



**POLITECNICO**  
MILANO 1863

# **Software Engineering 2**

## **Requirements Analysis and**

## **Specification Document**

**Author(s): Shreesh Kumar Jha - 11022306**

**Samarth Bhatia - 11059097**

**Satvik Sharma - 11054680**

Academic Year: 2024-2025

Version: 3.0

Release date: 21/12/2024



# Contents

<b>Contents</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Purpose . . . . .	2
1.1.1 Goals . . . . .	2
1.2 Scope . . . . .	3
1.2.1 World Phenomena . . . . .	3
1.2.2 Shared phenomena . . . . .	5
1.3 Definition, Acronyms, Abbreviations . . . . .	6
1.4 Revision History . . . . .	7
1.5 Reference Documents . . . . .	7
1.6 Document Structure . . . . .	8
<b>2 Overall Description</b>	<b>9</b>
2.1 Product perspective . . . . .	9
2.1.1 Scenarios . . . . .	9
2.1.2 System Context . . . . .	10
2.1.3 Class diagrams . . . . .	11
2.1.4 System Architecture . . . . .	12
2.1.5 State diagrams . . . . .	13
2.2 Product Functions . . . . .	21
2.2.1 Core Functions . . . . .	21
2.3 User Characteristics . . . . .	23
2.3.1 Students . . . . .	23
2.3.2 Companies . . . . .	23
2.3.3 University Administrators . . . . .	24
2.4 Assumptions, Dependencies, and Constraints . . . . .	24
2.4.1 Domain Assumptions . . . . .	24

2.4.2	Dependencies . . . . .	25
2.4.3	Constraints . . . . .	25
<b>3</b>	<b>Specific Requirements</b>	<b>27</b>
3.1	External interface requirements . . . . .	27
3.1.1	User Interfaces . . . . .	27
3.1.2	Hardware Interfaces . . . . .	30
3.1.3	Software Interfaces . . . . .	30
3.1.4	Communication Interfaces . . . . .	31
3.2	Functional Requirements . . . . .	31
3.2.1	Requirements Lists . . . . .	31
3.2.2	Priority and Criticality Matrix . . . . .	34
3.2.3	Dependencies . . . . .	34
3.2.4	Use case diagrams . . . . .	36
3.2.5	Use cases . . . . .	37
3.2.6	Performance Requirements . . . . .	59
3.3	Design constraints . . . . .	60
3.3.1	Design Constraints . . . . .	60
3.3.2	Software System Attributes . . . . .	61
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>63</b>
4.0.1	Alloy Specification . . . . .	63
4.0.2	Alloy Analysis Results . . . . .	69
4.0.3	Base World for InternHub . . . . .	74
<b>5</b>	<b>Effort Spent</b>	<b>77</b>
<b>6</b>	<b>References</b>	<b>79</b>
6.1	References . . . . .	79
6.2	Used Tools . . . . .	79
<b>List of Figures</b>		<b>81</b>
<b>List of Tables</b>		<b>83</b>

# 1 | Introduction

Internships form a crucial bridge between academic learning and real-world professional experience. However, many existing tools struggle to efficiently connect students with suitable internship opportunities, resulting in mismatches and administrative burdens for universities and companies alike. **InternHub – Students & Companies (S&C)** addresses this gap by unifying the entire internship cycle—from finding the right match to ensuring quality, accountability, and continuous improvement—within a single, integrated platform.

This **Requirements Analysis and Specification Document (RASD)** outlines the platform’s technical and functional specifications, serving both as a development roadmap and a contractual reference for stakeholders. By employing advanced matching algorithms, providing robust feedback mechanisms, and streamlining internship workflows, the S&C platform ensures a seamless experience that benefits students, companies, and universities. Through this document, each party gains a clear understanding of the platform’s capabilities and the value it delivers, setting the foundation for more efficient and effective internships.

## 1.1. Purpose

This Requirements Analysis and Specification Document (RASD) provides a comprehensive overview of the InternHub - Students & Companies (S&C) platform. Its primary purpose is to serve as both a guide for developers responsible for implementing the system specifications and as a contractual reference point for clients and contractors. Additionally, it offers a clear, precise, and unambiguous explanation of the platform's features and limitations, empowering students, companies, and academic institutions to confirm that the system meets their needs and requirements. The S&C platform's overarching goal is to transform how university students connect with companies for internships. To this end, it focuses on:

1. Establishing an efficient system that matches students with suitable internship opportunities.
2. Streamlining the entire internship lifecycle—from application through completion—to simplify both student and company workflows.
3. Utilizing smart recommendation algorithms to align student skills with company requirements, ensuring more accurate and beneficial matches.
4. Providing robust monitoring and feedback tools to enhance transparency, accountability, and continuous improvement.
5. Ensuring effective complaint management and maintaining high-quality standards throughout the internship process.

By offering a seamless and impactful experience, the S&C platform aims to serve as a trusted solution that addresses the requirements of all stakeholders—students, businesses, and universities—thereby ensuring a more efficient, productive, and rewarding internship ecosystem.

### 1.1.1. Goals

Below is a table that lists all the goals of the S&C platform:

ID	Description
G1	Enable students to create detailed profiles, including their CVs, skills, academic achievements, and interests.
G2	Allow companies to post comprehensive internship opportunities, detailing roles, requirements, benefits, and timelines.

G3	Provide intelligent recommendations that align student skills and preferences with internship opportunities.
G4	Equip universities with tools to effectively monitor, manage, and track student internship progress and performance.
G5	Implement feedback and rating systems to promote accountability and continuous improvement for both students and companies.
G6	Facilitate seamless communication between students, companies, and universities for better collaboration and coordination.
G7	Integrate a secure document management system for handling internship-related paperwork, such as contracts and certificates.
G8	Offer analytics and reporting tools to provide insights into internship trends, success rates, and areas for improvement.
G9	Support multilingual functionality to ensure accessibility for a diverse user base across regions.
G10	Implement a grievance redressal mechanism to resolve disputes and ensure fair treatment for all users.
G11	Provide training modules or resources to prepare students for internships, such as interview tips and skill-building exercises.

Table 1.1: Goals table.

## 1.2. Scope

### 1.2.1. World Phenomena

**World Phenomena:** The project addresses core operational areas that shape the interactions and processes among students, companies, and universities. These include:

- Student Internship Process:** Students search for and apply to internships, creating personal profiles and uploading their CVs.
  - ID: WP1 Students create profiles and upload their CVs.
  - ID: WP4 Students apply for internships.
- Company Internship Management:** Companies post internship opportunities, define requirements, and review applications to select suitable candidates.

- ID: WP2 Companies post internship opportunities.
  - ID: WP5 Companies review applications and select candidates.
3. **University Oversight:** Universities maintain a supervisory role, monitoring and managing internship activities to ensure quality and compliance.
- ID: WP3 Universities monitor internship progress.
4. **Interview Coordination and Selection:** The platform facilitates the scheduling and management of interviews and other selection procedures to ensure a smooth hiring process.
5. **Feedback and Quality Assurance:** Feedback is collected from students, companies, and universities to ensure continuous improvement, enhanced user experience, and adherence to quality standards.
- ID: WP6 Feedback is collected from students and companies.
6. **Complaint Handling:** A robust complaint-handling system ensures issues are addressed efficiently, maintaining trust and transparency in all interactions.
7. **Fair and Transparent Interactions:** The system is designed to foster an environment of fairness, transparency, and accountability among students, companies, and universities.

### 1.2.2. Shared phenomena

**Shared Phenomena:** The platform provides a suite of shared functionalities, ensuring seamless interaction and data exchange among all stakeholders (Students, Companies, and Universities):

#### 1. User Account Management and Profiles:

- **ID:** SP1 Students create accounts on the platform (Controller: Student, Observer: Platform).
- **ID:** SP2 Companies create accounts on the platform (Controller: Company, Observer: Platform).
- **ID:** SP3 Universities create accounts on the platform (Controller: University, Observer: Platform).

2. Students maintain detailed profiles, including CVs, skills, and achievements, while companies and universities manage their respective institutional profiles.

3. **Internship Postings and Applications:** Students can apply directly to posted internships, and companies can review and manage these applications.

- **ID:** SP4 Students apply for internships (Controller: Student, Observer: Platform).
- **ID:** SP5 Companies review and manage applications (Controller: Company, Observer: Platform).
- **ID:** SP6 Universities track student applications (Controller: University, Observer: Platform).

4. **Feedback and Rating System:** A comprehensive feedback mechanism enables stakeholders to exchange feedback, ratings, and reviews to ensure accountability and continuous improvement.

- **ID:** SP7 Feedback is exchanged between stakeholders (Controller: All, Observer: Platform).

5. **Interview Scheduling and Notifications:** Interviews are coordinated efficiently, with automated reminders and updates for both students and companies.

6. **Communication and Support Tools:** The platform offers seamless communication channels for queries, updates, and issue resolution among students, companies, and universities.

7. **Document Management:** Secure storage and sharing capabilities for documents, such as internship agreements, certificates, and other relevant files, ensure easy access and proper record-keeping.
8. **Analytics and Insights:** Real-time analytics and reporting tools help all parties make informed decisions, monitor internship progress, and assess performance metrics.
9. **Multilingual Support:** The platform supports multiple languages, accommodating a diverse global user base.
10. **Training and Preparation Resources:** Students have access to resources like training materials and interview preparation tools, aiding them in securing and succeeding in internships.

### 1.3. Definition, Acronyms, Abbreviations

Term/Acronym	Definition
S&C	Students & Companies Platform
RASD	Requirements Analysis & Specification Document
CV	Curriculum Vitae
UI	User Interface
API	Application Programming Interface
DBMS	Database Management System
SLA	Service Level Agreement
GDPR	General Data Protection Regulation

Table 1.2: Acronyms and terms used in the document.

## 1.4. Revision History

Version	Date	Description	Authors
0.1	8 December 2024	Initial Release	Shreesh Kumar Jha, Samarth Bhatia, Satvik Sharma
1.0	17 December 2024	Structure Fix and Added Use Cases	Shreesh Kumar Jha, Samarth Bhatia
2.0	19 December 2024	Alloy Modelling	Shreesh Kumar Jha, Samarth Bhatia
3.0	21 December 2024	Final Version	Shreesh Kumar Jha, Samarth Bhatia Satvik Sharma

Table 1.3: Revision History

## 1.5. Reference Documents

- Reference to Previous Year Student Projects for Structuring the Document
- Specification Document Assignment
- IEEE Standard Documentation For RASD

## 1.6. Document Structure

As shown below, the document is organized into six sections, each with a distinct focus:

**Introduction:** The project's goals, purpose, and a succinct analysis of common and worldwide occurrences are presented in the introduction, containing acronyms and definitions to help you grasp the problem domain.

**Overall Description:** Provides a thorough rundown of the issue, potential domains, and features of the product. explains limitations, dependencies, and assumptions as well.

**Specific Requirements:** Provides a detailed description of both functional and non-functional needs, including those pertaining to external interfaces.

**Formal Analysis Using Alloy:** Provides assertions and checks to validate the model outlined in previous parts.

**Effort Spent:** Describes how each team member contributed to the writing of this paper.

**References:** Provides a list of all the supplementary materials and references that were utilized to produce the document.

# 2 | Overall Description

## 2.1. Product perspective

### 2.1.1. Scenarios

**Scenario 1: A student registers** A university student named **Amelia Young** chooses to work for S&C while searching for an internship. After accessing the platform's home page, she selects the "**Sign-Up**" option and inputs her email address, name, and a strong password. She gets a confirmation email after completing the form. Amelia activates her account by clicking on the confirmation link. After finishing, she logs on to look for internship possibilities.

**Scenario 2: A Company posts an internship** Interns are needed for the data analytics team at **InnovateCorp**, a mid-sized software startup. After logging in to S&C, an HR representative from the organization fills out the "**Post Internship**" part, which includes the job title, description, length, and needed qualifications. They also establish a deadline for applications. After submission, S&C gives InnovateCorp a confirmation email and contacts the appropriate students based on their profiles.

**Scenario 3: A University administrator monitors internships** **Greenfield University** academic coordinator **Dr. Olivia Cruz** uses S&C to monitor the development of her students' internships. She goes to the "**Internship Monitoring**" section, where she may see comprehensive reports, examine student and company comments, and respond to complaints made by either side.

**Scenario 4: A Student applies for an internship** **Liam Chen**, a computer science major, looks for software development internships on S&C. He discovers a position at InnovateCorp that suits him after using filters to refine the results. After reading the job description, Liam selects "**Apply Now**" and sends in his resume. On his dashboard, the application status is updated.

**Scenario 5: A Company reviews applications** Notifications of new applications are sent to InnovateCorp's HR department. They access Liam's profile and resume by logging in to the "Application Management" area. Following screening, Liam is placed on their shortlist, and his dashboard is automatically updated with the interview time.

### 2.1.2. System Context

The Students & Companies (S&C) platform is a dynamic web-based solution designed to streamline the internship process by bridging the gap between students, companies, and universities. The platform caters to three primary user groups, each with distinct roles and requirements:

1. **University Students:** Students utilize the platform to search for internship opportunities, create detailed profiles with CVs and skills, and track their application progress.
2. **Companies:** Organizations post internship positions, outline role requirements, and select suitable candidates based on student profiles and recommendations.
3. **University Administrators:** Universities leverage the platform to oversee the entire internship process, monitor student performance, and ensure alignment with academic standards.
4. **Central Platform:** The S&C platform serves as a centralized hub, fostering efficient communication, reducing administrative overhead, and ensuring a seamless experience for all stakeholders involved.

### 2.1.3. Class diagrams

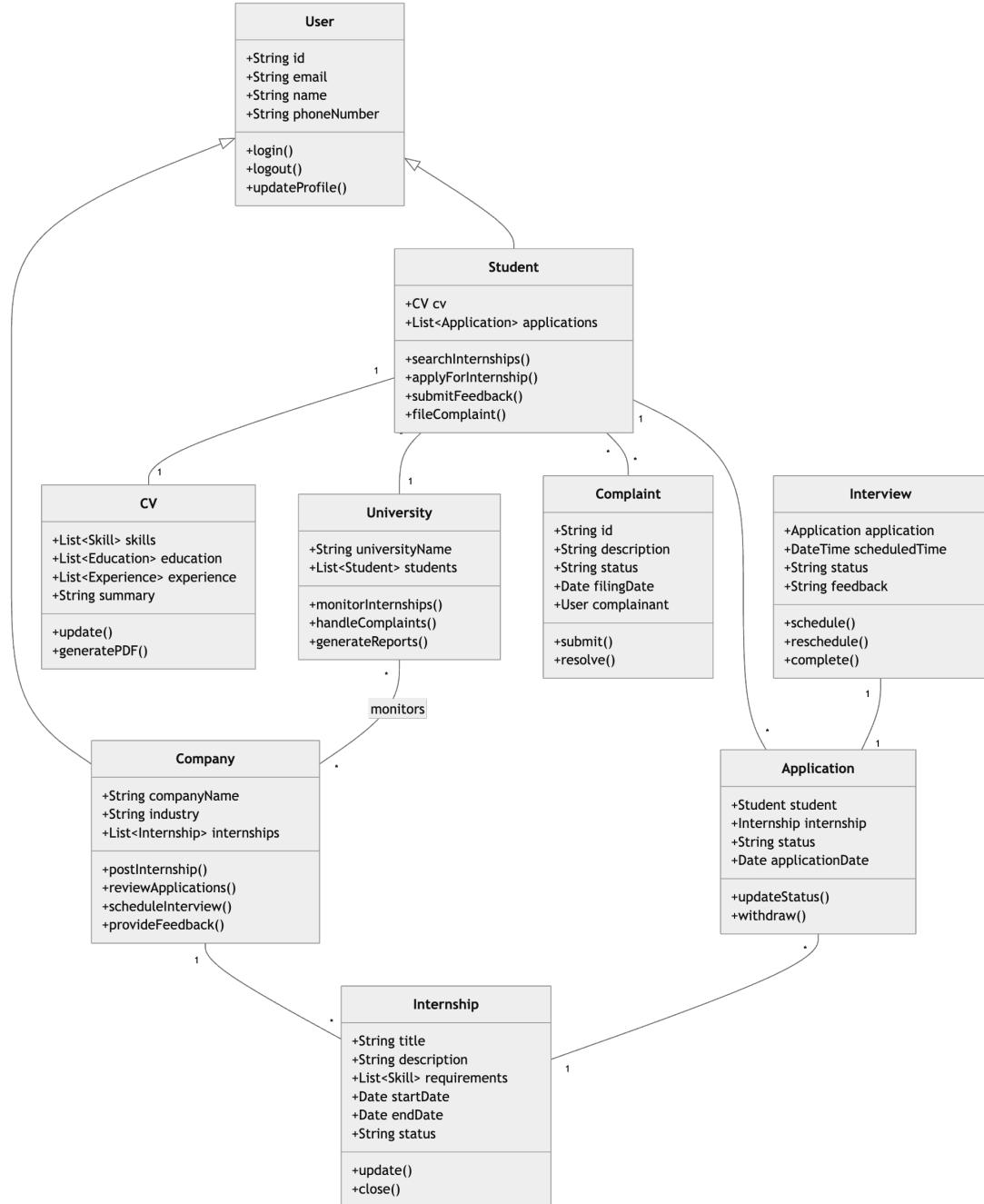


Figure 2.1: Class Diagram for InternHub

The system's core is the **User** class, which contains shared methods like **login()**, **logout()**, and **updateProfile()** as well as common attributes like id, email, name, and phoneNumber. The User class is extended by specialized classes like Student, University, and Company, each of which adds unique characteristics and features. Students can

look for internships, apply, provide feedback, and file complaints using the CV and application list included in the Student class. In the meantime, the University class uses functions like **monitorInternships()** and **generateReports()** to handle complaints, internships, and students. Employer-specific tasks like advertising internships, evaluating applications, setting up interviews, and giving feedback are made easier by the Company class.

Auxiliary classes such as Internship, which represents internship details, and Application, which links students to internships and tracks their status, support these core entities. The CV class creates professional resumes by combining a student's education, experience, and skills. To ensure smooth operations, the Complaint and Interview classes oversee the complaint lifecycle and interview procedures, respectively. Each class plays a unique role in the platform's workflow, ensuring scalability and promoting effective communication between businesses, universities, and students.

#### 2.1.4. System Architecture

The platform consists of:

The following elements make up the modular architecture of the S&C platform:

##### 1. Front-End Layer:

- **Student Portal:** A place where students can track applications, maintain profiles, and search for internships.
- **Company Portal:** Resources for employers to advertise openings, evaluate applicants, and provide feedback.
- **University Administrator Portal:** An interface enabling colleges to manage reports and monitor internships.
- **Public Interface:** Allows users to explore internships and learn more about the platform.

##### 2. Services for the Backend:

- Suggestion for a User Management Service.
- Matching Engine.
- Recommendation System.
- Interview Management Service.

- External Integrations for the Service Analytics Engine.

### 3. Email Support:

- Authentication Service for Document Storage Systems.
- Processing payments for compensated internships.
- Authentication Service.
- Payment Processing (for paid internships).

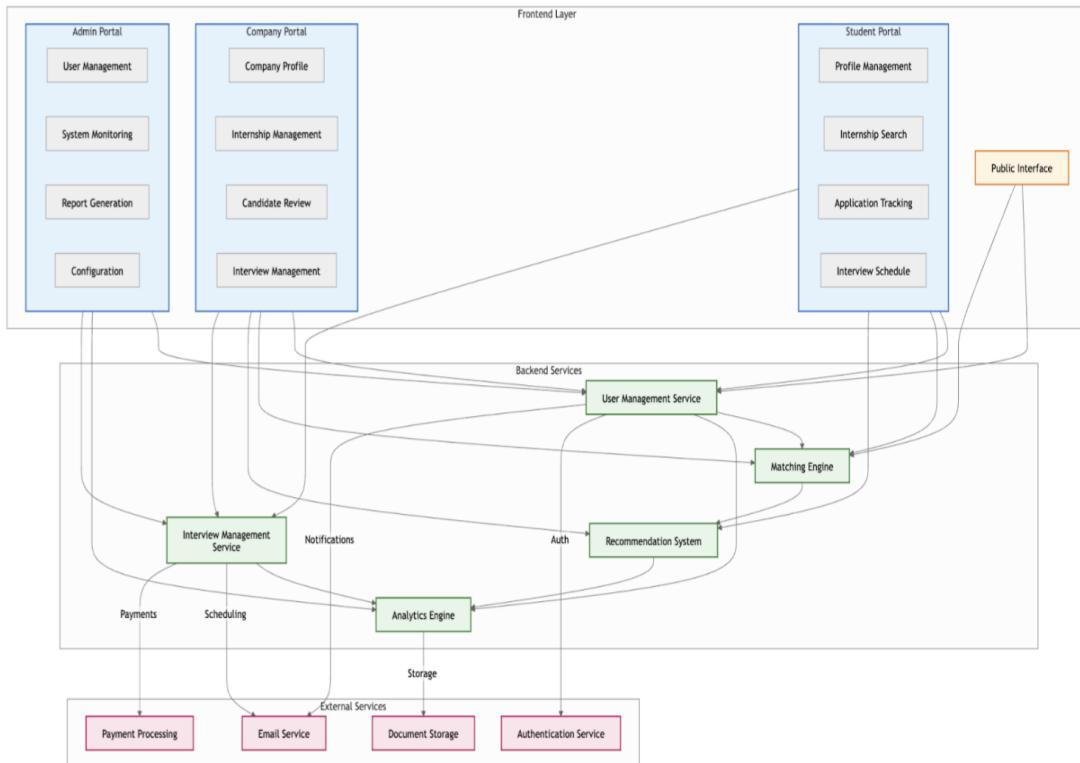


Figure 2.2: System Architecture Diagram for InternHub

#### 2.1.5. State diagrams

The State Diagrams of the S&C system, which depict every action a user could take, are shown in this section.

**SignUp:** When a user wants to register on S&C, they must fill out a registration form with their email address, password, name, and last name. S&C will send the user a verification email if the credentials they have provided are accepted (that is, if the password satisfies security requirements and the email address is not already in use). The new account is successfully created after the user confirms their registration using the email

link that was provided. S&C displays an error notice to the user and reroutes them to the signup form page if the credentials are incorrect.

Additionally, S&C will ask for more company-specific details including the company name, industry, and size if the user chooses the "Register as a Company" option during the registration process. To construct a comprehensive company profile, these details are necessary.

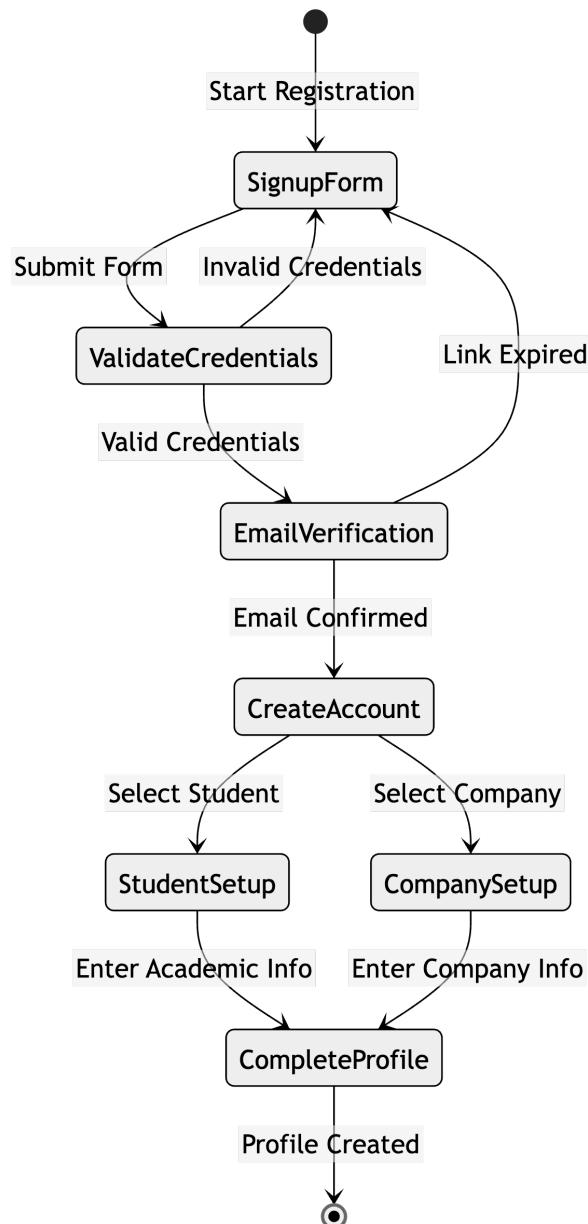


Figure 2.3: Signup state diagram.

**Login:** A registered user must fill out a login form using their email address and password in order to access their S&C account. S&C presents the User's dashboard, which highlights pertinent internships and applications according to their function, if the credentials supplied are correct and correspond to those of a registered User in the S&C database. S&C returns the user to the login form page after displaying an error notice if the credentials entered are incorrect.

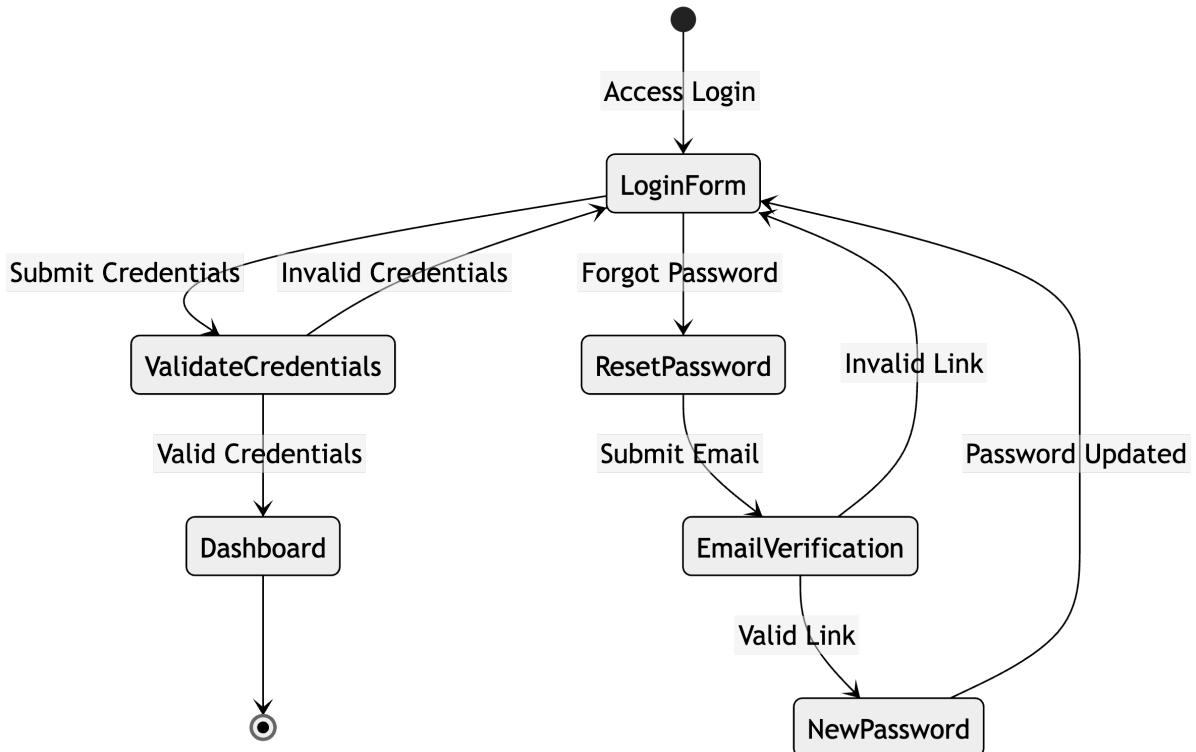


Figure 2.4: Login state diagram.

**Post Internship:** Companies must include a number of parameters in the create internship form when they plan to post a new internship on S&C. The job title, description, requirements, length of service, and any other information, such pay or perks, are examples of these factors. S&C shows the Company an error message and reroutes them back to the creation form if the system determines that any of these parameters are inappropriate. On the other hand, S&C creates the internship posting if every parameter satisfies the system's requirements. Notifications are subsequently delivered to matching students based on their profiles and preferences after the newly produced posting is published to the Company's dashboard.

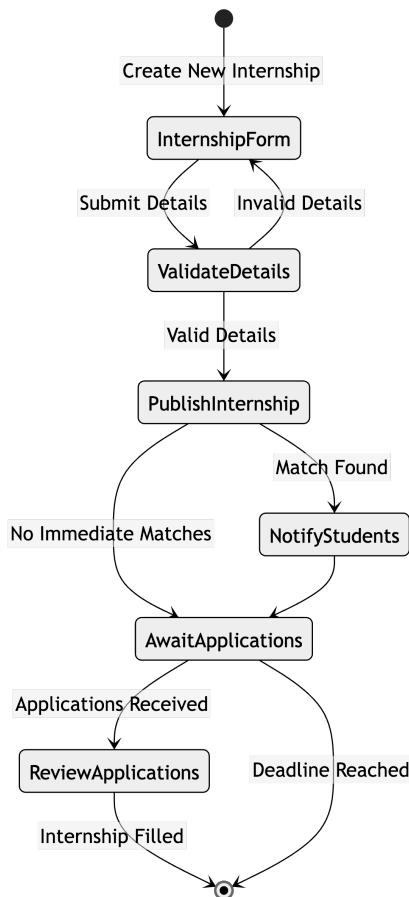


Figure 2.5: State Diagram for Post Internship

**Apply for Internship:** Students can use the platform to apply for internships when they locate one that interests them. The application form, which could contain extra questions unique to the role, must be filled out by the student. S&C confirms that the student satisfies the prerequisites and that the application is complete. The application is sent to the business if it is accepted, and the student can monitor its progress via their

dashboard. S&C lets the student finish the application after displaying an error message if any necessary information is lacking.

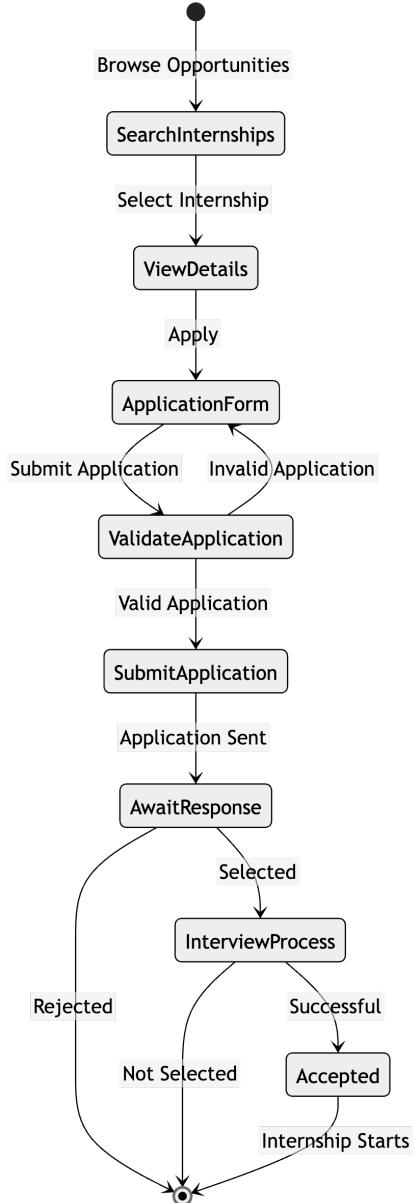


Figure 2.6: State Diagram for Applying for an Internship

**Interview Management:** A business can use S&C to start the interview process after reviewing applications and choosing applicants for interviews. The candidate is informed of the time slots that are available once the organization makes a selection. Other times can be suggested or accepted by the student. Notifications and calendar invitations are sent to both parties after confirmation. Following the interview, the business can document the results and move on with its choice.

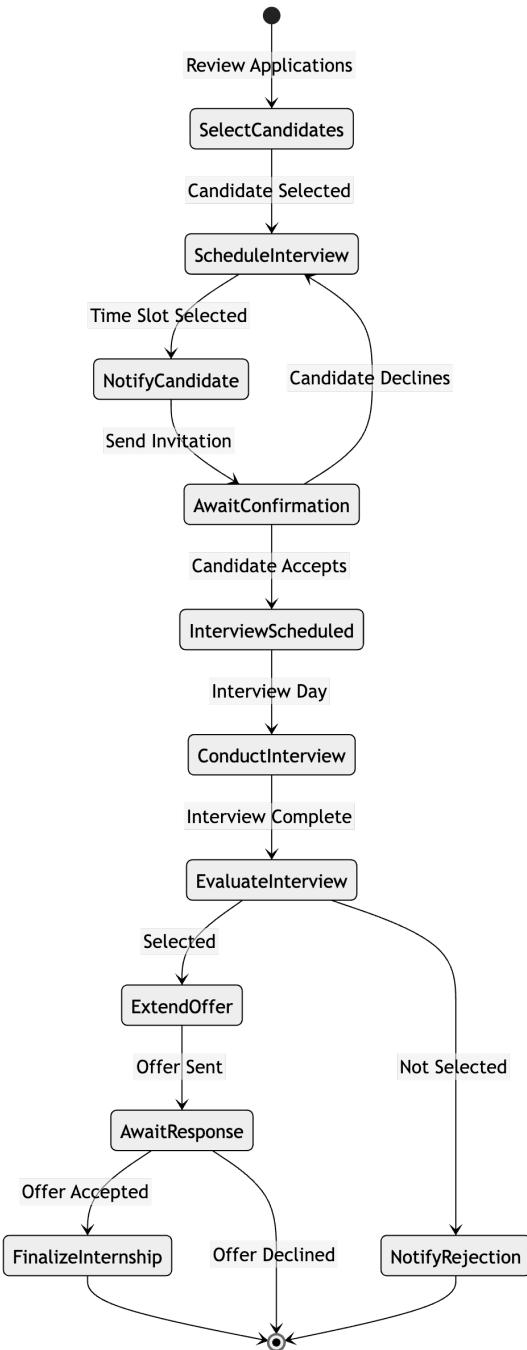


Figure 2.7: State Diagram for Interview Management

**Complaint Handling:** Through S&C, any party that experiences problems during the internship process can submit a complaint. A description of the problem and any pertinent documentation must be included in the complaint form. The relevant university official receives the complaint once it is filed. The university can look into the matter, get in touch with those involved, and try to find a solution. Every conversation and choice is recorded in the system.

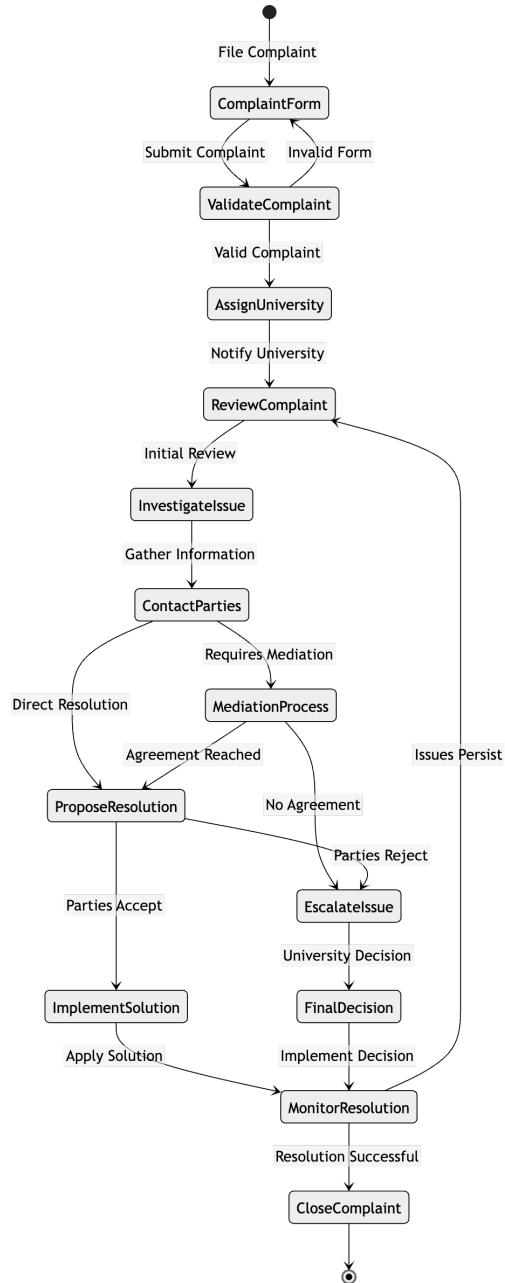


Figure 2.8: State Diagram for Complaint Handling

**Profile Management:** Anytime they choose, users can view and edit their profiles. For students, this entails revising their resume, preferences, and talents. For businesses, this entails revising internship requirements and company information. S&C verifies all modifications before they are saved. Additionally, in response to changes in profiles, the algorithm immediately updates matching recommendations.

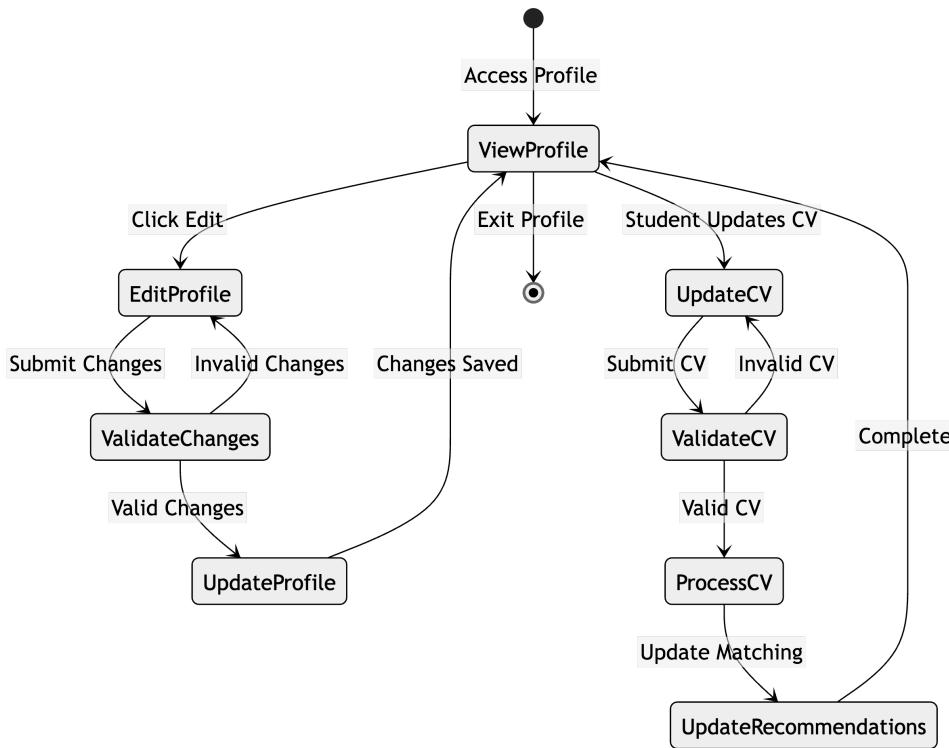


Figure 2.9: State Diagram for Profile Management

**Search and Filter:** The platform's search feature allows users to look for pertinent information. Students can use industry, geography, keywords, and other parameters to find internships. Employers can look up candidate profiles by experience, education, and skill set. Additional filters can be used to further refine search results, and users can remember their search choices for later use.

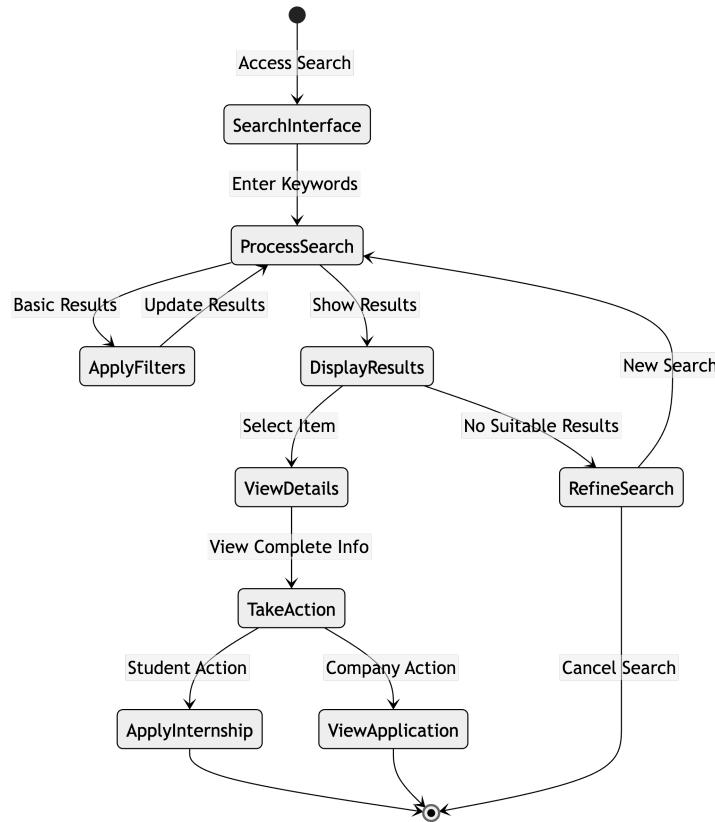


Figure 2.10: State Diagram for Search and Filter

## 2.2. Product Functions

### 2.2.1. Core Functions

#### For Students

- **Manage Profiles and CVs:** Students can create and update their profiles, as well as submit resumes for potential employers to view.
- **Search and Filter Internships:** Students can use search and filtering options to find internships based on interests, location, and duration.
- **Monitor Applications:** Students can track the status of their internship applications and receive notifications about any changes or decisions.
- **Schedule Interviews:** Depending on company availability, students can select interview slots for internship positions.
- **Submit Feedback:** After completing an internship, students can provide feedback and comments through the platform.

- **File Complaints:** Students can directly lodge issues or complaints related to internships via the portal.

## Management of Company Profiles

- **Manage Company Profiles:** Companies can create, update, and maintain profiles containing organizational details and contact information.
- **Post Internships:** Companies can list internship vacancies with detailed requirements and descriptions.
- **Screen Candidates:** Employers can review applications and filter candidates based on their backgrounds, profiles, and other criteria.
- **Conduct Interviews:** Employers can schedule and conduct interviews with short-listed candidates directly on the platform.
- **Gather Feedback:** Companies can give feedback on candidates after interviews or upon internship completion.
- **Monitor Performance:** During an intern's tenure, employers can track the intern's progress and performance.

## University Performance

- **Observe Student Activity:** Universities can monitor students' profiles, applications, and internship progress.
- **Oversee Internships:** Institutions can view details of ongoing internships and assess how they align with university objectives.
- **Handle Complaints:** Universities are authorized to review and resolve internship-related complaints raised by students or companies.
- **Generate Reports:** Institutions can produce reports on internship data, student achievements, and employer feedback.
- **Ensure Quality Assurance:** Universities can evaluate the quality of internships to ensure they meet institutional standards.

## 2.3. User Characteristics

### 2.3.1. Students

- **Demographics:** University students from diverse academic backgrounds and skill levels.
- **Motivations:** Actively seeking internships to enhance their academic learning, gain practical experience, and develop their careers.
- **Technical Proficiency:** Basic computer literacy, with varying levels of technical expertise depending on their academic discipline.
- **Access Requirements:** Use web browsers and mobile devices for seamless access to the platform.
- **Engagement Needs:** Require a user-friendly interface with clear guidance for application processes, personalized internship recommendations, and progress tracking.

### 2.3.2. Companies

- **Demographics:** Organizations ranging from startups to multinational corporations across various industries.
- **Key Users:**
  - **HR Personnel:** Responsible for posting internships, managing candidate applications, and coordinating interviews.
  - **Department Managers:** Assess technical and domain-specific skills of applicants.
  - **Technical and Non-Technical Staff:** Engage with interns for mentorship and project collaboration.
- **User Roles:** Flexible role-based access for different functionalities, such as job posting, candidate review, and administrative controls.
- **Engagement Needs:** Require tools for streamlining recruitment processes, accessing applicant analytics, and maintaining compliance with university policies.

### 2.3.3. University Administrators

- **Key Roles:**
  - **Academic Coordinators:** Oversee the alignment of internships with educational goals and curriculum requirements.
  - **Internship Program Managers:** Monitor internship program effectiveness, gather feedback, and ensure compliance with regulations.
  - **Student Advisors:** Guide students on selecting suitable internships and navigating application processes.
  - **System Administrators:** Maintain the platform, manage user accounts, and ensure data security.
- **Engagement Needs:** Require detailed dashboards for tracking student participation, company partnerships, and program outcomes. Need tools for generating reports and managing communications between stakeholders.

## 2.4. Assumptions, Dependencies, and Constraints

### 2.4.1. Domain Assumptions

ID	Description
DA1	Users have reliable internet access.
DA2	Students maintain updated CVs.
DA3	Companies provide accurate information about internships.
DA4	Universities actively monitor students' progress on S&C.
DA5	All users comply with the platform's terms and policies.

Table 2.1: Domain Assumptions.

### 2.4.2. Dependencies

ID	Description
D1	Reliable web hosting services.
D2	Functional email delivery system.
D3	Secure database system.
D4	File storage system for documents.
D5	Authentication services to manage user logins.

Table 2.2: Dependencies.

### 2.4.3. Constraints

ID	Description
C1	Must comply with GDPR and data protection regulations.
C2	Adhere to university-specific internship regulations.
C3	Ensure system performance under high traffic.
C4	Provide compatibility across major web browsers.
C5	Implement robust security standards for data handling.

Table 2.3: Constraints.



# 3 | Specific Requirements

## 3.1. External interface requirements

### 3.1.1. User Interfaces

Students, businesses, and university administrators will be able to access the InternHub - Students & Companies (S&C) platform's **user interface** via a **responsive web application** on any device with a current web browser and an internet connection.

#### Authentication Interface

1. User login and registration.
2. Password reset functionality.
3. Multi-factor authentication (optional).

The mockup displays two mobile-style screens side-by-side. The left screen is titled 'Welcome Back' and has a sub-instruction 'Sign in to your S&C account'. It features two buttons at the top: 'Student' (blue with white icon) and 'Company' (black with white icon). Below these are fields for 'Email' (placeholder 'Enter your email') and 'Password' (placeholder 'Enter your password'). There is a 'Remember me' checkbox and a 'Forgot password?' link. A large blue 'Sign In' button is at the bottom. At the very bottom, a link says 'Don't have an account? Sign up'. The right screen is titled 'Create Account' and has a sub-instruction 'Join S&C to start your journey'. It also has 'Student' and 'Company' buttons. Below them are fields for 'First Name' (placeholder 'Enter first name') and 'Last Name' (placeholder 'Enter last name'). There is a 'University' field (placeholder 'Enter your university'), an 'Email' field (placeholder 'Enter your email'), and a 'Password' field (placeholder 'Create password'). A large blue 'Create Account' button is at the bottom. At the very bottom, a link says 'Already have an account? Sign in'.

Figure 3.2: SignUp mockup.

Figure 3.1: Login mockup.

## Student Interface

1. **Dashboard Overview:** Shows notifications, recommended internships, and active applications.
2. **Profile Management:** Add or update qualifications and edit personal information.
3. **CV Builder:** Use built-in templates to create and manage resumes.
4. **Internship Search:** Look for and select internships according to talents, domain, and location.
5. **Application Tracker:** Check the progress of applications that have been submitted.
6. **Interview Calendar:** Organize interview dates with the calendar feature.
7. **Messaging System:** Interact with administrators and businesses.

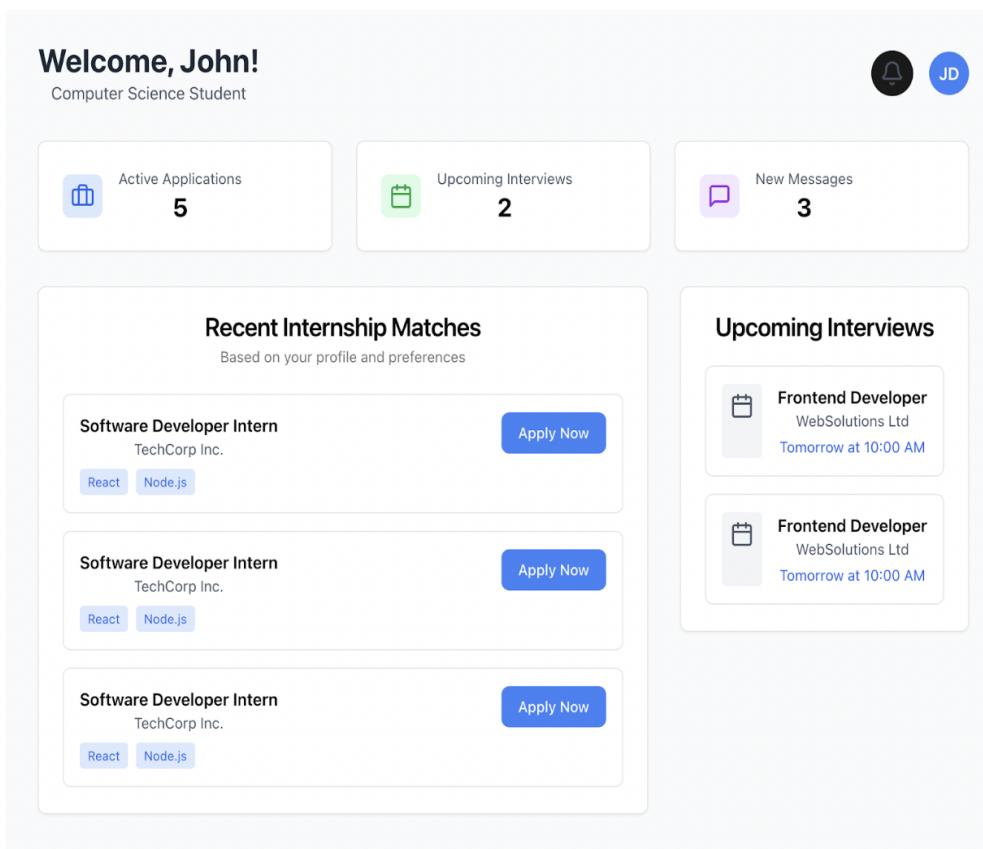


Figure 3.3: Student Interface Diagram

## Company Interface

1. **Dashboard Overview:** Displays applications, postings, and other data.
2. **Management of Company Profiles:** Modify logos and company information.
3. **Management of Internship Postings:** Produce and oversee internship advertisements.
4. **Managing Candidates:** Examine applications and create a shortlist of applicants.
5. **Interview Scheduler:** Use a scheduling tool to plan and monitor interviews.
6. **Analytics Dashboard:** Access information on posting engagement and candidate performance.

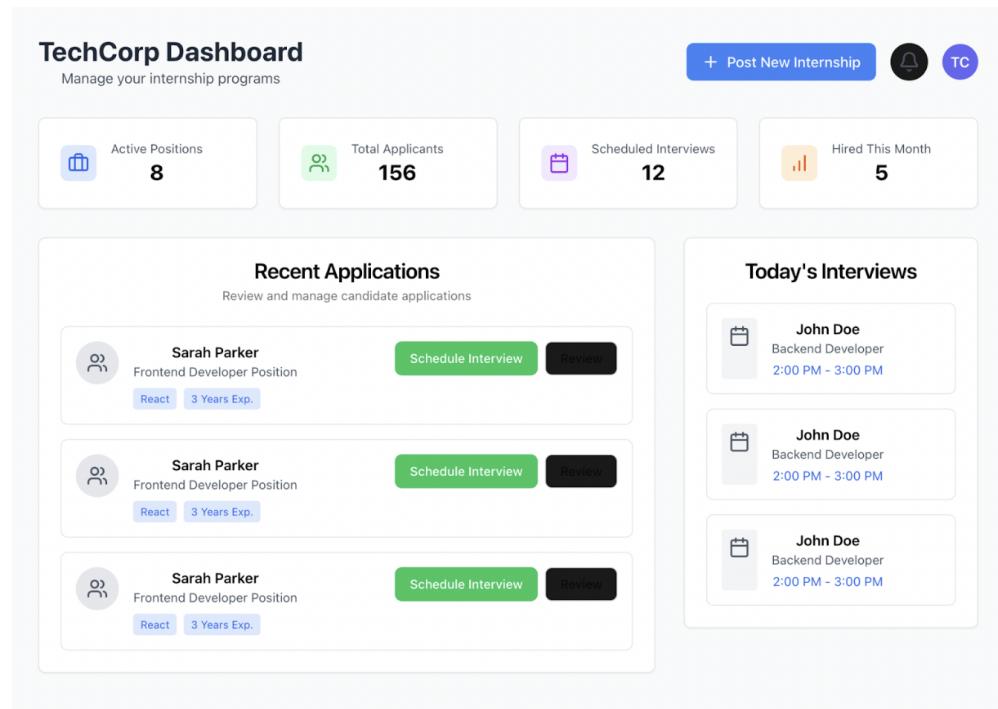


Figure 3.4: Company Interface Diagram

## Administrator Interface

1. **System Overview:** Keep an eye on the status and activities of the platform.
2. **User Management:** Oversee users, including administrators, businesses, and students.
3. **Handling Complaints:** Examine and address complaints that have been submitted.

4. **Report Creation:** Produce reports on internship progress and platform usage.
5. **Configuration Settings:** Modify settings for the entire system.

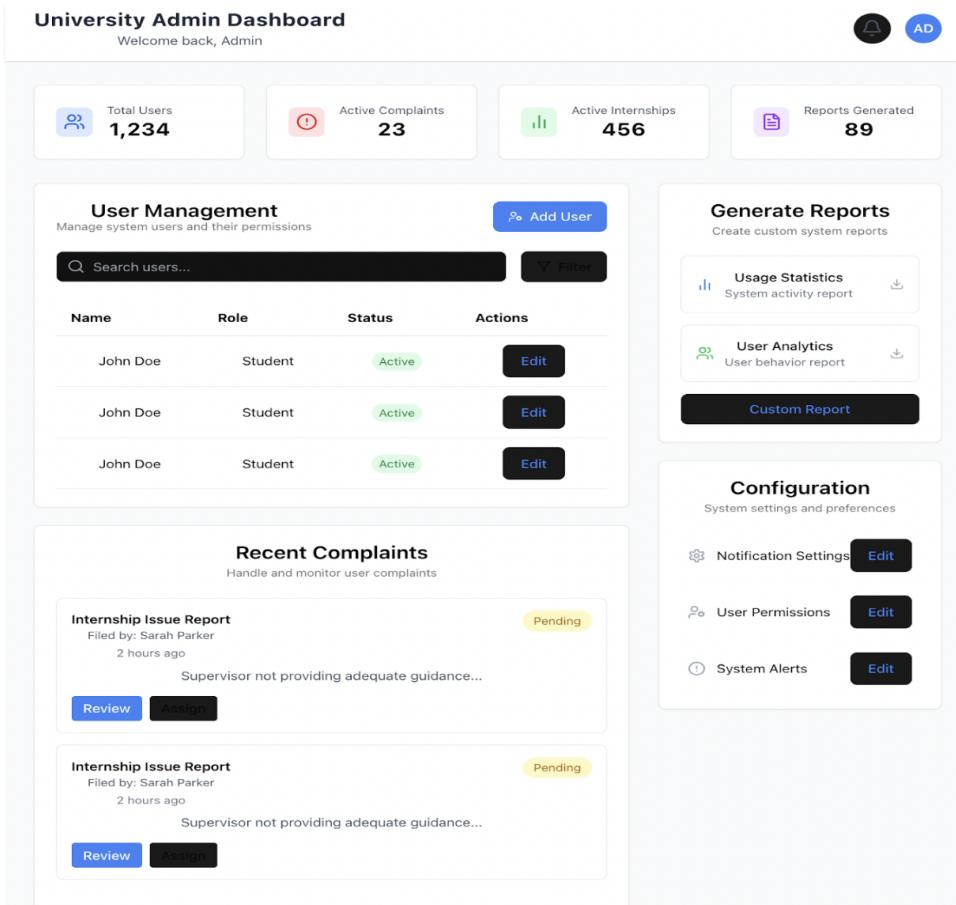


Figure 3.5: Administrator Interface Diagram

### 3.1.2. Hardware Interfaces

- Compatible with both desktop and mobile web browsers available today.
- Minimum screen resolution: 1280x720.
- Supports mobile devices with touch interfaces.
- Features for uploading and downloading resumes, job descriptions, and feedback files.

### 3.1.3. Software Interfaces

- **Database Management System:** Stores user information, internships, and system logs.

- **Email Server:** Provides emails for confirmation and notifications.
- **File Storage System:** Manages uploaded documents, including resumes.
- **Authentication System:** Manages role-based access and user login.
- **Analytics Engine:** Produces usage reports and offers insights.

### 3.1.4. Communication Interfaces

- **HTTPS Protocol:** Ensures secure communication between users and the platform.
- **RESTful API:** Enables seamless communication between the front-end and back-end systems.
- **WebSocket Connections:** Supports real-time notifications for updates and alerts.
- **Email SMTP:** Manages email correspondence for notifications and alerts.

## 3.2. Functional Requirements

### 3.2.1. Requirements Lists

#### F1. User Management

ID	Description
F1.1	The system shall allow students, companies, and university administrators to register with verified email addresses.
F1.2	The system shall provide secure authentication with optional two-factor verification.
F1.3	The system shall allow users to update their profile information including contact details and preferences.
F1.4	The system shall enforce role-based access control for students, companies, and administrators.
F1.5	The system shall support password reset functionality with email verification.
F1.6	The system shall maintain audit logs of all user authentication activities.
F1.7	The system shall allow users to manage notification preferences.
F1.8	The system shall enforce strong password policies.

## F2. CV Management

ID	Description
F2.1	The system shall provide customizable CV templates for students.
F2.2	The system shall allow students to create and store multiple versions of their CVs.
F2.3	The system shall enable students to update their CVs at any time.
F2.4	The system shall allow students to control CV visibility to specific companies.
F2.5	The system shall provide a skill management interface for students.
F2.6	The system shall validate skill entries against a standardized skill database.
F2.7	The system shall support document upload for certificates and portfolios.
F2.8	The system shall track CV view statistics for students.

## F3. Internship Management

ID	Description
F3.1	The system shall allow companies to create detailed internship postings.
F3.2	The system shall provide an application tracking system for companies.
F3.3	The system shall automatically notify students of application status changes.
F3.4	The system shall provide advanced search and filtering capabilities.
F3.5	The system shall allow companies to set application deadlines.
F3.6	The system shall enable bulk application processing for companies.
F3.7	The system shall support multiple rounds of application review.
F3.8	The system shall maintain a history of all internship postings.

## F4. Interview Management

ID	Description
F4.1	The system shall provide a calendar interface for interview scheduling.
F4.2	The system shall track interview status and progress.
F4.3	The system shall collect structured feedback from both parties.
F4.4	The system shall send automated interview reminders.

F4.5	The system shall support virtual interview link generation.
F4.6	The system shall allow rescheduling with mutual agreement.
F4.7	The system shall maintain interview history.
F4.8	The system shall support multiple interview rounds.

## F5. Recommendation System

ID	Description
F5.1	The system shall match students with internships based on skills alignment.
F5.2	The system shall suggest qualified candidates to companies.
F5.3	The system shall learn from user interactions to improve recommendations.
F5.4	The system shall generate personalized internship suggestions.
F5.5	The system shall consider location preferences in matching.
F5.6	The system shall factor in past application success patterns.
F5.7	The system shall update recommendations in real-time.
F5.8	The system shall explain recommendation reasoning to users.

## F6. Complaint Handling

ID	Description
F6.1	The system shall provide a structured complaint submission interface.
F6.2	The system shall route complaints to appropriate university administrators.
F6.3	The system shall track resolution progress.
F6.4	The system shall support appeals process.
F6.5	The system shall maintain complete complaint history.
F6.6	The system shall enable communication between parties.
F6.7	The system shall generate complaint resolution reports.
F6.8	The system shall enforce resolution timeframes.

### 3.2.2. Priority and Criticality Matrix

Requirement Category	Priority	Implementation Phase	Criticality
User Management	High	Phase 1	Critical
CV Management	High	Phase 1	Critical
Internship Management	High	Phase 1	Critical
Interview Management	Medium	Phase 2	Important
Recommendation System	Medium	Phase 2	Important
Complaint Handling	Low	Phase 3	Standard

Table 3.7: Priority and Criticality Matrix

### 3.2.3. Dependencies

#### F1 Dependencies

- **F1.1:** Must be completed before any other functionality.
- **F1.2:** Is required for all secure operations.
- **F1.4:** Impacts all other functional areas.

#### F2 Dependencies

- Requires F1 completion.
- **F2.1:** Must be completed before F2.2.
- **F2.5:** Is required for F5.1.

#### F3 Dependencies

- Requires F1 completion.
- **F3.1:** Must be completed before F3.2.
- **F3.4:** Is required for F5.4.

#### F4 Dependencies

- Requires F1 and F3 completion.
- **F4.1:** Must be completed before F4.2.

- **F4.3:** Feeds into F5.3.

## F5 Dependencies

- Requires F2 and F3 completion.
- **F5.1:** Must be completed before F5.4.
- Requires continuous data from F4.3.

## F6 Dependencies

- Requires all other systems to be operational.
- **F6.1:** Must be completed before F6.2.
- **F6.4:** Requires F6.3 completion.

### 3.2.4. Use case diagrams

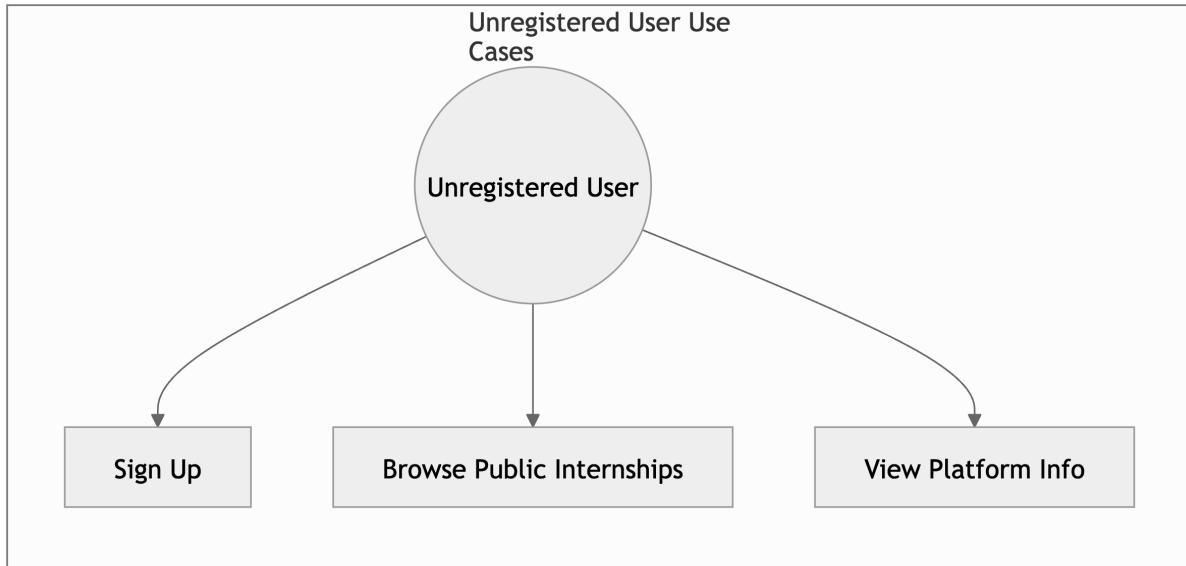


Figure 3.6: Use Cases Diagram for Unregistered Users.

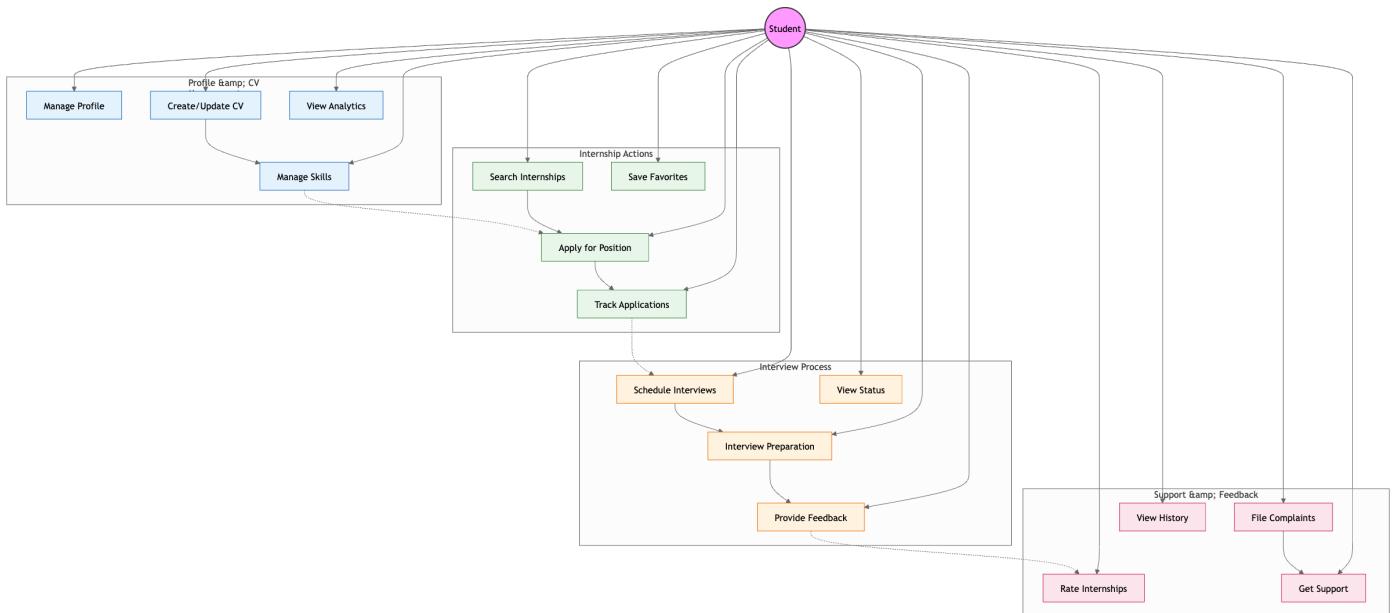


Figure 3.7: Use Cases Diagram for Students

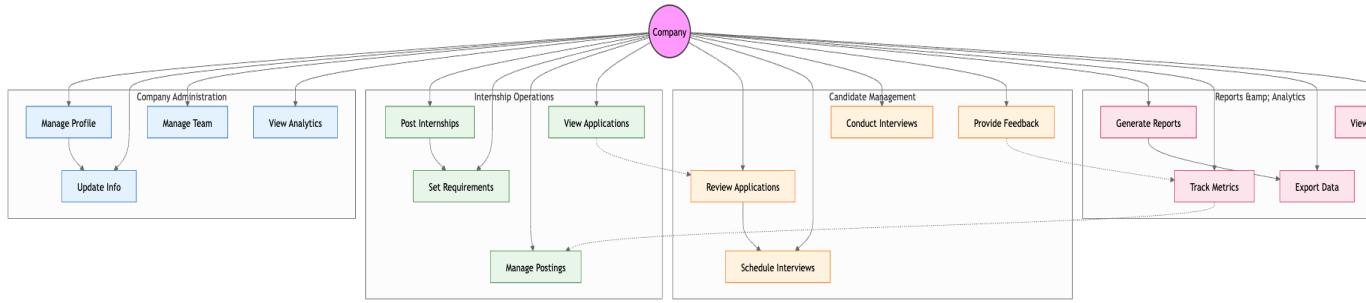


Figure 3.8: Use Cases Diagram for Companies.

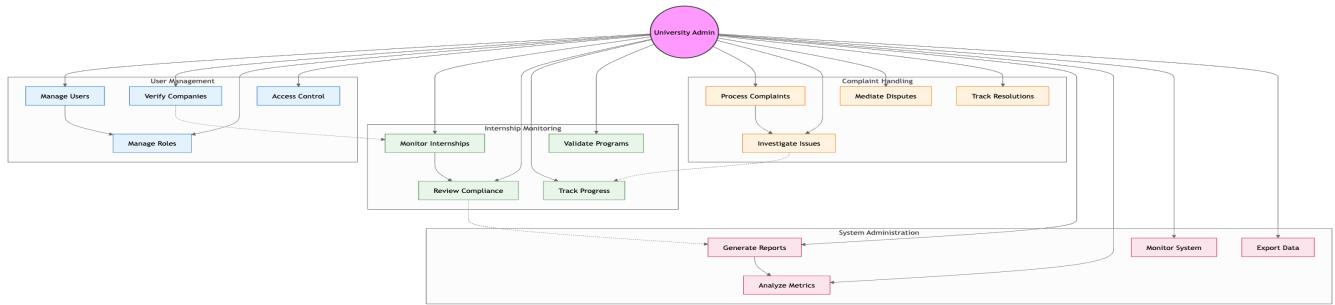


Figure 3.9: Use Cases Diagram for University Admin.

### 3.2.5. Use cases

The primary identified use cases are described and illustrated in this section. Each of them has a table with **entry conditions**, **event row**, **exit conditions**, and **exceptions**, as well as a sequence diagram that displays the messages sent back and forth between the called functions and the entities.

#### UC1. Student Registration

<b>Actor</b>	Company Representative, Email Provider
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Company not registered in S&amp;C</li> <li>• Representative has company email domain</li> <li>• Company meets platform requirements</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. S&amp;C shows the login form with registration option</li> <li>2. Representative clicks on "Create Account"</li> <li>3. S&amp;C displays registration form</li> <li>4. Representative selects "Register as Company"</li> <li>5. Representative enters initial information: <ul style="list-style-type: none"> <li>• Company Name</li> <li>• Company Website</li> <li>• Industry Type</li> <li>• Company Size</li> <li>• Company Email Domain</li> <li>• Representative Name</li> <li>• Representative Position</li> <li>• Password</li> </ul> </li> <li>6. S&amp;C validates company information</li> <li>7. S&amp;C performs preliminary company verification: <ul style="list-style-type: none"> <li>• Website domain matches email domain</li> <li>• Company exists in business registry (if applicable)</li> </ul> </li> <li>8. S&amp;C sends verification email</li> <li>9. Representative clicks verification link</li> <li>10. S&amp;C prompts for detailed company information: <ul style="list-style-type: none"> <li>• Company Description</li> <li>• Office Locations</li> <li>• Logo Upload</li> <li>• Required Documents</li> </ul> </li> <li>11. S&amp;C sends information for admin verification</li> <li>12. Admin reviews and approves company</li> <li>13. S&amp;C activates company account</li> <li>14. S&amp;C guides through internship posting process</li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• Company account is created and verified</li> <li>• Company profile is complete</li> <li>• Company can post internships</li> </ul>

<b>Exceptions</b>	<ol style="list-style-type: none"><li>1. <b>Company already registered:</b><ul style="list-style-type: none"><li>• Show existing company message</li><li>• Provide contact for account recovery</li></ul></li><li>2. <b>Invalid company email domain:</b><ul style="list-style-type: none"><li>• Request valid company email</li><li>• Provide business verification alternatives</li></ul></li><li>3. <b>Company verification failed:</b><ul style="list-style-type: none"><li>• Request additional verification documents</li><li>• Provide support contact</li></ul></li><li>4. <b>Admin rejection:</b><ul style="list-style-type: none"><li>• Notify reason for rejection</li><li>• Provide appeal process information</li></ul></li><li>5. <b>Incomplete required documents:</b><ul style="list-style-type: none"><li>• List missing documents</li><li>• Save partial progress</li><li>• Allow later completion</li></ul></li></ol>
-------------------	--

Table 3.8: Student Registration Use Case.

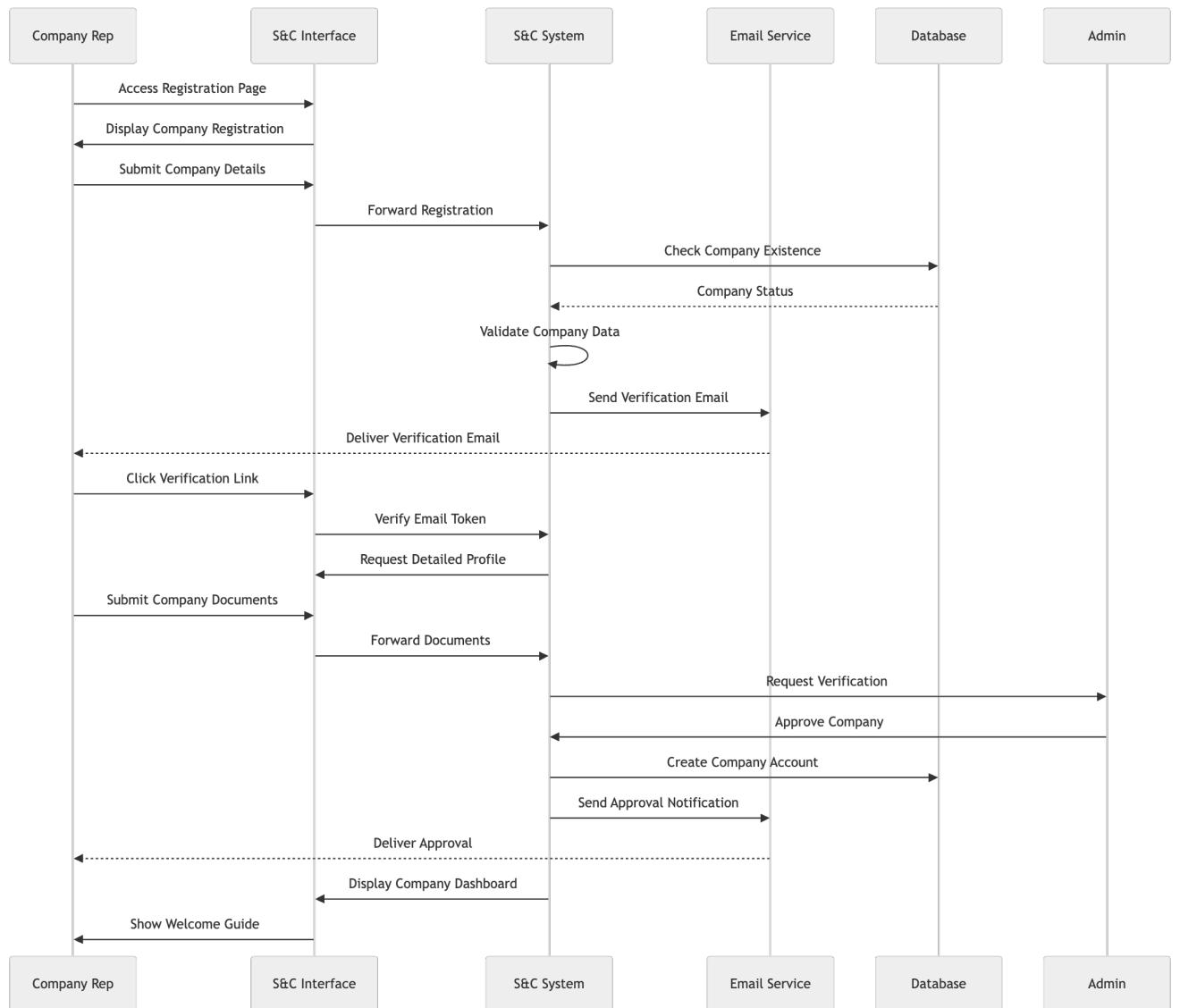


Figure 3.10: Sequence Diagram for Student Registration

## UC2. Company Registration

<b>Actor</b>	Company Representative, Email Provider
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Company not registered in S&amp;C</li> <li>• Representative has company email domain</li> <li>• Company meets platform requirements</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. S&amp;C shows the login form with registration option.</li> <li>2. Representative clicks on "Create Account".</li> <li>3. S&amp;C displays registration form.</li> <li>4. Representative selects "Register as Company".</li> <li>5. Representative enters initial information: <ul style="list-style-type: none"> <li>• Company Name</li> <li>• Company Website</li> <li>• Industry Type</li> <li>• Company Size</li> <li>• Company Email Domain</li> <li>• Representative Name</li> <li>• Representative Position</li> <li>• Password</li> </ul> </li> <li>6. S&amp;C validates company information.</li> <li>7. S&amp;C performs preliminary company verification: <ul style="list-style-type: none"> <li>• Website domain matches email domain</li> <li>• Company exists in business registry (if applicable)</li> </ul> </li> <li>8. S&amp;C sends verification email.</li> <li>9. Representative clicks verification link.</li> <li>10. S&amp;C prompts for detailed company information: <ul style="list-style-type: none"> <li>• Company Description</li> <li>• Office Locations</li> <li>• Logo Upload</li> <li>• Required Documents</li> </ul> </li> <li>11. S&amp;C sends information for admin verification.</li> <li>12. Admin reviews and approves company.</li> <li>13. S&amp;C activates company account.</li> <li>14. S&amp;C guides through internship posting process.</li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• Company account is created and verified.</li> <li>• Company profile is complete.</li> <li>• Company can post internships.</li> </ul>

<b>Exceptions</b>	<ol style="list-style-type: none"><li>1. <b>Company already registered:</b><ul style="list-style-type: none"><li>• Show existing company message.</li><li>• Provide contact for account recovery.</li></ul></li><li>2. <b>Invalid company email domain:</b><ul style="list-style-type: none"><li>• Request valid company email.</li><li>• Provide business verification alternatives.</li></ul></li><li>3. <b>Company verification failed:</b><ul style="list-style-type: none"><li>• Request additional verification documents.</li><li>• Provide support contact.</li></ul></li><li>4. <b>Admin rejection:</b><ul style="list-style-type: none"><li>• Notify reason for rejection.</li><li>• Provide appeal process information.</li></ul></li><li>5. <b>Incomplete required documents:</b><ul style="list-style-type: none"><li>• List missing documents.</li><li>• Save partial progress.</li><li>• Allow later completion.</li></ul></li></ol>
-------------------	---

Table 3.9: Company Registration use case.

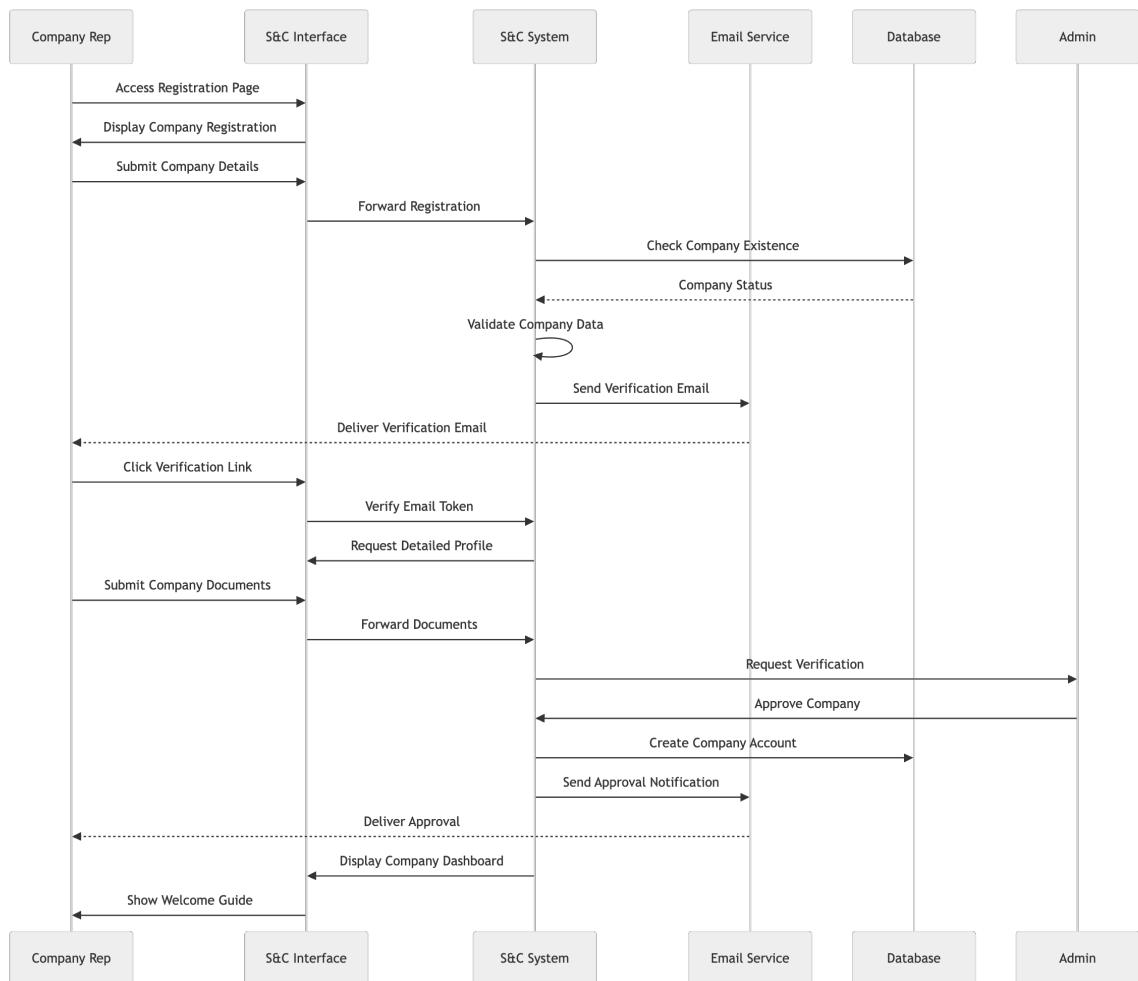


Figure 3.11: Sequence Diagram for Company Registration

### UC3. Internship Posting

<b>Actor</b>	Company Representative
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Company is verified and logged in</li> <li>• Company profile is complete</li> <li>• Company has posting privileges</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"><li>1. Company accesses internship management dashboard.</li><li>2. Clicks "Post New Internship" button.</li><li>3. S&amp;C displays internship creation form.</li><li>4. Company enters internship details:<ul style="list-style-type: none"><li>• Position Title</li><li>• Department</li><li>• Duration</li><li>• Start Date</li><li>• Required Skills</li><li>• Preferred Skills</li><li>• Educational Requirements</li><li>• Responsibilities</li><li>• Compensation Details</li><li>• Application Deadline</li><li>• Number of Positions</li><li>• Location/Remote Status</li></ul></li><li>5. Company sets additional preferences:<ul style="list-style-type: none"><li>• Interview Process Steps</li><li>• Required Documents</li><li>• Assessment Criteria</li></ul></li><li>6. S&amp;C validates all input fields.</li><li>7. Company previews posting.</li><li>8. Company submits for review.</li><li>9. S&amp;C performs automated checks:<ul style="list-style-type: none"><li>• Content appropriateness</li><li>• Completeness</li><li>• Compliance with platform rules</li></ul></li><li>10. S&amp;C processes posting:<ul style="list-style-type: none"><li>• Indexes for search</li><li>• Matches with student profiles</li><li>• Generates recommendations</li></ul></li><li>11. S&amp;C activates posting.</li><li>12. Notifies matching students.</li></ol>
-------------------	---

<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• Internship is posted and visible</li><li>• Matching students are notified</li><li>• Position appears in search results</li></ul>
<b>Exceptions</b>	<ol style="list-style-type: none"><li>1. <b>Incomplete Required Fields:</b><ul style="list-style-type: none"><li>• Highlight missing fields.</li><li>• Save as draft.</li></ul></li><li>2. <b>Invalid Date Combinations:</b><ul style="list-style-type: none"><li>• Show date validation error.</li><li>• Suggest valid date ranges.</li></ul></li><li>3. <b>Non-Compliant Content:</b><ul style="list-style-type: none"><li>• Flag specific issues.</li><li>• Provide compliance guidelines.</li></ul></li><li>4. <b>Duplicate Posting:</b><ul style="list-style-type: none"><li>• Show existing posting.</li><li>• Offer update option.</li></ul></li><li>5. <b>Posting Limit Reached:</b><ul style="list-style-type: none"><li>• Show upgrade options.</li><li>• Manage existing posts.</li></ul></li></ol>

Table 3.10: Internship Posting Use Case.

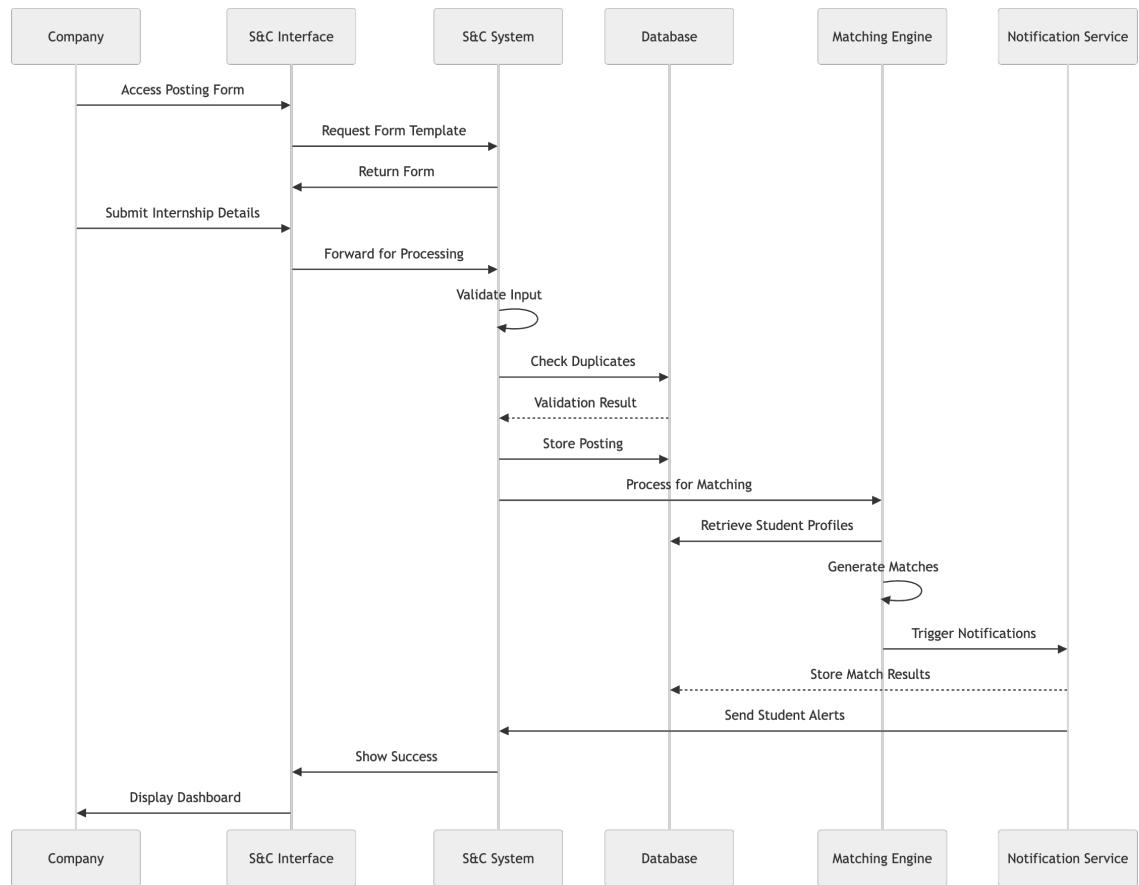


Figure 3.12: Sequence Diagram for Internship Posting

#### UC4. Student Application Process

<b>Actor</b>	Student
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Student is logged in</li> <li>• Profile and CV are complete</li> <li>• Meets basic internship requirements</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"><li>1. Student finds internship through:<ul style="list-style-type: none"><li>• Direct search</li><li>• Recommendations</li><li>• Email notification</li></ul></li><li>2. Student reviews internship details:<ul style="list-style-type: none"><li>• Company information</li><li>• Position requirements</li><li>• Terms and conditions</li></ul></li><li>3. Student initiates application:<ul style="list-style-type: none"><li>• Selects CV version</li><li>• Customizes cover letter</li><li>• Answers screening questions</li><li>• Provides additional documents</li></ul></li><li>4. S&amp;C performs preliminary checks:<ul style="list-style-type: none"><li>• Eligibility verification</li><li>• Previous applications</li><li>• Document completeness</li></ul></li><li>5. Student reviews application package.</li><li>6. Student submits application.</li><li>7. S&amp;C processes application:<ul style="list-style-type: none"><li>• Updates application counter</li><li>• Indexes for company search</li><li>• Generates application ID</li></ul></li><li>8. Company receives notification.</li><li>9. S&amp;C updates student dashboard.</li><li>10. Application tracking begins.</li></ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• Application is submitted</li><li>• Company is notified</li><li>• Student can track status</li></ul>

<b>Exceptions</b>	<ol style="list-style-type: none"><li><b>1. Incomplete Profile:</b><ul style="list-style-type: none"><li>• Redirect to profile completion.</li><li>• Save application draft.</li></ul></li><li><b>2. Missing Required Documents:</b><ul style="list-style-type: none"><li>• List required documents.</li><li>• Provide upload options.</li></ul></li><li><b>3. Previous Application Exists:</b><ul style="list-style-type: none"><li>• Show existing application.</li><li>• Offer update option.</li></ul></li><li><b>4. Deadline Passed:</b><ul style="list-style-type: none"><li>• Show expired notice.</li><li>• Suggest similar positions.</li></ul></li><li><b>5. Requirements Not Met:</b><ul style="list-style-type: none"><li>• Show specific gaps.</li><li>• Suggest skill development.</li></ul></li></ol>
-------------------	--

Table 3.11: Student Application Process Use Case.

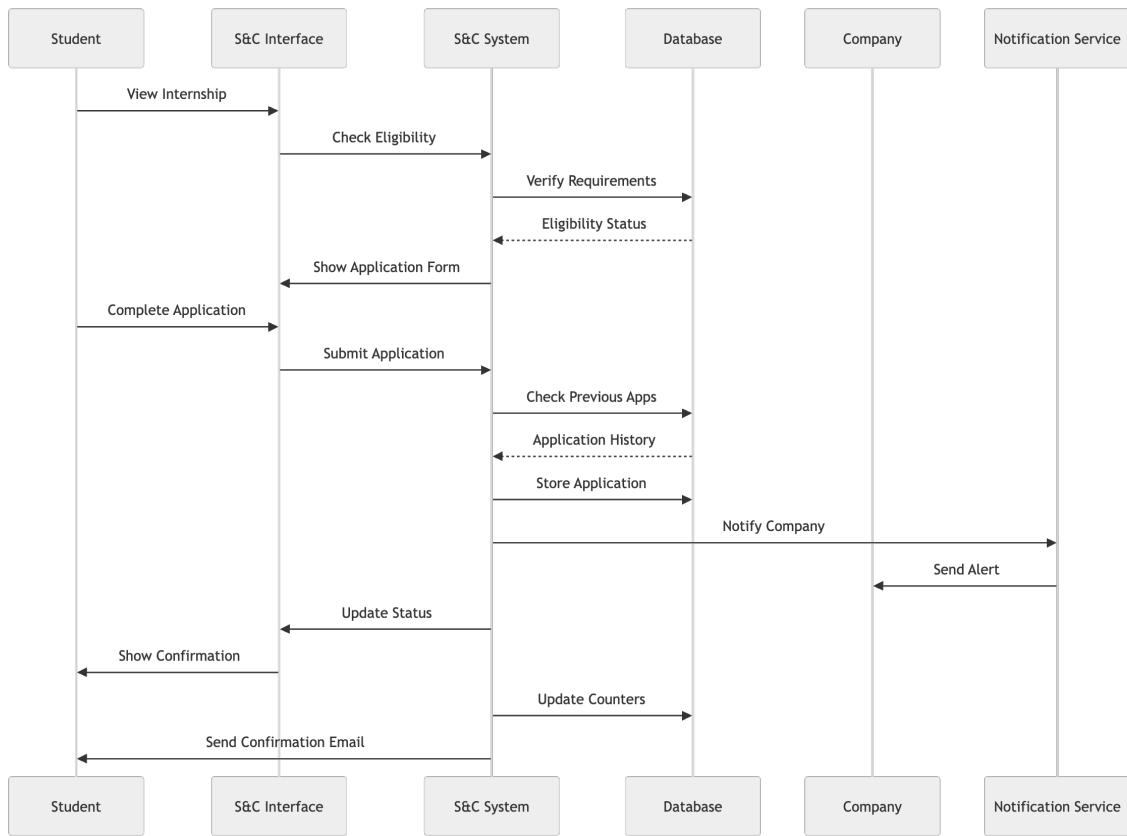


Figure 3.13: Sequence Diagram for Student Application Process

## UC5. Interview Management

<b>Actor</b>	Company Representative, Student
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>Application is shortlisted</li> <li>Both parties are active users</li> <li>Interview stage is initiated</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. Company reviews application.</li> <li>2. Company initiates interview process: <ul style="list-style-type: none"> <li>• Selects interview type</li> <li>• Proposes time slots</li> <li>• Sets interview format</li> </ul> </li> <li>3. S&amp;C sends interview request to student.</li> <li>4. Student receives notification: <ul style="list-style-type: none"> <li>• Views proposed slots</li> <li>• Checks interview details</li> <li>• Reviews preparations</li> </ul> </li> <li>5. Student responds: <ul style="list-style-type: none"> <li>• Accepts time slot or</li> <li>• Proposes alternatives</li> </ul> </li> <li>6. S&amp;C coordinates scheduling: <ul style="list-style-type: none"> <li>• Confirms final time</li> <li>• Sends calendar invites</li> <li>• Provides meeting links</li> </ul> </li> <li>7. Both parties prepare: <ul style="list-style-type: none"> <li>• Access interview materials</li> <li>• Review guidelines</li> <li>• Check technical setup</li> </ul> </li> <li>8. Interview conducted.</li> <li>9. Company provides feedback: <ul style="list-style-type: none"> <li>• Evaluation form</li> <li>• Notes and ratings</li> <li>• Decision input</li> </ul> </li> <li>10. S&amp;C processes results: <ul style="list-style-type: none"> <li>• Updates application status</li> <li>• Notifies student</li> <li>• Records outcomes</li> </ul> </li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• Interview is completed</li> <li>• Feedback is recorded</li> <li>• Next steps are initiated</li> </ul>

<b>Exceptions</b>	<ol style="list-style-type: none"><li>1. <b>Schedule Conflicts:</b><ul style="list-style-type: none"><li>• Rescheduling process</li><li>• Alternative slot suggestions</li></ul></li><li>2. <b>Technical Issues:</b><ul style="list-style-type: none"><li>• Backup contact methods</li><li>• Rescheduling options</li></ul></li><li>3. <b>No-Show Scenarios:</b><ul style="list-style-type: none"><li>• Record incident</li><li>• Reschedule policy</li></ul></li><li>4. <b>Cancellation Requests:</b><ul style="list-style-type: none"><li>• Process cancellation</li><li>• Update status</li></ul></li><li>5. <b>Feedback Deadline Missed:</b><ul style="list-style-type: none"><li>• Send reminders</li><li>• Escalation process</li></ul></li></ol>
-------------------	---

Table 3.12: Interview Management Use Case.

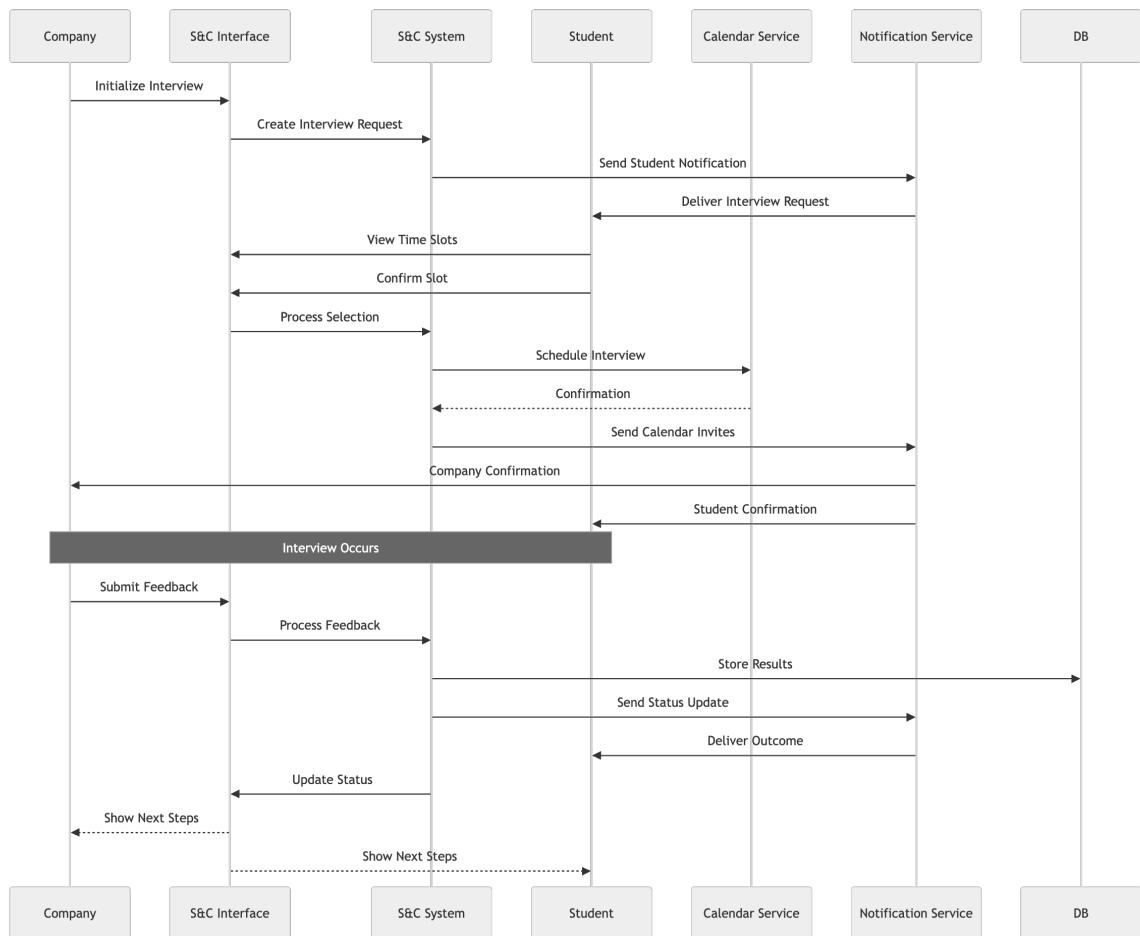


Figure 3.14: Sequence Diagram for Interview Management

## UC6. Complaint Handling

<b>Actor</b>	Student/Company, University Administrator
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>User is registered and active</li> <li>Incident is within platform scope</li> <li>Related internship is active/recent</li> </ul>

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. User initiates complaint: <ul style="list-style-type: none"> <li>• Selects complaint type</li> <li>• Identifies involved parties</li> <li>• Describes incident</li> <li>• Provides evidence</li> </ul> </li> <li>2. S&amp;C processes submission: <ul style="list-style-type: none"> <li>• Assigns complaint ID</li> <li>• Categorizes severity</li> <li>• Routes to appropriate admin</li> </ul> </li> <li>3. Administrator receives case: <ul style="list-style-type: none"> <li>• Reviews details</li> <li>• Assesses priority</li> <li>• Initiates investigation</li> </ul> </li> <li>4. S&amp;C facilitates investigation: <ul style="list-style-type: none"> <li>• Gathers additional information</li> <li>• Contacts involved parties</li> <li>• Documents communications</li> </ul> </li> <li>5. Administrator actions: <ul style="list-style-type: none"> <li>• Reviews all evidence</li> <li>• Consults policies</li> <li>• Determines resolution</li> </ul> </li> <li>6. Resolution implementation: <ul style="list-style-type: none"> <li>• Notifies all parties</li> <li>• Records decisions</li> <li>• Implements actions</li> </ul> </li> <li>7. Follow-up process: <ul style="list-style-type: none"> <li>• Monitors compliance</li> <li>• Collects feedback</li> <li>• Updates records</li> </ul> </li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• Complaint is resolved</li> <li>• Actions are implemented</li> <li>• Resolution is documented</li> </ul>

<b>Exceptions</b>	<ol style="list-style-type: none"><li><b>1. Insufficient Information:</b><ul style="list-style-type: none"><li>• Request additional details.</li><li>• Hold complaint status.</li></ul></li><li><b>2. Multiple Parties Involved:</b><ul style="list-style-type: none"><li>• Expand investigation scope.</li><li>• Coordinate responses.</li></ul></li><li><b>3. Policy Violations Found:</b><ul style="list-style-type: none"><li>• Escalate to higher authority.</li><li>• Implement immediate actions.</li></ul></li><li><b>4. Appeal Submitted:</b><ul style="list-style-type: none"><li>• Review appeal grounds.</li><li>• Reassign to different admin.</li></ul></li><li><b>5. Resolution Deadline Missed:</b><ul style="list-style-type: none"><li>• Escalate to supervisor.</li><li>• Update timeline.</li></ul></li></ol>
-------------------	---

Table 3.13: Complaint Handling Use Case.

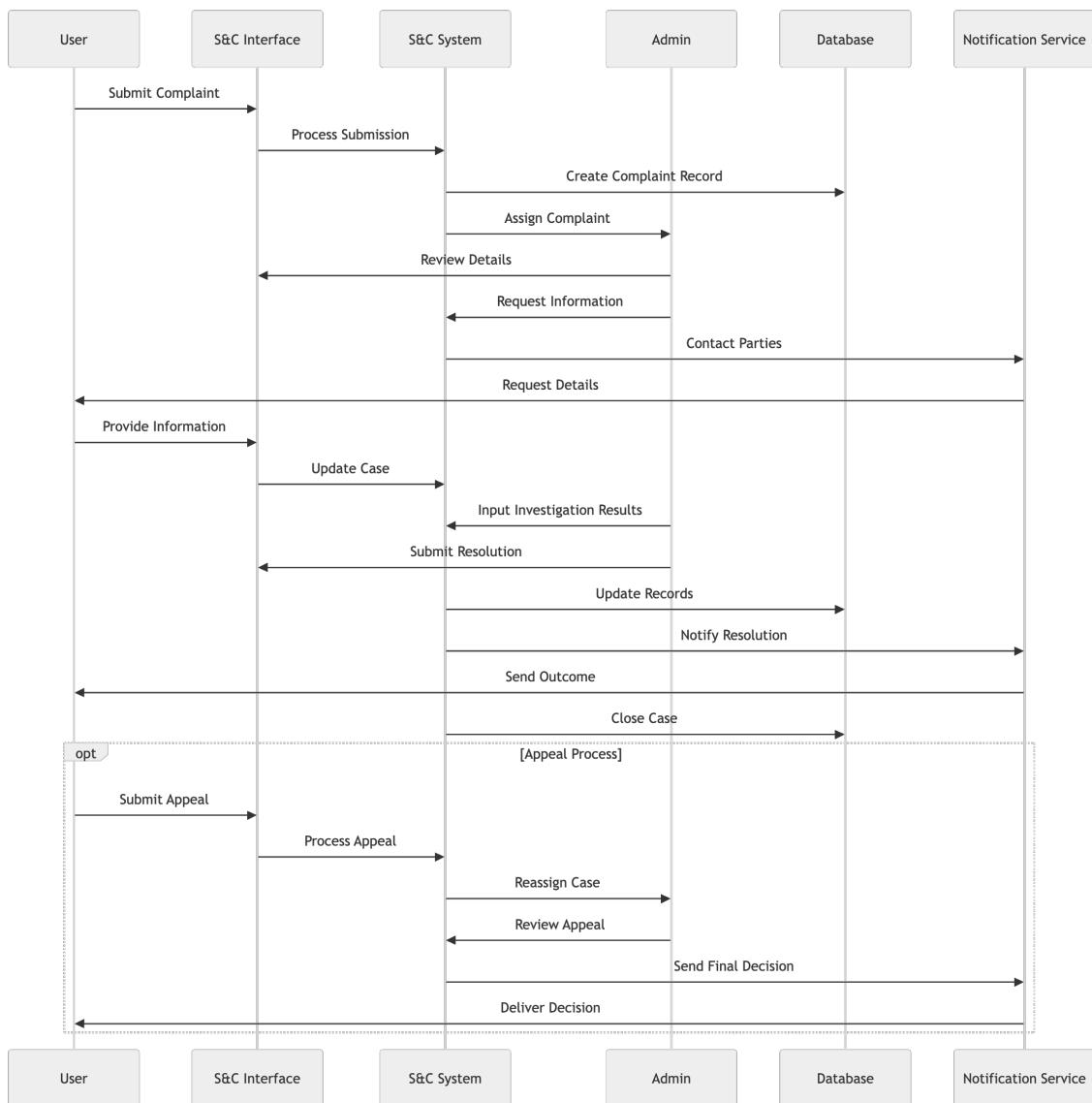


Figure 3.15: Sequence Diagram for Complaint Handling

### UC7. Admin Privileges

<b>Actor</b>	University Administrator
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Admin is logged in with verification privileges</li> <li>• Items pending verification exist</li> <li>• Access to verification tools available</li> </ul>

Event Flow	1. Admin accesses verification dashboard: <ul style="list-style-type: none"><li>• Views pending items</li><li>• Checks verification queue</li><li>• Reviews priority items</li></ul> 2. For each verification request: <ul style="list-style-type: none"><li>• Reviews company details</li><li>• Checks documentation</li><li>• Validates credentials</li><li>• Assesses compliance</li></ul> 3. Document verification: <ul style="list-style-type: none"><li>• Company registration</li><li>• Business licenses</li><li>• Insurance certificates</li><li>• Tax documentation</li></ul> 4. Compliance check: <ul style="list-style-type: none"><li>• University policies</li><li>• Legal requirements</li><li>• Industry standards</li><li>• Safety regulations</li></ul> 5. Decision process: <ul style="list-style-type: none"><li>• Approves application</li><li>• Requests modifications</li><li>• Rejects with reason</li></ul> 6. Post-approval actions: <ul style="list-style-type: none"><li>• Sets verification status</li><li>• Assigns trust score</li><li>• Enables features</li><li>• Sets review date</li></ul> 7. Communication: <ul style="list-style-type: none"><li>• Notifies company</li><li>• Updates records</li><li>• Documents decision</li></ul>
------------	---

<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• Verification decision made</li><li>• Company notified</li><li>• Records updated</li></ul>
<b>Exceptions</b>	<ol style="list-style-type: none"><li>1. <b>Missing Documentation:</b><ul style="list-style-type: none"><li>• Request specific documents</li><li>• Set pending status</li></ul></li><li>2. <b>Compliance Issues:</b><ul style="list-style-type: none"><li>• Detail requirements</li><li>• Provide guidance</li></ul></li><li>3. <b>Verification Timeout:</b><ul style="list-style-type: none"><li>• Extend deadline</li><li>• Notify parties</li></ul></li><li>4. <b>Suspicious Activity:</b><ul style="list-style-type: none"><li>• Flag for investigation</li><li>• Suspend processing</li></ul></li><li>5. <b>Appeal Request:</b><ul style="list-style-type: none"><li>• Review appeal</li><li>• Escalate if needed</li></ul></li></ol>

Table 3.14: Admin Privileges Use Case.

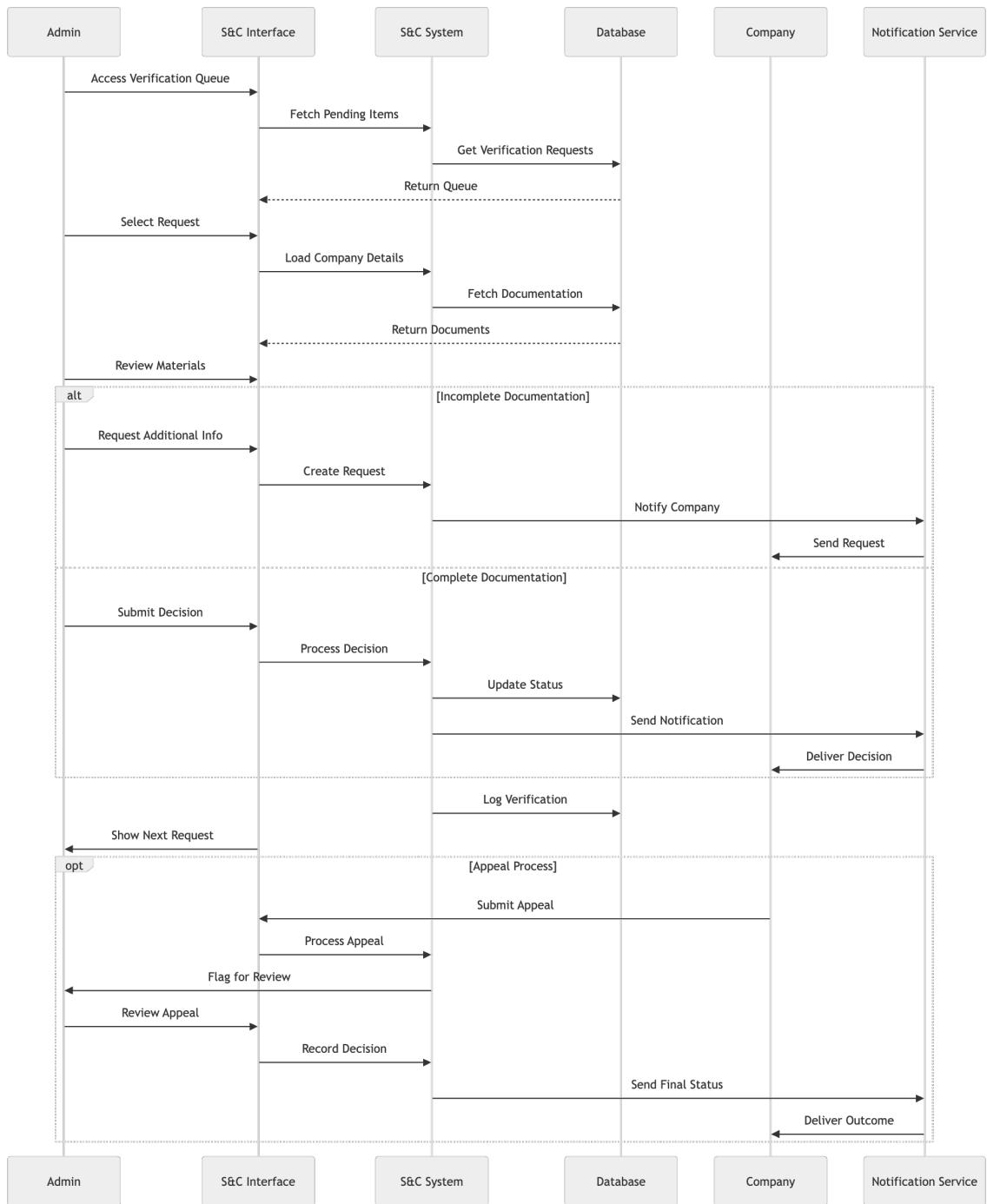


Figure 3.16: Sequence Diagram for Admin Privileges

### 3.2.6. Performance Requirements

#### Response Time

- **Page Load:** Less than **5 seconds** for typical pages during testing.
- **Search Results:** Less than **2 seconds** for small-scale datasets.
- **File Upload:** Less than **10 seconds** for files up to **10MB**.
- **Real-time Updates:** Less than **1 second** for key interactive features.

#### System Capacity

- **Concurrent Users:** Support for up to **50 simultaneous users**.
- **Database Transactions:** Designed to handle **10 transactions per second** under typical usage.
- **File Storage:** Capacity for up to **10GB**, suitable for project scope and testing.
- **Backup Frequency:** **Weekly or manual backups** for data protection during development.

#### Availability

- **Uptime:** Target of **95%**, considering potential downtime for development and testing.
- **Scheduled Maintenance:** As needed during the project lifecycle.
- **Backup Recovery:** Recovery within **12 hours** for small-scale data.
- **Error Rate:** Less than **1% for prototype-level functionality**.

### 3.3. Design constraints

#### 3.3.1. Design Constraints

##### Standards Compliance

- **GDPR Compliance:** The platform must comply with the **General Data Protection Regulation** to ensure the protection of user data and privacy. This includes explicit consent mechanisms, data anonymization, and secure data processing protocols.
- **WCAG 2.1 Accessibility:** Adheres to the **Web Content Accessibility Guidelines 2.1** to ensure inclusivity. This involves providing support for screen readers, keyboard navigation, and ensuring sufficient color contrast.
- **ISO/IEC 27001 Security:** Implements security management practices aligned with **ISO/IEC 27001** to safeguard information assets and prevent data breaches. This includes encryption, secure authentication, and incident response planning.
- **Browser Standards:** Ensures **cross-browser compatibility** by adhering to **HTML5, CSS3, and modern JavaScript standards**. This allows the platform to work seamlessly on popular browsers like Chrome, Firefox, and Edge.

##### Development Constraints

- **Web-based Architecture:** The system will be fully web-based, designed to run on a server and accessible via web browsers without the need for additional software installations.
- **Responsive Design:** The platform must be responsive, ensuring an optimal user experience on devices of varying screen sizes, including desktops, tablets, and mobile phones.
- **Modular Components:** The architecture will be modular, facilitating easier debugging, testing, and future enhancements by breaking functionality into independent, reusable components.
- **API-First Approach:** Development will follow an API-first methodology, prioritizing a well-documented API layer that enables easy integration with external systems and scalability for future mobile app development.

### 3.3.2. Software System Attributes

#### Reliability

- **Error handling** to ensure data integrity.
- **Input validation** for all user entries.
- **Transaction integrity** for financial processes.
- **System recovery** after failures.

#### Availability

- **Redundant Systems:** Ensure continuous, round-the-clock functioning of the platform.
- **Failover Mechanism:** Enables the system to handle server disruptions by automatically switching to backup systems.

#### Security

- **Robust authentication systems.**
- **Authorization for sensitive operations** based on roles.
- **Data encryption** from beginning to end.

#### Maintainability

- **Modular design** for easier updates.
- **Comprehensive documentation** for developers and users.
- **Version control** for all software components.
- **Automated testing** frameworks.

#### Portability

- **Cross-browser compatibility.**
- **Mobile responsiveness.**
- **Platform independence** to run on various operating systems.
- **Easy deployment** for scaling and updates.



# 4 | Formal Analysis Using Alloy

## 4.0.1. Alloy Specification

Here is the Alloy specification of the system described in the RASD:

---

```
// ----- Core Signatures -----
abstract sig User {
    id: one ID,
    email: one ID,
    name: one ID,
    phoneNumber: one ID
}

// A student user who has a CV and can submit internship applications .
sig Student extends User {
    cv: one CV,
    applications: set Application
} {
    // Every student must have a CV.
    some cv
}

// A company user that offers internships .
sig Company extends User {
    companyName: one ID,
    industry: one ID,
    internships: set Internship // Internships provided by the company.
}

// A university user managing students and handling complaints.
sig University extends User {
    universityName: one ID,
    students: set Student, // The students enrolled in the university .
    complaints: set Complaint // Complaints registered with the university .
}
```

```
}
```

```
// A CV containing a summary, skills, education, and experience details .
```

```
sig CV {
    skills: set Skill,
    education: set Education,
    experience: set Experience,
    summary: one ID
}
```

```
// An internship with specific details such as title , description , requirements , and dates .
```

```
sig Internship {
    title: one ID,
    description: one ID,
    requirements: set Skill, // Skills required for the internship .
    startDate: one Date,
    endDate: one Date,
    status: one Status
}
```

```
// An application linking a student to an internship .
```

```
sig Application {
    student: one Student,
    internship: one Internship,
    applicationDate: one Date,
    status: one Status
}
```

```
// An interview scheduled for a specific application .
```

```
sig Interview {
    application: one Application,
    scheduledTime: one DateTime, // Time slot for the interview.
    status: one Status,
    feedback: lone ID
}
```

```
// A complaint filed by a user with a description and status .
```

```
sig Complaint {
    id: one ID,
    description: one ID, // Complaint details.
    status: one Status,
    filingDate: one Date,
    complainant: one User // The user who raised the complaint.
}
```

```

// Represents textual identifiers for various elements.
sig ID {}

// ----- Enumerations -----

enum Skill { JAVA, PYTHON, CPP, JAVASCRIPT, DATABASE }
enum Education { BACHELORS, MASTERS, PHD }
enum Experience { JUNIOR, MID, SENIOR }
enum Status { OPEN, CLOSED, PENDING, APPROVED, REJECTED }
enum DateTime { MORNING, AFTERNOON, EVENING }

// Represents months as a hierarchy rather than an enumeration.
abstract sig Date {}

one sig JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC extends Date {}

// ----- Utility Function + Predicate -----

// Assigns a numerical index to each month for easy comparison.
fun dateIndex[d: Date]: Int {
    d = JAN => 1
    else d = FEB => 2
    else d = MAR => 3
    else d = APR => 4
    else d = MAY => 5
    else d = JUN => 6
    else d = JUL => 7
    else d = AUG => 8
    else d = SEP => 9
    else d = OCT => 10
    else d = NOV => 11
    else 12
}

// Validates that one date comes before another.
pred dateOrder[d1, d2: Date] {
    dateIndex[d1] < dateIndex[d2]
}

// ----- Facts & Constraints -----

// Enforces unique IDs for all users.
fact UserIDAreUnique {
    all disj u1, u2: User | u1.id != u2.id
}

```

```
}
```

```
// Ensures the system always contains at least one student, company, and university.
```

```
fact AtLeastOneOfEach {
    some Student
    some Company
    some University
}
```

```
// Every student must be associated with exactly one university .
```

```
fact StudentUniversityRelationship {
    all s: Student | one u: University | s in u.students
}
```

```
// Ensures internships are owned by exactly one company.
```

```
fact InternshipOwnership {
    all i: Internship | one c: Company | i in c.internships
}
```

```
// Ensures internships have a valid time frame, with the start date before the end date.
```

```
fact InternshipDates {
    all i: Internship | dateOrder[i.startDate, i.endDate]
}
```

```
// A student can only apply for internships where their CV covers all required skills .
```

```
fact ValidApplications {
    all a: Application |
        a.internship.requirements in a.student.cv.skills
}
```

```
// Guarantees that open internships have at least one applicant .
```

```
fact OpenInternshipsMustHaveApplicants {
    all i: Internship |
        i.status = OPEN implies some a: Application | a.internship = i
}
```

```
// Prevents duplicate applications for the same internship by the same student.
```

```
fact UniqueApplications {
    all disj a1, a2: Application |
        (a1.student != a2.student) or (a1.internship != a2.internship)
}
```

```
// If an application is approved, the corresponding internship cannot remain open.
```

```
fact ApprovedApplicationClosesInternship {
```

```

all a: Application |
    a.status = APPROVED implies a.internship.status != OPEN
}

// Ensures interviews are only scheduled if the student's skills meet the internship requirements.

fact ValidInterview {
    all i: Interview |
        i.application.student.cv.skills in i.application.internship.requirements
}

// Complaints (if not rejected) must have a valid complainant.

fact ComplaintResolution {
    all c: Complaint |
        c.status != REJECTED implies some c.complainant
}

// Ensures all complaints have unique IDs and exactly one complainant.

fact ValidComplaints {
    all c: Complaint | one u: User | c.complainant = u
    all disj c1, c2: Complaint | c1.id != c2.id
}

// Suggests that students whose skills match the requirements of an internship should apply.

fact RecommendedApplications {
    all s: Student, i: Internship |
        (i.requirements in s.cv.skills) implies
            (some a: Application | a.student = s and a.internship = i)
}
}

// ----- Simple Run Command -----

// Generates an instance of the model up to a scope of 5.
run {} for 5

// ----- Analysis Code (Assertions + Predicates) -----

// Assertion: Prevents students from applying to the same internship multiple times.
assert UniqueApplicationsAssert {
    all disj a1, a2: Application |
        a1.student = a2.student and a1.internship = a2.internship
        implies a1 = a2
}
check UniqueApplicationsAssert for 5

```

```

// Assertion: Validates that open internships have at least one applicant.
assert OpenInternshipsAssert {
    all i: Internship |
        i.status = OPEN implies some a: Application | a.internship = i
}
check OpenInternshipsAssert for 5

// Assertion: Ensures students are only interviewed if they meet the requirements.
assert ValidInterviewAssert {
    all i: Interview |
        i.application.student.cv.skills in i.application.internship.requirements
}
check ValidInterviewAssert for 5

// Assertion: Complaints (if not rejected) must have valid complainants.
assert ComplaintResolutionAssert {
    all c: Complaint |
        c.status != REJECTED implies some c.complainant
}
check ComplaintResolutionAssert for 5

// Assertion: Approved applications mean the internship is not open anymore.
assert ApprovedApplicationClosesInternshipAssert {
    all a: Application |
        a.status = APPROVED implies a.internship.status != OPEN
}
check ApprovedApplicationClosesInternshipAssert for 5

// Assertion: If a student meets all required skills, an application should exist.
assert RecommendedApplicationsAssert {
    all s: Student, i: Internship |
        (i.requirements in s.cv.skills) implies
            (some a: Application | a.student = s and a.internship = i)
}
check RecommendedApplicationsAssert for 5

```

---

*//-----*  
*// Scenarios to RUN and visualize*  
*//-----*

```

/**
 * Scenario: A student is applying for an internship.
 * The internship is open, and the student has the required skills for it.

```

```

* We also ensure that the application is properly linked to both the student
* and the internship .
*/
pred StudentAppliesToOpenInternship {
    some s: Student, i: Internship |
        i.status = OPEN
        and i.requirements in s.cv.skills
        and some a: Application | a.student = s and a.internship = i
}
/** Execute this scenario with up to 5 instances */
run StudentAppliesToOpenInternship for 5

/**
* Scenario: An internship gets approved for a student.
* Once approved, the internship is no longer open for applications .
* This helps verify that the status transitions are consistent .
*/
pred ApprovedInternshipScenario {
    some a: Application |
        a.status = APPROVED
        and a.internship.status != OPEN
}
/** Execute this scenario with up to 5 instances */
run ApprovedInternshipScenario for 5

/**
* Scenario: A complaint has been submitted and isn't rejected .
* If the complaint is still under consideration or resolved ,
* we need to ensure theres a valid complainant tied to it .
*/
pred ValidComplaintScenario {
    some c: Complaint |
        c.status != REJECTED
}
/** Execute this scenario with up to 5 instances */
run ValidComplaintScenario for 5

```

---

#### 4.0.2. Alloy Analysis Results

The Alloy Analyzer was used to validate the constraints, assertions, and facts defined in the model. The following results were obtained:

## Assertion 1: UniqueApplicationsAssert

**Description:** This assertion ensures that no student can apply to the same internship more than once.

**Check Command:**

```
check UniqueApplicationsAssert for 5
```

**Result:**

```
Executing "Check UniqueApplicationsAssert for 5"
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch
11978 vars. 1040 primary vars. 22948 clauses. 22ms.
No counterexample found. Assertion may be valid. 3ms.
```

## Assertion 2: OpenInternshipsAssert

**Description:** Ensures that all internships marked as OPEN have at least one associated applicant.

**Check Command:**

```
check OpenInternshipsAssert for 5
```

**Result:**

```
Executing "Check OpenInternshipsAssert for 5"
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch
11958 vars. 1035 primary vars. 22710 clauses. 23ms.
No counterexample found. Assertion may be valid. 4ms.
```

## Assertion 3: ValidInterviewAssert

**Description:** Validates that students are only interviewed if their skills match the internship requirements.

**Check Command:**

```
check ValidInterviewAssert for 5
```

**Result:**

```
Executing "Check ValidInterviewAssert for 5"  
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
12040 vars. 1035 primary vars. 22907 clauses. 23ms.  
No counterexample found. Assertion may be valid. 49ms.
```

**Assertion 4: ComplaintResolutionAssert**

**Description:** Ensures that all complaints not marked REJECTED have valid complainants.

**Check Command:**

```
check ComplaintResolutionAssert for 5
```

**Result:**

```
Executing "Check ComplaintResolutionAssert for 5"  
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
11918 vars. 1035 primary vars. 22639 clauses. 23ms.  
No counterexample found. Assertion may be valid. 3ms.
```

**Assertion 5: ApprovedApplicationClosesInternshipAssert**

**Description:** Ensures that once an application is APPROVED, the associated internship is no longer OPEN.

**Check Command:**

```
check ApprovedApplicationClosesInternshipAssert for 5
```

**Result:** Verified. The model enforces that APPROVED

```
Executing "Check ApprovedApplicationClosesInternshipAssert for 5"  
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
11949 vars. 1035 primary vars. 22747 clauses. 25ms.  
No counterexample found. Assertion may be valid. 2ms.
```

**Assertion 6: RecommendedApplicationsAssert**

**Description:** If a student's skills match an internship's requirements, an application should exist.

### Check Command:

```
check RecommendedApplicationsAssert for 5
```

### Result:

Executing "Check RecommendedApplicationsAssert for 5"

Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
 12128 vars. 1040 primary vars. 23002 clauses. 23ms.  
 No counterexample found. Assertion may be valid. 7ms.

## Scenario: Student Applies to an Open Internship

**Description:** A scenario where a student with matching skills applies to an OPEN internship.

### Predicate:

```
pred StudentAppliesToOpenInternship {
    some s: Student, i: Internship |
        i.status = OPEN
        and i.requirements in s.cv.skills
        and some a: Application |
            a.student = s and a.internship = i
}
run StudentAppliesToOpenInternship for 5
```

**Result:** Successful. The Alloy Analyzer generates a valid instance where a student with the required skills applies to an OPEN internship.

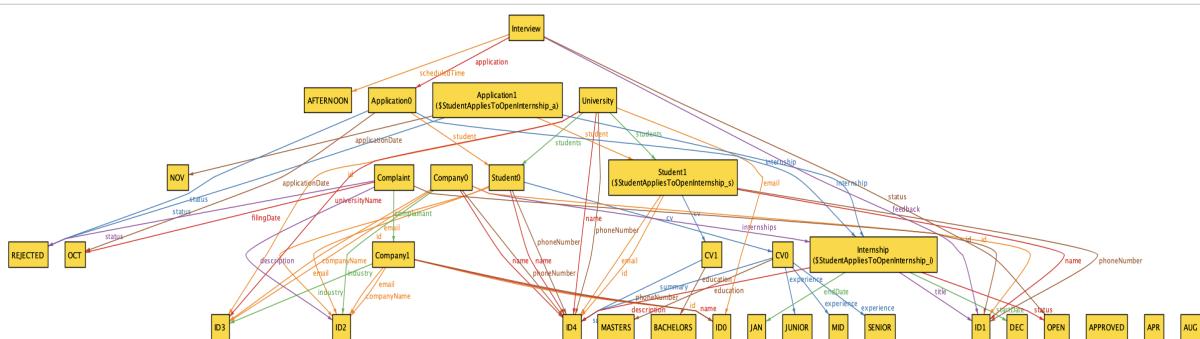


Figure 4.1: Scenario for Student Applies to an *Open Internship*.

## Scenario: Approved Internship Transition

**Description:** Once an application is APPROVED, the corresponding internship should no longer be OPEN.

**Predicate:**

```
pred ApprovedInternshipScenario {
    some a: Application |
        a.status = APPROVED
        and a.internship.status != OPEN
}
run ApprovedInternshipScenario for 5
```

**Result:** Successful. The Alloy Analyzer verifies the status transition of internships when an application is approved.

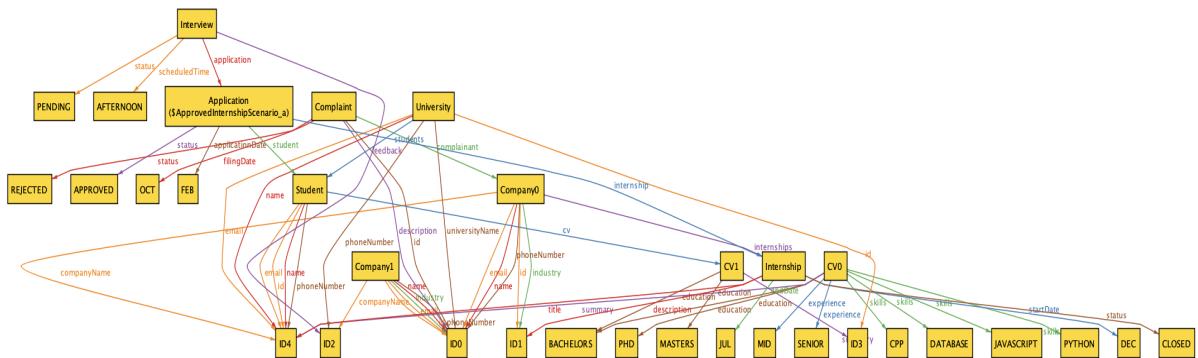


Figure 4.2: Scenario for Approved Internship.

## Scenario: Valid Complaint with Complainant

**Description:** A complaint that is not marked REJECTED must have a valid complainant associated with it.

**Predicate:**

```
pred ValidComplaintScenario {
    some c: Complaint |
        c.status != REJECTED
}
run ValidComplaintScenario for 5
```

**Result:** Verified. The Alloy Analyzer confirms that all valid complaints are associated with a complainant.

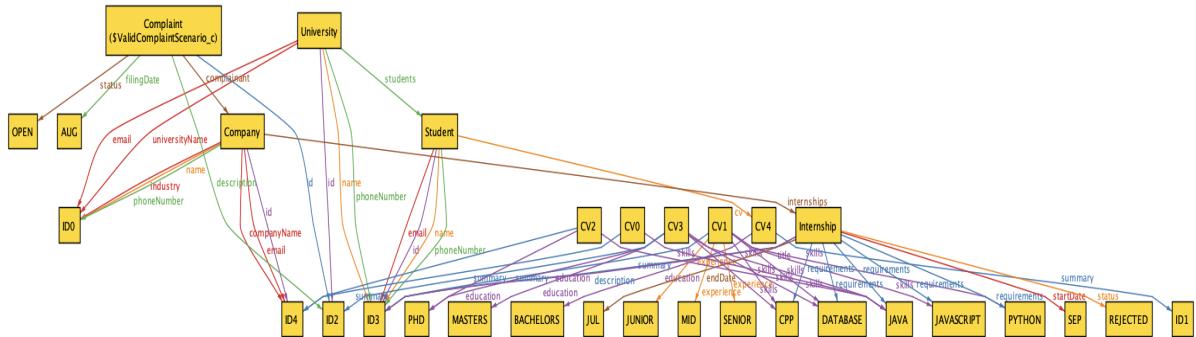


Figure 4.3: Scenario for *Valid Complaint*.

#### 4.0.3. Base World for InternHub

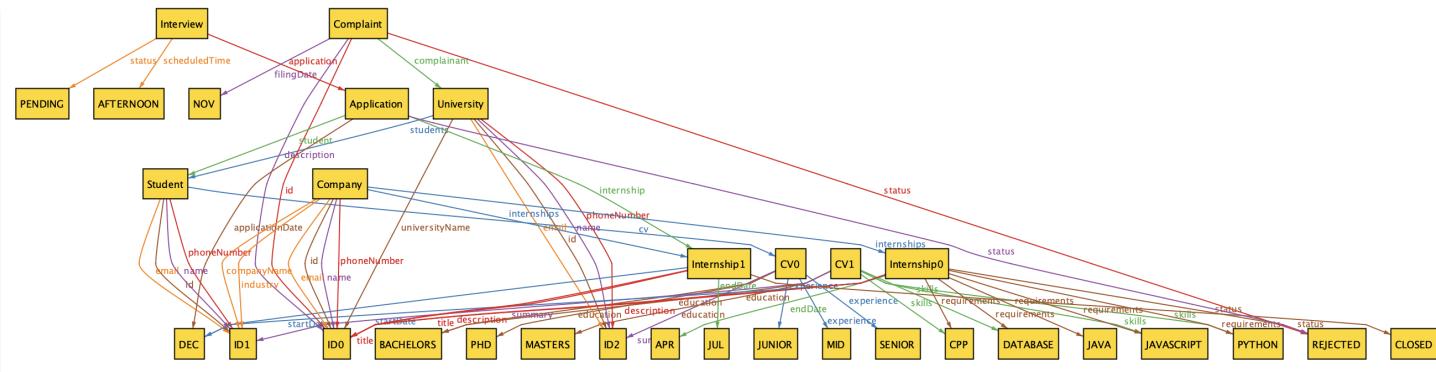


Figure 4.4: Instance for *BaseWorld Simplified*.

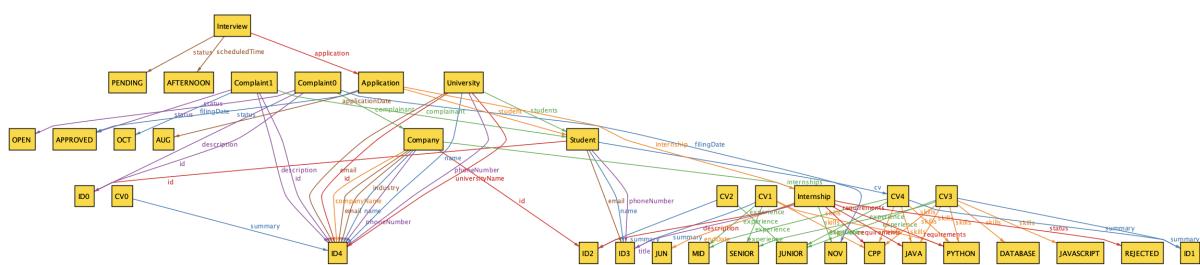


Figure 4.5: Simplified Instance for *BaseWorld Simplified 2*.

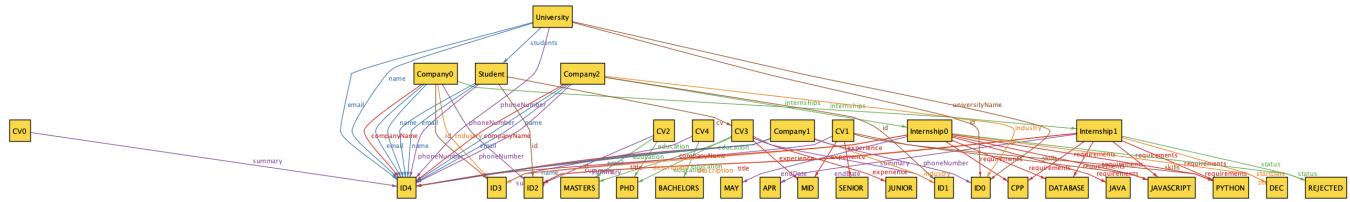


Figure 4.6: Simplified Instance for *BaseWorld Simplified 3*.



# 5 | Effort Spent

<b>Team Member</b>	<b>Task</b>	<b>Hours Spent</b>
Shreesh Kumar Jha	Requirements Analysis, Frontend Mockup, Frontend Setup, Mockup Implementation, Alloy Modelling	56
Samarth Bhatia	Requirements Analysis, UML Diagram, Frontend Setup, Mockup Implementation	51
Satviki Sharma	Requirements Analysis, UML Diagrams, Backend Setup, Backend Implementation	51

Table 5.1: Effort spent by each member of the group.



# 6 | References

## 6.1. References

- IEEE Standard 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.
- ISO/IEC/IEEE 29148:2018 - Systems and software engineering - Life cycle processes - Requirements engineering.
- Daniel Jackson, *Software Abstractions: Logic, Language, and Analysis*.
- Assignment RDD AY 2024-2025.pdf.
- Software Engineering 2 Course Materials, A.Y. 2024-2025.

## 6.2. Used Tools

- GitHub for version control and collaborative development.
- L<sup>A</sup>T<sub>E</sub>X with *Overleaf* and *Neovim* for writing and formatting this document.
- Mermaid for UML Diagrams
- *Alloy* for formal specification and model analysis.
- *Google Documents* for brainstorming, collaborative notes, and sharing drafts.



# List of Figures

2.1	Class Diagram for InternHub . . . . .	11
2.2	System Architecture Diagram for InternHub . . . . .	13
2.3	Signup state diagram. . . . .	14
2.4	Login state diagram. . . . .	15
2.5	State Diagram for Post Internship . . . . .	16
2.6	State Diagram for Applying for an Internship . . . . .	17
2.7	State Diagram for Interview Management . . . . .	18
2.8	State Diagram for Complaint Handling . . . . .	19
2.9	State Diagram for Profile Management . . . . .	20
2.10	State Diagram for Search and Filter . . . . .	21
3.1	Login mockup. . . . .	27
3.2	SignUp mockup. . . . .	27
3.3	Student Interface Diagram . . . . .	28
3.4	Company Interface Diagram . . . . .	29
3.5	Administrator Interface Diagram . . . . .	30
3.6	Use Cases Diagram for Unregistered Users. . . . .	36
3.7	Use Cases Diagram for Students . . . . .	36
3.8	Use Cases Diagram for Companies. . . . .	37
3.9	Use Cases Diagram for University Admin. . . . .	37
3.10	Sequence Diagram for Student Registration . . . . .	40
3.11	Sequence Diagram for Company Registration . . . . .	43
3.12	Sequence Diagram for Internship Posting . . . . .	46
3.13	Sequence Diagram for Student Application Process . . . . .	49
3.14	Sequence Diagram for Interview Management . . . . .	52
3.15	Sequence Diagram for Complaint Handling . . . . .	55
3.16	Sequence Diagram for Admin Privileges . . . . .	58
4.1	Scenario for Student Applies to an <i>Open Internship</i> . . . . .	72
4.2	Scenario for <i>Approved Internship</i> . . . . .	73

4.3	Scenario for <i>Valid Complaint</i> . . . . .	74
4.4	Instance for <i>BaseWorld Simplified</i> . . . . .	74
4.5	Simplified Instance for <i>BaseWorld Simplified 2</i> . . . . .	74
4.6	Simplified Instance for <i>BaseWorld Simplified 3</i> . . . . .	75

# List of Tables

1.1	Goals table.	3
1.2	Acronyms and terms used in the document.	6
1.3	Revision History	7
2.1	Domain Assumptions.	24
2.2	Dependencies.	25
2.3	Constraints.	25
3.7	Priority and Criticality Matrix	34
3.8	Student Registration Use Case.	39
3.9	Company Registration use case.	42
3.10	Internship Posting Use Case.	45
3.11	Student Application Process Use Case.	48
3.12	Interview Management Use Case.	51
3.13	Complaint Handling Use Case.	54
3.14	Admin Privileges Use Case.	57
5.1	Effort spent by each member of the group.	77

