missing information, and extra administrative work for universities and companies. A standardized digital system ensures that all submissions follow a consistent structure, making processing faster, simpler, and more accurate.

2. **Reliable Communication Between All Parties**

   In manual systems, communication between students, supervisors, company managers, and university coordinators is often scattered across emails, phone calls, or in-person interactions. Important updates can be missed, causing confusion and delays. A digital system enables direct messaging, notifications, and alerts, ensuring all stakeholders receive timely information about applications, assessments, and placements.

3. **Timely Reviews and Approvals**

   Manual review processes for applications and assessments are slow and inconsistent. Supervisors and company managers may take days or weeks to provide feedback or approve requests. A web-based system allows for automated and tracked workflows, ensuring decisions are made promptly and reducing uncertainty for students and companies.

4. **Accurate Documentation and Placement Records**

   Tracking student progress, submitted documents, and internship placements manually is prone to errors and missing information. A centralized system provides secure, accessible, and accurate records, enabling universities to monitor compliance, generate reports, and ensure students meet all internship requirements.

1.3 Objectives

1.3.1 General objective

The general objective of the project is to design and develop InternConnect, a web-based system that streamlines internship management for students, universities, and companies in Ethiopia.

1.3.2 Specific Objectives

To achieve the general objective of developing InternConnect, this project has the following specific objectives:

- **Gather requirements** from stakeholders, including students, university coordinators, supervisors, and company managers, through interviews, surveys, and available documentation.

- **Organize and analyze requirements** into functional and non-functional specifications to ensure the system meets the needs of all stakeholders.

- **Design a web-based application** that incorporates role-based access, internship application tracking, assessment management, and reporting functionalities.

- **Implement the system** using the chosen technology stack (MERN: MongoDB, Express, React, Node.js) to ensure a scalable, responsive, and secure system.

- **Test and evaluate the system** using a set of predefined test cases to verify its functionality, usability, and reliability, and ensure it meets the requirements gathered during analysis.

1.4. Scope And Limitations of the Project

1.4.1 Scope

The scope of InternConnect covers managing internship activities for students, universities, and partner companies in Ethiopia. The system is designed to provide a centralized system that brings together all the key processes involved in internship management. This includes the submission of applications, tracking of student progress, assessment and evaluation of performance, communication between stakeholders, and generation of reports for administrative purposes.

InternConnect is intended to serve as a coordinating tool that simplifies interactions between students, university staff, and companies. By providing a single system for all internship-related activities, it helps to reduce delays, minimize errors, and ensure that records are accurately maintained. The system is designed to support multiple users with different roles, allowing each stakeholder to perform their tasks efficiently while maintaining proper oversight and accountability.

Additionally, the system facilitates better organization of data and workflow management, helping universities and companies keep track of placements and approvals. InternConnect also aims to provide easy access to information for students regarding available opportunities, their application status, and required tasks, thereby improving the overall internship experience.

In summary, the scope of InternConnect encompasses the full cycle of internship management, from application submission to completion, and focuses on enhancing efficiency, coordination, and record-keeping for all stakeholders involved.

1.4.2 Limitations

The InternConnect system has the following limitations:

- The system does not support internship processes with international companies; it is designed specifically for Ethiopian universities and local organizations.
  .
- It does not handle financial transactions or include a payment gateway for fees, stipends, or related activities
  .
- The system does not provide advanced analytics or predictive placement recommendations for students or companies; decision-making remains manual in these aspects.

1.5 System Development Methodology.

1.5.1 Software Development Models

For the development of InternConnect, the project will adopt the Agile Software Development Life Cycle (SDLC) [7]. Agile emphasizes close collaboration between the development team and stakeholders, and it uses an iterative approach to gradually build and improve software. Work is organized into short, repeatable cycles called sprints, usually lasting between two and four weeks, which allow for continuous progress and frequent review.

The methodology relies on self-managed, cross-functional teams that gather requirements, design features, and implement solutions in small increments. Agile is particularly suitable for InternConnect because it allows flexibility in responding to changes, which is important when dealing with the varied needs of students, supervisors, company managers, and university coordinators.

Using Agile, the development team can incorporate feedback from stakeholders continuously, adjust priorities as needed[7], and deliver features that meet user requirements more effectively. Each sprint will include planning, development, and testing phases, ensuring that issues related to functionality, performance, and security are identified and addressed promptly. This approach ensures that InternConnect evolves into a robust and user-friendly internship management system that meets the expectations of all its users.

1.5.2 Investigation (Fact-Finding) Methods

For the development of InternConnect, various investigation methods will be used to gather information and ensure the system meets the needs of all stakeholders, including students, university coordinators, supervisors, and partner companies. These methods include:

- **Questionnaires:** Structured questionnaires will be distributed to students, university staff, and company representatives to collect quantitative data on their requirements, challenges, and expectations regarding the internship process. This approach allows the team to gather information from a larger audience and identify common needs or pain points.

- **Interviews:** One-on-one interviews will be conducted with students, supervisors, company managers, and university coordinators to obtain detailed insights into their experiences and expectations. Interviews allow for in-depth discussion of specific features, workflows, and challenges, providing valuable qualitative data to guide system design.

- **Document Analysis:** Existing records, forms, and manual internship tracking processes will be reviewed to understand how the current system operates and identify inefficiencies. This analysis helps clarify the problems the new system should address and informs the design of more effective workflows.

By combining these methods, the project team can gather both quantitative and qualitative data[8], ensuring that InternConnect is designed to meet user needs, improve efficiency, and support the internship process effectively.

1.5.3 System Development Tools

This table presents the system development tools and technologies used in the implementation of the InternConnect system. It summarizes the frontend, backend, database, development tools, and deployment systems selected to support efficient development, testing, and deployment of the web-based application.

Table 1.5.3 System Development Tools

Frontend

Tool/Technology	Purpose
React.js / Next.js	Building responsive, modular user interfaces

Backend

Tool/Technology	Purpose
Node.js with Express	Scalable API development

Database

Tool/Technology	Purpose
MongoDB	Structured and flexible data storage

ORM

Tool/Technology	Purpose
Prisma	Efficient database modeling and communication

Development Tools

Tool/Technology	Purpose
Git / GitHub	Version control
Figma	UI/UX design
Postman	API testing

Deployment

Tool/Technology	Purpose
Docker	Containerization for consistent deployment
Vercel	Cloud hosting and deployment

1.6 Significance of the Project

The InternConnect project addresses practical problems observed in internship management within universities and organizations in Ethiopia. Its significance is explained based on the direct benefits it provides to each stakeholder involved in the internship process.

- **Support for Students:**
  InternConnect helps students by providing a single system where they can apply for internships, follow up on application decisions, view assigned tasks, request assessments, and generate internship reports. This reduces reliance on multiple communication channels and manual document submission.

- **Structured Internship Supervision:**
  The system supports supervisors by allowing them to assign tasks, monitor student progress, and complete assessments within the system. This ensures that student performance is documented and reviewed based on clearly recorded activities.

- **Improved University Oversight:**
  Universities can use the system to track student placements, monitor internship progress, and verify submission of required documents. This allows coordinators to follow internship activities without relying on informal reports or physical files.

- **Support for Partner Companies:**
  Companies benefit from a standardized process for reviewing applications, assigning supervisors, and approving assessments. This red136es repeated communication and helps companies keep records of interns under their supervision.

- **Contribution to Digital Academic Systems:**
  InternConnect contributes to the adoption of web-based systems in higher education by replacing manual internship handling with a structured digital process aligned with current university needs.

1.7 Beneficiaries

- **Students:**
  Students benefit from having a single system for managing internship applications, task follow-ups, assessment requests, and report generation, which simplifies their internship experience.

- **University Coordinators and Departments:**
  Coordinators benefit from direct access to internship data, including placement status, supervisor assignments, and assessment records, supporting academic follow-up and documentation.

- **Companies and Supervisors:**
  Companies and supervisors benefit from organized application reviews, task assignments, and assessment submission without relying on email chains or paper-based records.

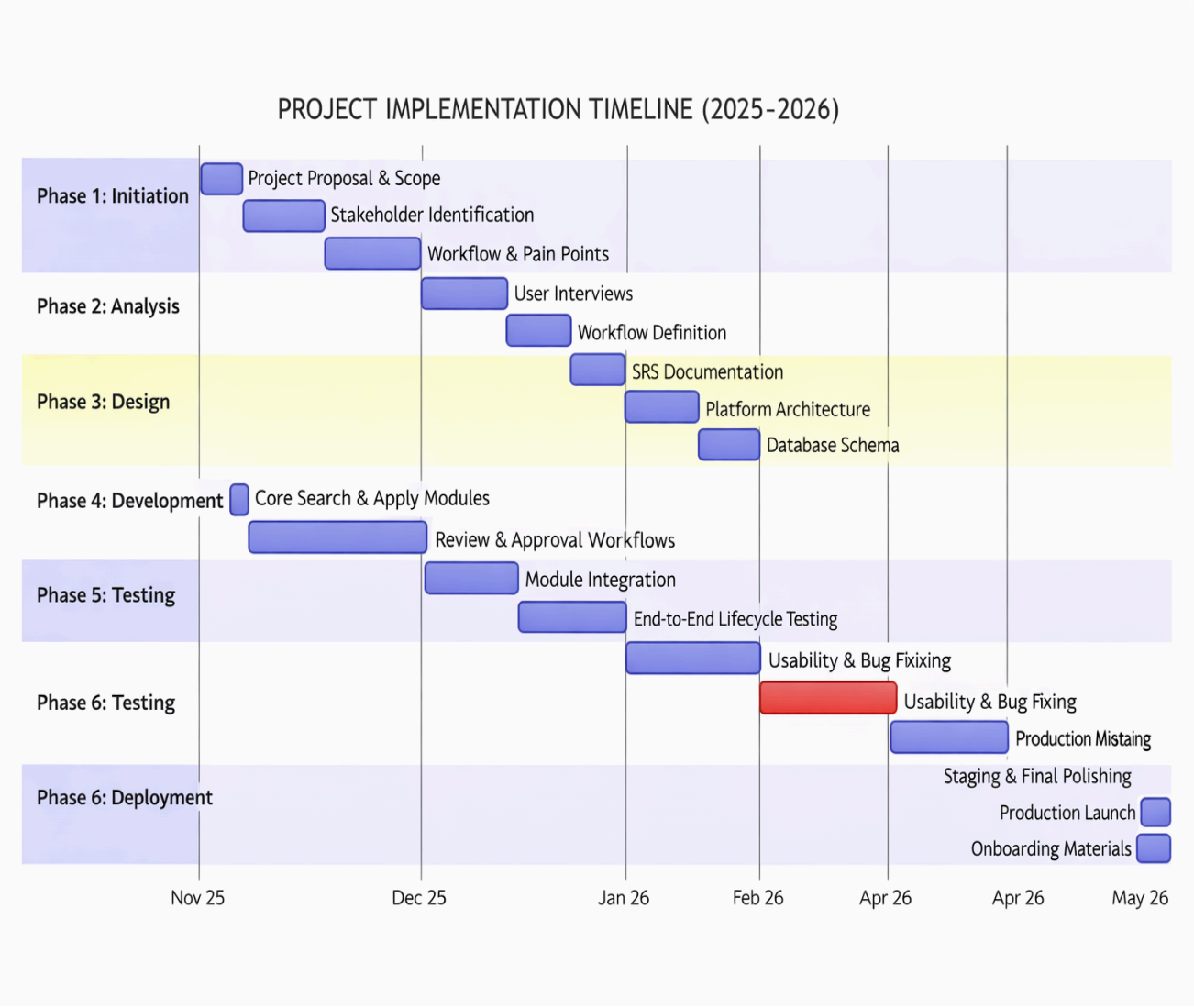
- **System Administrators:**
  Administrators benefit from tools that allow them to manage user access, control system usage, and generate university-level reports.

- **Higher Education Institutions:**
  Institutions benefit from centralized internship records that can be used for academic evaluation, reporting, and planning of future internship programs.

1.8 Project Time Schedule

The InternConnect System will be developed over a 6-month period (26 weeks) with the following major phases and milestones.

===



CHAPTER TWO

REQUIREMENT ANALYSIS

2.1 Introduction

Requirement analysis is a critical phase in the software development life cycle because it defines what a system should accomplish and the conditions under which it should operate. Without clearly identifying requirements early in the project, it becomes difficult to develop a system that meets both user needs and organizational objectives. For the InternConnect system, requirement analysis focuses on understanding the expectations of students, university coordinators, supervisors, and partner companies. By analyzing and documenting these needs, developers can ensure that the final system effectively supports internship management processes in Ethiopian universities. [8]

The process of requirement analysis involves identifying both functional and non-functional requirements. Functional requirements describe the specific tasks the system must perform[9], such as application submission, task assignment, progress tracking, and assessment reporting. Non-functional requirements, on the other hand, define quality and operational constraints, including system performance, security, usability, reliability, and accessibility. Together, these requirements determine how the InternConnect system behaves and how well it fulfills its intended purpose.

A thorough requirement analysis provides multiple benefits. It reduces the risk of scope changes and prevents costly rework in later stages of development. It also facilitates better communication among stakeholders by establishing a shared understanding of system expectations. Clearly documented requirements serve as a guide for designers and developers and provide measurable criteria for testers to verify that the system meets its objectives. [9]

In this project, various methods are used to gather requirements, including interviews with students, supervisors, and university coordinators; questionnaires to collect structured information; and document analysis of current internship procedures. These techniques allow the project team to collect both quantitative and qualitative data, ensuring that all relevant perspectives are considered.

By combining careful investigation with systematic documentation, this chapter establishes a solid foundation for developing InternConnect. It ensures that the system will provide practical solutions to the challenges faced by students, universities, and companies during the internship process, leading to improved organization, communication, and record-keeping throughout the internship cycle.

2.2 Current System

2.2.1 Overview of Existing System

The current internship management process in many Ethiopian universities relies on manual procedures and limited digital support. Students must first visit their departments to obtain a letter of recommendation, which they submit in person to companies offering internship opportunities. Companies review these letters manually, approve the internship, and assign supervisors by hand.

Tracking of student work progress is also manual. Supervisors record student activities on paper, and updates are sent to the department when needed. Upon completion of the internship, students submit their final reports or completion letters back to the department, where staff record the results in physical files.

This system is primarily focused on allowing students to apply for internships and submit reports but does not provide any centralized or systematic way to track placements, monitor student progress, or manage communications between stakeholders. Each participant, students, supervisors, companies, and university staff must rely on manual processes, which are time-consuming and prone to inconsistencies.

2.2.2 Problems of the Existing System

The current internship management system exhibits several critical weaknesses that limit its effectiveness:

- **Susceptibility to Data Loss:** Physical documents and records are easily misplaced, lost, or damaged. This creates gaps in documentation and increases the risk of errors in student evaluations and internship approvals.

- **Limited Oversight of Student Progress:** Supervisors and university staff cannot easily monitor ongoing work. Progress tracking is inconsistent, often recorded informally, and students may submit incomplete or inaccurate reports, which can affect grading and assessment.

- **Manual Communication and Coordination:** Communication between students, companies, supervisors, and departments relies on face-to-face interactions, phone calls, or emails. This often leads to delays, miscommunication, and missing information.

- **Vulnerability to Academic Dishonesty:** The current paper-based process is prone to manipulation or falsification of documents and results, reducing trust in the system.

- **Fragmented Data Management:** Student applications, approvals, supervisor assignments, and completion records are stored in multiple locations with no integration. This makes reporting, performance evaluation, and long-term analysis of internship outcomes difficult.
- **Difficulty Accessing Internship Opportunities:** Due to the lack of a centralized system, students may miss available internship programs or fail to know which opportunities are open to them. This limits participation and reduces the overall effectiveness of the internship process.

2.3 Requirement Gathering

2.3.1 Requirement Gathering Methodologies

Requirement-gathering methodologies for the InternConnect system involve a combination of techniques to ensure a clear understanding of the needs of students, universities, supervisors, and partner companies.

- **Interviews:** Conducting interviews with key stakeholders including students, university coordinators, supervisors, and company managers helps uncover specific requirements and insights related to internship management, such as task tracking, application workflow, and reporting needs.

- **Questionnaires and Surveys:** Structured surveys are distributed to a larger audience of students and companies to collect quantitative data on common challenges, expectations, and preferred features. This helps identify trends and ensure the system addresses the needs of most users.

- **Workshops and Collaborative Sessions:** Organizing group discussions with university staff and company representatives allows stakeholders to brainstorm and align on the goals of the system. These sessions also help refine requirements based on shared perspectives.

- **Observation:** Observing how students currently apply for internships, how supervisors track work progress, and how departments manage records provides context on practical challenges and highlights inefficiencies in existing processes.

- **Document Analysis:** Reviewing current internship procedures, departmental guidelines, and company internship policies ensures that the system is aligned with existing regulations and best practices, while identifying gaps that the new system can address.

Requirements Categories:

- **Functional Requirements:** Specific tasks and actions the system must perform, such as student registration, internship application submission, task tracking, assessment submission, and report generation.

- **Non-Functional Requirements:** Attributes of the system, including usability, performance, security, accessibility, and reliability.

2.3.2 Results Found

The requirement-gathering process for InternConnect yielded several high-level findings that directly shape the design and functionality of the system:

- **Student Needs:** Students expressed the need for a single system to view available internship opportunities, track application status, and monitor assigned tasks. They also emphasized the importance of generating reports and receiving timely updates on approvals and assessments.

- **Supervisor Needs:** Supervisors require an easy way to assign tasks, monitor student progress, and submit assessments in a structured format that can be tracked by the university.

- **University Coordinator Needs:** Coordinators highlighted the need for centralized tracking of student applications, placement approvals, and completion records. They also wanted tools for generating reports for departmental evaluation.

- **Company Needs:** Companies requested standardized application formats, a method to review applications efficiently, assign supervisors, and provide feedback on student performance.

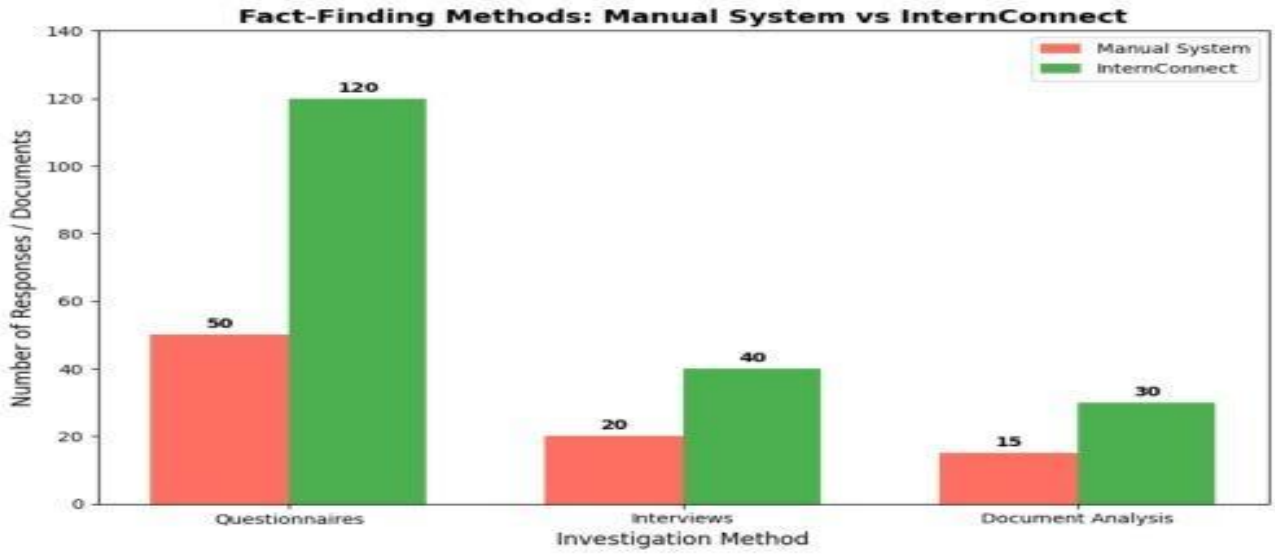
These insights were collected through:

- **University Departments:** Engaging with students, coordinators, and academic staff provided insights into daily challenges in internship management.

- **Partner Companies:** Discussions with company managers revealed common difficulties in reviewing applications and assigning supervisors.

- **Internship Events and Career Fairs:** Observing internship recruitment and outreach programs helped identify gaps in communication and tracking of student progress.

Through these findings, the project establishes a foundation for a system that improves coordination, reduces delays, and ensures accurate documentation throughout the internship process. The chapter highlights the techniques used to gather requirements, the importance of stakeholder input, and the iterative approach needed to adapt the system to evolving needs and practical realities of internship management in Ethiopia.



2.4 Proposed System

2.4.1 Overview

The proposed InternConnect system is designed to provide a comprehensive digital solution for managing internship activities for students, universities, and partner companies in Ethiopia. This system will serve as an integrated system that streamlines the entire internship process, from application submission to performance tracking and reporting. Key features of the proposed system include centralized application management, supervisor task assignment, progress tracking, assessment submissions, and report generation, allowing all stakeholders to coordinate effectively in one system.

InternConnect is intended to be an all-in-one system that makes internship management more organized and less prone to errors. Students will have access to dashboards where they can view available internship opportunities, track the status of their applications, monitor assigned tasks, and generate reports upon completion. Supervisors can assign tasks, monitor student progress, and submit assessments directly through the system, while companies can review applications, approve interns, and assign supervisors without relying on manual processes.

The system also provides university coordinators and administrators with tools to manage user accounts, track placements, and generate reports for departmental evaluation. By centralizing internship activities, InternConnect reduces delays, prevents data loss, ensures accurate documentation, and provides a structured workflow that supports better decision-making. This integrative approach enables students, companies, and universities to participate in a coordinated internship program, improving overall management and accountability.

2.4.2 Functional Requirements

The proposed InternConnect system will include several essential functional requirements to ensure efficient management of internships for students, universities, and partner companies.

FR 1: The system shall allow students to register an account by providing the required personal and academic information.

FR 2: The system shall allow registered users to log in to and log out of the system.

FR 3: The system shall allow students to view available internship opportunities and apply for internships.

FR 4: The system shall allow students to view and track the status of their internship applications.

FR 5: The system shall allow students to view assigned tasks and generate or submit internship reports.

FR 6: The system shall allow supervisors to assign tasks to students.

FR 7: The system shall allow supervisors to monitor student progress and submit performance assessments.

FR 8: The system shall allow supervisors to approve or reject internship-related requests.

FR 9: The system shall allow company managers to review student internship applications and assign supervisors.

FR 10: The system shall allow company managers to approve or reject internship applications.

FR 11: The system shall allow university coordinators to send internship invitations and track the invitation status.

FR 12: The system shall allow university coordinators to generate internship placement and completion reports.

FR 13: The system shall allow the system administrator to activate, deactivate, and manage user accounts.

FR 14: The system shall allow the system administrator to monitor system performance and resolve system issues.

2.4.3 Non-Functional Requirements

Non-functional requirements describe the conditions and constraints under which the InternConnect system must operate. Unlike functional requirements, these requirements do not define specific actions of the system but focus on how well the system performs its functions. They address aspects such as system speed, data protection, availability, ease of use, and long-term operation. Defining these requirements helps ensure that the system not only works correctly but also provides a stable, secure, and usable environment for students, universities, company managers, and administrators. Accordingly, the non-functional requirements of the system are listed below[10]

2.4.3.1 User Interface and User Experience

- The user interface of the InternConnect system will be designed with a focus on simplicity and ease of use. Clear labels, organized layouts, and consistent navigation will be used to ensure that users such as students, supervisors, and coordinators can complete their tasks without confusion. The system will be understandable even for users with limited experience using web-based applications.

2.4.3.2 Documentation

- Comprehensive documentation will be prepared throughout the system development process. This documentation will include requirement specifications, system design descriptions, implementation details, testing procedures, and user guidelines. Proper documentation will support system maintenance, future updates, and ease of use for administrators and end users.

2.4.3.3 Hardware Consideration

- The InternConnect system will be designed to operate smoothly on commonly available computing devices. It will be compatible with major web browsers and will not require specialized hardware. The system will ensure stable communication between the client and server to support timely processing of internship-related requests.

2.4.3.4 Performance Characteristics

- The system will support access from different devices, including desktop computers, laptops, tablets, and smartphones. It will be capable of handling multiple users at the same time without causing noticeable delays in task execution, application submission, or report generation.

2.4.3.5 Error Handling and Extreme Conditions

- The system will validate user inputs to reduce errors during registration, application submission, and assessment processes. Meaningful error messages will be displayed when incorrect or incomplete data is entered. In case of unexpected system issues, basic recovery mechanisms will be in place to prevent data loss.

2.4.3.6 Quality Issues

- The InternConnect system will be available for use at all times, provided an internet connection is available. Multiple users will be able to access and use the system simultaneously while maintaining consistent system behavior and data accuracy.

2.4.3.7 Security Issues

- User access to the system will require authentication based on assigned roles. Sessions will automatically expire after a period of inactivity to reduce unauthorized access. Administrative functions will be restricted to authorized system administrators only.

2.4.3.8 Physical Environment

- The InternConnect application will be hosted on the Internet, allowing users to access it from different locations. No physical installation will be required on user devices other than a web browser and an active internet connection.

2.3.6.1 Scenario

A scenario represents a specific instance of a use case, describing a concrete sequence of actions that fulfills a particular goal within the system. Scenarios offer detailed insights into how users interact with the system, highlighting key processes and their realizations[9].

Table 2.1 Scenario – Login

Scenario ID	SC-02
Scenario Name	Student Registration and Profile Management
Participating Actor(s)	Student (Dawit)
Precondition	Student has valid academic information
Main flow	1. Students navigate to the registration page. 2. Fills out academic and personal details. 3. Uploads CV and supporting documents. 4. Saves profile. 5. The system confirms the profile is stored. 6. Students can later edit or save as draft.

Table 2.2 Scenario – Student Registration and Profile Management

Scenario ID	SC-02
Scenario Name	Student Registration and Profile Management
Participating Actor(s)	Student (alex)
Precondition	Student has valid academic information

Main flow	1. Students navigate to the registration page. 2. Fills out academic and personal details. 3. Uploads CV and supporting documents. 4. Saves profile. 5. The system confirms the profile is stored. 6. Students can later edit or save as draft.

Table 2.3 Scenario – Apply for Internship

Scenario ID	**SC-03**
Scenario Name	Apply for Internship
Participating Actor(s)	Student (alex)
Precondition	Student profile is completed; internship posting is active
Main flow	1. Students log in and view available internships. 2. Selects an internship. 3. Fills any required application forms. 4. Submits application. 5. The system stores the application and confirms submission. 6. Students can view status after submission.

Table 2.4 Scenario – View Application Status

Scenario ID	SC-04
Scenario Name	View Application Status
Participating Actor(s)	Student (Dawit)
Precondition	Student has submitted at least one internship application
Main flow	1. Students log in. 2. Navigates to the "Application Status" section. 3. The system retrieves the current status of applications. 4. Student views status (read-only).

Table 2.5 Scenario – Search Organization

Scenario ID	SC-05
Scenario Name	Search Organization
Participating Actor(s)	University Coordinator (Selam)

Precondition	University Coordinator is logged in
Main flow	1. Coordinator enters search criteria for organizations. 2. The system displays matching organizations. 3. Coordinator selects organization for further actions.

Table 2.6 Scenario – Send Invitation

Scenario ID	**SC-06**
Scenario Name	Send Invitation
Participating Actor(s)	University Coordinator (Selam)
Precondition	Organization exists in system
Main flow	1. Coordinator selects organization. 2. Sends an invitation to participate in internships. 3. System records invitation status.

	4. Organization receives notification of invitation.

Table 2.7 Scenario – View Internship Requests

Scenario ID	**SC-07**
Scenario Name	View Internship Requests
Participating Actor(s)	Company Manager (Hanna)
Precondition	Internship requests have been submitted
Main flow	1. Company Manager logs in. 2. Navigates to "Internship Requests" dashboard. 3. Views submitted requests along with applicant details.

Table 2.8 Scenario – Approve / Reject Internship Request

Scenario ID	**SC-08**

Scenario Name	Approve / Reject Internship Request
Participating Actor(s)	Company Manager (Hanna)
Precondition	Pending internship request exists
Main flow	1. Manager reviews submitted request. 2. Decides to approve or reject requests. 3. System updates request status. 4. The applicant is notified of the decision.

Table 2.9 Scenario – Assign Supervisor

Scenario ID	**SC-09**
Scenario Name	Assign Supervisor
Participating Actor(s)	Company Manager (Hanna)
Precondition	Internship request approved
Main flow	1. The manager selects students assigned to internships. 2. Assigns a supervisor. 3. System records supervisor assignment.

	4. Students and supervisors receive notifications.

Table 2.10 Scenario – Assign Task and Monitor Progress

Scenario ID	**SC-10**
Scenario Name	Assign Task and Monitor Progress
Participating Actor(s)	Supervisor (Daniel)
Precondition	Student assigned to supervisor
Main flow	1. The supervisor assigns tasks to students. 2. Student updates task progress. 3. Supervisor monitors updates and reviews task completion. 4. The system logs all progress for reporting.

Table 2.11 Scenario – Request and Fill Assessment

Scenario ID	SC-11
Scenario Name	Request and Fill Assessment
Participating Actor(s)	Student (Dawit), Supervisor (Daniel)
Precondition	Student is assigned to internship
Main flow	1. The student requests assessment. 2. Supervisor receives requests and fills evaluation forms. 3. Submits assessment. 4. System records assessment. 5. Student notified of completion.

Table 2.12 Scenario – Activate / Deactivate User

Scenario ID	SC-12
Scenario Name	Activate / Deactivate User
Participating Actor(s)	System Administrator (Mekdes)

Precondition	Admin logged in
Main flow	1. Admin selects user accounts. 2. Choose to activate or deactivate the account. 3. System updates account status. 4. The user receives notification.

Table 2.13 Scenario – Generate Reports and Analytics

Scenario ID	**SC-13**
Scenario Name	Generate Reports and Analytics
Participating Actor(s)	University Admin (Selam), Management (Kebede)
Precondition	Reporting privileges granted
Main flow	1. User selects report type. 2. Applies filters. 3. Generates reports. 4. The system displays reports for viewing/export.

Table 2.14 Scenario – Audit and Compliance Review

Scenario ID	SC-14
Scenario Name	Audit and Compliance Review
Participating Actor(s)	Auditor (Betelhem), System Admin (Mekdes)
Precondition	Audit access granted
Main flow	1. Define audit scope. 2. Access system logs. 3. Review activities. 4. Record findings. 5. The system generates audit reports.

2.3.6.2 Use Case Model

The use case model describes how various actors interact with the InternConnect system to accomplish specific goals. It captures the functional requirements of the system from the users' perspective and defines the core services that the system provides. The use case model acts as a foundational element for system analysis, design, and subsequent implementation.

InternConnect supports multiple user roles, each with distinct responsibilities and access privileges. The primary actors interacting with the system include:

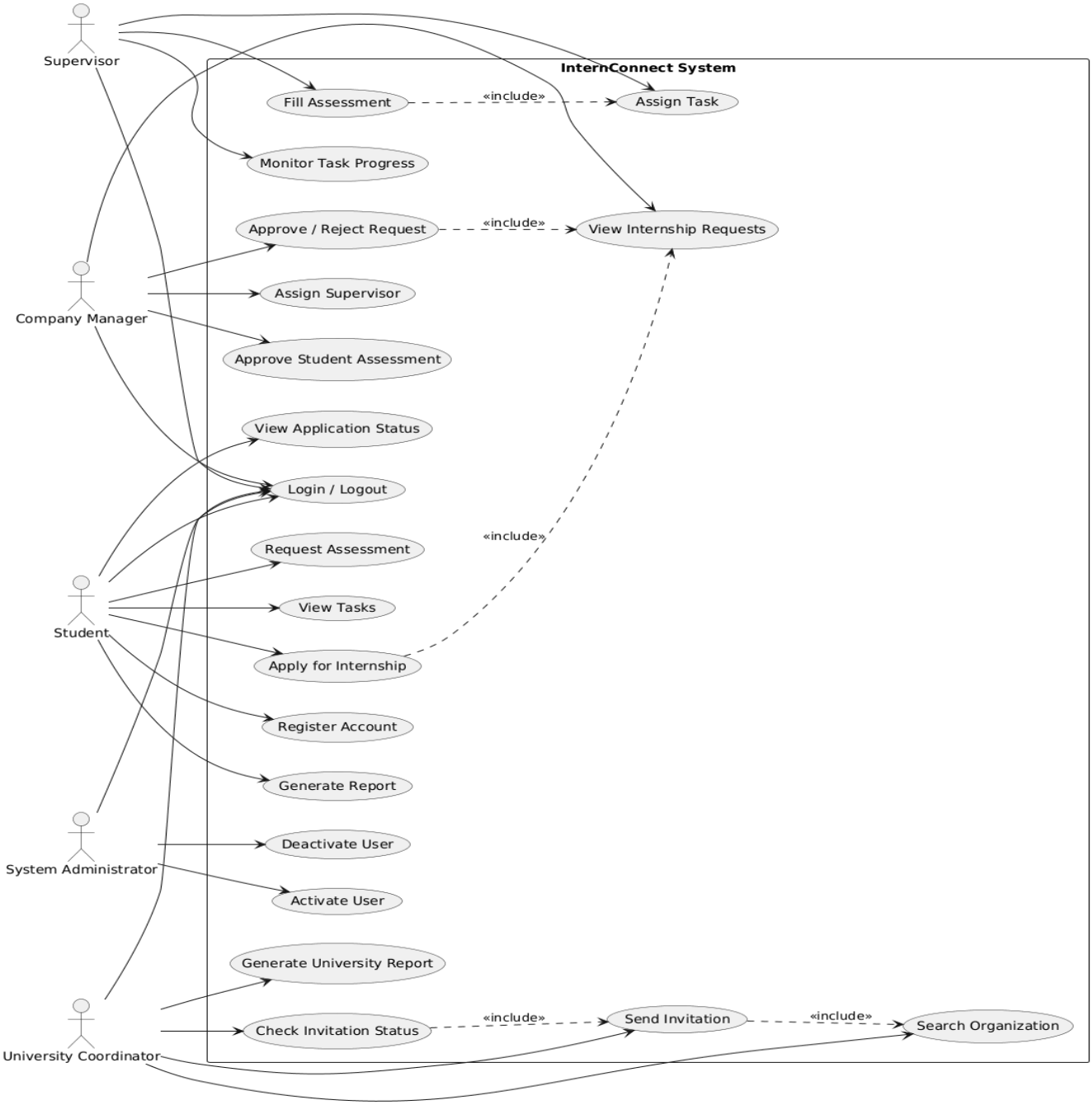
- Student
- Company Representative
- Internship Coordinator
- University Coordinator
- System Administrator

Each actor interacts with one or more use cases based on their role within the internship management process. For instance, students are responsible for creating profiles and applying for internships, while company representatives post internship opportunities and evaluate applications. Internship coordinators and department heads oversee the approval process and ensure alignment with academic requirements, whereas system administrators manage users and system configurations.

The use case model visually represents these interactions using a use case diagram. The diagram illustrates the system boundary, actors, and major functional use cases such as user registration, internship posting, application management, approval workflows, and reporting. These use cases collectively define the operational scope of InternConnect and ensure that all stakeholder requirements are addressed.

The use case diagram (to be included) serves as a high-level functional map of the system and provides a shared understanding for developers, stakeholders, and academic evaluators.

2.3.6.2.1 Use Case Diagram



2.3.6.2.2 Description of Selected Use Case Models

This subsection provides detailed descriptions of the key use cases identified in the system. Each use case is documented using a structured template that includes the goal, actors, pre-conditions, flow of events, alternative and exception flows, post-conditions, and applicable business rules.

Table 2-7: User Authentication and Authorization

Item	Description
Use Case ID	UC001
Use Case Name	User Authentication and Authorization
Goal	Allow students, universities, and companies to securely access the system with role-based permissions
Participating Actor(s)	Student, University Admin, Company Recruiter, System Admin
Pre-conditions	User has valid credentials; Account is active

Flow of Events	1. User opens login page2. Enter email and password3. System validates credentials4. User role identified5. Appropriate dashboard loaded6. Login activity logged
Alternative Flow	Two-factor authentication required
Exception Flow	Invalid credentials entered; Account suspended
Post-conditions	User session established or access denied
Business Rules	Strong password policy; Session timeout; Account lockout after failed attempts

Table 2-8: Student Registration and Profile Management

Item	Description
Use Case ID	UC002
Use Case Name	Student Registration and Profile Management
Goal	Enable students to create and manage academic and personal profiles

Participating Actor(s)	Student
Pre-conditions	Student should login to the system Student has valid academic information
Flow of Events	1. Student registers account 2. Enter academic details 3. Uploads CV and documents 4. Saves profile
Alternative Flow	Profile saved as draft
Exception Flow	Required fields missing or invalid
Post-conditions	Student profile stored and available
Business Rules	One active profile per student

Table 2-9 :Post Internship Opportunity Use Case

Item	Description
Use Case ID	UC005
Use Case Name	Post Internship Opportunity
Goal	Allow companies to publish internship opportunities for students

Participating Actor(s)	Company Recruiter / Company Manager
Pre-conditions	The user must login to the system Company account is approved and active
Flow of Events	1. Company Recruiter logs into the system. 2. Navigates to “Post Internship”. 3. Enters internship details (title, description, requirements, duration, location). 4. Sets application deadline. 5. Submits internship posting. 6. The system validates the information. 7. Internship is published.
Post-conditions	Internship opportunity is visible and searchable by students
Business Rules	Application deadline is mandatory; Only approved companies can post internships

Table 2-10: Apply for Internship

Item	Description
Use Case ID	UC003

Use Case Name	Apply for Internship
Goal	Allow students to apply for available internships
Participating Actor(s)	Student
Pre-conditions	User should login to the system. Completed student profile; Internship is active
Flow of Events	1. Student selects internship2. Submits application3. System stores application
Alternative Flow	Application saved as draft
Exception Flow	Application deadline passed
Post-conditions	Application recorded in system
Business Rules	One application per internship per student

Table 2-11: View Application Status

Item	Description
Use Case ID	UC004

Use Case Name	View Application Status
Goal	Allow students to track internship application progress
Participating Actor(s)	Student
Pre-conditions	User should login to the system Student has submitted an application
Flow of Events	1. Student logs in 2. Opens application status page 3. System displays current status
Alternative Flow	Not applicable
Exception Flow	Application not found
Post-conditions	Application status viewed
Business Rules	Status is read-only for students

Table 2-12: Search Organization

Item	Description
Use Case ID	UC005
Use Case Name	Search Organization

Goal	Allow universities to find partner organizations
Participating Actor(s)	University Coordinator
Pre-conditions	University coordinator logged in
Flow of Events	1. Enter search criteria 2. System displays matching organizations
Alternative Flow	No matching organization found
Exception Flow	Invalid search criteria

Table 2-13: Send Invitation

Item	Description
Use Case ID	UC006
Use Case Name	Send Invitation
Goal	Invite companies to participate in internship programs
Participating Actor(s)	University Coordinator

Pre-conditions	Logged in Organization exists in system
Flow of Events	1. Select organization 2. Send invitation
Alternative Flow	Invitation resent
Exception Flow	Invalid organization selected
Post-conditions	Invitation sent

Table 2-14: View Internship Requests

Item	Description
Use Case ID	UC007
Use Case Name	View Internship Requests
Goal	Allow company managers to review internship requests
Participating Actor(s)	Company Manager
Pre-conditions	User should login to the system Internship requests submitted

Flow of Events	1. Login 2. Open requests dashboard
Alternative Flow	Not applicable
Exception Flow	No requests available

Table 2-15: Approve / Reject Internship Request

Item	Description
Use Case ID	UC008
Use Case Name	Approve / Reject Internship Request
Goal	Decide on internship participation requests
Participating Actor(s)	Company Manager
Pre-conditions	Login , Pending request exists
Flow of Events	1. Review request 2. Approve or reject request
Alternative Flow	Request deferred

Exception Flow	Request already processed

Table 2-16: Assign Supervisor

Item	Description
Use Case ID	UC009
Use Case Name	Assign Supervisor
Goal	Assign a supervisor to an accepted student
Participating Actor(s)	Company Manager
Pre-conditions	login Internship request approved
Flow of Events	1. Select student 2. Assign supervisor
Alternative Flow	Supervisor reassigned
Exception Flow	Supervisor unavailable

Table 2-17: Assign Task and Monitor Progress

Item	Description
Use Case ID	UC010
Use Case Name	Assign Task and Monitor Progress
Goal	Manage internship tasks and progress
Participating Actor(s)	Supervisor
Pre-conditions	login Student assigned to supervisor
Flow of Events	1. Assign task 2. Monitor progress
Alternative Flow	Task updated
Exception Flow	Task overdue

Table 2-18: Request and Fill Assessment

Item	Description
Use Case ID	UC011
Use Case Name	Request and Fill Assessment
Goal	Evaluate student internship performance

Participating Actor(s)	Student, Supervisor
Pre-conditions	login Student assigned to internship
Flow of Events	1. Student requests assessment 2. Supervisor fills assessment 3. Submit assessment
Alternative Flow	Assessment revised before submission
Exception Flow	Assessment deadline missed
Post-conditions	Assessment recorded
Business Rules	Assessment immutable after submission

Table 2-19: Activate / Deactivate User

Item	Description
Use Case ID	UC012
Use Case Name	Activate / Deactivate User
Goal	Control system access
Participating Actor(s)	System Administrator
Pre-conditions	Admin logged in

Flow of Events	1. Select user 2. Activate or deactivate account
Alternative Flow	Reactivate user
Exception Flow	Unauthorized operation
Post-conditions	User status updated
Business Rules	Admin-only operation

Table 2-20: Generate Reports and Analytics

Item	Description
Use Case ID	UC013
Use Case Name	Generate Reports and Analytics
Goal	Provide analytical insights on internships
Participating Actor(s)	University Admin, Management
Pre-conditions	ogged in Reporting privileges granted
Flow of Events	1. Select report 2. Apply filters 3. Generate report
Alternative Flow	Scheduled report generation

Exception Flow	No data available
Post-conditions	Report available for viewing/export
Business Rules	Read-only access

Table 2-21: Audit and Compliance Review

Item	Description
Use Case ID	UC014
Use Case Name	Audit and Compliance Review
Goal	Ensure system transparency and compliance
Participating Actor(s)	Auditor, System Administrator
Pre-conditions	login Audit access granted
Flow of Events	1. Define audit scope 2. Review logs 3. Record findings
Alternative Flow	Follow-up audit scheduled
Exception Flow	Logs missing or corrupted
Post-conditions	Audit report generated

Business Rules	Logs are immutable; Read-only auditor access

Table 2-22: View Internship Opportunities

Item	Description
Use Case ID	UC015
Use Case Name	View Internship Opportunities
Goal	Allow students to browse available internship opportunities
Participating Actor(s)	Student
Pre-conditions	Student is logged into the system
Flow of Events	1. Students navigate to the internship listings page. 2. The system displays all active internships. 3. Students apply filters (location, duration, field).

	4. Students view internship details.
Alternative Flow	No internships match the filter criteria
Exception Flow	System unable to retrieve internship listings
Post-conditions	Internship opportunities displayed to the student
Business Rules	Only active internships are visible; Listings are read-only

Table 2-23: Withdraw Internship Application

Item	Description
Use Case ID	UC016
Use Case Name	Withdraw Internship Application
Goal	Allow students to withdraw a submitted internship application
Participating Actor(s)	Student
Pre-conditions	Logged in Student has an active internship application

Flow of Events	1. Student logs into the system. 2. Navigates to application history. 3. Selects an application. 4. Chooses “Withdraw Application”. 5. System confirms withdrawal.
Alternative Flow	Student cancels withdrawal action
Exception Flow	Application already processed or deadline passed
Post-conditions	Application status updated to “Withdrawn”
Business Rules	Withdrawal allowed only before application review

Table 2-24: Accept / Reject Internship Offer

Item	Description
Use Case ID	UC017
Use Case Name	Accept / Reject Internship Offer
Goal	Allow students to respond to internship offers
Participating Actor(s)	Student
Pre-conditions	Logged in , Internship offer has been issued by the company

Flow of Events	1. Students receive offer notification. 2. Student opens offer details. 3. Students accept or reject the offer. 4. The system records the decision.
Alternative Flow	Offer expires without response
Exception Flow	Offer withdrawn by company
Post-conditions	Offer status updated; Company notified
Business Rules	One response per offer; Decision is final

Table 2-25: Internship Completion

Item	Description
Use Case ID	UC018
Use Case Name	Internship Completion
Goal	Formally close an internship and record completion status
Participating Actor(s)	Student, Supervisor, University Admin
Pre-conditions	Logged in , Internship period completed

Flow of Events	1. Student submits final internship report. 2. Supervisor reviews and confirms completion. 3. University Admin validates completion. 4. System marks internship as completed.
Alternative Flow	Report revision requested
Exception Flow	Final report not submitted
Post-conditions	Internship marked as completed and archived
Business Rules	Completion requires supervisor approval and final report

Table 2-26: Notifications Management

Item	Description
Use Case ID	UC019
Use Case Name	Notifications Management
Goal	Notify users about important system events
Participating Actor(s)	Student, Company Manager, Supervisor, University Admin

Pre-conditions	Logged in User has an active system account
Flow of Events	1. System detects an event (application update, approval, task assignment). 2. The system generates notifications. 3. Notification delivered to user dashboard/email.
Alternative Flow	User disables optional notifications
Exception Flow	Notification delivery failure
Post-conditions	Notification delivered and logged
Business Rules	Critical notifications cannot be disabled; Notifications are read-only

2.3.6.3 Object Model

The object model describes the static structure of the InternConnect system by identifying key objects (entities), their attributes, and relationships among them. It focuses on how data is organized rather than the dynamic behavior of the system. This model is critical for database design, API development, and ensuring consistency across the system.

InternConnect connects students, universities, and companies to facilitate internship opportunities, application tracking, and reporting. The object model identifies core entities and their relationships to support these functionalities.

User

Object	Attribute	Data Type	Description	Constraints
User	_id	ObjectId	Unique system user identifier	Primary Key
	email	String	User email address	Unique, Indexed, Required
	password	String	Hashed account password	BCrypt, Required
	firstName	String	User first name	Required
	lastName	String	User last name	Required
	userType	String	Role of the user	ENUM: student, university, company, admin
	status	String	Account status	ENUM: active, suspended, deleted
	createdAt	Date	Account creation time	Auto-generated
	updatedAt	Date	Last update time	Auto-generated

Student

Object	Attribute	Data Type	Description	Constraints
Student	_id	ObjectId	Student profile identifier	Primary Key

	userId	ObjectId	Reference to User	FK → User._id
	studentId	String	University-issued student ID	Unique, Required
	academicProfile	Object	Academic details	Optional
	cvUrl	String	Uploaded CV location	Optional
	verified	Boolean	Academic verification status	Default: false

University

Object	Attribute	Data Type	Description	Constraints
University	_id	ObjectId	University identifier	Primary Key
	userId	ObjectId	Linked user account	FK → User._id
	name	String	University name	Required, Unique
	Accreditation Code	String	Official accreditation code	Required
	verified	Boolean	Approval status	Default: false

Company

Object	Attribute	Data Type	Description	Constraints
Company	_id	ObjectId	Company identifier	Primary Key
	userId	ObjectId	Linked user account	FK → User._id
	name	String	Company name	Required, Unique
	industry	String	Business sector	Required
	verified	Boolean	Approval status	Default: false

Internship

Object	Attribute	Data Type	Description	Constraints
internship	_id	ObjectId	internship identifier	Primary Key
	title	String	internship title	Required
	description	String	internship details	Required
	companyId	ObjectId	Offering company	FK → Company._id

	Application Deadline	Date	Application closing date	Required
	status	String	Internship availability	ENUM: open, closed

Application

Object	Attribute	Data Type	Description	Constraints
Application	_id	ObjectId	Application identifier	Primary Key
	studentId	ObjectId	Applying student	FK → Student._id
	internshipId	ObjectId	Target internship	FK → internship._id
	status	String	Application state	ENUM: pending, shortlisted, accepted, rejected
	submittedAt	Date	Submission timestamp	Required

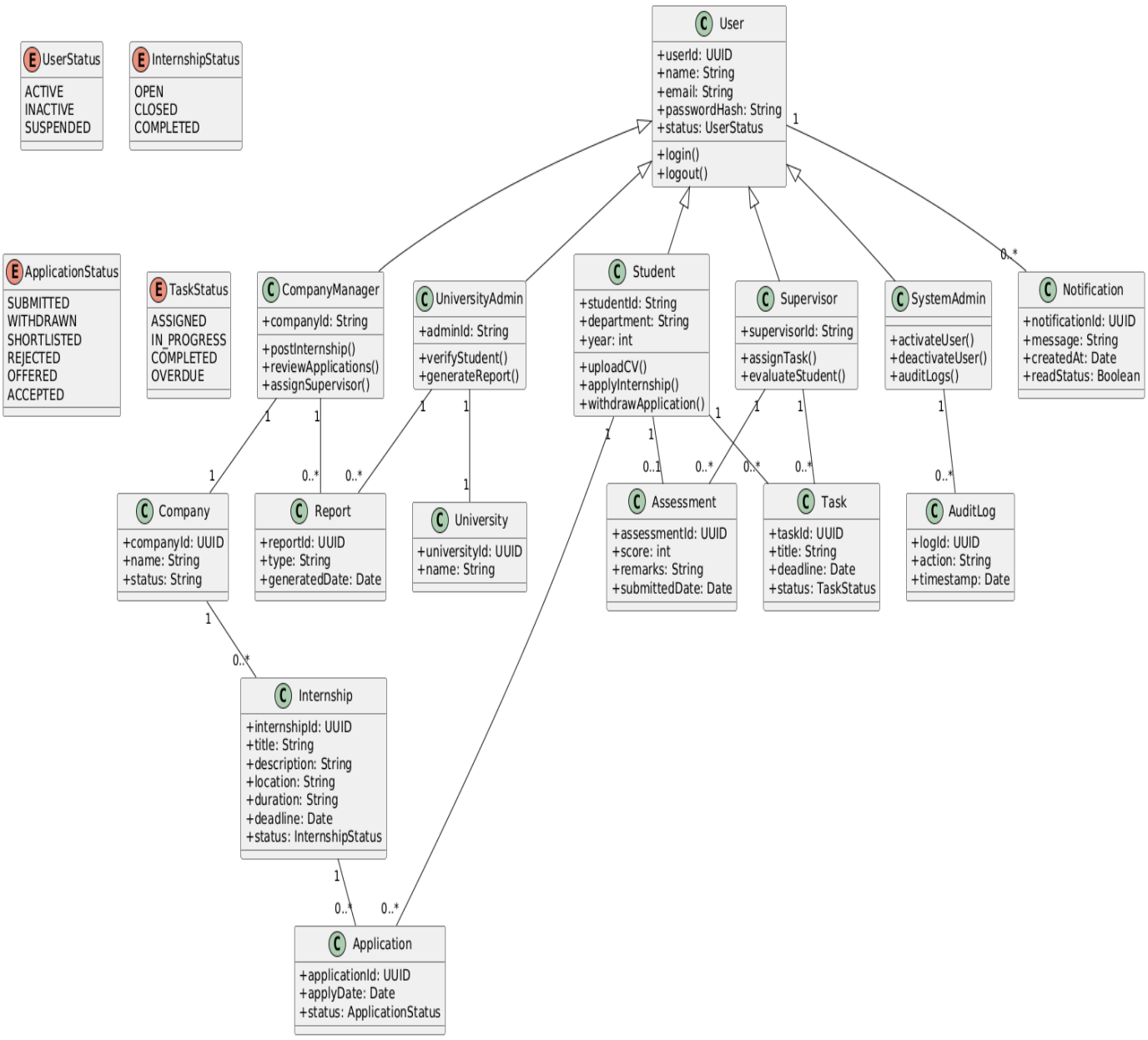
Report

Object	Attribute	Data Type	Description	Constraints
Report	_id	ObjectId	Report identifier	Primary Key

	type	String	Report category	ENUM: university, company, audit
	generatedBy	ObjectId	Report creator	FK → User._id
	generatedAt	Date	Report creation time	Required

2.3.6.3.2 Class Modeling

Class modeling represents the relationships among the identified objects using a class diagram. It illustrates classes, their attributes, methods, and associations such as inheritance, aggregation, and composition. The class diagram provides a blueprint for system design and supports database design and implementation.



2.3.6.4 Dynamic Modeling – InternConnect System

Dynamic modeling describes how the InternConnect system behaves over time by capturing interactions between system components, users, and data during core processes such as internship posting, application submission, verification, shortlisting, and tracking. Unlike static

models that define the structure, dynamic models focus on behavior, control flow, and state changes during real operational scenarios.

In the InternConnect project, dynamic modeling illustrates how students, universities, and companies interact with the system, and how the system responds to those interactions. This modeling highlights decision points, message flows between components, and lifecycle transitions of key objects such as internships, applications, and profiles.

Dynamic modeling is documented using:

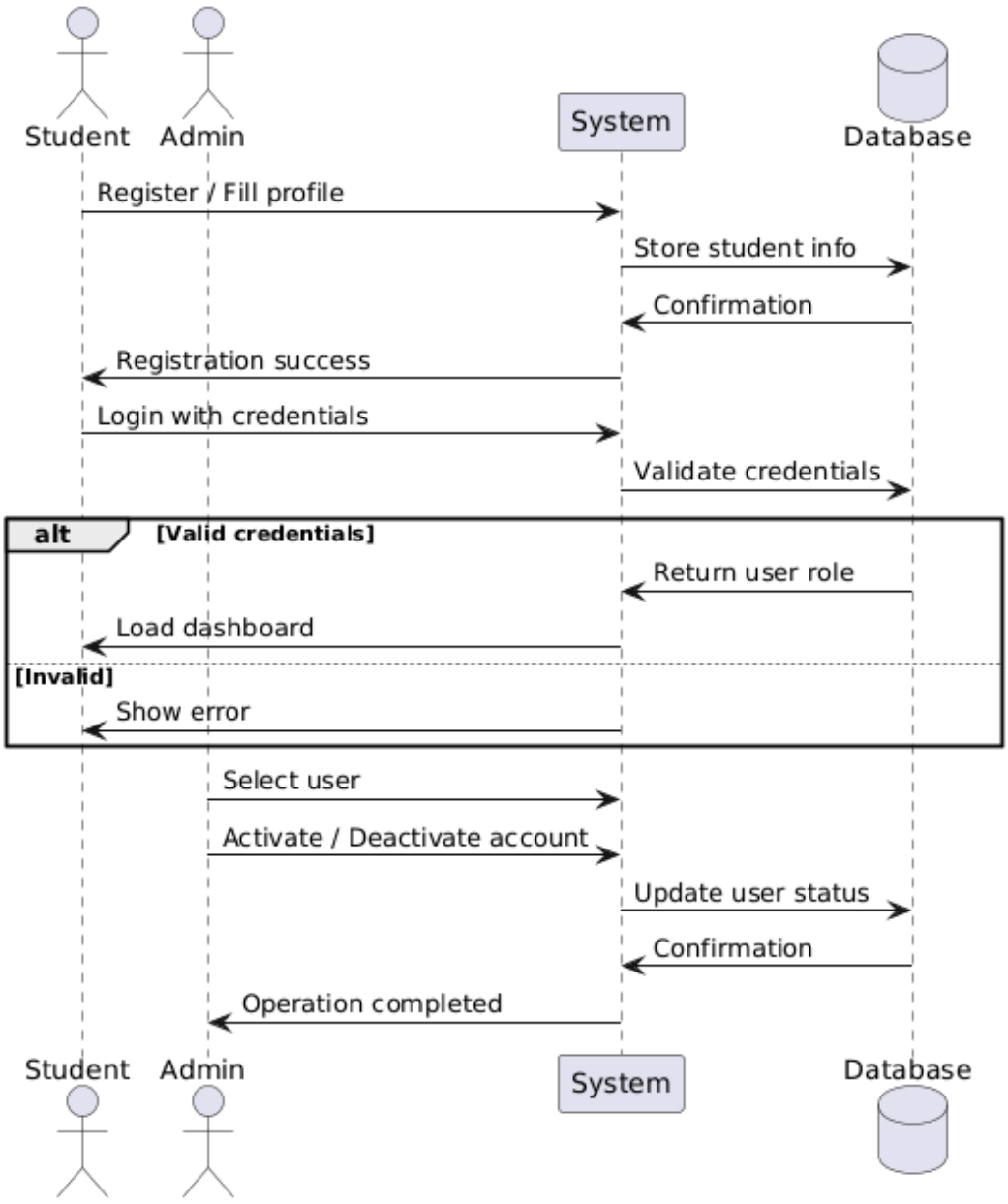
1. **Sequence Diagrams** - Show chronological message exchange between actors and system modules.
2. **Activity Diagrams** - Depict the flow of actions and decision points in a process.
3. **State Diagrams** - Represent the lifecycle and state transitions of key objects like applications and internships.

These models help identify responsibilities of each component, validate system behavior, and ensure processes execute in the correct sequence.

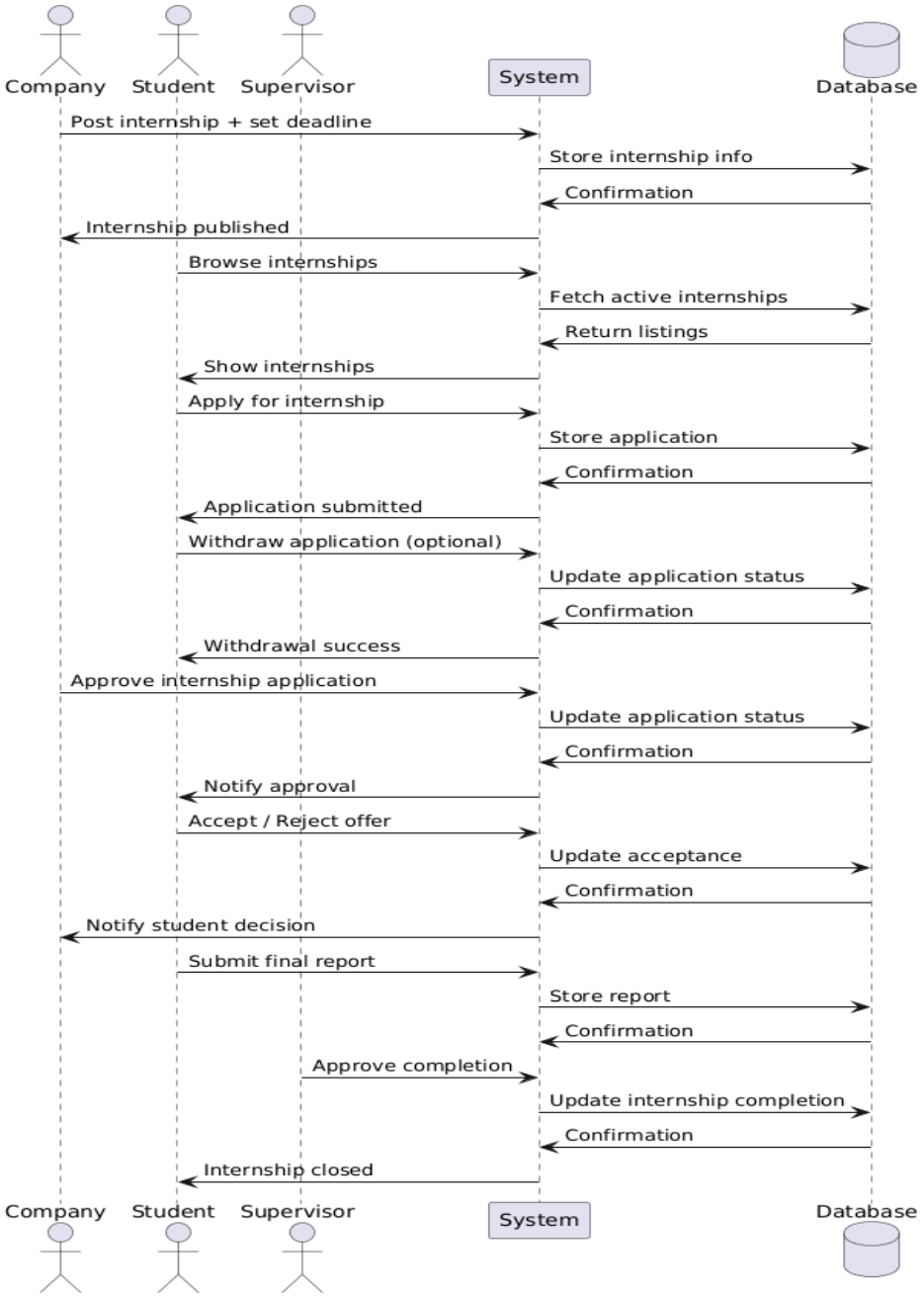
2.3.6.4.1 Sequence Diagrams

A sequence diagram is a Unified Modeling Language (UML) interaction diagram that represents the logic of how objects within a system interact, used in business analysis and software engineering at various system levels.

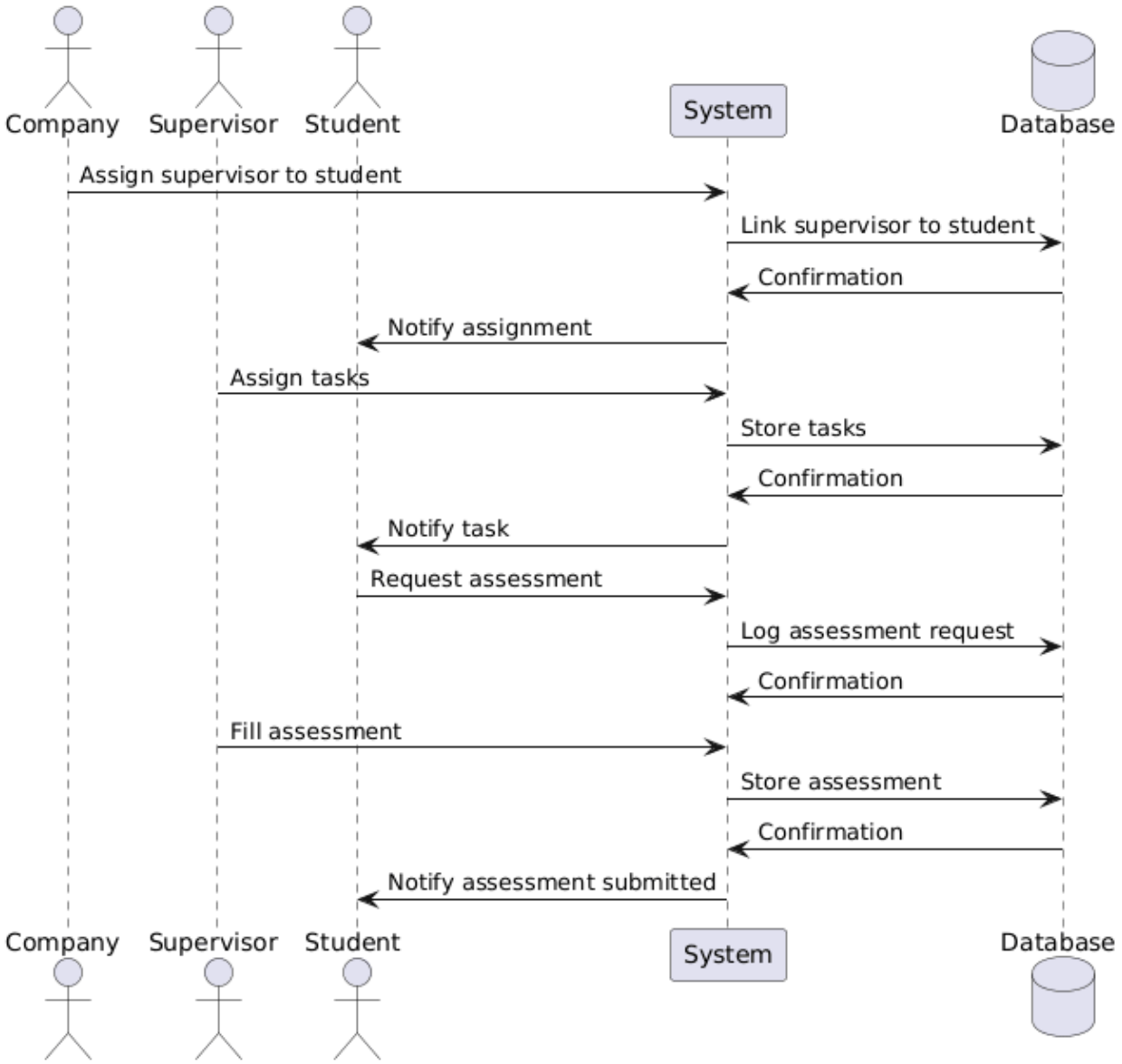
1. **User Management (Login, Registration, Activate/Deactivate)**



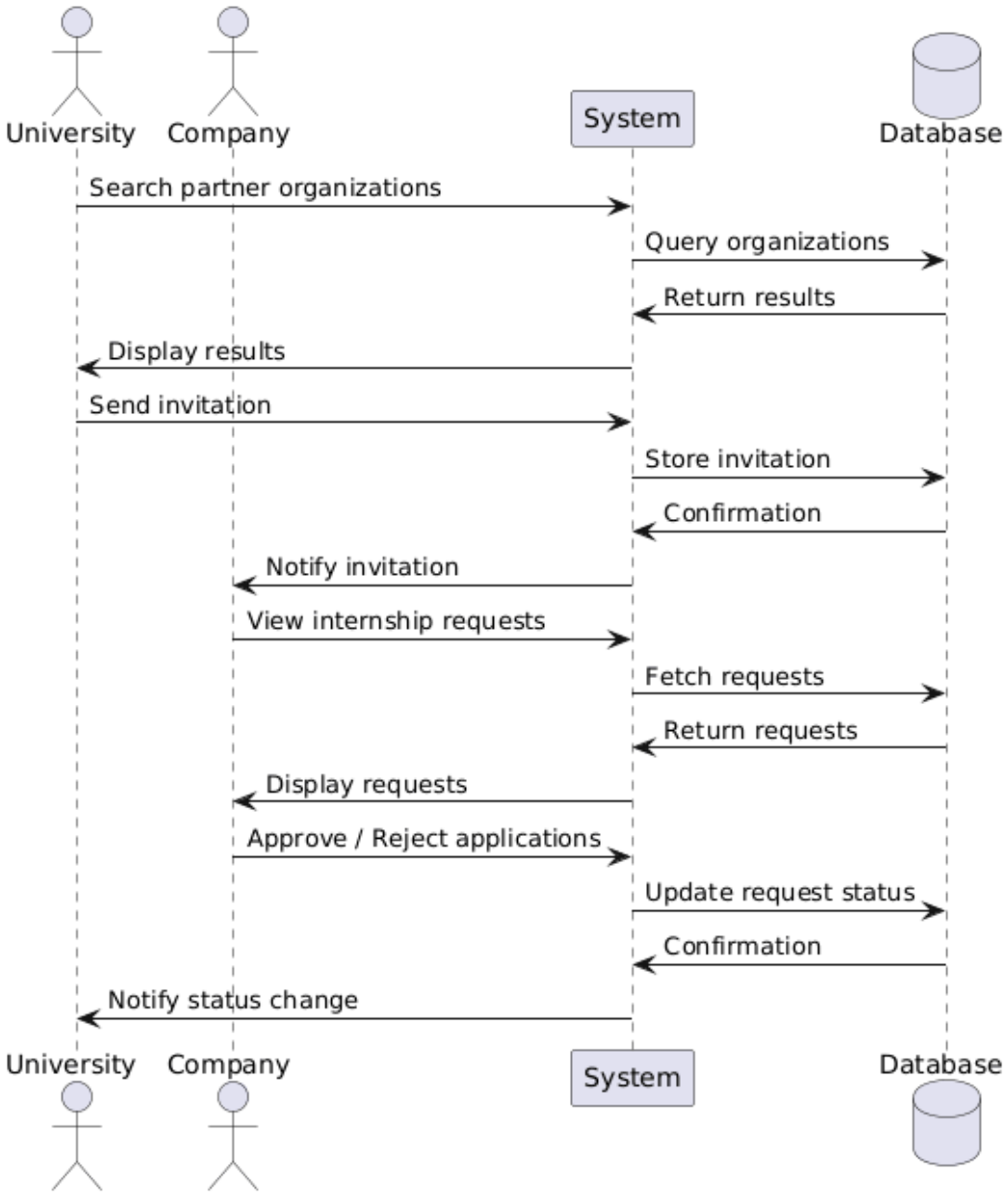
2. Internship Lifecycle (Post, Apply, Withdraw, Accept/Reject, Complete)



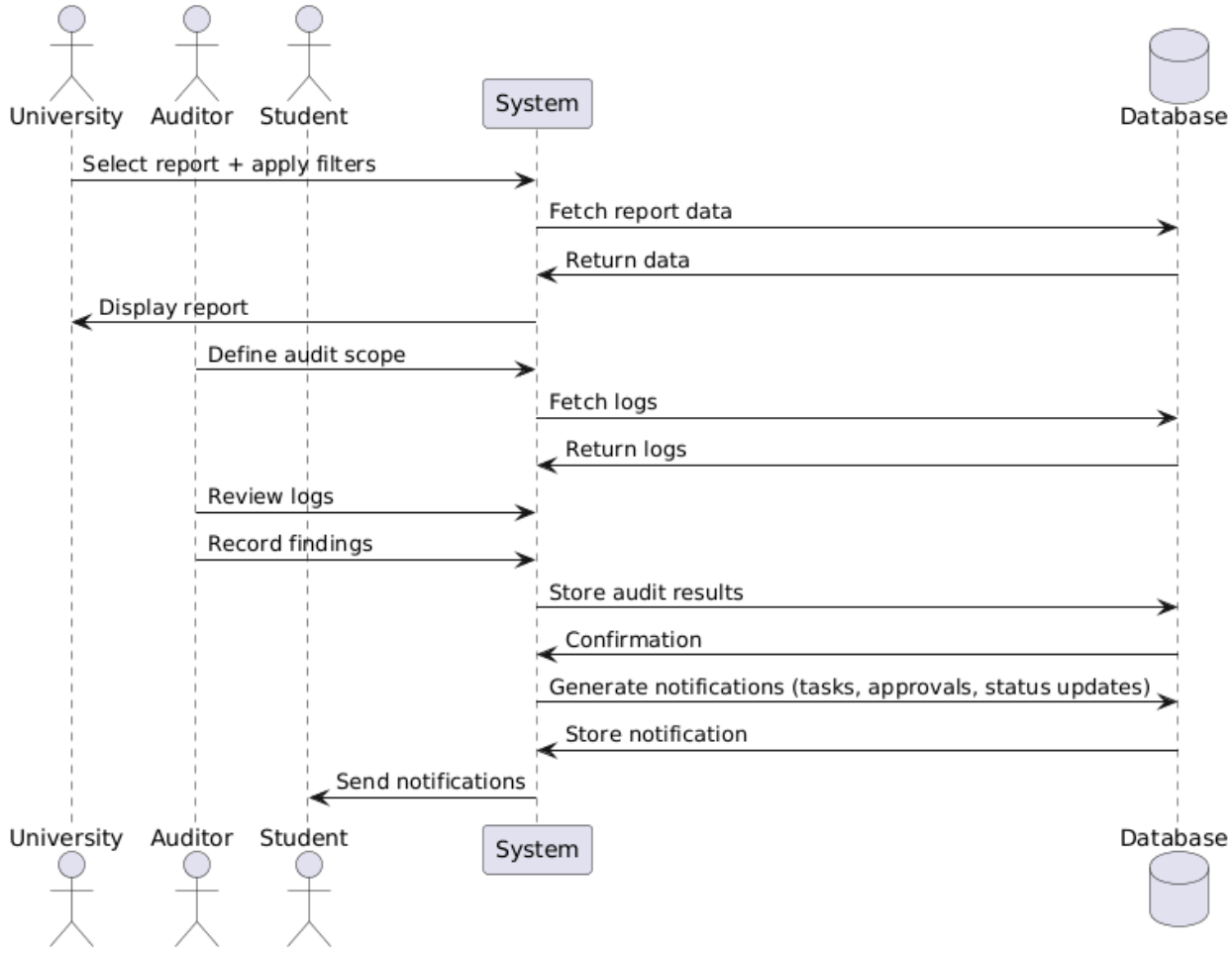
3. Internship Management (Assign Supervisor, Tasks, Assessment)



4. University & Company Collaboration (Search, Invite, Approve/Reject Requests)



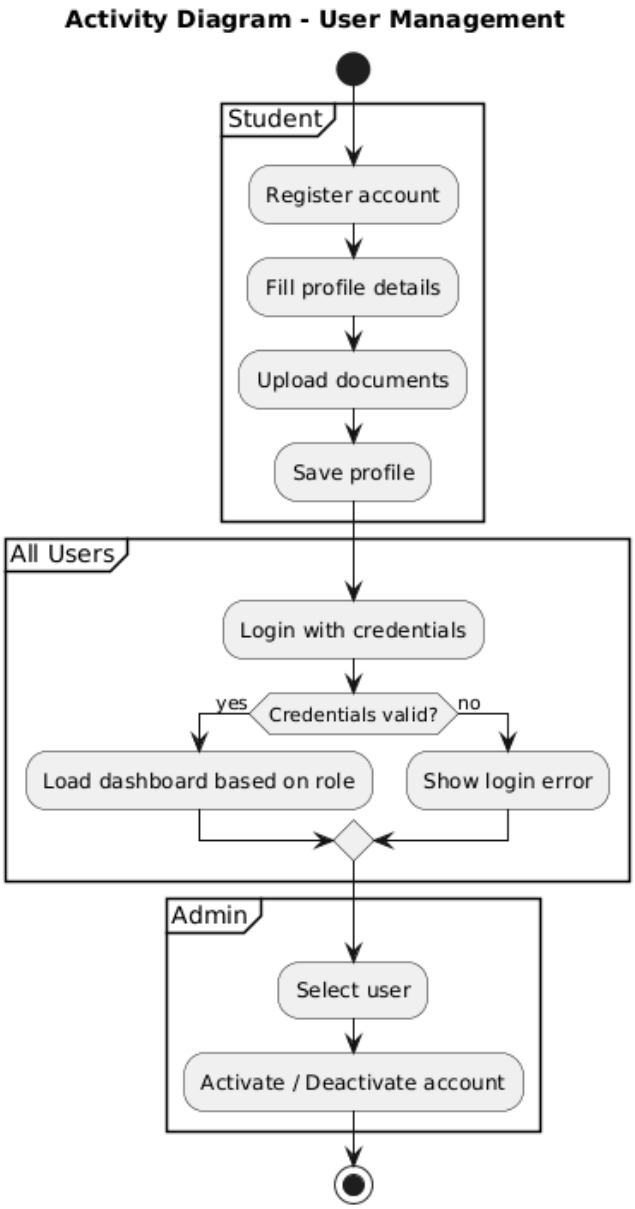
5. Reporting & Notifications (Reports, Audit, Notifications)



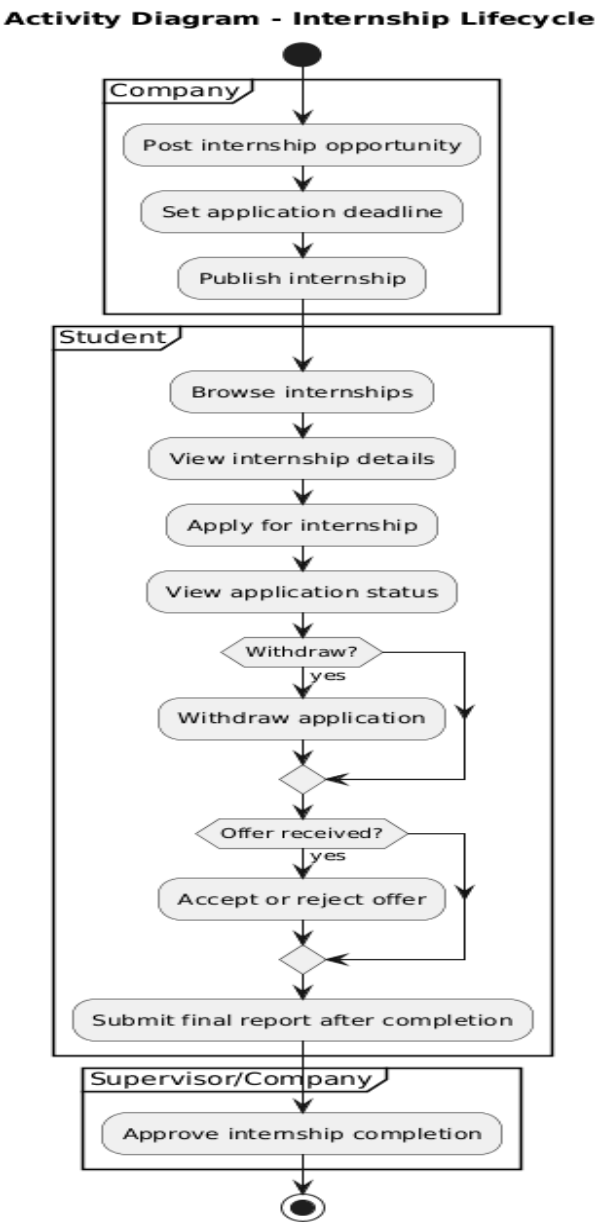
2.3.6.4.2 Activity Diagrams (Process Flow Diagrams) – InternConnect

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.

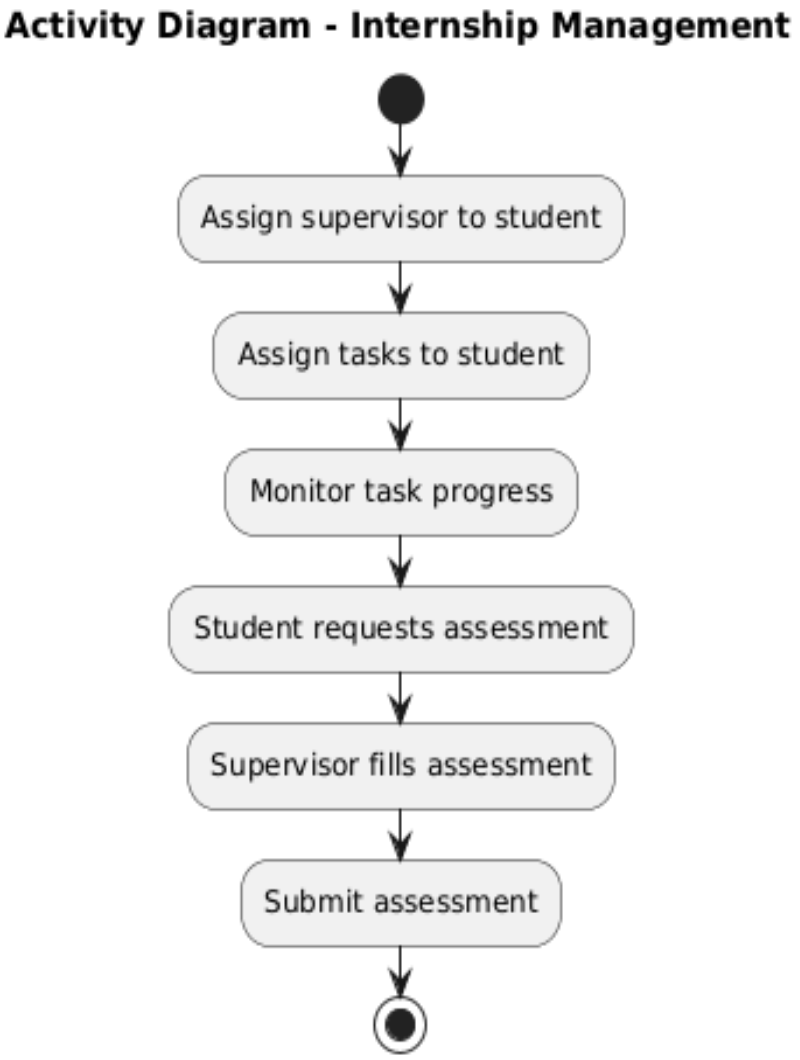
1. **User Management**



2. Internship lifecycle

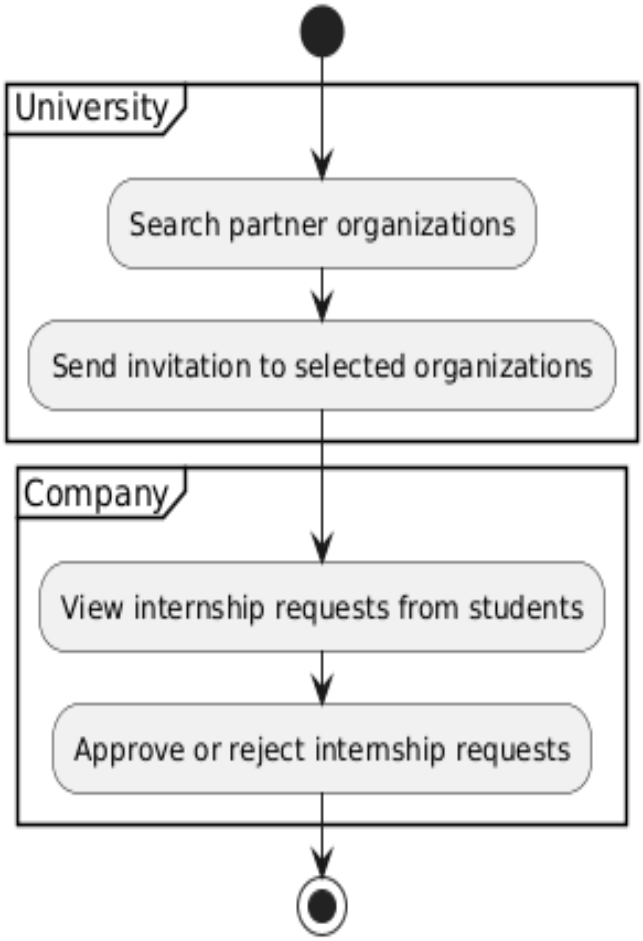


3. Internship management



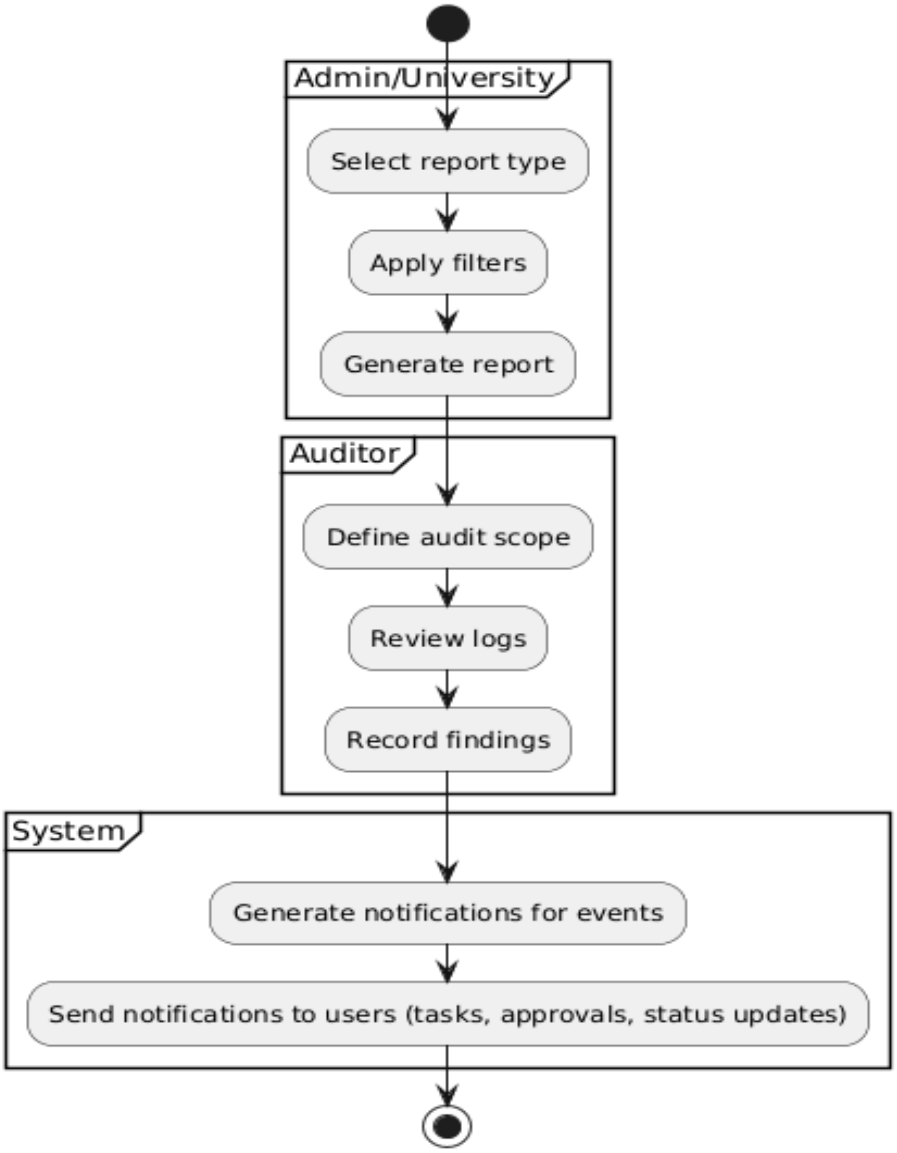
4. University and company collaboration

Activity Diagram - University & Company Collaboration



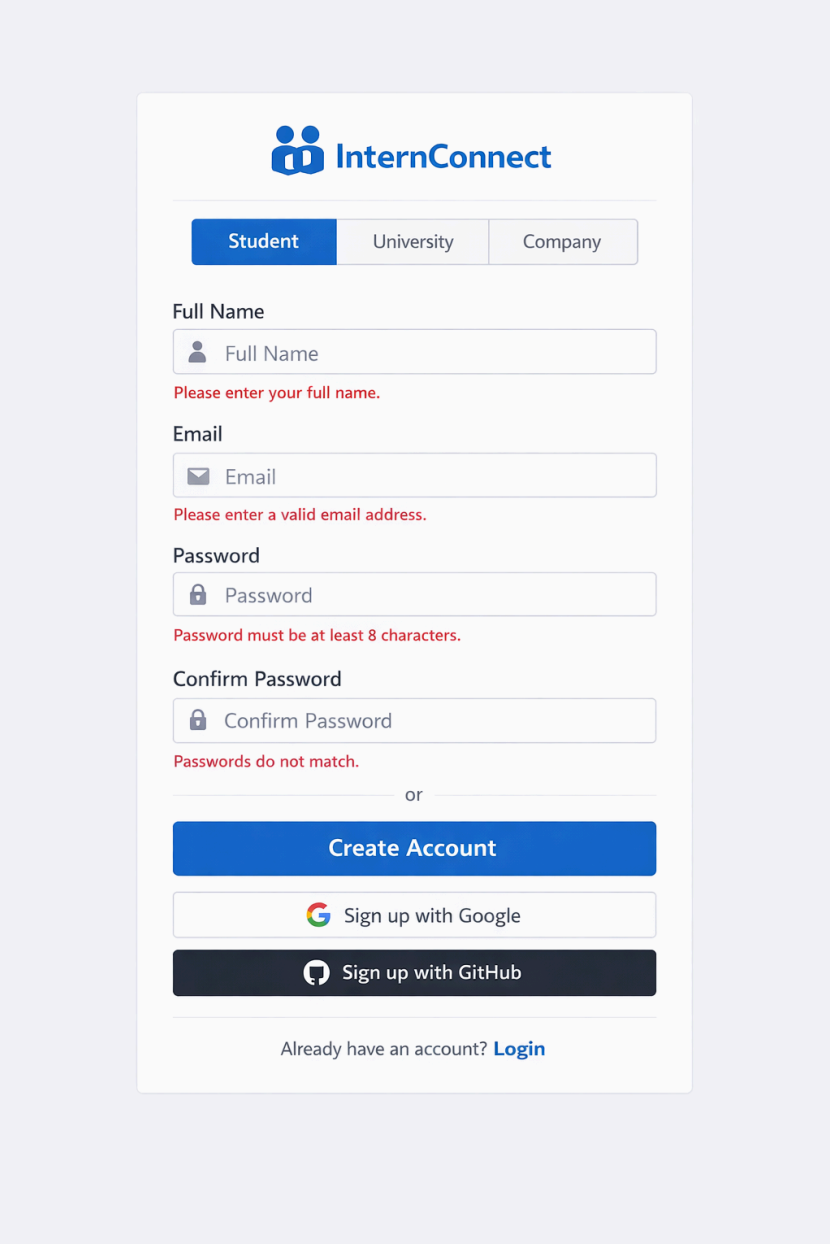
5. Reporting and compliance

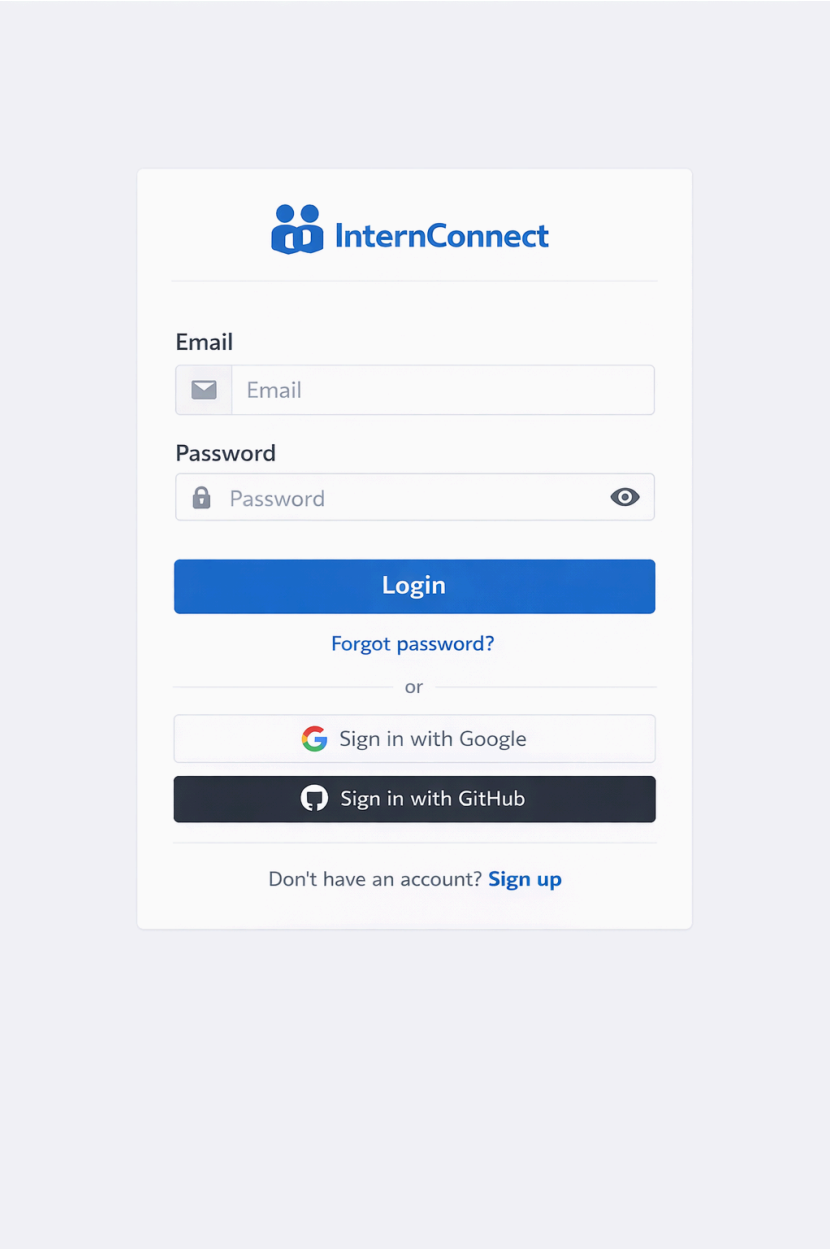
Activity Diagram - Reporting & Compliance

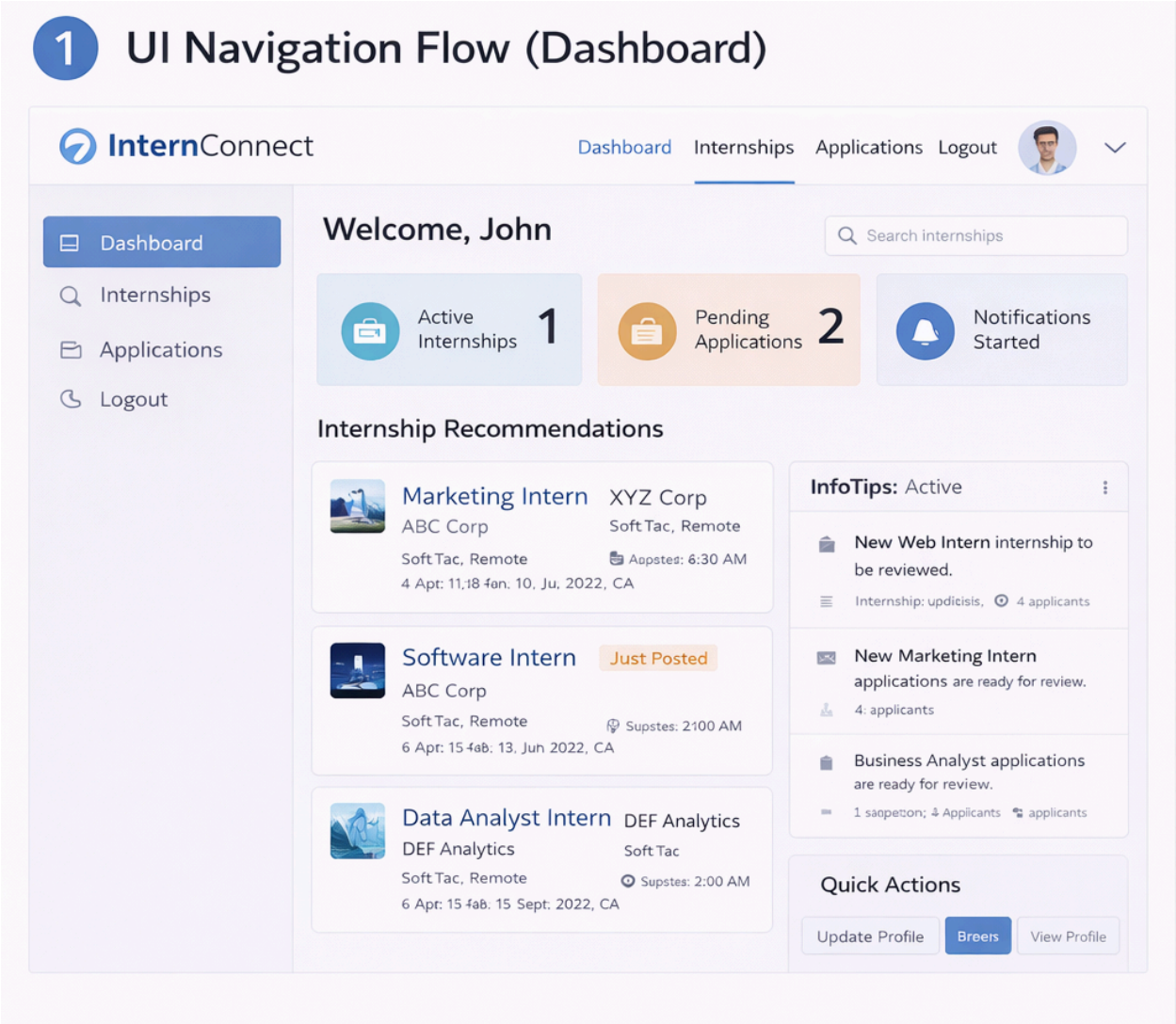


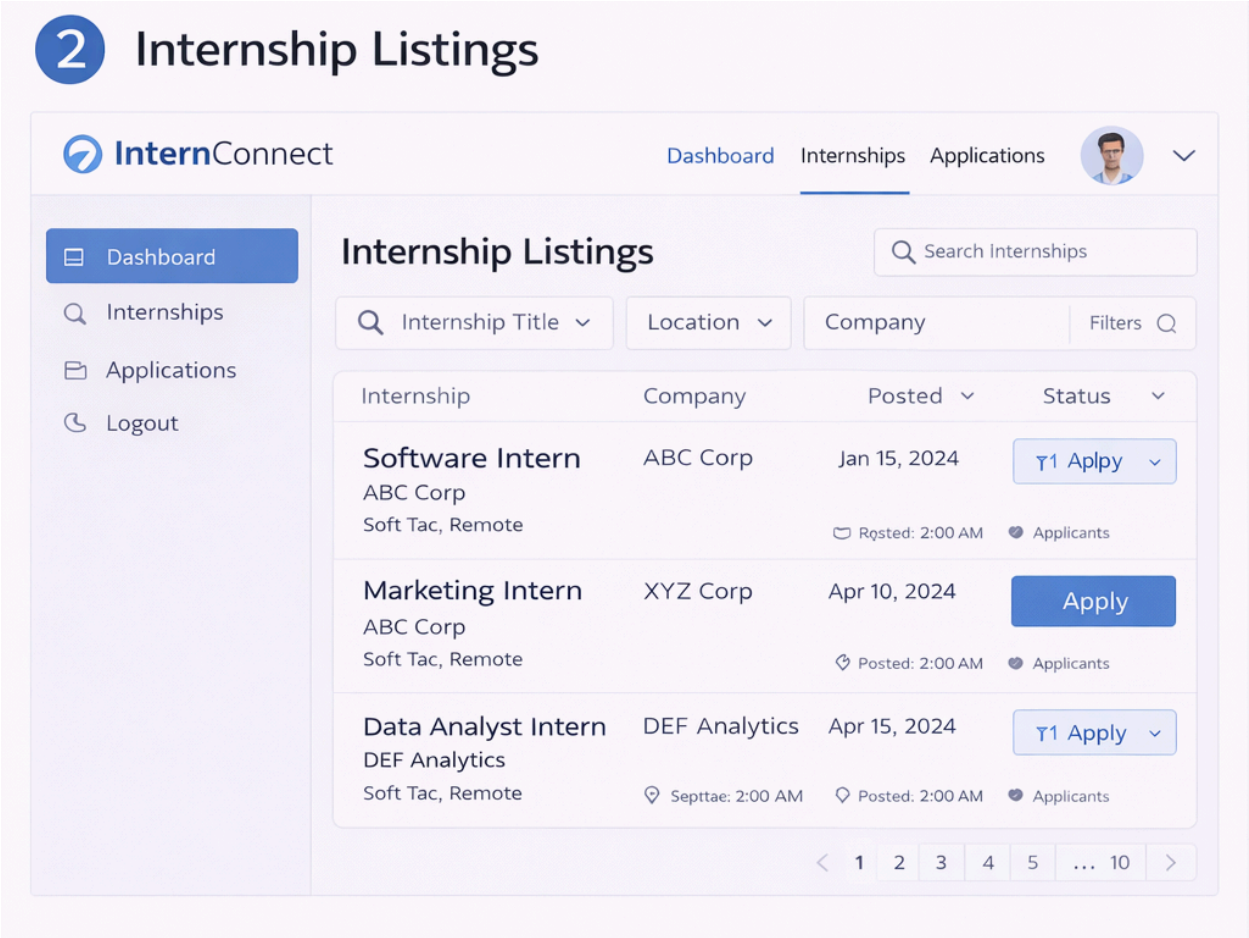
2.3.6.5 User Interface - InternConnect

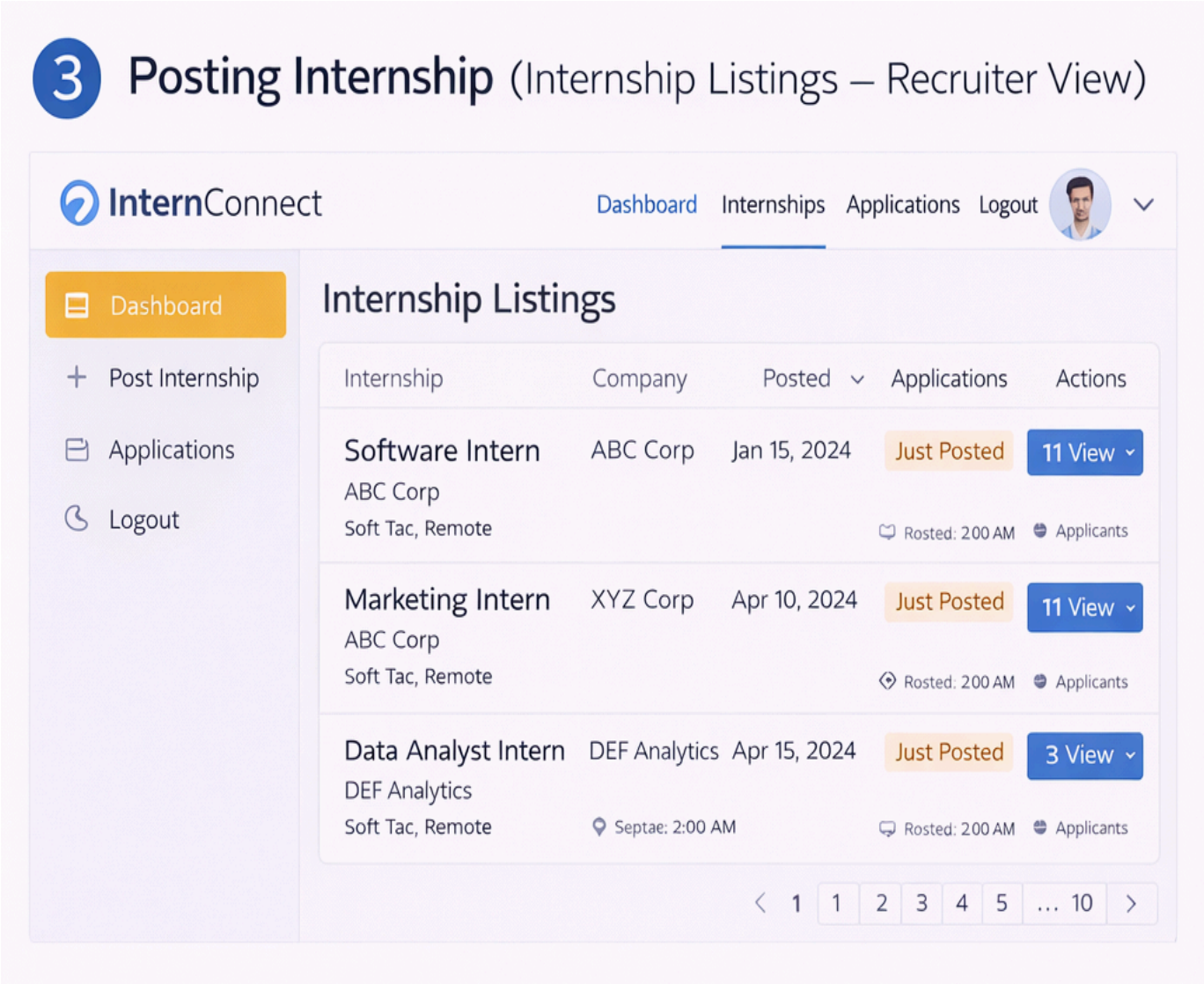
This section describes the user interface (UI) of InternConnect, illustrating how students, university administrators, company recruiters, and system admins navigate the system. The design emphasizes usability, clarity, and task efficiency. The UI supports role-based access, ensuring that each type of user sees only the relevant menus, dashboards, and forms.



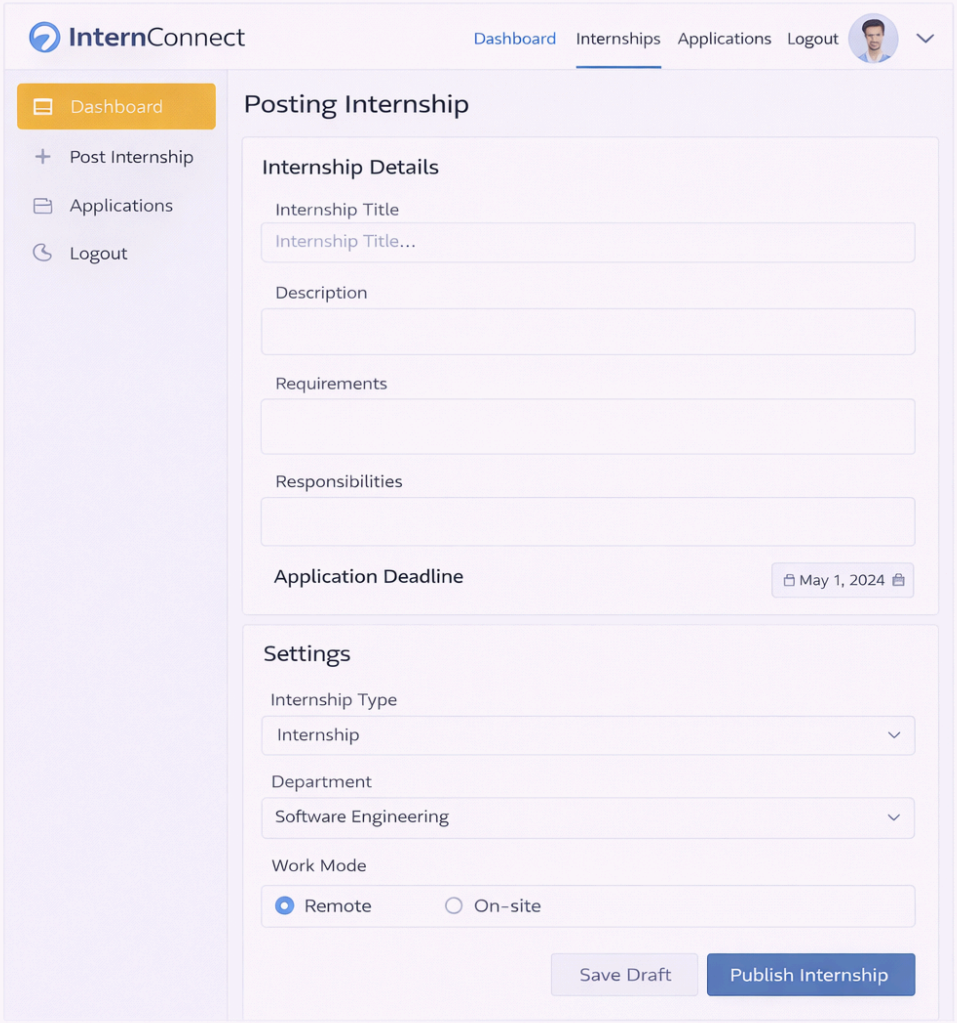

InternConnect
Email
Email
Password
Password
Login
Forgot password?
or
Sign in with Google
Sign in with GitHub
Don't have an account? Sign up


1
UI Navigation Flow (Dashboard)
InternConnect
Dashboard
Internships
Applications
Logout
Dashboard
Internships
Applications
Logout
Welcome, John
Search internships
Active Internships 1
Pending Applications 2
Notifications Started
Internship Recommendations
Marketing Intern
XYZ Corp
ABC Corp
Soft Tac, Remote
Soft Tac, Remote
Aopstes: 6:30 AM
4 Apt: 11,18 fan. 10, Ju, 2022, CA
Software Intern
Just Posted
ABC Corp
Soft Tac, Remote
Supstes: 2100 AM
6 Apt: 15 fab: 13, Jun 2022, CA
Data Analyst Intern
DEF Analytics
DEF Analytics
Soft Tac
Soft Tac, Remote
Supstes: 2:00 AM
6 Apt: 15 fab: 15 Sept: 2022, CA
InfoTips: Active
New Web Intern internship to be reviewed.
Internship: updicisis, 4 applicants
New Marketing Intern applications are ready for review.
4: applicants
Business Analyst applications are ready for review.
1 saopetion; 4 Applicants applicants
Quick Actions
Update Profile
Broers
View Profile


2 Internship Listings
InternConnect
Dashboard
Internships
Applications
Dashboard
Internships
Applications
Logout
Internship Listings
Search Internships
Internship Title
Location
Company
Filters
Internship
Company
Posted
Status
Software Intern
ABC Corp
Soft Tac, Remote
ABC Corp
Jan 15, 2024
Rqsted: 2:00 AM
Applicants
Marketing Intern
ABC Corp
Soft Tac, Remote
XYZ Corp
Apr 10, 2024
Apply
Posted: 2:00 AM
Applicants
Data Analyst Intern
DEF Analytics
Soft Tac, Remote
DEF Analytics
Apr 15, 2024
Apply
Septtae: 2:00 AM
Posted: 2:00 AM
Applicants
1 2 3 4 5 ... 10


3 Posting Internship (Internship Listings – Recruiter View)
InternConnect
Dashboard
Internships
Applications
Logout
Dashboard
Post Internship
Applications
Logout
Internship Listings
Internship
Company
Posted
Applications
Actions
Software Intern
ABC Corp
Soft Tac, Remote
ABC Corp
Jan 15, 2024
Just Posted
11 View
Rosted: 200 AM
Applicants
Marketing Intern
ABC Corp
Soft Tac, Remote
XYZ Corp
Apr 10, 2024
Just Posted
11 View
Rosted: 200 AM
Applicants
Data Analyst Intern
DEF Analytics
Soft Tac, Remote
DEF Analytics
Apr 15, 2024
Just Posted
3 View
Septae: 2:00 AM
Rosted: 200 AM
Applicants
1 1 2 3 4 5 ... 10

4 Posting Internship (Create / Edit Internship Form)



CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

System design defines how the InternConnect platform is structured to meet the functional and non-functional requirements identified during the analysis phase. It provides a comprehensive view of the system from a functional and architectural perspective, detailing how system components, modules, interfaces, and data structures interact to support internship management processes. This phase translates user and organizational requirements into a concrete technical blueprint, covering software architecture, subsystem decomposition, database design, hardware and software mapping, and access control mechanisms. By clearly defining these elements, the system design ensures that InternConnect operates as a cohesive, scalable, and maintainable platform that supports students, universities, and partner companies effectively.

Effective system design is essential for ensuring seamless interaction between system components and reliable system behavior under real-world conditions. For InternConnect, the design emphasizes modularity, well-defined interfaces, and structured data flow to reduce integration complexity and improve system robustness. In addition to functional behavior, the design addresses non-functional requirements such as performance, reliability, and security. Role-based access control and secure data handling mechanisms are incorporated to protect sensitive academic and organizational data. By addressing both functional and quality

requirements at this stage, the system design establishes a strong foundation for implementation and future system evolution without requiring major architectural changes.

3.2 Design Goals

Design goals outline the qualities developers should optimize to meet non-functional requirements. Derived from the system requirements, the goals define InternConnect's optimized qualities and operational characteristics. The major quality perspectives guiding these objectives are functionality, performance, dependability, reliability, scalability, and maintainability. This framework ensures that the system not only delivers the intended internship management features but also operates efficiently, reliably, and with ease of maintenance, aligning with overarching non-functional criteria.

3.2.1 Functionality goals

InternConnect aims to provide a comprehensive and integrated system that delivers:

1. Complete internship lifecycle management (posting → application → selection → completion) .
2. User-centric feature set with role-specific interfaces (students, companies, universities, admins)
3. Robust verification and validation of academic eligibility and company credentials .
4. Rich analytics and reporting for data-driven decision-making .
5. Comprehensive audit trails and compliance support.

3.2.1.1 Performance Goals

The design aims to achieve peak performance characterized by swift response times, seamless throughput, and efficient resource utilization to meet user expectations and support operational

scale. InternConnect demands high performance to support interactions between distributed students, companies, and universities.

3.2.1.1.1 Response Time

- Page Load Times: < 2 seconds for all UI pages.
- API Response Times: < 500ms median response time for standard queries.
- Search Operations: < 1 second (95th percentile) for internship searches with filters.
- Report Generation: < 30 seconds for standard reports; < 2 minutes for complex analytics.
- File Operations: < 5 seconds for 10MB CV upload/download.

3.2.1.2 Throughput and Concurrency

- Support 2,000 -5,000 concurrent active users during peak periods.
- Handle 1,000 - 3000 requests per second at maximum load.
- Process 500 - 2,500 database transactions per second.
- Maintain performance during internship application deadlines and system-wide announcements.

3.2.1.3 Dependability Goals

The System design aims to ensure the system is reliable, available, secure, and resilient to faults while maintaining data integrity and user trust. Dependency consists of four critical dimensions for InternConnect.

1. Reliability Goals

Reliability is a critical quality goal for the InternConnect system, ensuring dependable and consistent operation for all users. The system targets an uptime of 99.5% during regular academic calendar periods and 99.9% during critical internship application deadlines.

2. Availability Goals

Availability is a key quality objective of the system, ensuring continuous access for users at all times. The system is designed to provide 24/7 accessibility, with scheduled maintenance limited to two hours per week during off-peak periods to minimize user disruption. To support uninterrupted service delivery, the database infrastructure targets a high availability level of 99.99%, ensuring reliable data access and supporting critical system operations.

3. Security Goals

The design is aimed at preventing unauthorized data access, mitigating risks associated with the OWASP Top 10 vulnerabilities, and ensuring compliance with applicable data protection regulations.

4. Error Handling and Resilience Goal

The system is designed to support graceful degradation while providing meaningful feedback to users in the event of failures, with a target production error rate of less than one percent. These measures ensure that system disruptions are managed effectively, minimizing data loss and maintaining continuity of service under adverse conditions.

3.2.2 End-User Experience Goals

This is aimed at delivering an intuitive, efficient, and transparent system that fosters trust and satisfaction among all user types. The system emphasizes a user-friendly interface that accommodates users with varying levels of technical proficiency through responsive, mobile-first design, clear navigation structures, and consistent visual elements.

Efficiency is prioritized to enable users to complete internship-related tasks quickly and with minimal effort. Optimized workflows, search and filtering capabilities, one-click actions, and support for keyboard shortcuts reduce task completion time and improve productivity.

Transparency is achieved by providing clear visibility into application processes, decisions, and timelines. Real-time status tracking, timeline displays, decision feedback, audit trails, and transparent data usage disclosures ensure users remain informed and confident in system operations. Finally, convenience is enhanced through a centralized, web-based system that integrates communication channels, document management, calendar scheduling, and full mobile accessibility, enabling users to access and manage internship opportunities seamlessly from any location.

3.3 Proposed Software Architecture

3.3.1 Overview

The Interconnect System adopts a three-tier distributed architecture with separation of concerns across presentation, business logic, and data layers. Through a web browser that is installed on their computers, phone, tablets, users will interact with the system. In the server-client-database environment, the graphical user interface, which is available through the internet, makes it easier for users to communicate with the database system. The system integrates students, universities, and companies into a unified digital ecosystem using a three-tier distributed architecture implemented with the MERN (MongoDB, Express, React, Node.js) technology stack.

The design follows separation of concerns principles, with clear boundaries between the presentation tier (React frontend), business logic tier (Node.js/Express backend), and data persistence tier (MongoDB database). This architectural approach ensures modularity, scalability, maintainability, and supports future enhancements. The Interconnect System is designed using a classic three-tier architecture pattern, which separates the application into three logical layers as depicted below.

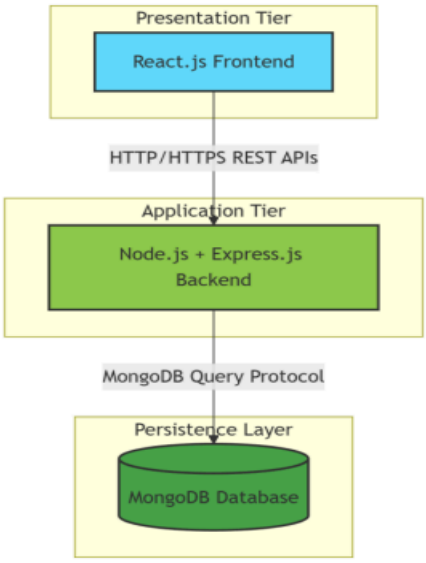


Figure 3.1: 3-Tier architectural diagram

3.3.2 Tier Responsibilities

1. Presentation Tier

The front-end is to be developed using React, serving as the primary interface for stakeholders. This tier is responsible for rendering user interface components and layouts while managing client-side state and navigation. Key responsibilities include handling user input validation, managing form submissions, and overseeing authentication token storage. By handling route management and data display, this tier ensures a seamless and responsive user experience.

2. Business Logic Tier

The intermediary layer is powered by Node.js and Express, acting as the core engine for system operations. This tier manages request routing and API endpoints while strictly enforcing authentication, authorization, and business rules. It orchestrates complex workflows, such as the internship lifecycle, and handles integration with external services. Furthermore, this layer is responsible for data transformation between the client and the database, error management, and the generation of comprehensive audit trails and logs.

3. Data Tier

The foundation of the architecture is the MongoDB data tier, which provides persistent storage for all system data. It ensures data integrity through schema validation and optimizes performance via indexing and aggregation pipelines for complex queries. This tier supports critical operations through transactional integrity and maintains system resilience through dedicated backup and recovery capabilities.

3.3.3 Architectural Principles followed

1. Separation of Concerns

The principle of Separation of Concerns ensures that each tier within the system maintains distinct and isolated responsibilities. This architectural approach partitions functions into specific layers to avoid overlapping duties and maintain structural clarity. The system is structured into three primary layers. The Presentation Tier is exclusively responsible for user interface rendering and managing user interactions. The Business Logic Tier serves as the core of the system, containing all application logic, workflows, and integrations. Finally, the Data Tier is dedicated to managing data storage, retrieval, and the maintenance of data consistency[10].

Organizing the system through these distinct boundaries enables several operational advantages. It allows for the independent development and testing of each tier, which streamlines the production lifecycle. Furthermore, this separation supports flexible technology choices within constraints and makes maintenance and troubleshooting easier. Ultimately, the structure establishes clear interfaces between all components to ensure orderly communication across the system.

1. Stateless Services

The business logic tier is designed as a collection of stateless services, where each request is processed independently. This architectural choice ensures that no session data is stored within

the application memory. Instead, session state is managed through the use of secure tokens, such as JSON Web Tokens (JWT).

By removing the reliance on local memory for session storage, the system enables horizontal scaling across multiple instances. This design allows the application to handle increased loads by distributing incoming requests across various service nodes without losing continuity.

2. API-First Design principle

The system adheres to an API-First Design principle, where communication between tiers occurs exclusively via well-defined APIs. This approach ensures that all interactions are structured and predictable across the entire architecture.

Interactions between the frontend and backend are handled through a RESTful API, while communication between the backend and the database utilizes MongoDB driver protocols. To maintain consistency and stability, the system relies on version-controlled API contracts and follows a documentation-driven development process. This methodology ensures that interfaces are clearly defined and standardized before implementation.

3. Security at Multiple Levels

The system implements security across multiple layers to ensure comprehensive protection of information and operations. At the network level, all data is secured using HTTPS and TLS encryption to protect information while in transit. Within the application layer, security is maintained through JWT authentication and role-based authorization, ensuring that only verified users can access specific functions.

Data security is further reinforced by encrypting sensitive fields and utilizing hashed passwords to prevent unauthorized access to user credentials. Finally, the audit layer provides an additional

level of oversight by maintaining comprehensive logging of all system activities, which ensures accountability and supports security monitoring.

3.3.4 System Decomposition

The system decomposition represents the strategic division of the InternConnect system into cohesive, manageable subsystems that align with the three-tier distributed architecture and MERN technology stack. This structured breakdown transforms complex functional requirements into discrete, interdependent modules with clearly defined responsibilities, interfaces, and deployment boundaries .The InternConnect System is decomposed into the following interconnected subsystems[11].

3.3.4.1. Authentication and Authorization Subsystem

The Authentication and Authorization module is a critical subsystem within the InternConnect architecture, responsible for managing user identity verification and enforcing granular permission levels across all system operations. This subsystem ensures that system resources are accessed only by verified entities according to their specific roles.

To ensure a secure and scalable identity layer, the subsystem is decomposed into several discrete components:

- JWT Token Manager: Handles the generation, signing, and verification of JSON Web Tokens to facilitate stateless authentication between the React frontend and the Node.js backend.
- Password Encryption Service: Utilizes industry-standard hashing algorithms (such as bcrypt) to secure user credentials at rest.
- Role-Based Access Control (RBAC) Engine: The logic layer that evaluates user roles against requested resources to determine access rights.
- Session Management: An optional integration potentially utilizing Redis to manage active user states and token revocation.

- Two-Factor Authentication (2FA): A planned architectural enhancement to provide an additional layer of security for administrative accounts.

Table 3.1 RBAC Role Hierarchy

Role	Permissions / Actions
Student	- Register and manage account/profile- View and browse internship opportunities- Apply for internships- Withdraw applications- Track application status and internship progress- Request assessment- Provide feedback on internships- Receive notifications
Company / Recruiter / Manager	- Register and manage company profile- Post, edit, or withdraw internship opportunities- Review internship applications- Approve / reject internship applications- Shortlist candidates- Assign supervisors to interns- Track student progress- Send offers to students
University Admin / Coordinator	- Register and verify students- Search and invite partner companies- Monitor student internship placements- Track internship progress- Generate analytical reports on student placements and internships

Supervisor	- Assign tasks to students- Monitor task progress- Fill and submit assessments- Provide feedback on student performance
System Admin	- Activate / deactivate users- Manage all accounts and permissions- Verify external entities (companies, universities)- Audit and compliance logging- Oversee system-wide notifications and reports

3.3.4.2 Student Management Subsystem

The Student Management Subsystem manages the end-to-end lifecycle of student data. Its primary responsibilities are maintaining accurate student profiles, facilitating academic verification, and providing real-time tracking of internship progress.

The subsystem acts as an intermediary between the Presentation Tier where students input their data and the Data Tier, where MongoDB stores records with strict schema validation. By isolating services, document handling (which may involve external storage such as AWS S3) does not interfere with high-frequency profile updates or registration tasks.

This subsystem is divided into five specialized services to ensure modularity, data integrity, and maintainability:

- **Student Registration Service:** Handles initial onboarding, capturing essential student identification and account setup details.
- **Profile Management Service:** Manages the persistence and updates of student-specific data, including skills, resumes, and portfolios.

- **Academic Verification Service:** Coordinates workflow between students and University Admins to confirm enrollment status and academic eligibility for internships.
- **Document Management Service:** Handles secure upload, storage, and retrieval of critical files such as transcripts, recommendation letters, and internship agreements.
- **Internship Tracking Service:** Provides continuous monitoring for active placements, allowing students and administrators to log milestones and progress.

3.3.4.3 Company Management Subsystem

The Company Management Subsystem is the primary interface for industry partners. It manages the full lifecycle of employers from onboarding and verification to posting internships and evaluating candidates.

The subsystem is composed of **six specialized services**:

- **Company Registration Service:** Facilitates account creation and collects organizational metadata.
- **Verification Service:** Validates the legitimacy of companies via submitted legal and corporate documentation.
- **Internship Posting Service:** Enables creation, definition, and publication of internship listings, including eligibility criteria.
- **Application Review Service:** Centralized access for companies to review student applications, resumes, and academic credentials.
- **Candidate Shortlisting Service:** Allows recruiters to filter, categorize, and move applicants through evaluation stages.
- **Offer Management Service:** Handles internship offer issuance, tracks student acceptance or decline, and closes the recruitment process.

3.3.4.4 University Management Subsystem

The University Management Subsystem manages student verification, ensures internship quality, and produces institutional reports. It integrates the following components:

- **University Registration and Student Verification Services**
- **Compliance Monitoring Service**
- **Reporting Service**
- **Coordinator Management Service**

The subsystem operates through **three primary workflows**:

1. **Student Verification:** Validates a student's academic status, approves or rejects eligibility, and notifies the student.
2. **Internship Monitoring:** Tracks student placements, monitors ongoing progress, verifies completion, and records final outcomes.
3. **Reporting:** Collects relevant data, aggregates statistics, and generates comprehensive reports for institutional use.

3.3.4.5 Application Management Subsystem

The Application Management Subsystem manages the entire lifecycle of internship applications from submission to completion. It consists of five core services:

- **Application Submission Service:** Handles intake of new internship applications.
- **Eligibility Verification Service:** Assesses whether applicants meet the necessary criteria.
- **Status Tracking Service:** Monitors application progression through all stages.
- **Notification Service:** Sends updates to students, supervisors, and company managers.
- **Feedback Management Service:** Collects and organizes evaluative input throughout the process.

3.3.4.6 Reporting and Analytics Subsystem

The Reporting and Analytics Subsystem generates actionable insights on internship placements, application trends, and system performance. It transforms raw data into meaningful intelligence via **five primary services**:

- **Data Aggregation and Analytics Computation:** Collects and analyzes system-wide data to identify trends.
- **Report Generation and Visualization:** Structures computed data into readable formats, including charts and tables.
- **Export Service:** Enables users to download reports in PDF, Excel, or CSV formats.

3.3.4.7 Notification Subsystem

The Notification Subsystem delivers timely alerts to system users regarding critical events. It ensures stakeholders remain informed through multi-channel delivery and management services:

- **Delivery Channels:** Email (SMTP/SendGrid), In-App notifications, and optional SMS (Twilio).
- **Management Services:** Notification Template Service standardizes messages; Delivery Tracking Service monitors success.

Notifications are triggered by:

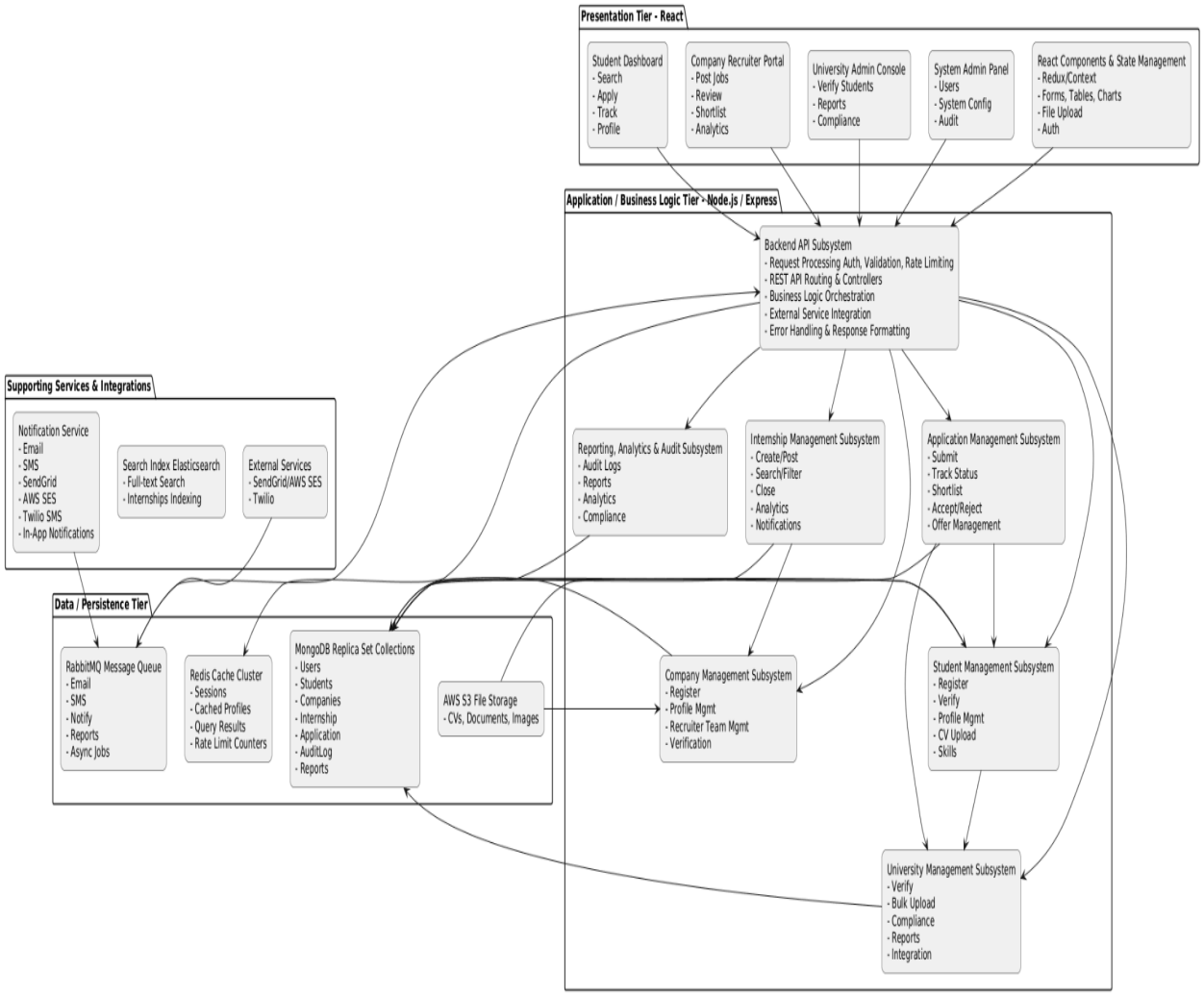
- **Application Lifecycle Updates:** Application status changes, deadlines.
- **Verification Results:** Student verification notifications.
- **Task Assignments:** Supervisor-assigned tasks or feedback.

3.3.4.8 Audit and Compliance Subsystem

The Audit and Compliance Subsystem tracks all system operations, ensures regulatory adherence, and provides data for forensic analysis. It functions as the system oversight layer, utilizing **five specialized services**:

- **Audit Logging and Data Integrity Services:** Record system activity and maintain accurate, unaltered data.
- **Compliance and Access Auditing:** Verify operations meet standards and monitor user permissions.
- **Regulatory Report Generator:** Automates creation of reports required for legal or institutional audits.

Figure 3.2 High-Level Three-Tier Architecture with Subsystems



3.3.5 Database Design and Schema

The InternConnect system's persistent data layer represents the critical foundation upon which all system operations, reporting, analytics, and compliance activities depend. This section provides a comprehensive framework for understanding the data persistence strategy, architectural principles, and implementation decisions that enable the system to reliably store, retrieve, and manage internship-related information across multiple stakeholder types and operational scenarios.

Why MongoDB for InternConnect?

The selection of MongoDB as the primary data store reflects a deliberate architectural decision optimized for the specific requirements of internship management workflows and operational patterns observed during requirements gathering. The following are the reasons why MongoDB is used for the system.

- Document-Oriented Data Model: Internship management involves complex, hierarchical data structures that map naturally to JSON documents. Student profiles encompass multiple arrays (skills, certifications, interests), company records contain nested recruiter information, and internship postings include detailed requirement specifications. MongoDB's native document model eliminates the impedance mismatch that occurs when mapping object-oriented application code to relational schemas, reducing complexity and improving development velocity[12].
- Flexible Schema Evolution: Academic institutions and recruiting practices evolve continuously. New fields may need to be added to student profiles (e.g., work authorization,status,language,proficiency),
  internship requirements may expand (e.g., preferred technologies, soft skills), and business rules may change(e.g.,new verification criteria). MongoDB's schema-less design allows these changes to be deployed without database migration scripts or downtime, supporting agile development and rapid iteration.
- High Read Throughput for Search Operations: Internship discovery is a read-heavy operation where students perform numerous searches with filters and sorting. MongoDB's optimized query engine, combined with strategic indexing (text indexes for full-text search, compound indexes for multi-field filtering), delivers fast search results. The aggregation pipeline enables complex analytics queries to be expressed and optimized efficiently.

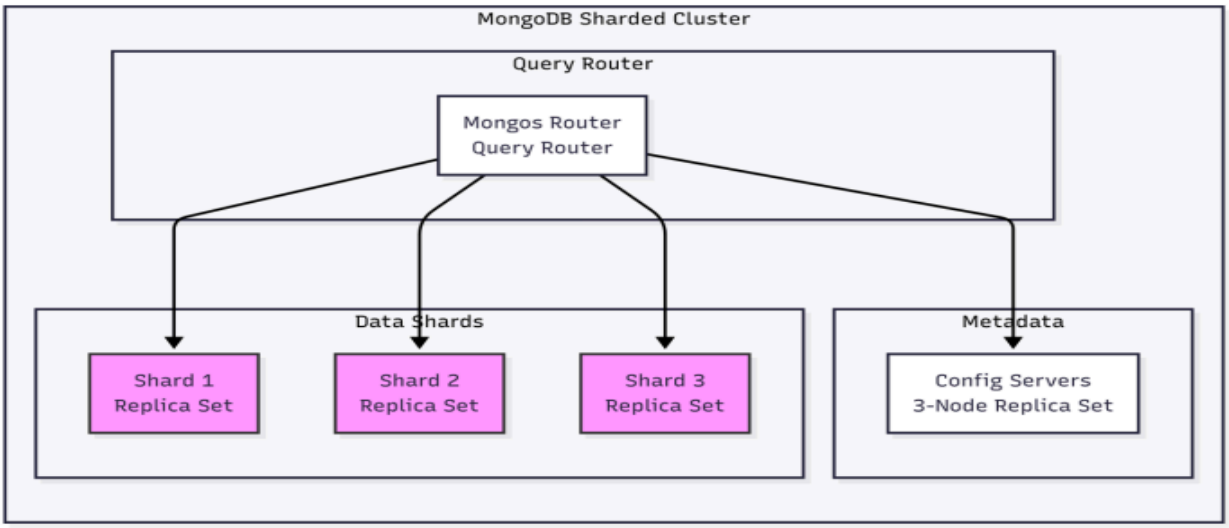
- Horizontal Scalability through Sharding: As the user base and data volumes grow, MongoDB's native sharding mechanism distributes data across multiple database servers,

enabling linear scalability. Queries can be transparently distributed to relevant shards, maintaining performance as the dataset grows beyond the capacity of a single machine[13].

3.3.5.1 Storage Model

The Primary Database of the system is MongoDB Atlas Sharded Cluster.

Figure 3.3: Storage Model



3.3.5.2 MongoDB Collection Schemas

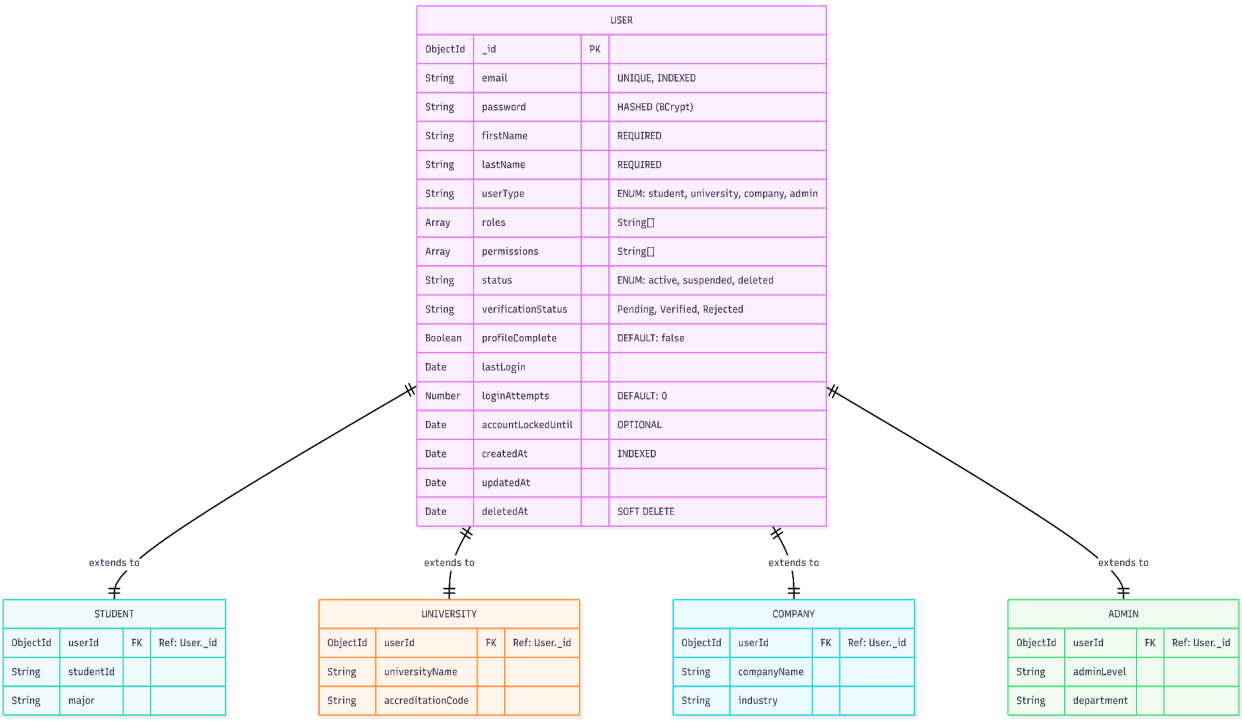
The InternConnect data model is organized into ten primary collections, each representing a distinct entity type or operational domain.

3.3.5.2.1 Users Collection (Foundational)

Stores base user information (email, password hash, roles) common to all user types. This collection implements inheritance, where specific user types (Student, University, Company)

reference the User document via userId foreign key. Indexing on email (unique) ensures fast authentication lookups. Indexing on userType enables role-based filtering for administrative operations[14].

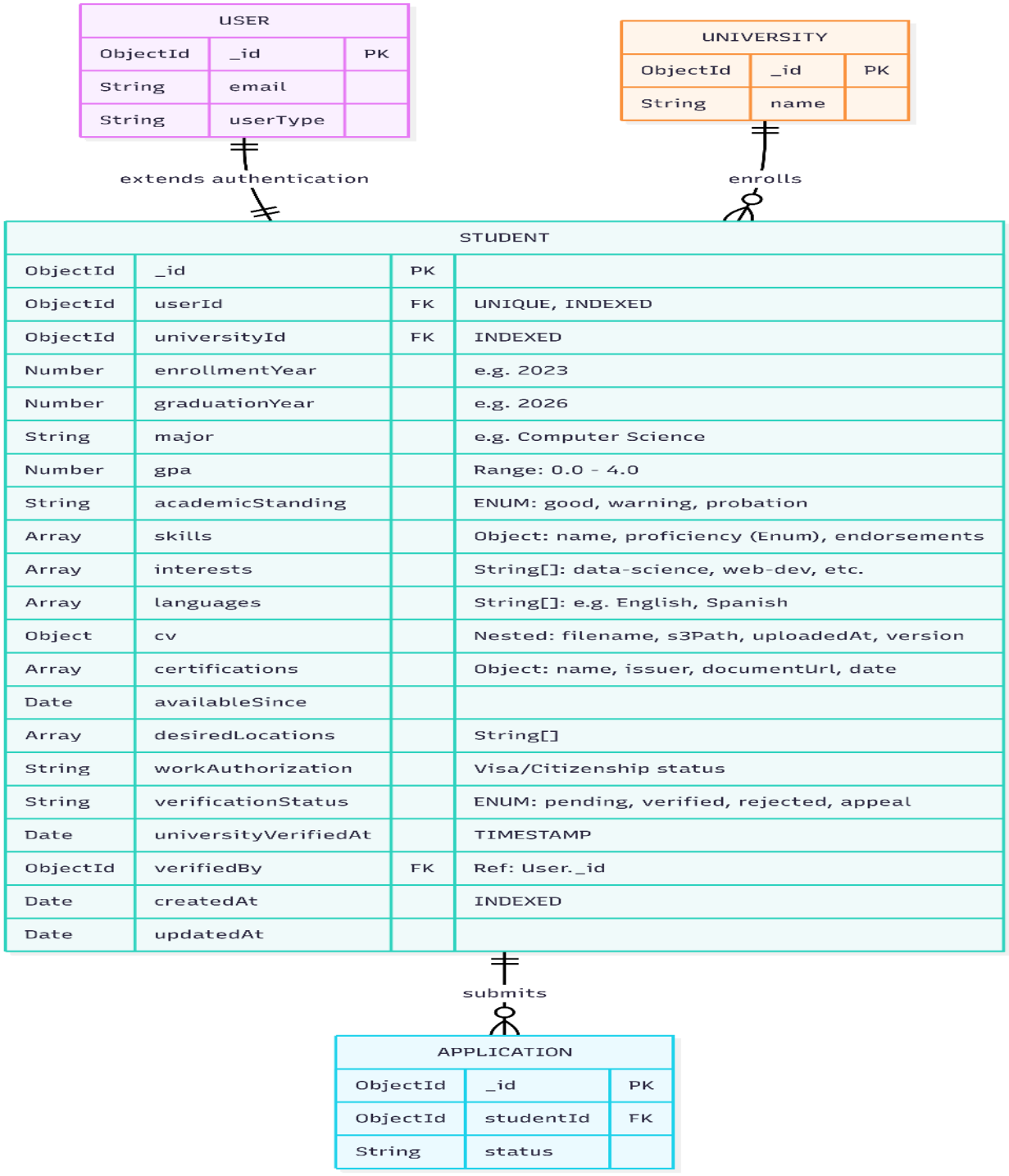
Figure 3.4 User collection Schema



3.3.5.2.2. Student Collection (Domain-Specific)

It contains student-specific attributes including academic profile (university, major, GPA, academic standing), skills with proficiency levels, interests in internship categories, uploaded CV metadata, certifications, and verification status. This collection grows rapidly as students engage with the system to complete their profiles and apply for internships.

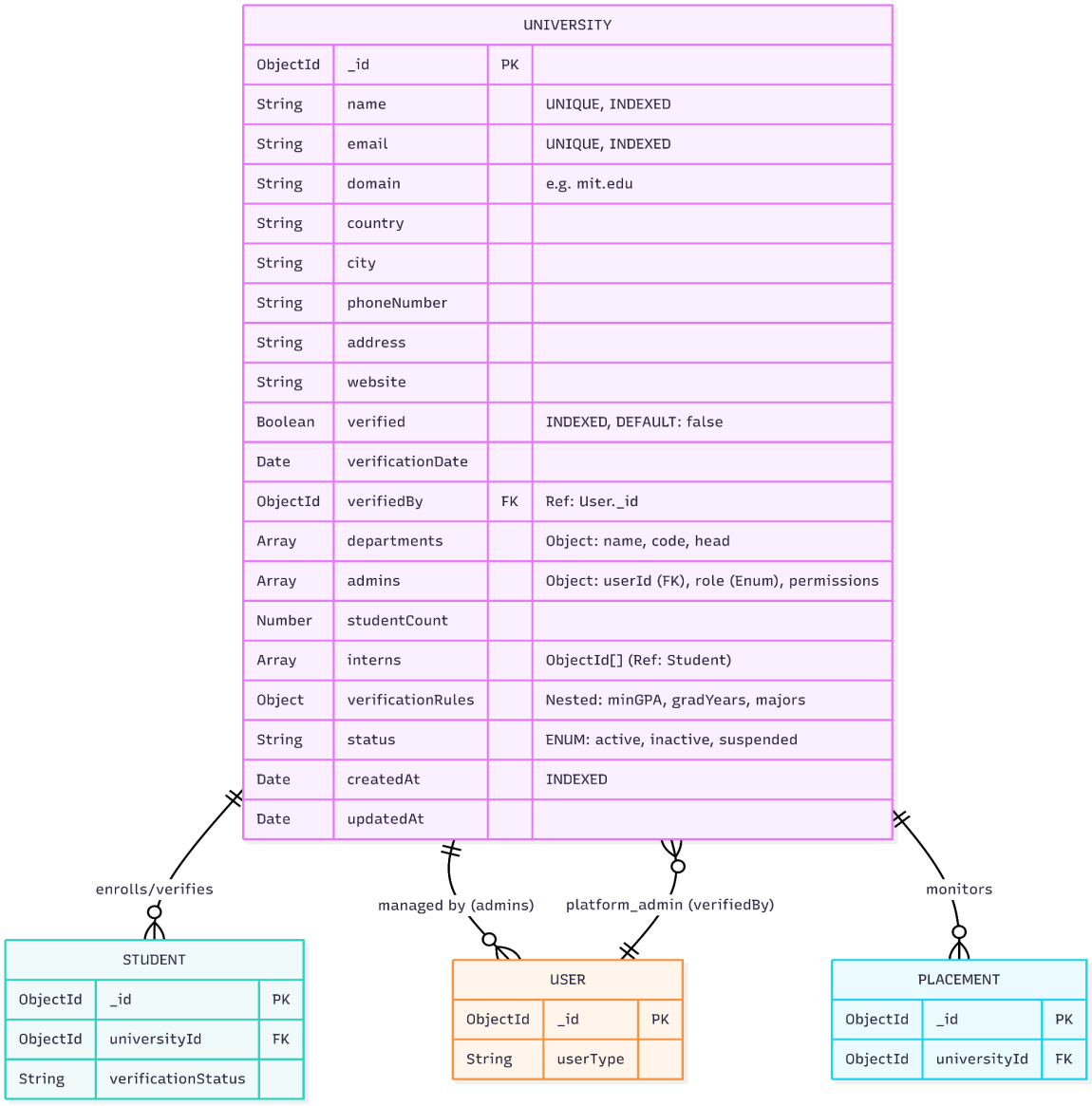
Figure 3.5 Student collection Schema


USER
ObjectId _id PK
String email
String userType
UNIVERSITY
ObjectId _id PK
String name
extends authentication
enrolls
STUDENT
ObjectId _id PK
ObjectId userId FK UNIQUE, INDEXED
ObjectId universityId FK INDEXED
Number enrollmentYear e.g. 2023
Number graduationYear e.g. 2026
String major e.g. Computer Science
Number gpa Range: 0.0 - 4.0
String academicStanding ENUM: good, warning, probation
Array skills Object: name, proficiency (Enum), endorsements
Array interests String[]: data-science, web-dev, etc.
Array languages String[]: e.g. English, Spanish
Object cv Nested: filename, s3Path, uploadedAt, version
Array certifications Object: name, issuer, documentUrl, date
Date availableSince
Array desiredLocations String[]
String workAuthorization Visa/Citizenship status
String verificationStatus ENUM: pending, verified, rejected, appeal
Date universityVerifiedAt TIMESTAMP
ObjectId verifiedBy FK Ref: User._id
Date createdAt INDEXED
Date updatedAt
submits
APPLICATION
ObjectId _id PK
ObjectId studentId FK
String status

3.3.5.2.3 University Collection (Organizational)

Represents academic institutions with organizational details (name, domain, departments), verification authority status (ability to verify student academic standing), administrator user references, and verification rule definitions. Universities grow at a slower rate than students, typically limited by the number of academic institutions adopting the system.

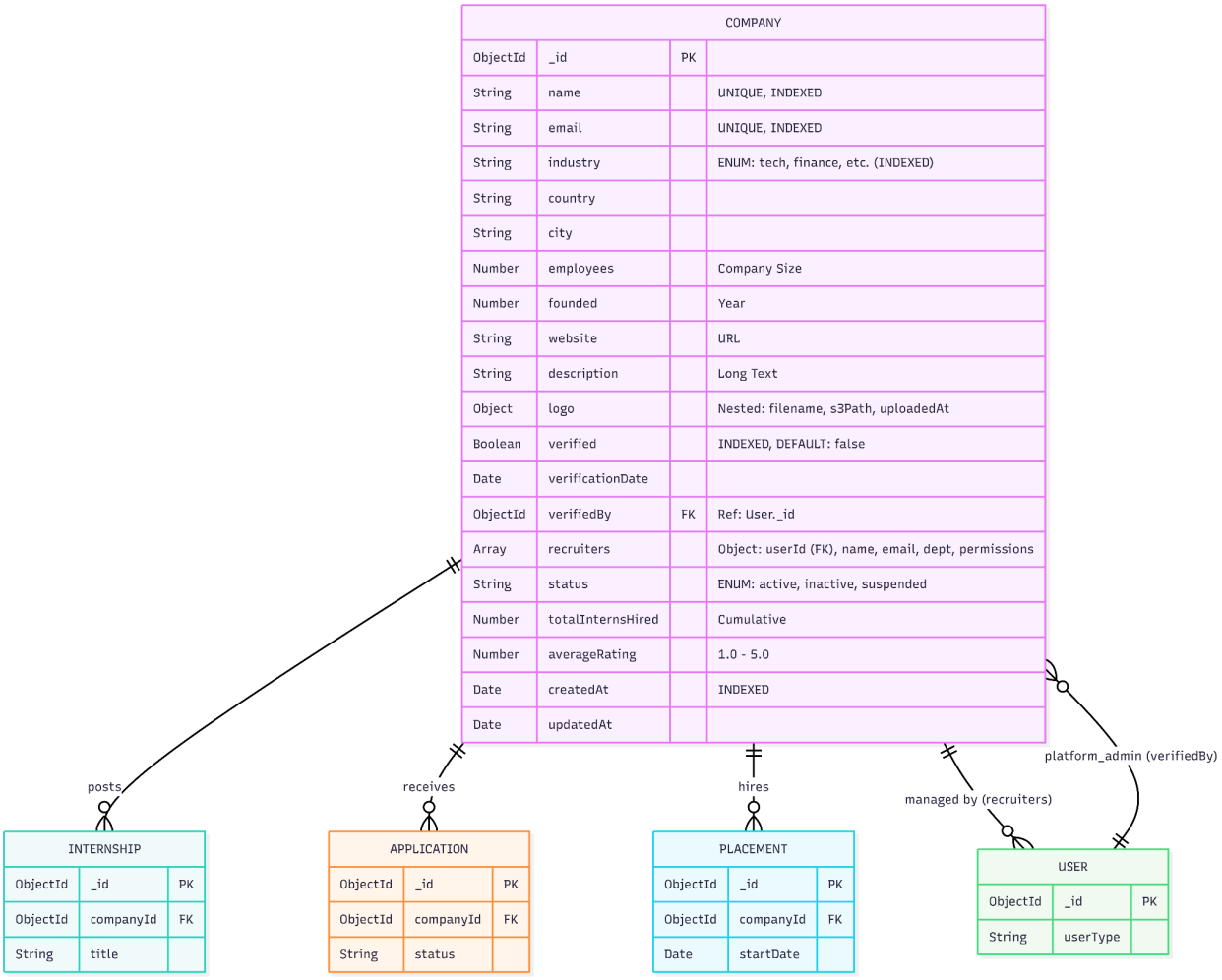
Figure 3.6 University Collection Schema



3.3.5.2.4 Company Collection (Organizational)

Represents recruiting organizations with company details (name, industry, size), recruiter management (contact information, permissions), verification status (approved to post internships), and hiring metrics. Companies typically onboard at a faster rate than universities during scaling phases.

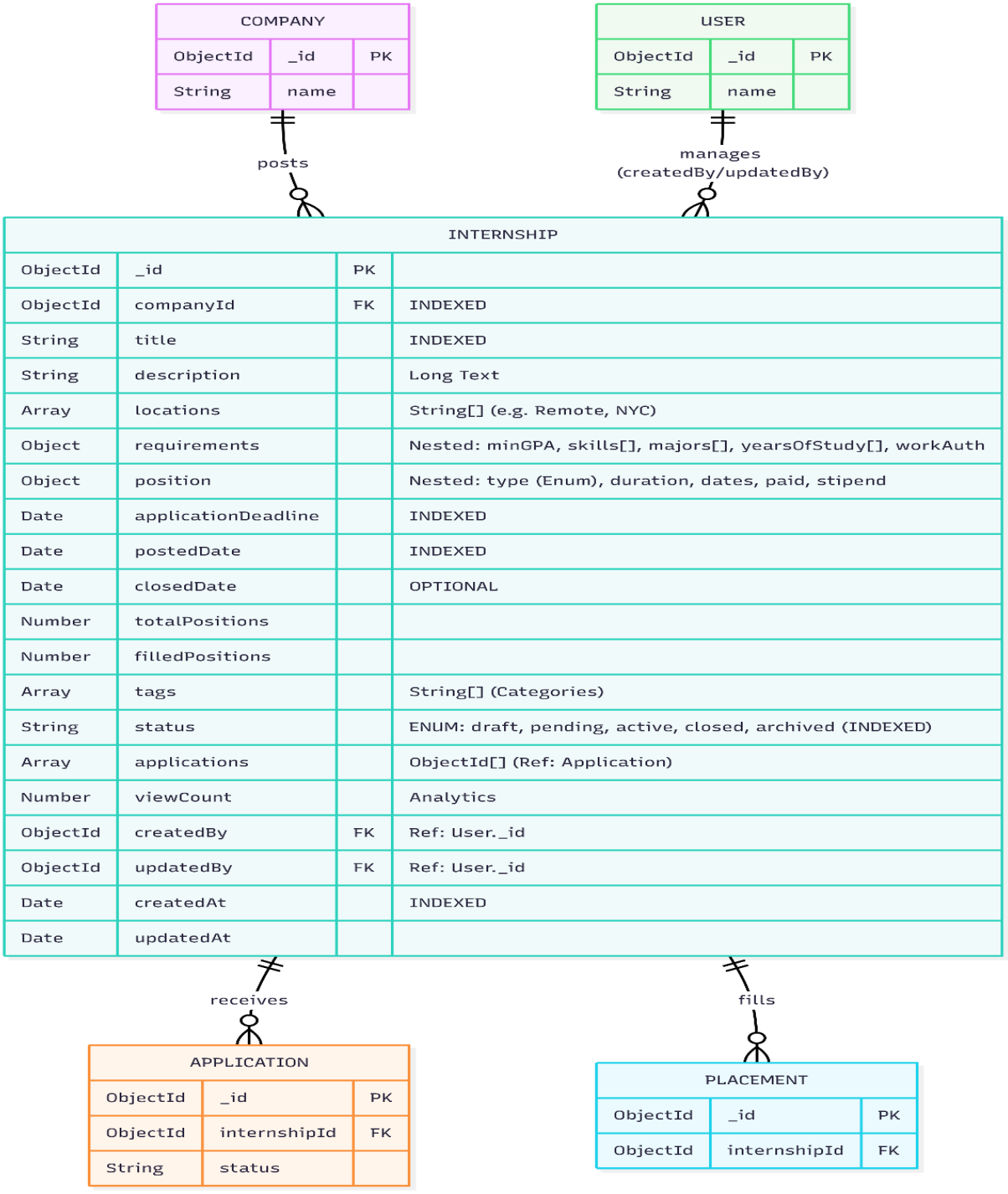
Figure 3.7 Company Collection Schema



3.3.5.2.5 Internship Collection Core

operational collection storing internship opportunity postings. Each document captures internship details (title, description, requirements, position type, locations, timeline, compensation), posting metadata, application tracking, status lifecycle, and visibility metrics. This collection experiences high read traffic during search/discovery phases and moderate write traffic during posting and closure.

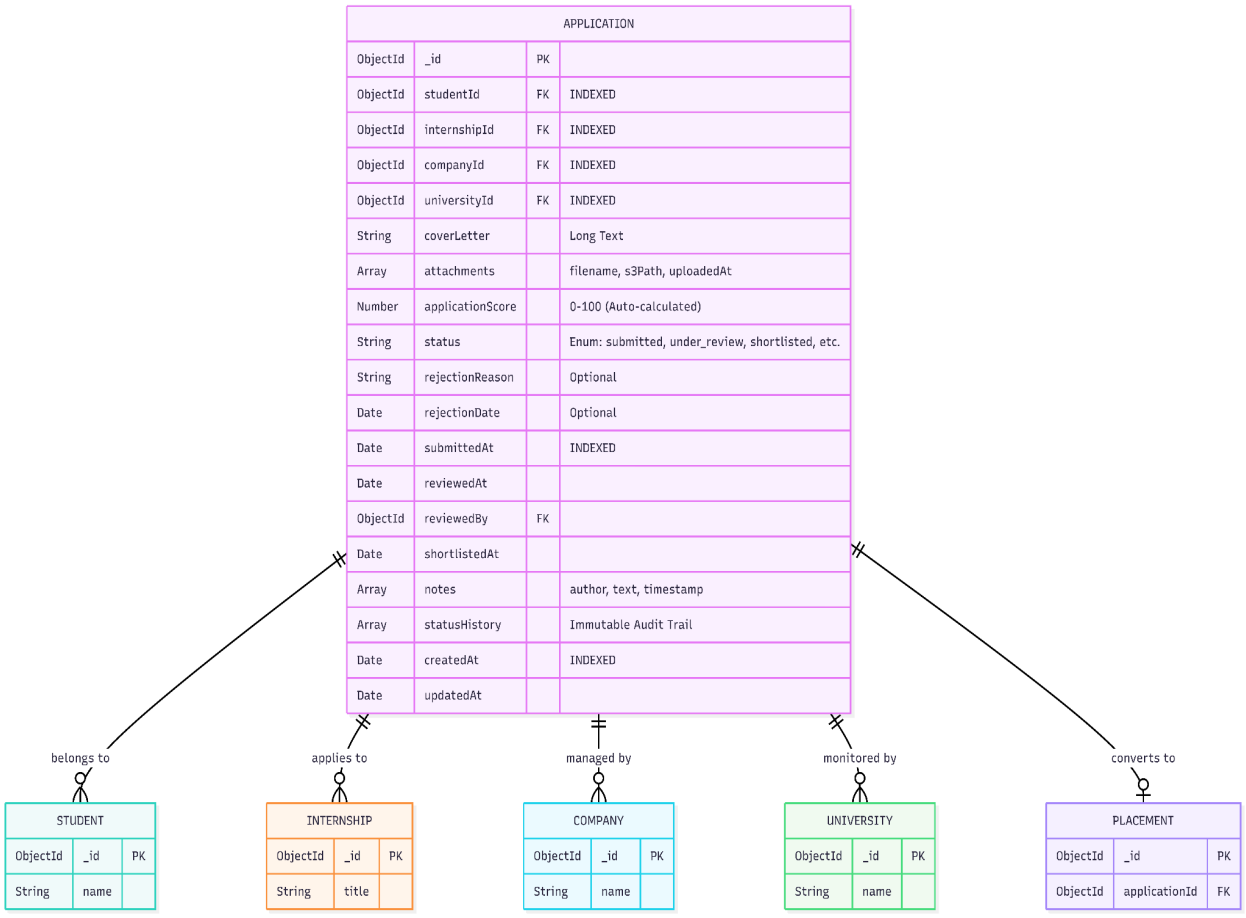
Figure 3.8 Internship Collection Schema


COMPANY
ObjectId _id PK
String name
USER
ObjectId _id PK
String name
posts
manages
(createdBy/updatedBy)
INTERNSHIP
ObjectId _id PK
ObjectId companyId FK INDEXED
String title INDEXED
String description Long Text
Array locations String[] (e.g. Remote, NYC)
Object requirements Nested: minGPA, skills[], majors[], yearsOfStudy[], workAuth
Object position Nested: type (Enum), duration, dates, paid, stipend
Date applicationDeadline INDEXED
Date postedDate INDEXED
Date closedDate OPTIONAL
Number totalPositions
Number filledPositions
Array tags String[] (Categories)
String status ENUM: draft, pending, active, closed, archived (INDEXED)
Array applications ObjectId[] (Ref: Application)
Number viewCount Analytics
ObjectId createdBy FK Ref: User._id
ObjectId updatedBy FK Ref: User._id
Date createdAt INDEXED
Date updatedAt
receives
fills
APPLICATION
ObjectId _id PK
ObjectId internshipId FK
String status
PLACEMENT
ObjectId _id PK
ObjectId internshipId FK

3.3.5.2.6 Application Collection

Tracks student internship applications, including submission details, applicant information, status history, reviewer notes, scoring, and decision rationale. This collection grows rapidly during peak application periods (typically end of academic year for summer internships). Applications are immutable once created (status history captures all changes), supporting audit requirements.

Figure 3.9 Application Collection Schema

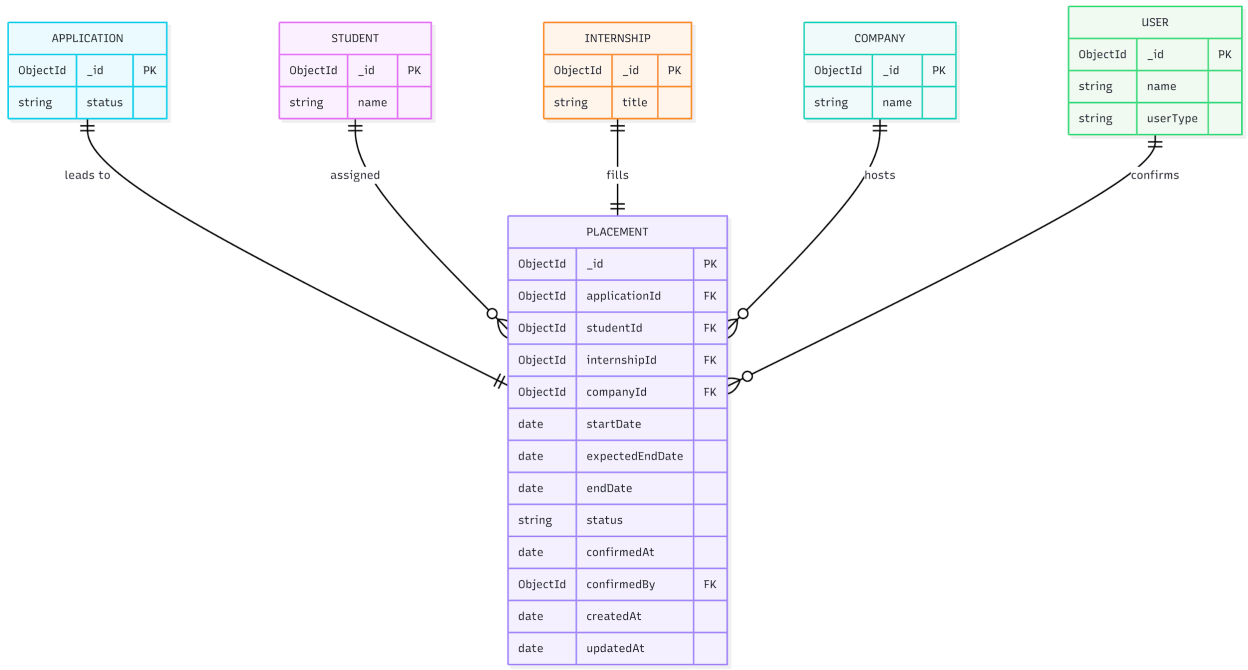


3.3.5.2.7 Placement Collection

Records confirmed internship placements, tracking start dates, end dates, completion status, progress updates, and feedback from both students and employers. Placements are the

measurable outcome of the system's core function (connecting students with internships). This collection supports tracking internship progress, collecting feedback for learning assessment, and measuring placement success metrics.

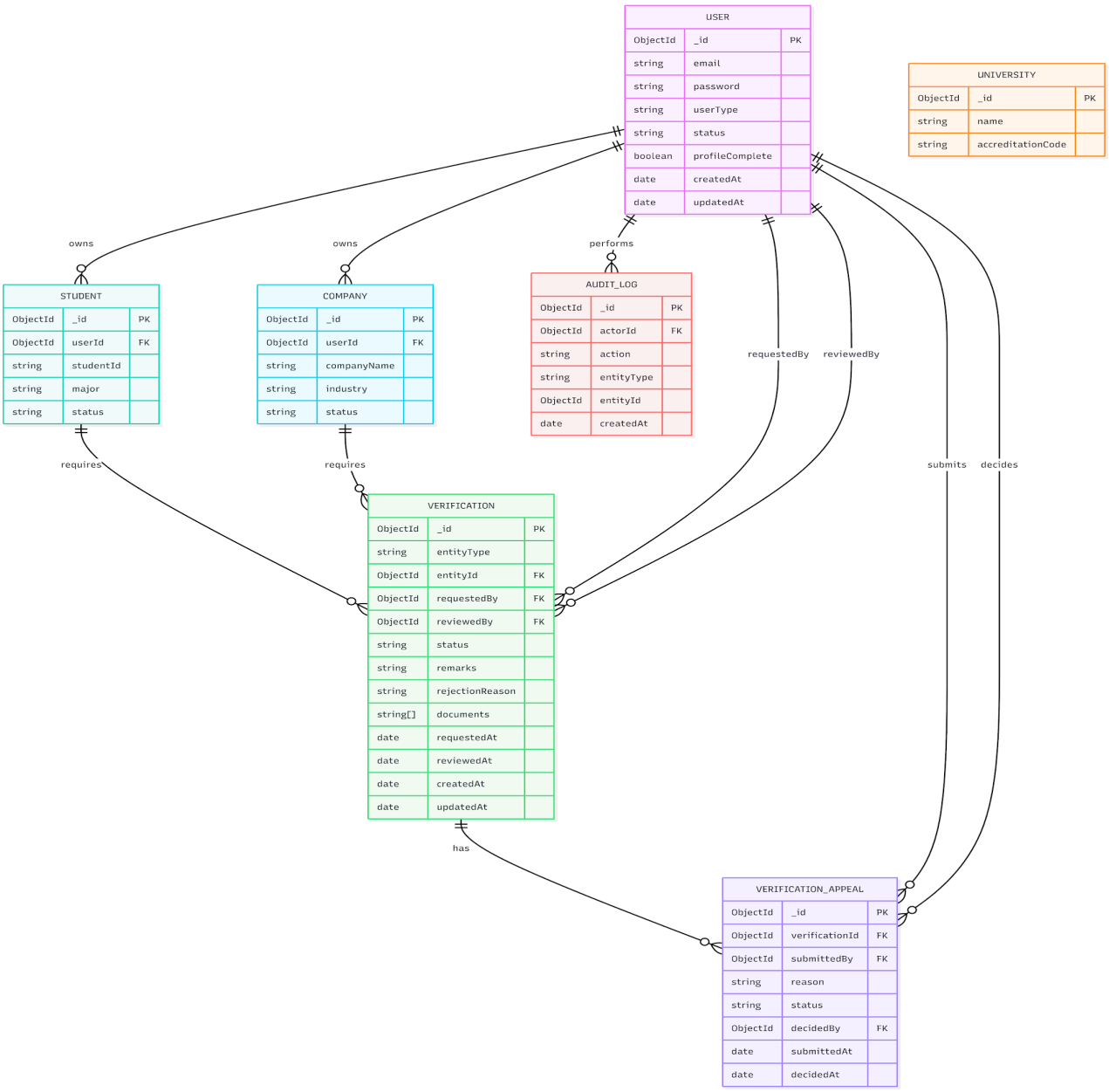
Figure 3.10 Placement Collection Schema



3.3.5.2.8. Verification Collection (Critical for Compliance)

Maintains academic verification records linking students to universities that have verified their academic standing and eligibility. Also tracks company verification (approval to post internships). This collection enforces the verification workflow and maintains an audit trail of verification decisions, appeals, and expiration.

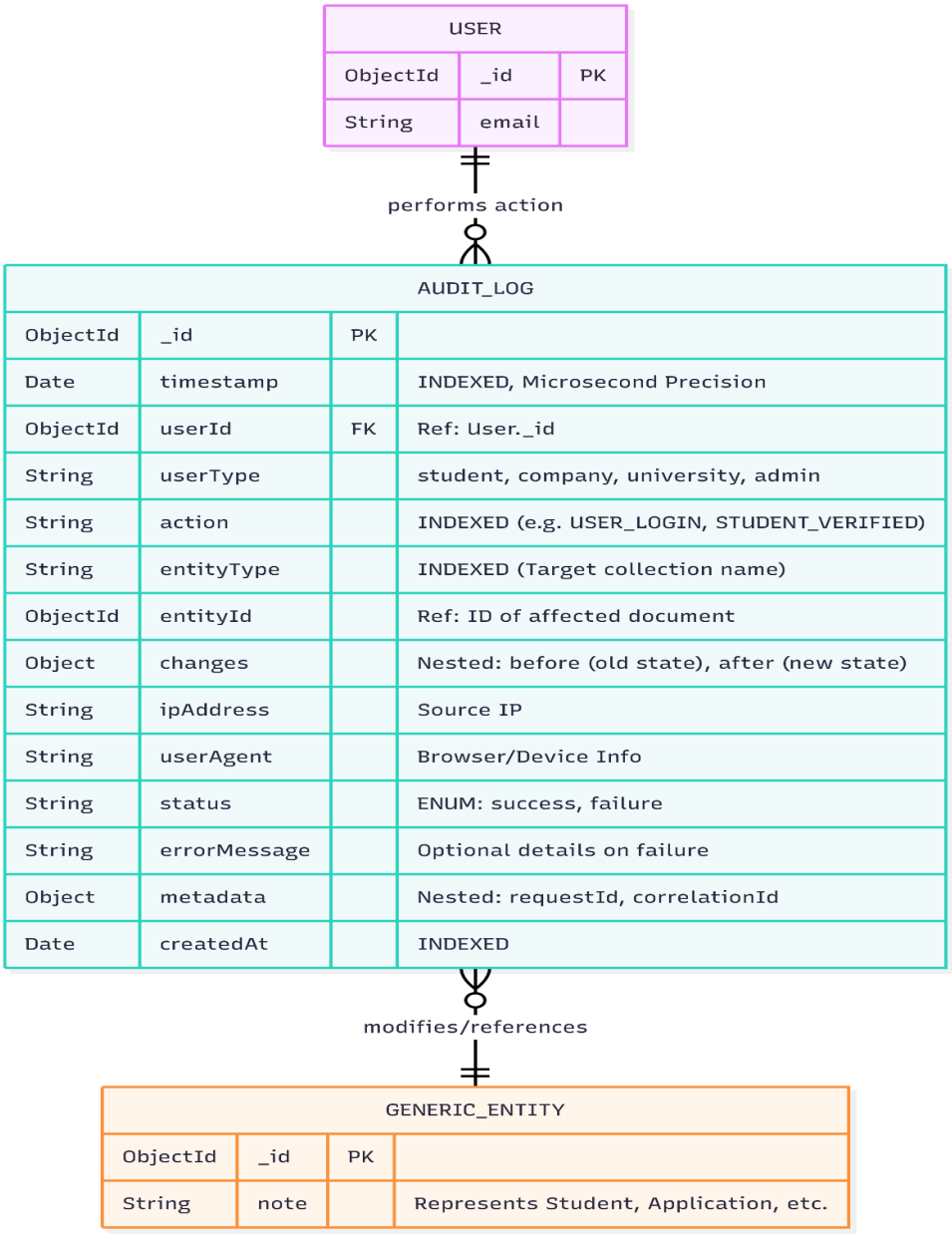
Figure 3.11 Verification Collection Schema



3.3.5.2.9 AuditLog Collection (Append-Only, Immutable)

Records every significant system activity: user logins, profile modifications, application status changes, verification decisions, data exports, and administrative actions. This collection is append-only (no updates, only inserts) to maintain immutability for compliance purposes. Storage volumes are significant, requiring archival policies.

Figure 3.12 AuditLogCollection Schema



3.3.5.2.10 Notification Collection (Transient)

Stores in-app notification records and email queue entries. Notifications are time-limited (automatically expire after 90 days for in-app, immediately after send for email). This collection experiences high write volume during peak periods (applications submitted, status changes announced) but has short retention requirements.

Figure 3.13 Notification Collection Schema

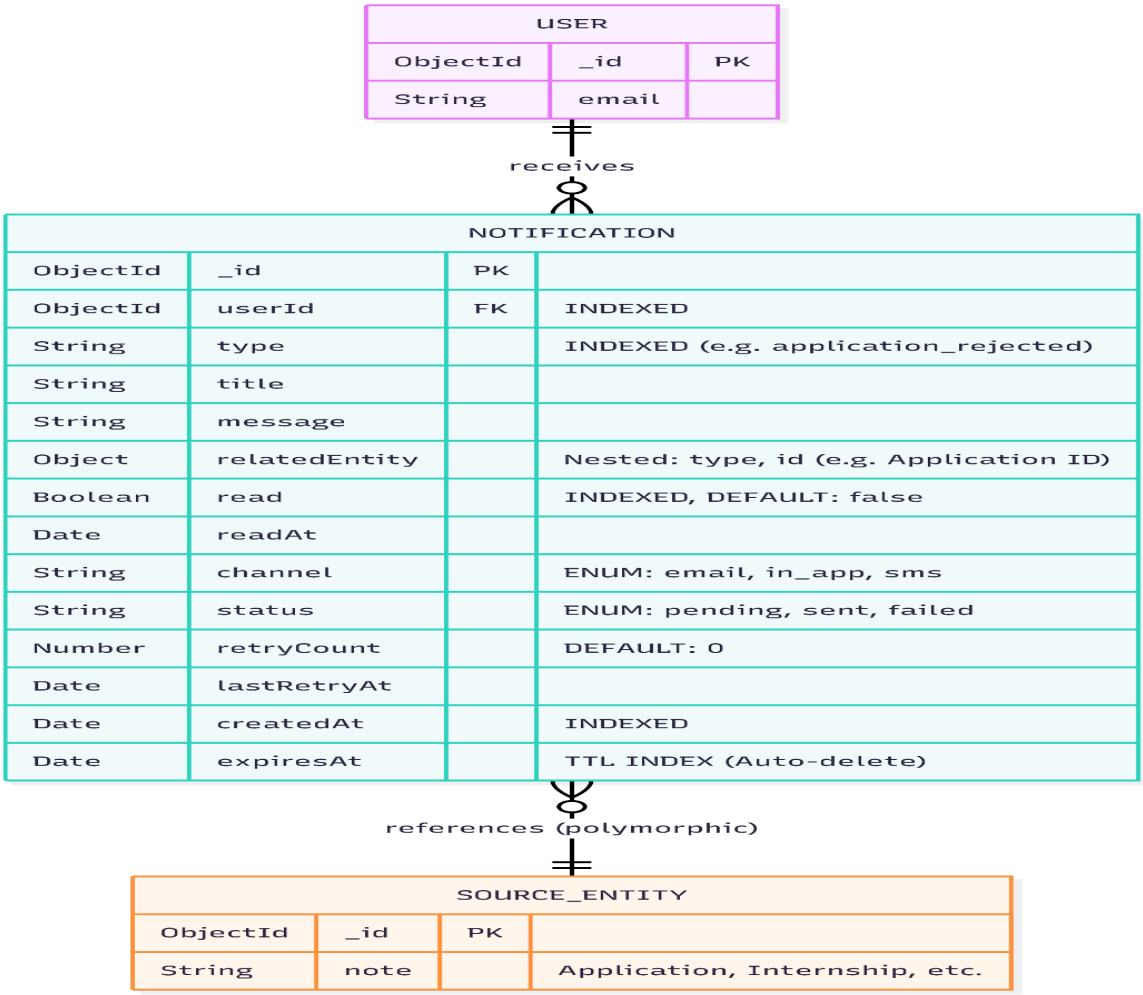
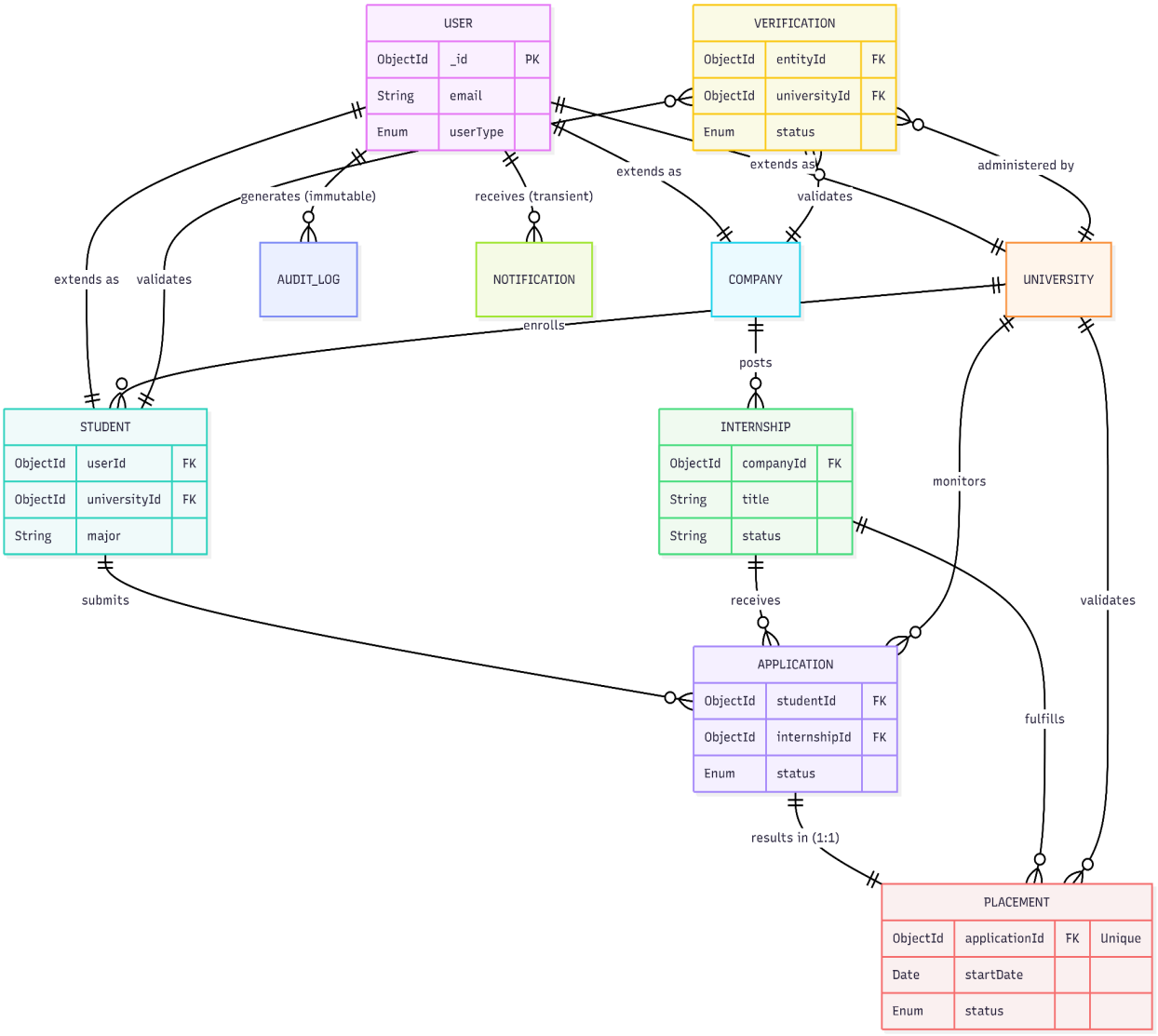


Figure 3.14 Collection Interaction Diagram

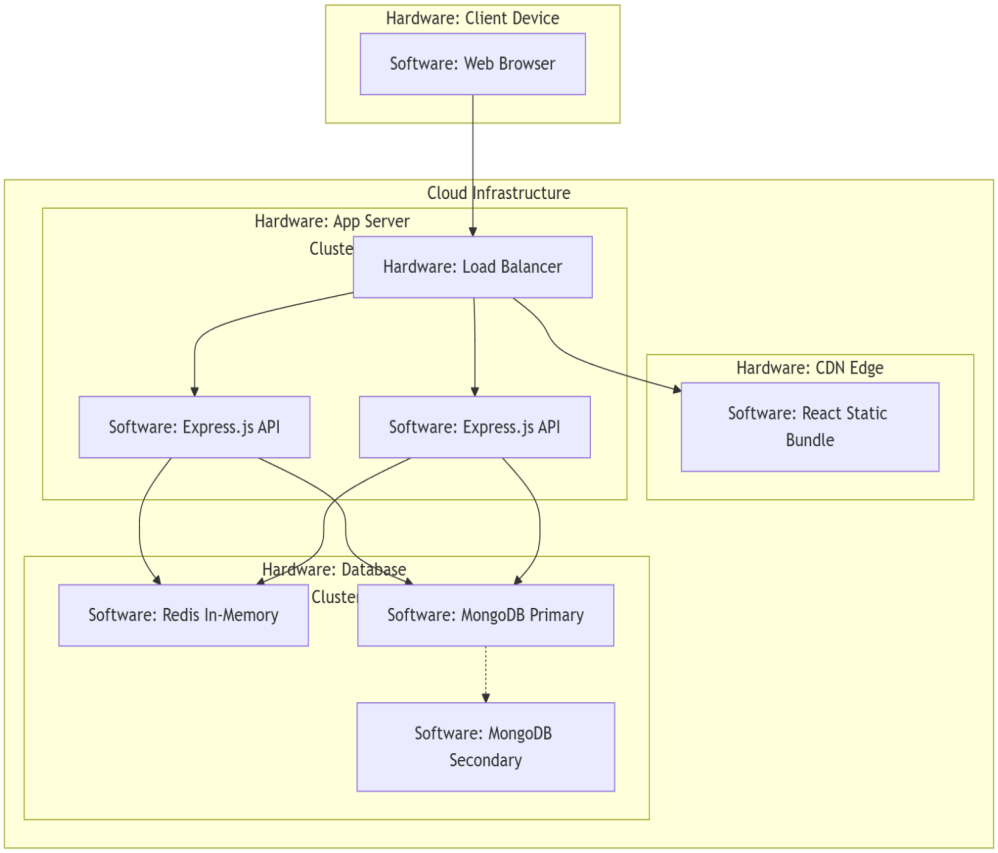


3.3.6 Hardware/Software Mapping

The Hardware/Software Mapping for the InternConnect system defines the strategic deployment of logical software components onto physical and virtualized computing resources. This architectural view is essential for bridging abstract design with physical infrastructure, ensuring that each layer of the MERN stack is allocated to environment-specific hardware optimized for its unique processing requirements.

By applying the following mapping, the InternConnect system achieves a robust alignment between software needs and hardware capabilities. This separation allows for independent scaling of tiers; application servers can be horizontally scaled to meet processing demand, while the data tier can be vertically or horizontally scaled to manage storage volume and query performance without affecting the front-end delivery[15].

Figure 3.15 Hardware to Software Mapping Diagram



3.3.7 Package

The InternConnect system follows a modular, package-based architecture organized into

functional domains. Each package encapsulates specific responsibilities, manages its own data structures and services, and communicates with other packages through well-defined interfaces.

The packages are designed following the principles of separation of concerns, modularity, and maintainability.

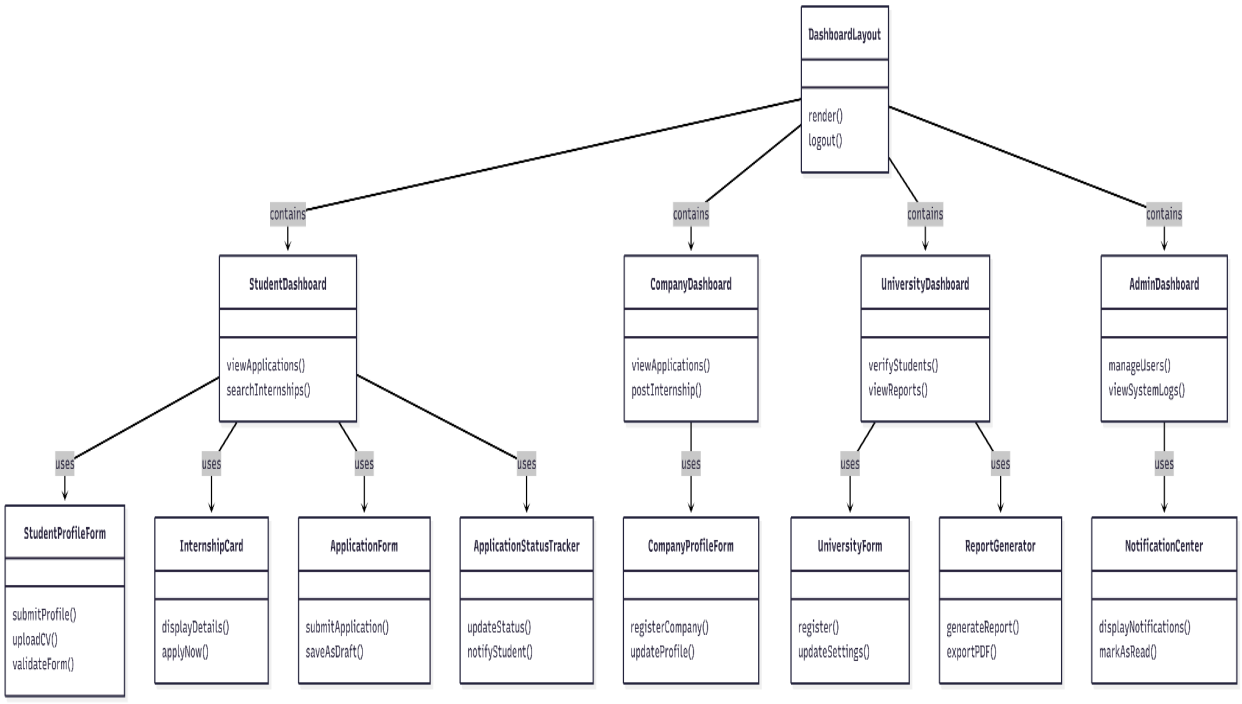
The architecture is divided into four main sections. The Presentation Layer Package contains the components responsible for the user interface and how people interact with the system. Below this, the Application Layer Packages handle the business logic and manage the various workflows. The Data Layer Packages are dedicated to storing, retrieving, and managing information. Finally, the Cross-Cutting Concern Packages provide essential services that support the entire system, such as security, activity logging, and connections to external tools

3.3.7.1 Presentation Layer Package

Package ID: PKG_01- User Interface Package

Purpose: Provides the web-based user interface for all system actors (students, companies, universities, and administrators). The package includes dashboards, forms, navigation components, and data visualization elements. Provides the web-based user interface for all system actors (students, companies, universities, and administrators). The package includes dashboards, forms, navigation components, and data visualization elements.

Figure 3.16 UI Package Class Diagram



3.3.7.2 Application Layer Package

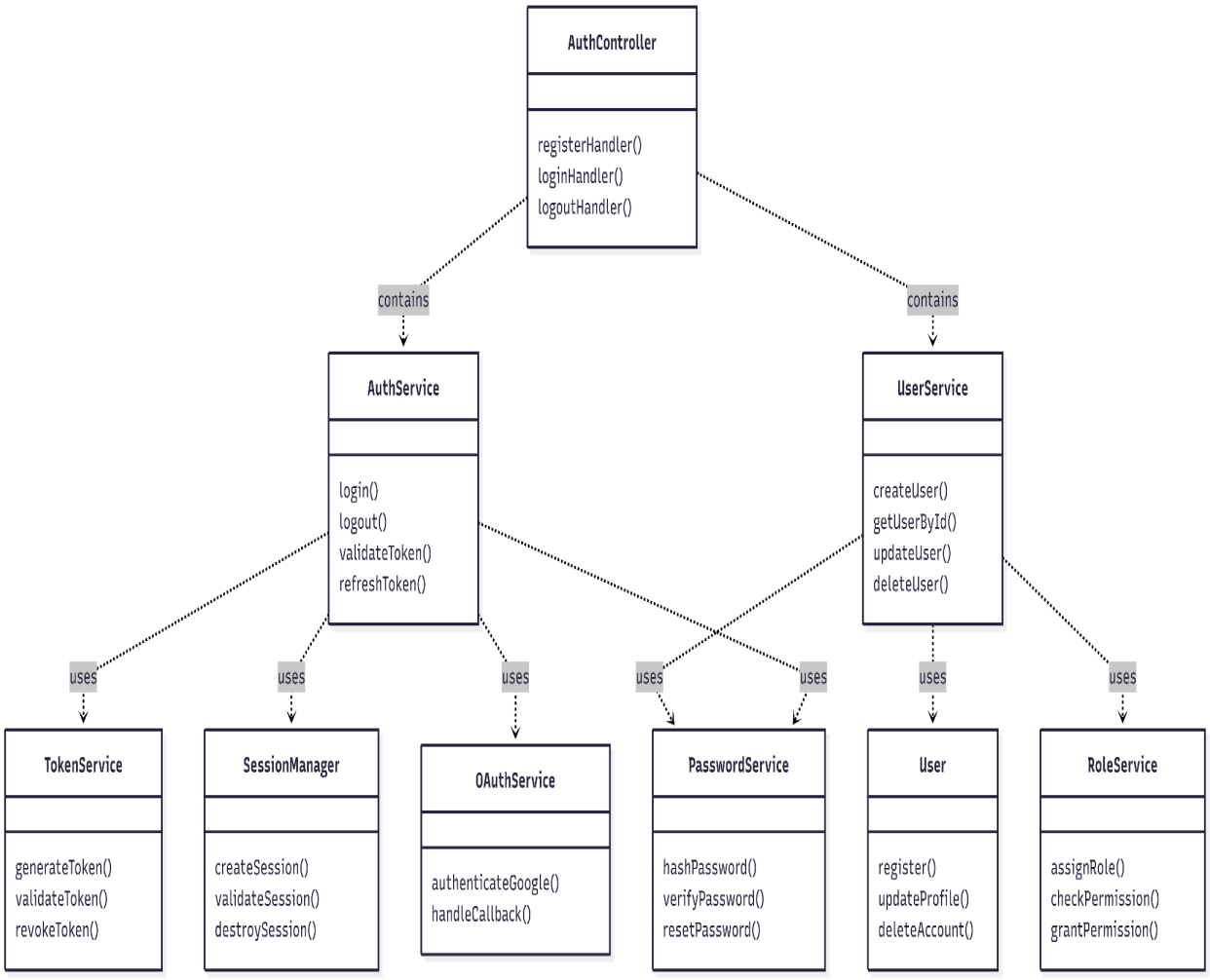
Package ID: PKG_02 - Authentication & User Management Package

Purpose: Manages user authentication, authorization, account management, and security-related operations. Ensures only authorized users can access the system and appropriate resources based on their roles.

Key Responsibilities:

1. User registration and account creation
2. Login/logout functionality
3. JWT token generation and validation
4. Password management and reset
5. Role-based access control (RBAC)
6. OAuth 2.0 integration for external authentication
7. Session management

Figure 3.17 Authentication & User Management Package Class Diagram



Package ID: PKG_03 Student Management Package

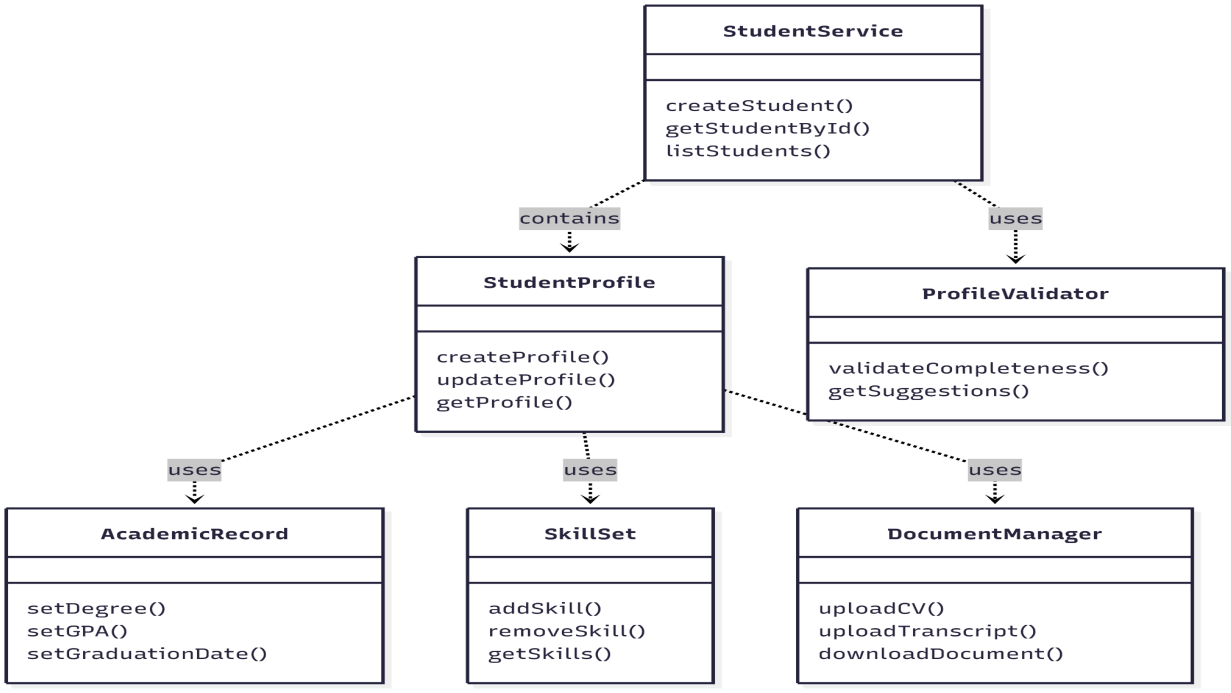
Purpose: Manages student profile information, academic details, and student related operations throughout the internship lifecycle.

Key Responsibilities:

1. Student profile creation and management
2. Academic information storage and retrieval

3. Document upload and management (CV, transcripts)
4. Skill and qualification tracking
5. Student eligibility verification
6. Profile search and filtering

Figure 3.18 Student Management Package Class Diagram



Package ID: PKG_04 Internship Management Package

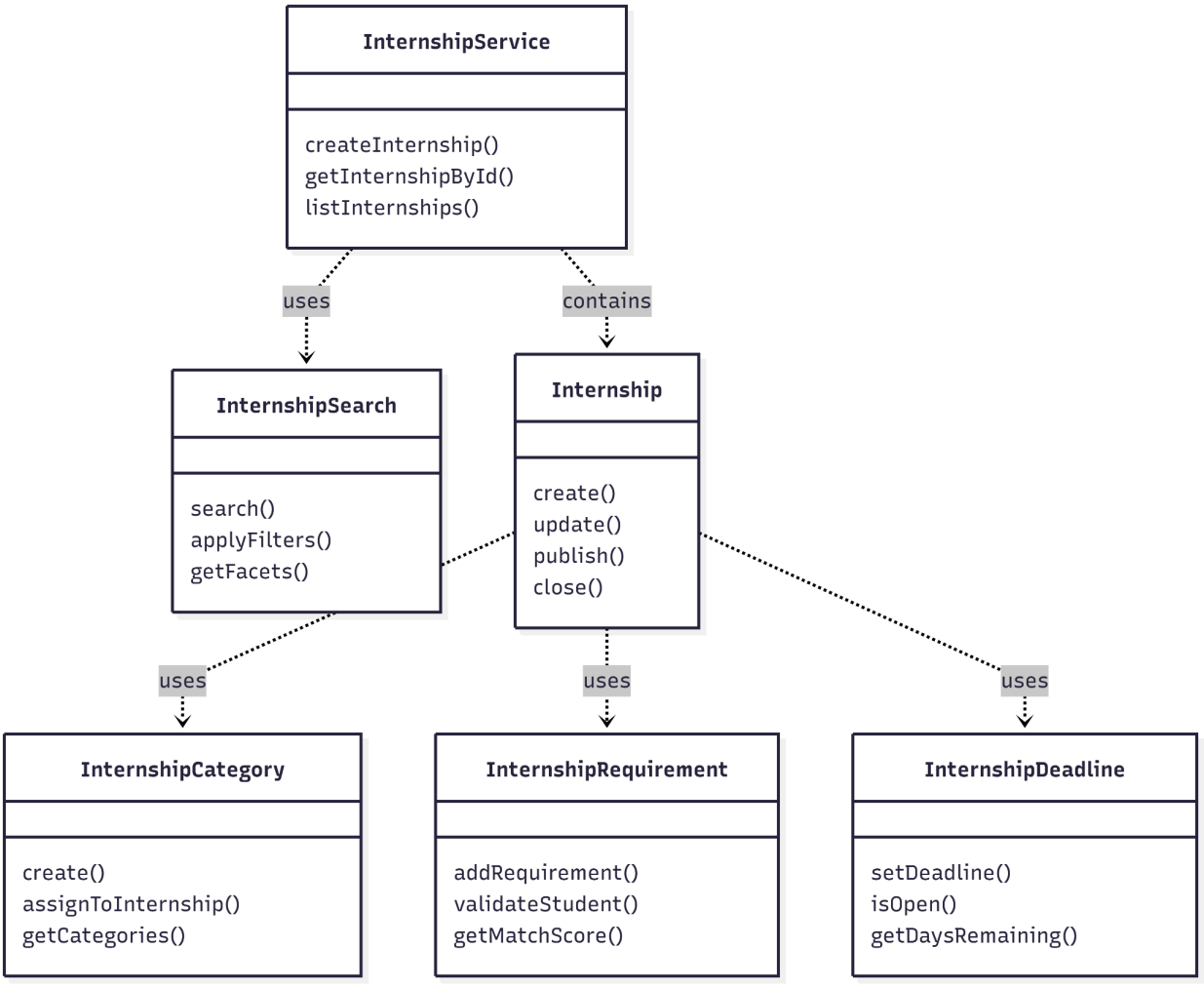
Purpose: Manages internship opportunities, postings, and related metadata. Provides functionality for companies to post internships and students to discover opportunities.

Key Responsibilities:

1. Internship posting and management
2. Internship search and filtering

3. Internship metadata management (duration, requirements, compensation
4. Internship visibility and promotion
5. Internship categorization and tagging
6. Internship deadline management

Figure 3.19 Internship Management Package Class Diagram



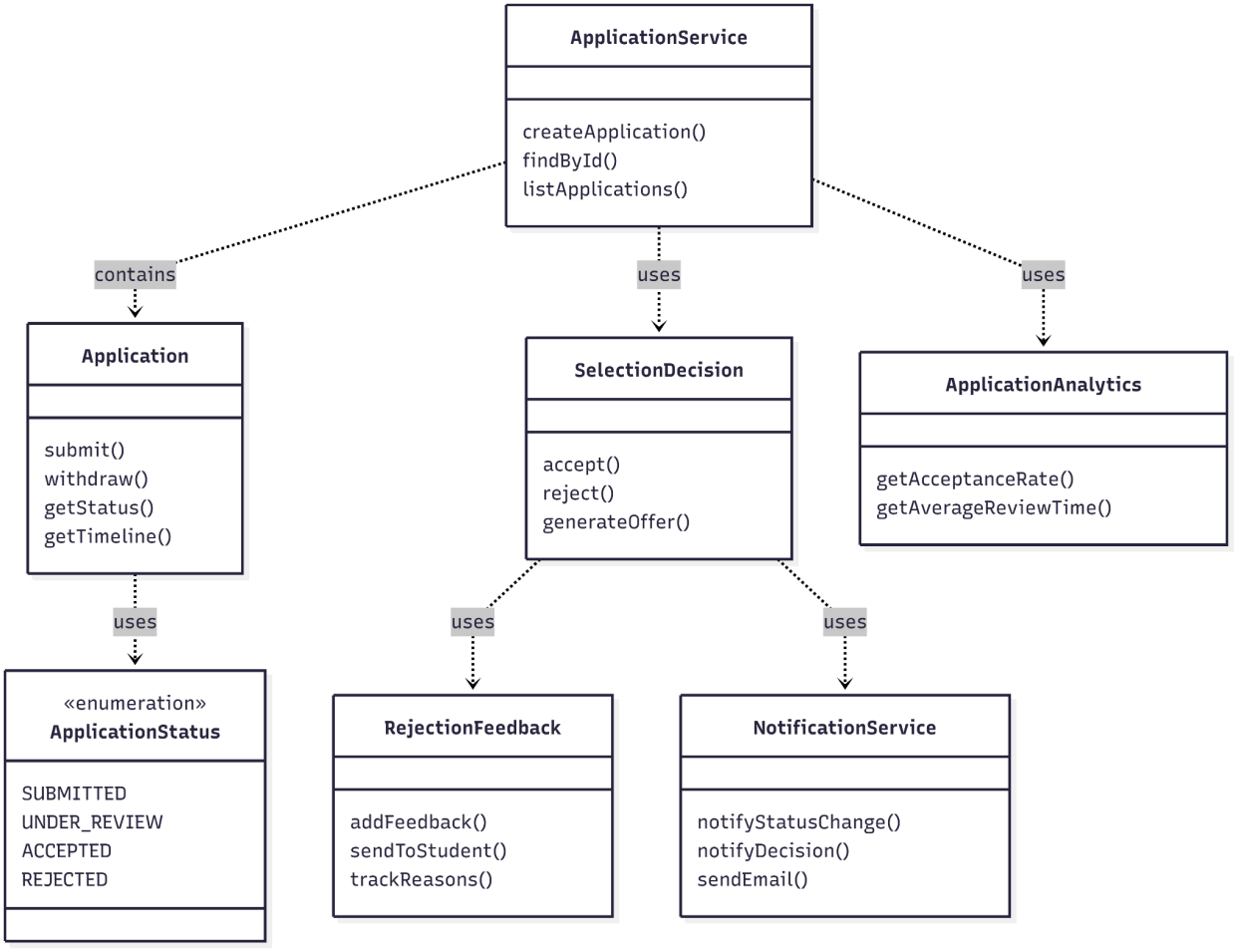
Package ID: PKG_05Application & Selection Package

Purpose: Manages the complete internship application lifecycle from submission through final decision, including review workflows and status tracking.

Key Responsibilities:

1. Application submission and validation
2. Application status tracking
3. Selection decision management
4. Rejection handling and feedback

Figure 3.20 Application Package Class Diagram



Package ID: PKG_06 Organization Management Package

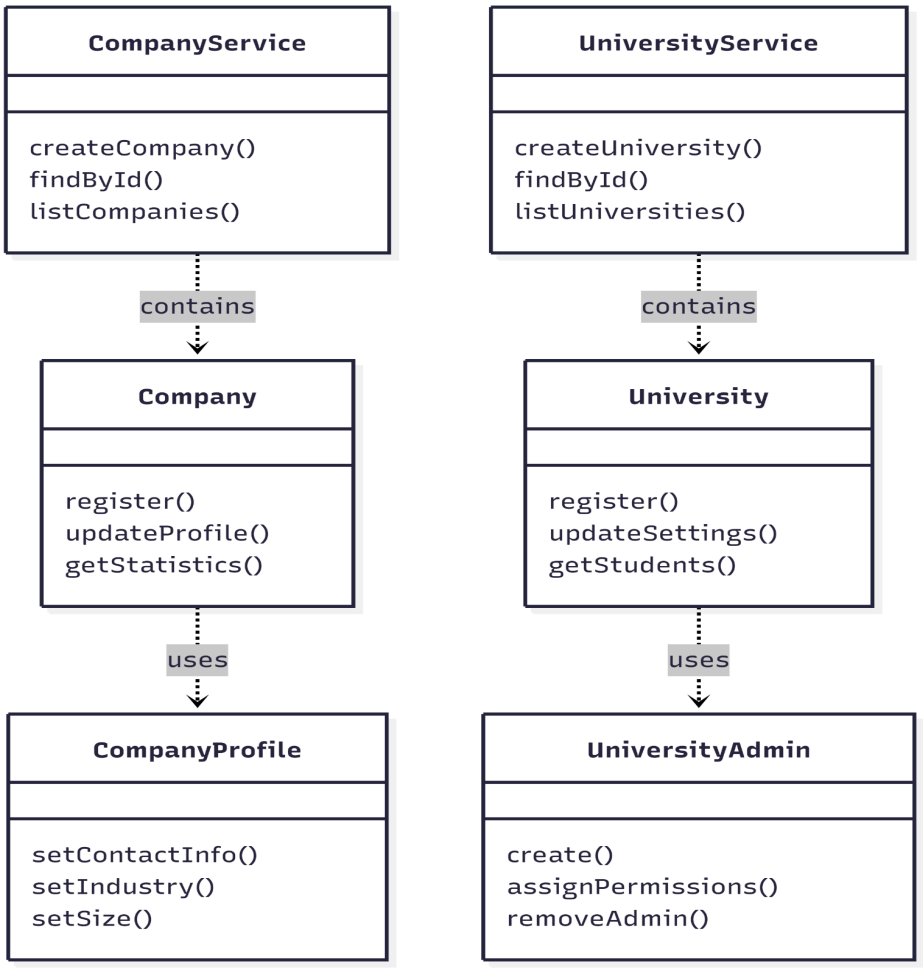
Purpose: Manages company and university profiles, registration, verification, and organization-specific settings and analytics.

Key Responsibilities:

1. Company registration and profile management
2. University administration and configuration
3. Organization verification and approval

4. Organization settings management

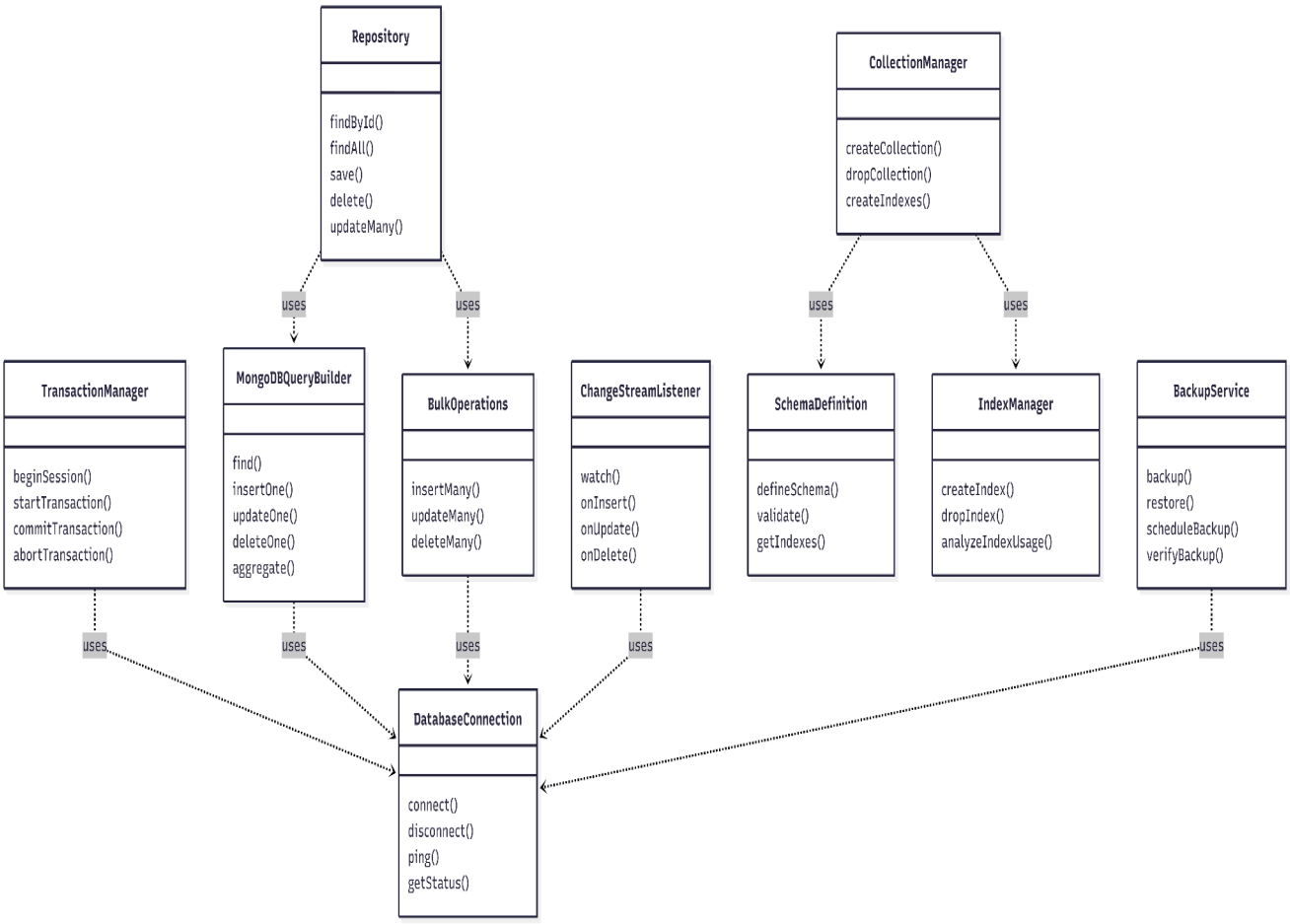
Figure 3.21 Organization Package Class Diagram



Package ID: PKG_07- Data Persistence Package

Purpose: Manages data persistence, database operations, and data consistency. Provides abstraction for database access across the system. Manages data persistence, database operations, and data consistency using MongoDB as the primary document database. Provides abstraction for database access across the system with document-oriented data modeling.

Figure 3.22 Data persistence Package Class Diagram



3.3.7.3 Cross-Cutting Concern Packages

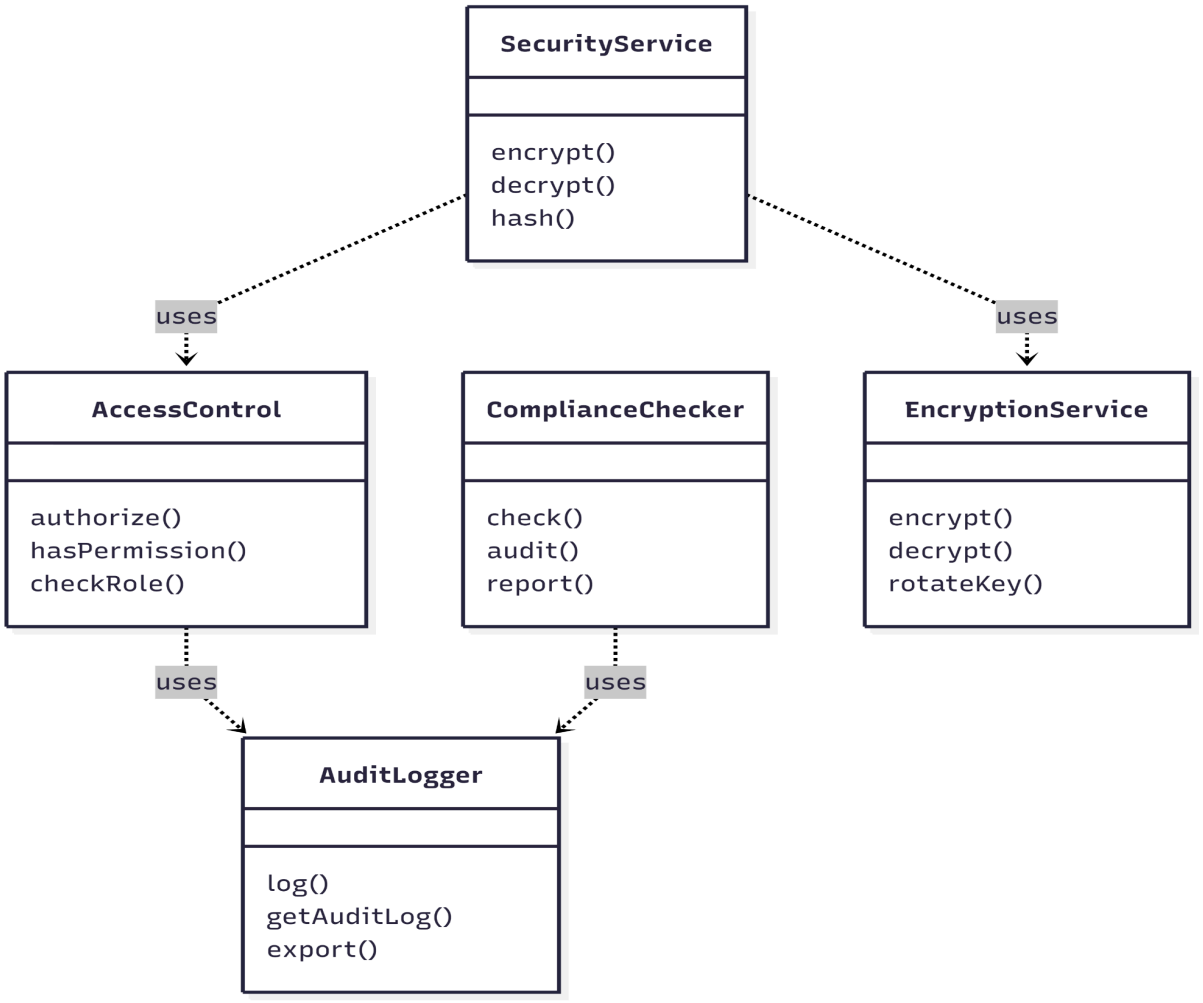
Package ID: PKG_08 - Security & Compliance Package

Purpose: Implements security mechanisms, access control, compliance enforcement, and audit logging across the system.

Key Responsibilities:

1. Role-based access control enforcement
2. Data encryption
3. API authentication and authorization
4. Audit trail maintenance
5. Compliance monitoring

Figure 3.23 Security and Compliance Package Class Diagram



Package ID: PKG_09 Notification & Communication Package

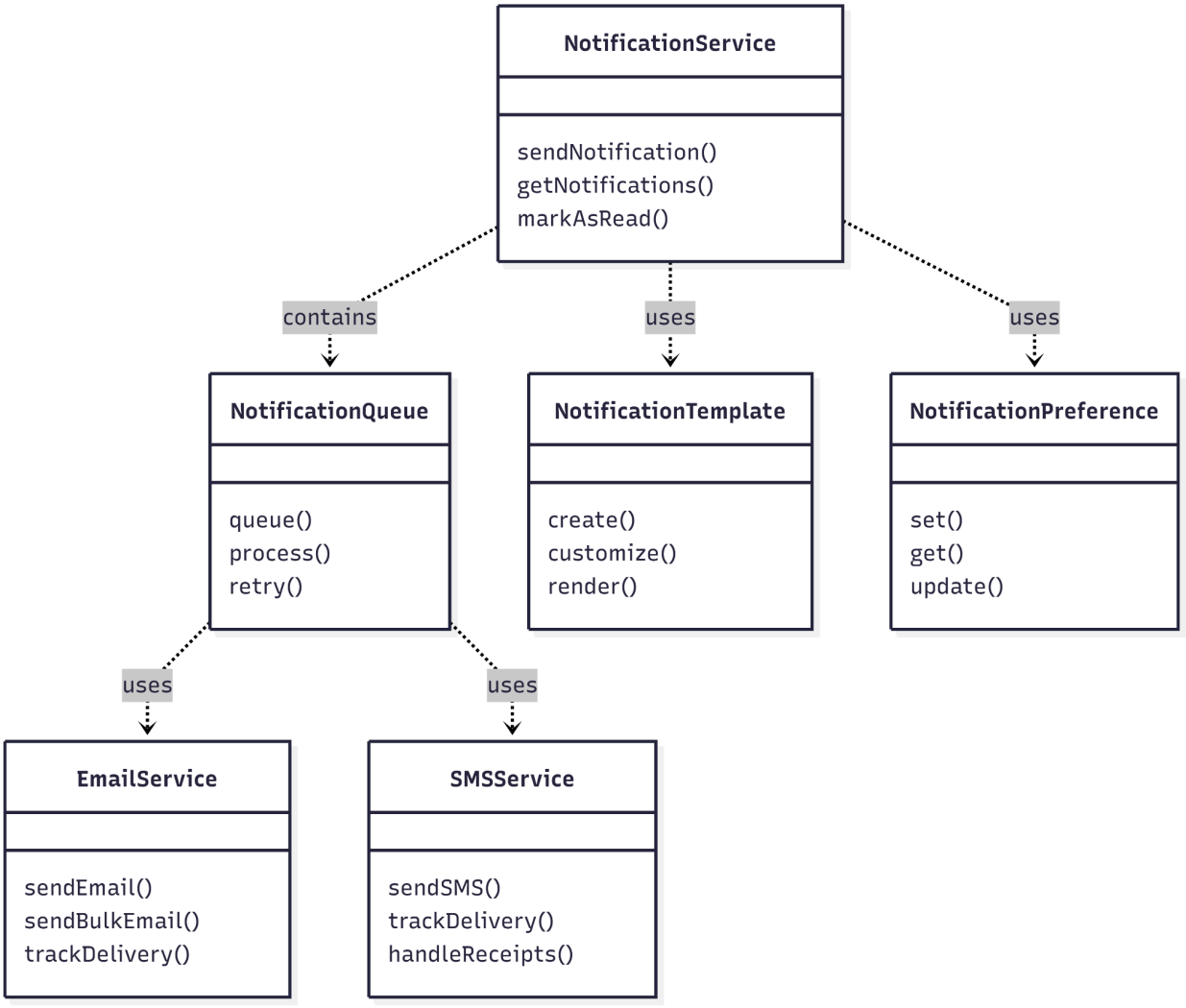
Purpose: Manages all forms of communication and notifications including email, SMS, and in-app notifications.

Key Responsibilities

1. Email notification delivery
2. SMS notification delivery
3. In-app notification management
4. Notification scheduling and queuing

5. Delivery tracking and retry logic
6. Notification preferences management

Figure 3.23 Notification and Communication Package Class Diagram



Package ID: PKG_10 Reporting & Analytics Package

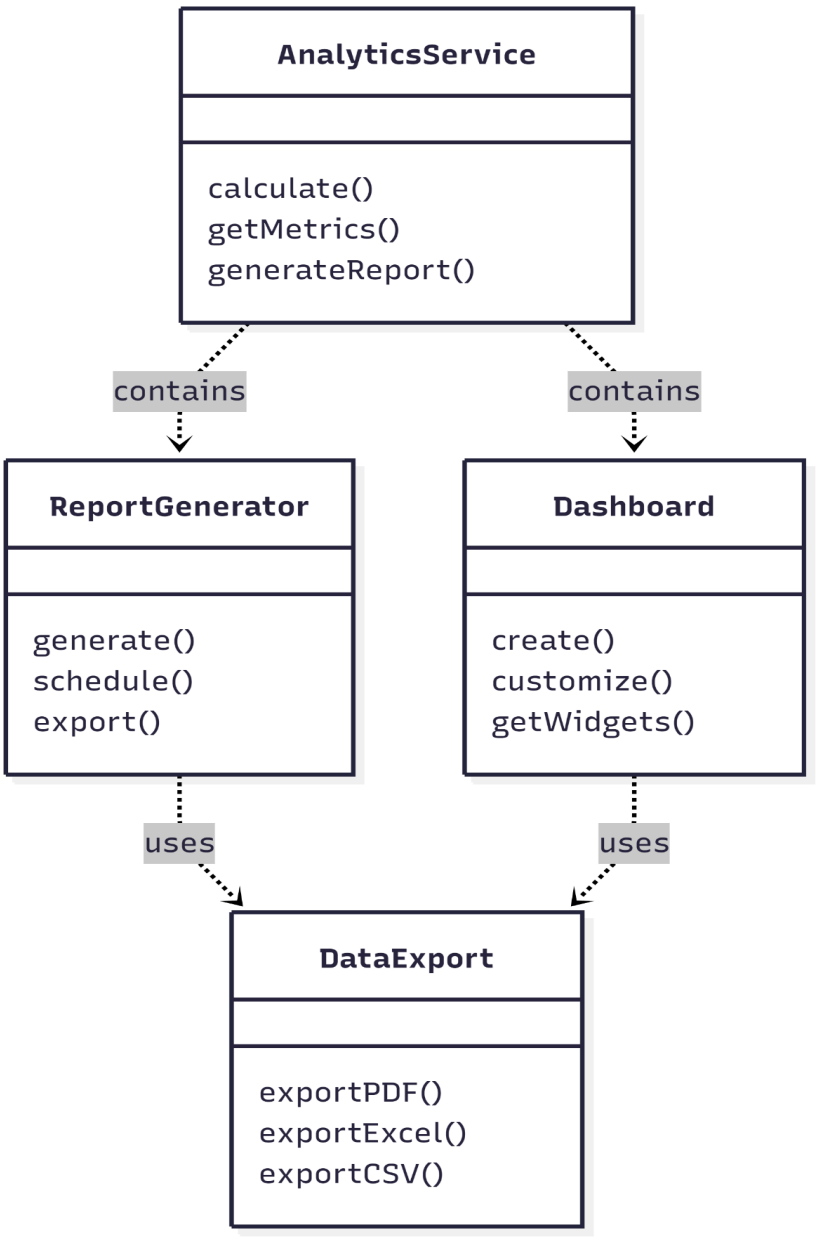
Purpose: Provides comprehensive analytics, reporting, and business intelligence capabilities across the system.

Key Responsibilities:

1. Dashboard analytics and KPI tracking
2. Report generation and scheduling
3. Data visualization

4. Export capabilities (PDF, Excel, CSV)

Figure 3.24 Reporting and Analytics Class Diagram



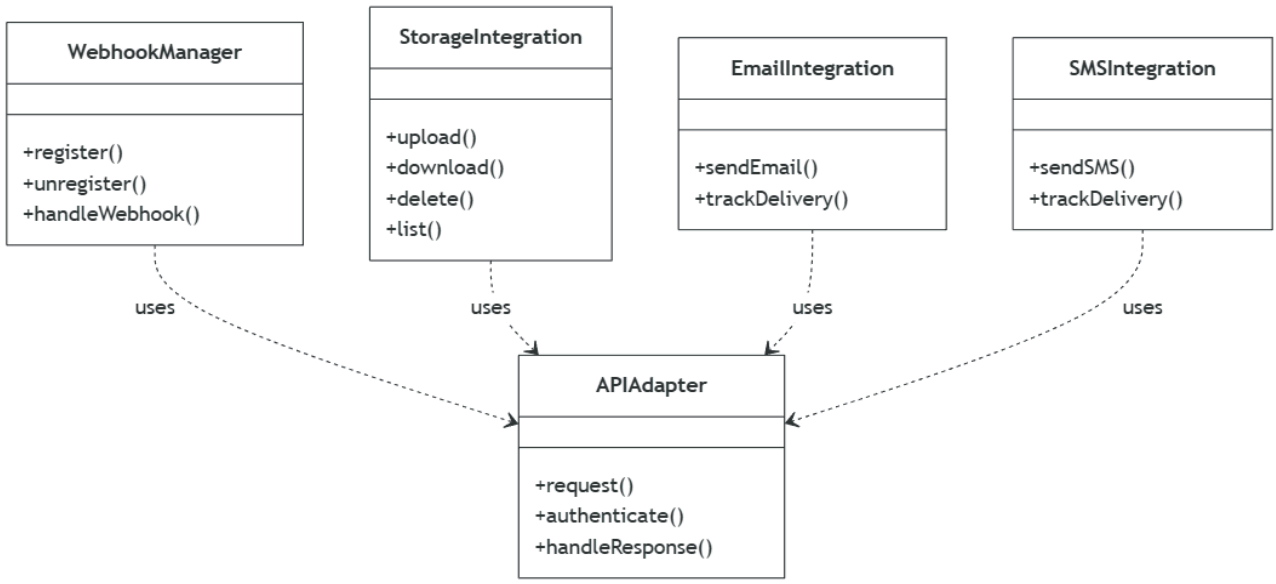
Package ID: PKG_11- External Integration Package

Purpose: Manages integrations with external services including payment processing, cloud storage, and email providers.

Key Responsibilities:

1. Cloud storage integration
2. Email service provider integration
3. SMS provider integration
4. Webhook management

Figure 3.24 External Integration Package Class Diagram


WebhookManager
+register()
+unregister()
+handleWebhook()
StorageIntegration
+upload()
+download()
+delete()
+list()
EmailIntegration
+sendEmail()
+trackDelivery()
SMSIntegration
+sendSMS()
+trackDelivery()
uses
uses
uses
uses
APIAdapter
+request()
+authenticate()
+handleResponse()

3.3.7.4 Package Dependency

Figure 3.24 External Integration Package Class Diagram

REFERENCE

1. **[1]** Professional development and internships in Ethiopian higher education https://www.nisc.co.za/products/abstracts/34307/students-experience-and-commentary-on-the-internship-programme-of-civil-engineering-training-in-public-universities-of-ethiopia
2. **[2]** Challenges of manual internship management in Ethiopia https://hetriil.moe.gov.et/hetriil/intership-offers
3. **[3]** MERN stack and web-based internship platforms https://www.macrothink.org/journal/index.php/jet/article/view/12945
4. **[4]** InternConnect overview and digital solutions for internship management https://immap.org/news/redefining-opportunities-in-humanitarian-tech-in-ethiopia-through-an-immersive-internship-program/
5. **[5]** Fragmented processes in university internship management https://www.iosrjournals.org/iosr-jmce/pages/16%281%29Series-2.html
6. **[6]** Need for streamlined university internship systems https://kje.kue.edu.et/index.php/kje/article/view/152
7. **[7]** Agile Software Development Life Cycle (SDLC) methodology https://www.agilealliance.org/agile101/
8. **[8]** Requirement gathering and analysis concepts https://www.geeksforgeeks.org/software-engineering-requirement-gathering-process/
9. **[9]** Functional and non-functional requirements in software development https://www.tutorialspoint.com/software_engineering/software_requirements.htm
10. **[10]** Three-tier architecture and separation of concerns https://www.geeksforgeeks.org/dbms/introduction-of-3-tier-architecture-in-dbms-set-2/

11. **[11]** Multi-tier architecture, scaling, and structural decomposition
    https://www.geeksforgeeks.org/multi-tier-architecture-in-software-engineering/
12. **[12]** MongoDB document-oriented data model
    https://www.mongodb.com/docs/manual/core/data-modeling/
13. **[13]** MongoDB sharding and horizontal scalability
    https://www.mongodb.com/docs/manual/sharding/
14. **[14]** MongoDB indexing and inheritance patterns
    https://learn.mongodb.com/courses/schema-design-patterns/
15. **[15]** Multi-tier architecture and independent tier scaling
    https://www.geeksforgeeks.org/multi-tier-architecture-in-software-engineering/