



**POLITECNICO**  
MILANO 1863

# **Software Engineering 2**

## **Design Documentation**

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

**Samarth Bhatia - 11059097**

**Satvik Sharma - 11054680**

Academic Year: 2024-2025

Version: 2.0

Release date: 04/01/2025



# Contents

<b>Contents</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Purpose . . . . .	1
1.1.1 Scope . . . . .	2
1.1.2 Goals . . . . .	2
1.2 Definition, Acronyms, Abbreviations . . . . .	3
1.3 Revision History . . . . .	3
1.4 Reference Documents . . . . .	4
1.5 Document Structure . . . . .	4
<b>2 Architectural Design</b>	<b>7</b>
2.1 Overview . . . . .	7
2.2 Component View . . . . .	9
2.2.1 High-Level Diagram . . . . .	9
2.2.2 Low-Level Diagram . . . . .	11
2.2.3 Evaluation Manager . . . . .	12
2.2.4 Dashboard Manager . . . . .	14
2.2.5 Profile Manager . . . . .	15
2.2.6 Internship Manager . . . . .	16
2.2.7 Application Manager . . . . .	17
2.3 Deployment View . . . . .	19
2.4 Runtime View . . . . .	20
2.4.1 Sign-Up Process . . . . .	20
2.4.2 Login Process . . . . .	21
2.4.3 Post Internship . . . . .	22
2.4.4 Interview Management . . . . .	23
2.4.5 Complaint Handling . . . . .	24
2.4.6 Profile Management . . . . .	25

2.4.7	Search and Filter . . . . .	26
2.5	Component Interfaces . . . . .	26
2.6	Architectural Styles and Patterns . . . . .	29
2.6.1	Layered Architecture . . . . .	29
2.6.2	Client-Server Architecture . . . . .	30
2.6.3	Event-Driven Architecture . . . . .	30
2.6.4	Design Patterns . . . . .	31
2.7	Other Design Decisions . . . . .	34
2.7.1	Availability . . . . .	34
2.7.2	Scalability . . . . .	35
2.7.3	Security . . . . .	35
2.7.4	Notification Handling . . . . .	35
2.7.5	Document Management . . . . .	36
2.7.6	Data Persistence . . . . .	36
2.7.7	Performance Optimization . . . . .	36
<b>3</b>	<b>User Interface Design</b>	<b>39</b>
3.1	Overview . . . . .	40
3.2	User Interfaces . . . . .	41
3.2.1	UI1. Login . . . . .	41
3.2.2	UI2. SignUp . . . . .	43
3.2.3	UI3. Company Dashboard . . . . .	44
3.2.4	UI4. Post Internship . . . . .	45
3.2.5	UI5. Student Dashboard . . . . .	46
3.2.6	UI6. Build CV . . . . .	47
3.2.7	UI7. Messaging System . . . . .	48
<b>4</b>	<b>Requirements Traceability</b>	<b>49</b>
4.1	System Requirements . . . . .	49
4.1.1	Login Manager . . . . .	49
4.1.2	Registration Manager . . . . .	49
4.1.3	Internship Manager . . . . .	49
4.1.4	Application Manager . . . . .	50
4.1.5	Interview Manager . . . . .	50
4.1.6	Complaint Manager . . . . .	50
4.1.7	Profile Manager . . . . .	51
4.1.8	Search and Filter . . . . .	51
4.1.9	Dashboard Manager . . . . .	51

4.1.10	Notification Manager . . . . .	52
4.1.11	Model . . . . .	52
<b>5</b>	<b>Implementation, Integration and Test Plan</b>	<b>53</b>
5.1	Overview and Implementation Plan . . . . .	53
5.2	Features Identification . . . . .	53
5.2.1	[F1] Login and Registration Features . . . . .	53
5.2.2	[F2] Profile Management Features . . . . .	54
5.2.3	[F3] Internship Management Features . . . . .	54
5.2.4	[F4] Search and Filter Features . . . . .	54
5.2.5	[F5] Application and Interview Features . . . . .	54
5.2.6	[F6] Complaint Handling Features . . . . .	55
5.2.7	[F7] Notification Features . . . . .	55
5.2.8	[F8] Dashboard Features . . . . .	55
5.2.9	Development Dependencies . . . . .	55
5.3	Implementation Strategy . . . . .	56
5.3.1	Overview and Integration Plan . . . . .	56
5.4	System Testing Strategy . . . . .	62
5.4.1	Functional Testing . . . . .	62
5.4.2	Load Testing . . . . .	63
5.4.3	Performance Testing . . . . .	63
5.4.4	Stress Testing . . . . .	63
5.4.5	User Interface Testing . . . . .	64
5.4.6	Comprehensive Testing Approach . . . . .	64
<b>6</b>	<b>Effort Spent</b>	<b>65</b>
<b>7</b>	<b>References</b>	<b>67</b>
7.1	References . . . . .	67
<b>List of Figures</b>		<b>69</b>
<b>List of Tables</b>		<b>71</b>



# 1 | Introduction

## 1.1. Purpose

This Design Document provides a comprehensive overview of the **InternHub - Students & Companies (S&C)** platform. Its primary purpose is to serve as a guide for developers responsible for implementing the system's architecture and design and as a reference for clients and stakeholders to ensure alignment with the agreed-upon requirements and goals. Additionally, it offers a clear, precise, and unambiguous explanation of the platform's features, design decisions, and constraints. This ensures that all stakeholders, including students, companies, and academic institutions, have a shared understanding of how the system will function and deliver value. 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. Scope

By connecting students, businesses, and academic institutions, **InternHub - Students & Companies (SC)** is a platform that aims to improve and expedite the internship experience. It seeks to match students with appropriate internships, allowing universities to manage the full internship lifecycle and businesses to identify the best candidates.

Companies may post extensive internship openings, analyze applications, and oversee the selection process on the site, while students can create detailed profiles, apply for internships, and follow their progress. Universities serve as supervisors, guaranteeing adherence to academic standards and offering systems for observation and criticism.

The platform aims to streamline processes and promote openness and cooperation among stakeholders with features like intelligent recommendations, feedback and quality assurance tools, and strong communication channels. In order to guarantee a smooth, equitable, and effective experience for every user, it also incorporates analytics, training materials, and a grievance management system.

### 1.1.2. 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. 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.

## 1.3. Revision History

Version	Date	Description	Authors
1.0	03 January 2025	Initial Release	Shreesh Kumar Jha, Samarth Bhatia
2.0	04 January 2025	LaTeX Format and Minor Fixes	Shreesh Kumar Jha, Samarth Bhatia

**Table 1.3:** Revision History

## 1.4. Reference Documents

- Reference to Previous Year Student Projects for Structuring the Document
- Specification Document Assignment

## 1.5. Document Structure

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

**Introduction:** In the first section, the importance of the Design Document is established, and acronyms and abbreviations are defined and explained in detail. The **Intern-Hub - Students & Companies (S&C)** platform's goals, scope, and purpose are also described.

**Architectural Design:** A thorough explanation of the system's primary parts and how they work together is given in the second section. In order to guarantee scalability, dependability, and efficiency, this section also covers important design choices, architectural styles, patterns, and paradigms.

**User Interface Design:** The platform's user interface is described in the third section, which includes mockups and thorough explanations of the main pages and user workflows. The goal of this part is to make sure that everyone involved has an easy-to-use and accessible experience.

**Requirements Traceability:** The fourth step ensures that all functionality and restrictions are sufficiently addressed by the design decisions made by mapping the system's design back to the established requirements.

**Implementation, Integration, and Testing Plan:** The strategy for implementing the platform's components and integrating them into a unified system is described in the fifth section. Additionally, it offers a thorough testing strategy to confirm the platform's performance and functioning.

**Effort Spent:** In the sixth section are included information about the number of hours each group member has worked for this document.

**References:** The publications, resources, and standards consulted in the production of this Design Document are listed in the last section. It is a tool for comprehending the rationale behind and background of the design choices.



# 2 | Architectural Design

## 2.1. Overview

Here we represent an overview of how the entire **InternHub** S&C architecture is composed of:

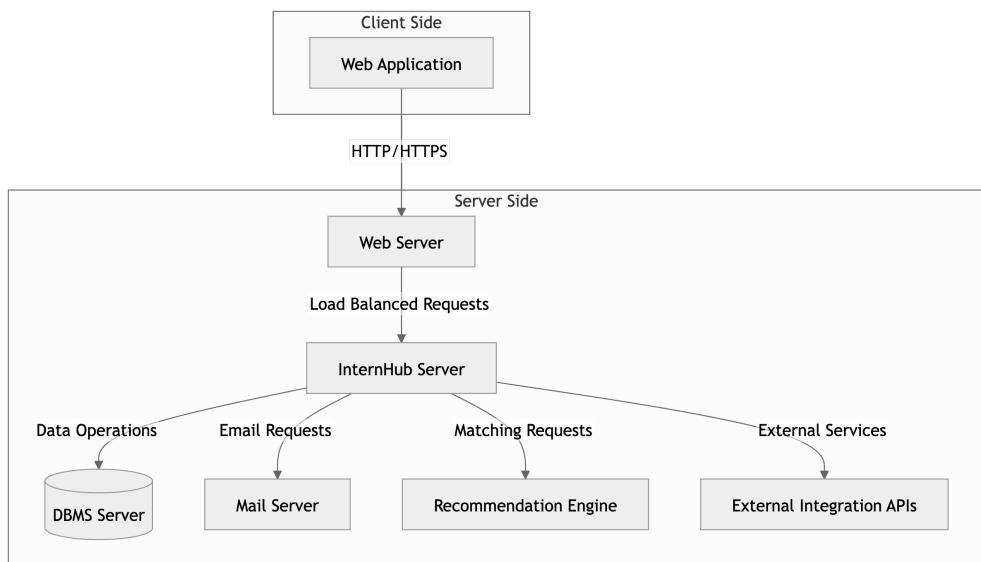


Figure 2.1: InternHub Overview

## Client Side

- **Web Application:** The main user interface is the web application, which makes the platform accessible to all users—students, companies, and university administration. It makes it possible to do things like register, maintain profiles, post and apply for internships, handle complaints, and examine analytics.

## Server Side

- **Web Server:** Carries out user communications, taking in and processing their requests. For incoming requests, it offers load balancing and divides them among

several S&C Server replicas. To guarantee safe access, it also controls user sessions.

- **S&C Server:** The center of the platform, which houses all of the interactions. It makes it easier for the database, Web Server, and external APIs to communicate with one another. To manage heavy traffic and guarantee availability, the S&C Server is replicated over several computers.
- **DBMS Server:** Serves as the primary store for user, internship, application, feedback, and complaint data. It facilitates effective data retrieval and storage for all platform features.
- **Mail Server:** Manages email correspondence, including notifications for internships, changes, and user registration confirmation emails. By informing stakeholders, it improves the user experience.
- **Recommendation Engine API:** Uses advanced algorithms to recommend suitable internships to students based on their profiles and preferences, as well as potential candidates to companies.
- **Analytics Engine API:** Enables stakeholders to make data-driven decisions by offering insights and producing reports about internships, applications, and platform usage.
- **External Tools API:** Enables the safe storage and retrieval of internship-related files, including contracts and certificates, by facilitating integration with document management systems and other outside services.

## 2.2. Component View

### 2.2.1. High-Level Diagram

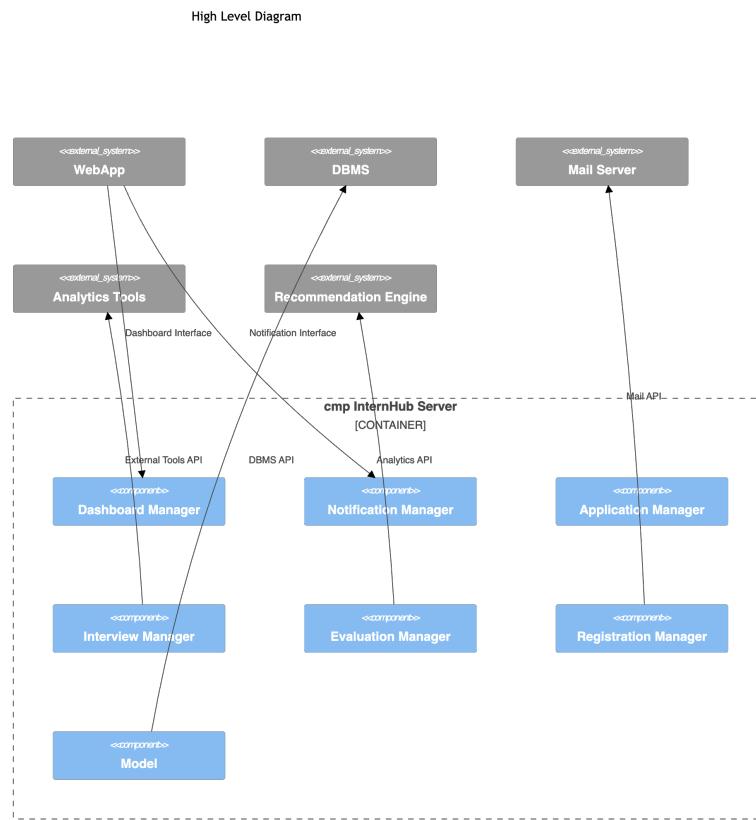


Figure 2.2: High Level Diagram

The external components of S&C are depicted in the high-level component diagram of S&C above, along with their methods of communication with the S&C server.

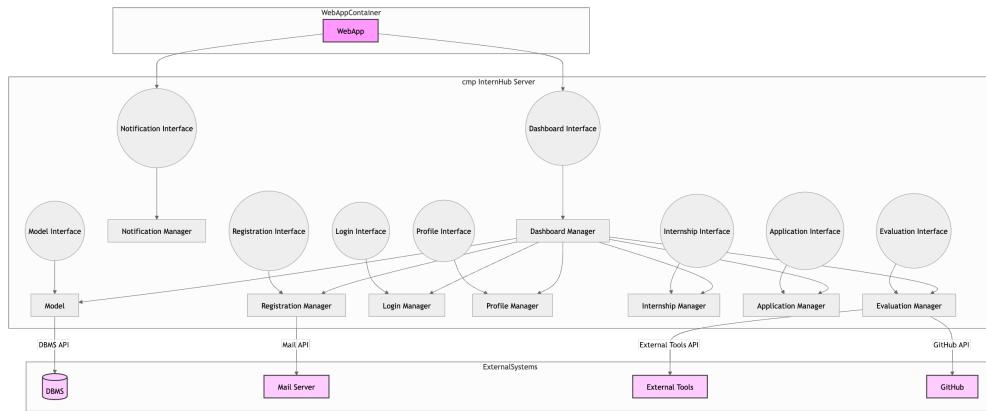
- **WebApp:** Facilitates connectivity with the S&C Server via the Dashboard Interface, the main channel for client-server communication, by acting as the external access point for users (students, businesses, and university administrators). Through the Notification Interface, the S&C Server additionally notifies users of updates, reminders, and internship matches.
- **DBMS:** Serves as a storehouse for user profiles, applications, feedback, complaints, and internship posts. Through the DBMS API, which is controlled by the Model

Component, it interacts with the S&C Server.

- **Mail Server:** Manages email correspondence, including alerts for internships and confirmation emails sent after user registration. Through the Mail API, which is connected to the User Registration Manager component, the Mail Server can communicate with the S&C Server.
- **Recommendation Engine:** Employs sophisticated algorithms to suggest internships to prospective employers and students. Through the Recommendation API, which is incorporated into the Matching Engine, it interacts with the S&C Server.
- **Analytics Engine:** Provides data, analytics, and insights on platform activities, including system utilization, application success rates, and internship patterns. It uses the Analytics API to communicate with the S&C Server.
- **Document Management System:** Makes it easier to store and retrieve internship-related paperwork securely, including contracts, certifications, and feedback reports. Through the Document Management API, which is incorporated into the File Manager component, it interfaces with the S&C Server.
- **External System:**
  - WebApp connects to the Dashboard Manager and Notification Manager using respective interfaces.
  - DBMS connects to the Model using the DBMS API.
  - Mail Server communicates with the Registration Manager through the Mail API.
  - Recommendation Engine and Analytics Tools interact with the Notification Manager.
- **InternHub Server:** The InternHub Server consists of core components like:
  1. Dashboard Manager
  2. Notification Manager
  3. Application Manager
  4. Registration Manager
  5. Evaluation Manager
  6. Interview Manager

## 7. Model for centralized data access

### 2.2.2. Low-Level Diagram



**Figure 2.3: Low Level Diagram**

The figure above represents the detailed architecture of the InternHub – Students & Companies (S&C) platform, showing the key components within the InternHub Server and their interactions.

- **Dashboard Manager:** The essential element that plans out all user-platform communication. The Dashboard Manager routes user queries to the appropriate parts of InternHub, while the Dashboard Interface is how users engage with the platform. User activities including managing profiles, looking for internships, and receiving notifications all take place there.
- **Model Component:** Acts as a bridge interface to the DBMS Server and represents the data on the server. It guarantees that all components use the DBMS API to safely and effectively access data.
- **Registration Manager:** Manages the registration of new users, including companies, universities, and students. The Registration Manager handles requests made through the Registration Interface. It interacts with the Model Component to add new user information to the database and with the Mail Server via the Mail API to deliver confirmation emails.
- **Login Manager:** Oversees the registered users' login procedure. The Login Manager handles user login requests sent through the Login Interface. It collects user data from the Model Component and verifies credentials.

- **Profile Manager:** Enables users to search for and browse other users' profiles in addition to managing their own. In order to retrieve or change profile data, the Profile Manager interacts with the Model Component via the Profile Interface.
- **Internship Manager:** Oversees all internship-related activities, such as job advertisements, applications, and status reports.
- **Application Manager:** Manages internship applications, including submission, alerts, and status monitoring. It interacts with the Model Component to retrieve and update data and handles requests through the Application Interface.
- **Evaluation Manager:** Handles the evaluation of students' success during internships.
- **Notification Manager:** In charge of managing every notification that users get. It handles requests through the Notification Interface, including alerting users to platform events, application updates, and internship postings. It interacts with other managers for event triggers and the Model Component for storing notification data.

### 2.2.3. Evaluation Manager

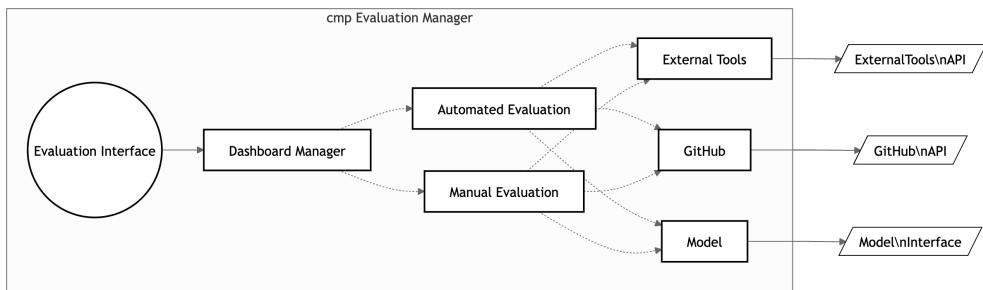


Figure 2.4: Evaluation Manager

The InternHub – Students & Companies (S&C) platform's Evaluation Manager is designed to evaluate and oversee students' performance reviews during internships. To manage various evaluation techniques, it is divided into two sub-components: Automated Evaluation and Manual Evaluation.

#### Automated Evaluation Component

The Automated Evaluation component is utilized by the InternHub system to automatically assess a student's performance or task completion during their internship. The

workflow is as follows:

1. Through the Evaluation Interface, the system notifies the Automated Evaluation Component when an internship milestone or work submission takes place.
2. Through the External Tools API, the Automated Evaluation component forwards the submitted work to outside tools for evaluation, testing, or scoring (e.g., project testing or task completion verification).
3. The Automated Evaluation component uses the Model Interface to communicate with the Model Component after receiving the evaluation results from the external tools.
4. In turn, the Model Component uses the DBMS API to update the student's score in the database, guaranteeing that the evaluation results are appropriately recorded in the appropriate database area.

## Manual Evaluation Component

The Manual Evaluation component allows companies or university administrators to manually assess a student's performance during their internship. The process is as follows:

1. The evaluator (company or administrator) initiates a request via the WebApp, which connects to the Dashboard Interface.
2. Through the Evaluation Interface, the request is sent to the Manual Evaluation Component.
3. The GitHub API or other integrated technologies are used by the Manual Evaluation component to retrieve the student's submitted work or internship data.
4. The Manual Evaluation component uses the Model Interface to provide the updated results to the Model Component following the evaluator's manual analysis and scoring.
5. The Model Component ensures that the outcomes of the manual evaluation are accurately documented by updating the scores or evaluations in the database using the DBMS API.

## 2.2.4. Dashboard Manager

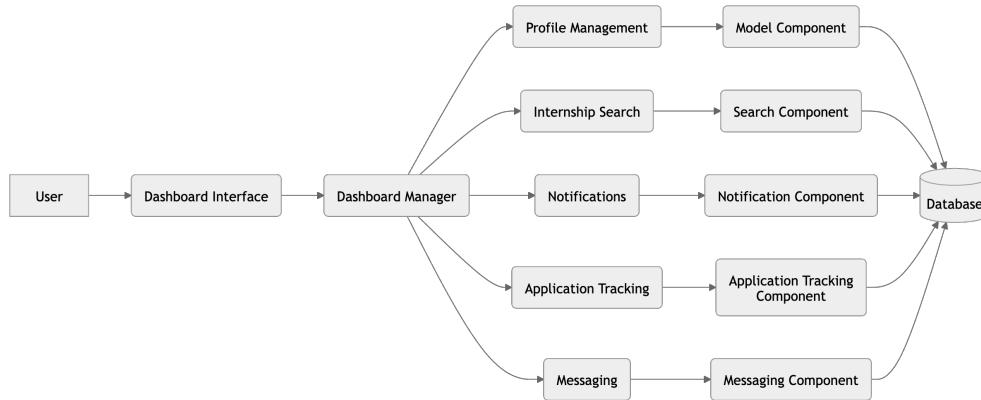


Figure 2.5: Dashboard Manager

The Dashboard Manager is a pivotal component in the InternHub – Students & Companies (S&C) platform, managing user interactions and orchestrating communication between various subsystems. The following outlines its subcomponents and their interactions:

**Profile Management** Users can manage and update their profiles, which include contact details, preferences, and resumes.

- In order to retrieve or change user data, the Model Component communicates with the Database and receives requests pertaining to profile management.

**Internship Search** Allows users to search and filter internships based on location, preferences, and skills.

- These requests are handled by the Search Component, which queries the database and provides users with pertinent results through the Dashboard Interface.

**Notifications** Oversees the distribution of user notifications, including critical system messages, profile recommendations, and updates to internship applications.

- Interacts with the Notification Component, which provides notification logs and status updates to the database.

**Application Tracking** Allows users to keep track of the acceptance, rejection, and company updates on their internship applications.

order to retrieve and present current application statuses, the Application Tracking Component communicates with the database.

**Messaging** Facilitates direct communication between administration, businesses, and students to address questions or discuss internships.

- Communicates with the Messaging Component, which guarantees that every conversation is recorded and safely kept in the database.

### 2.2.5. Profile Manager

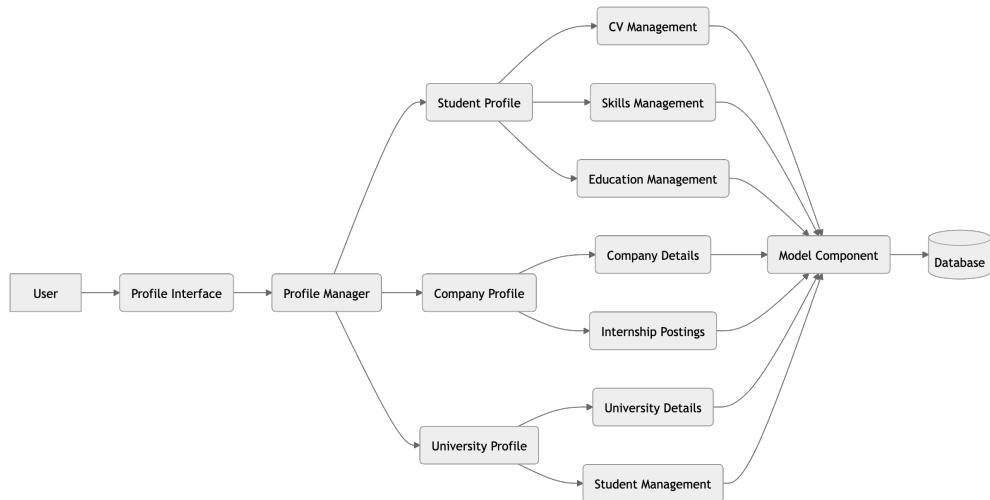


Figure 2.6: Profile Manager

The InternHub – Students & Companies (S&C) platform’s Profile Manager is a key feature that helps institutions, businesses, and students manage and streamline profile-related tasks. To efficiently handle user profile data, the component ensures seamless communication between the Model Component and the database. These are the main subcomponents and their responsibilities:

**Student Profile** Manages student-specific details, including academic achievements, skills, and internship preferences.

- **CV Management:** Handles the creation, modification, and storage of student CVs.
- **Skills Management:** Allows students to add, update, or remove skills based on their expertise.

- **Education Management:** Manages educational history and academic records.

**Company Profile** Responsible for maintaining company-related data, such as organizational details, internship postings, and company preferences.

- **Company Details:** Stores and updates company information, such as name, size, industry, and contact details.
- **Internship Postings:** Enables companies to create, modify, and manage internship listings.

**University Profile** Oversees university-specific information and ensures administrators have tools to manage student profiles and internships.

- **University Details:** Handles institutional information, including department details and associated staff.
- **Student Management:** Allows universities to monitor and manage student data and progress.

### 2.2.6. Internship Manager

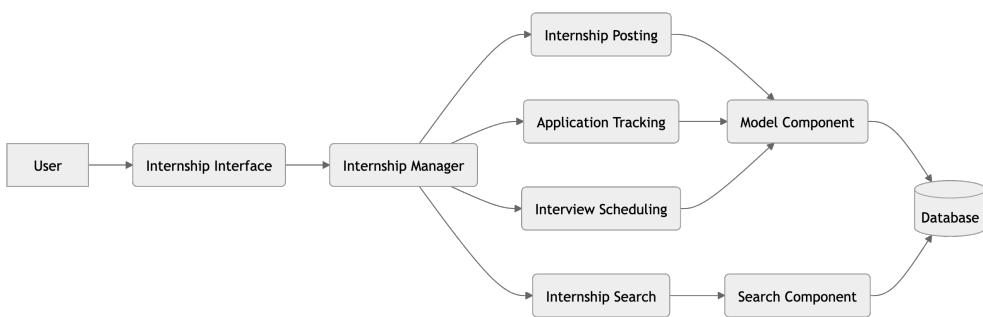


Figure 2.7: Internship Manager

Managing the internship lifecycle is the primary responsibility of the InternHub – Students & Companies (S&C) platform’s Internship Manager. It facilitates effective handling of internship-related features by managing user interactions with the platform. Below is a summary of its constituents and interrelations:

#### Internship Posting

- Allows companies to create, update, and manage internship opportunities.

- Interacts with the Model Component to store and retrieve data from the Database.
- Ensures all postings are accessible to students via the Internship Search feature.

### Application Tracking

- Enables students and companies to monitor the status of internship applications in real-time.
- Communicates with the Model Component to update and retrieve application details from the Database.
- Provides notifications to users regarding application progress or results.

### Interview Scheduling

- Handles the scheduling of interviews between students and companies as part of the internship process.
- Retrieves and updates scheduling data in the Database through the Model Component.

### Internship Search

- Helps students search for internships based on criteria such as location, skills, or duration.
- Uses the Search Component to query the Database and fetch matching results.
- Provides personalized recommendations based on student profiles and preferences.

#### 2.2.7. Application Manager

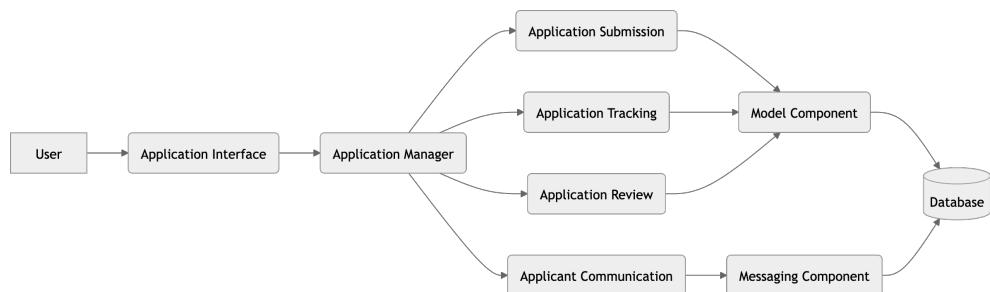


Figure 2.8: Application Manager

The Application Manager is responsible for handling all operations related to internship applications, ensuring seamless interaction between students, companies, and the platform.

### **Application Submission**

- Manages the submission of internship applications by students.
- Interacts with the Model Component to store the application data in the Database.

### **Application Tracking**

- Allows students and companies to monitor the status of submitted applications.
- Communicates with the Model Component to retrieve and update application status in the Database.

### **Application Review**

- Enables companies to review applications submitted by students.
- Retrieves application details from the Model Component and updates the status post-review in the Database.

### **Applicant Communication**

- Facilitates direct communication between applicants and companies regarding applications.
- Utilizes the Messaging Component to send and receive messages, with all interactions logged in the Database.

### 2.3. Deployment View

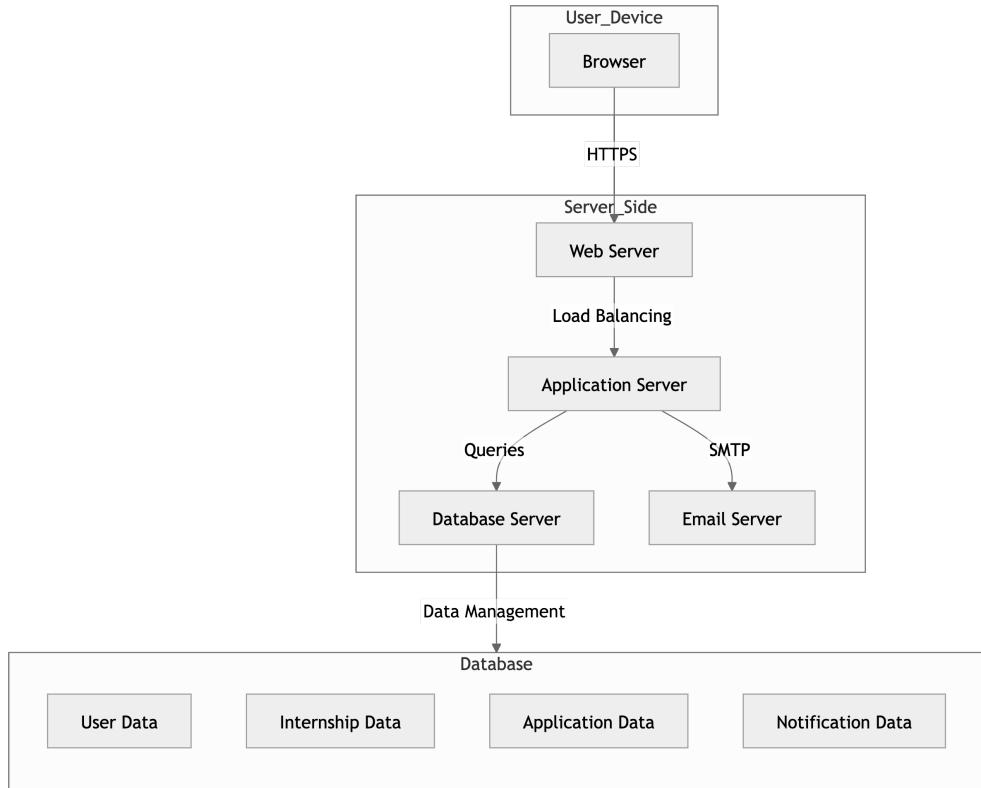


Figure 2.9: Deployment View

The platform's deployment architecture is composed of the following layers:

#### Client Side

- User devices, which are mostly accessed through web browsers, make up the client side. These devices stand in for platform users, which include companies, educational institutions, and students.
- Users engage with platform services such as applications, notifications, internship posts, profile management, registration, and login using their browsers.
- HTTPS ensures data privacy and protection by securing communication between the user and the platform.

#### Server Side

- **Web Server:** serves as the gateway for all requests from clients. By allocating incoming requests to several application server replicas, it performs load balancing,

guarantees secure HTTPS connections, and controls the delivery of static content (HTML, CSS, and JavaScript).

- **Application Server:** acts as the main backend, coordinating with other parts and processing business logic. Notification delivery, internship management, and user management are all handled by it. For data storage and retrieval, it interfaces with the DBMS Server; for automated communication, it integrates with the Email Server.
- **DBMS Server:** keeps application, user, and internship-related data. Notifications are recorded, and effective data updates, retrieval, and storage are guaranteed.
- **Email Server:** uses the SMTP protocol to handle outward messages, including application updates and registration confirmations.

## 2.4. Runtime View

### 2.4.1. Sign-Up Process

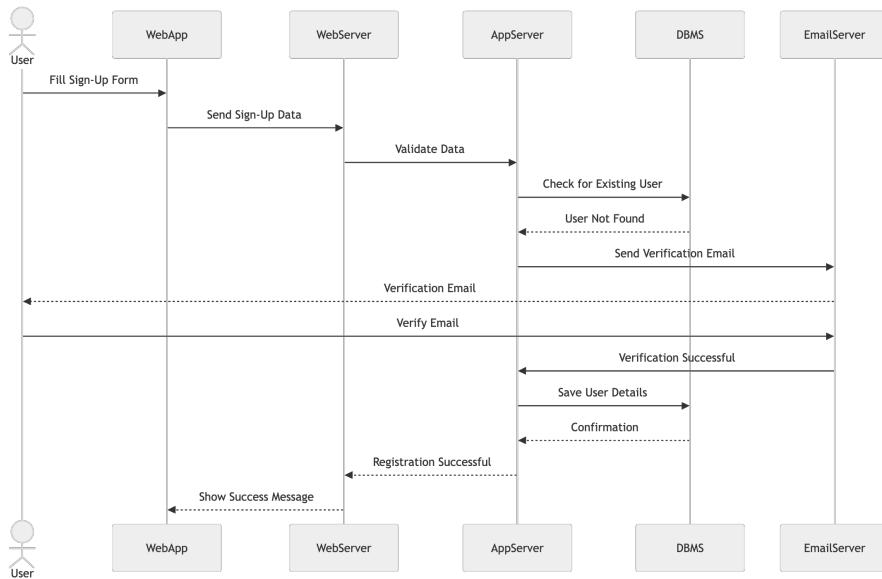


Figure 2.10: Runtime SignUp

The platform's sign-up process allows new users (students, companies, or university administrators) to register. The process is as follows:

1. Through the WebApp's sign-up form, the user provides their name, email address, password, and user role.

2. After confirming the inputs, the Web Server securely sends the data to the Application Server.
3. The DBMS Server is used by the Application Server's Registration Manager component to search the database for duplicate records.
4. The Email Server sends a verification email if the user is unique.
5. The Application Server validates successful registration and saves the user's information in the database after confirming the email.

#### 2.4.2. Login Process

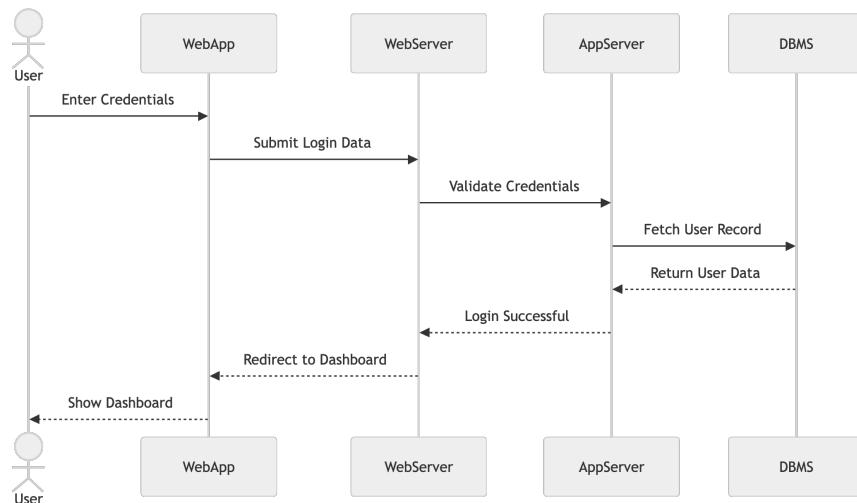


Figure 2.11: Runtime Login

The login process ensures secure access for registered users. The steps are as follows:

1. The user inputs their credentials on the WebApp's login form.
2. The Application Server receives the data safely from the Web Server.
3. Through the DBMS Server, the Application Server's Login Manager compares the credentials against database entries.
4. The system creates a session token and logs the user into the platform if the credentials match.
5. To ensure easy and safe access, the user is taken to their dashboard.

### 2.4.3. Post Internship

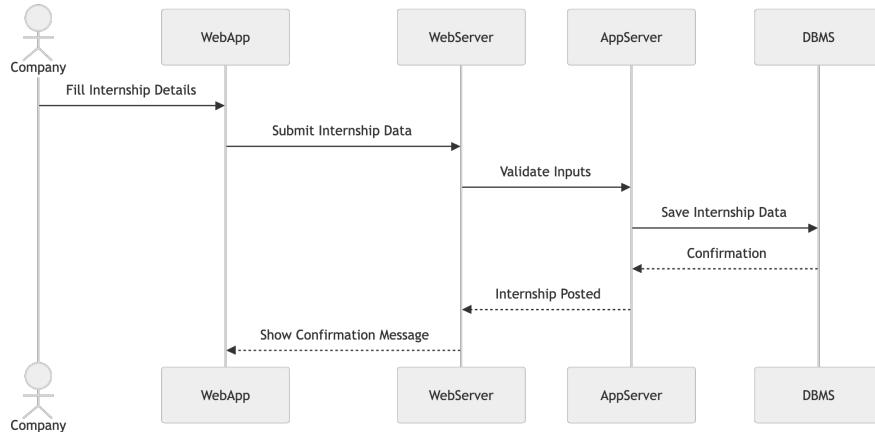


Figure 2.12: Runtime Post Internship

Posting an internship is an essential function for company users. The process is as follows:

1. By completing the internship details using the WebApp, including the job title, description, requirements, and duration, the company user starts the process.
2. After receiving this data, the Web Server sends it to the Application Server for validation.
3. The new internship posting is stored in the database by the Internship Manager component, which handles data processing and communication with the DBMS Server.
4. A confirmation message is sent to the company user when the data has been successfully stored.

This process ensures that the posted internship is immediately visible to qualified students looking for opportunities.

#### 2.4.4. Interview Management

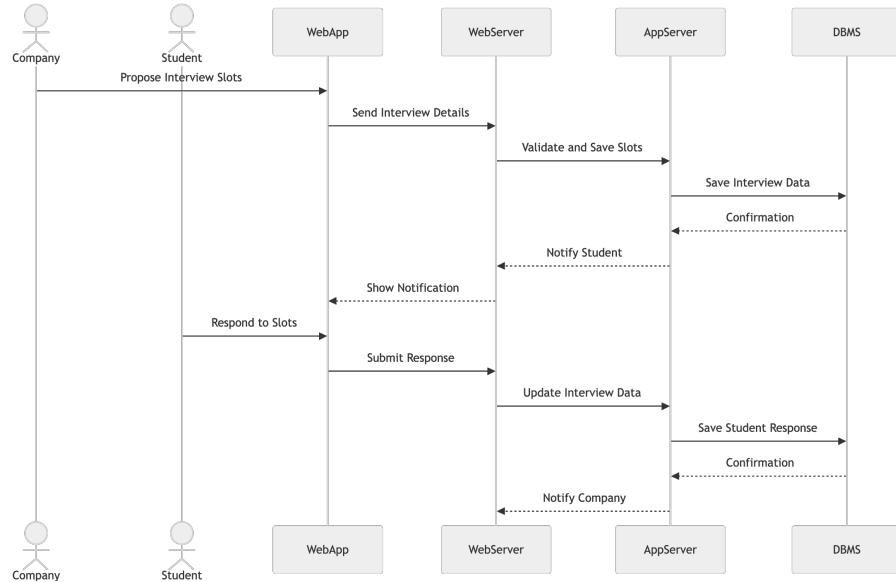


Figure 2.13: Runtime Interview Management

The interview management process streamlines scheduling and coordination between students and companies. The steps involved are:

1. Employers use the WebApp to suggest interview schedules. After passing via the Web Server, the data is sent to the Application Server.
2. The DBMS Server is used by the Interview Manager component to store the suggested timeslots in the database after verifying them.
3. The Notification Manager and Email Server are used to inform students of the recommended timeslots.
4. Through the WebApp, students can reply to the suggested timeslots, and the Interview Manager updates their answers in the database.
5. The business receives notifications verifying the student's choice.

This efficient process ensures seamless scheduling and communication for both parties.

### 2.4.5. Complaint Handling

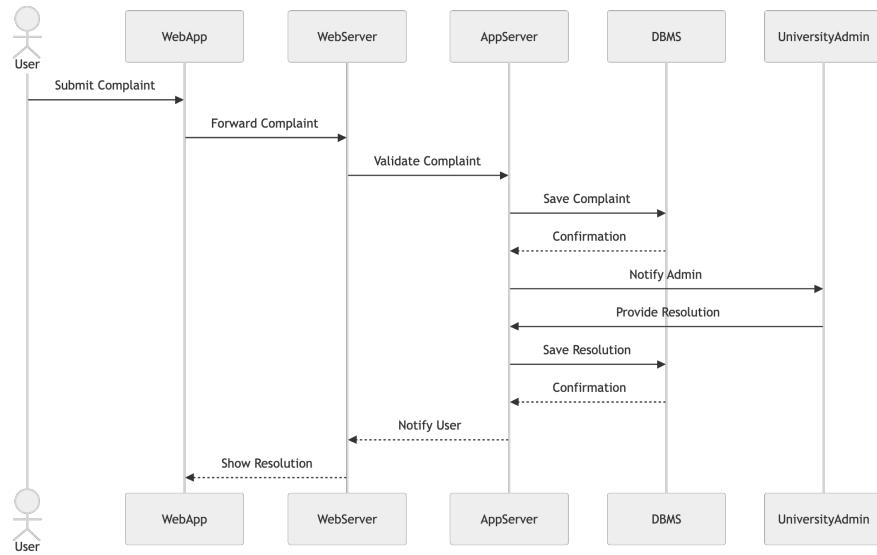


Figure 2.14: Complaint Handling

The complaint-handling process is designed to efficiently address user complaints. The steps are as follows:

1. Complaints submitted by users via the WebApp are routed to the Web Server and subsequently to the Application Server.
2. The Complaint Manager component uses the DBMS Server to store the complaint in the database after verifying it.
3. After being informed of the complaint, the relevant university representative investigates it and offers a remedy.
4. The user is informed of the resolution through the Email Server and Notification Manager.
5. For future use, the resolution is noted in the database.

This transparent process ensures that all complaints are handled promptly and efficiently.

### 2.4.6. Profile Management

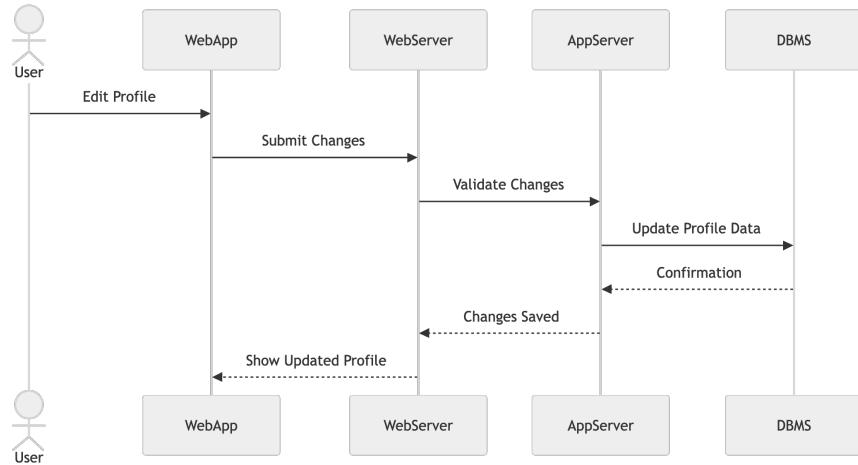


Figure 2.15: Profile Management

Profile management allows users to update and maintain their personal information. The process includes:

1. Updates are started by users by inputting information including contact data, talents, and educational background on the WebApp's profile page.
2. The Application Server receives these changes from the Web Server.
3. In order to store the updated data in the database, the Profile Manager contacts the DBMS Server and confirms the changes.
4. A success message is displayed to the user when the modifications have been saved.

This feature ensures that user profiles remain up-to-date and relevant for internship matching and application processes.

### 2.4.7. Search and Filter

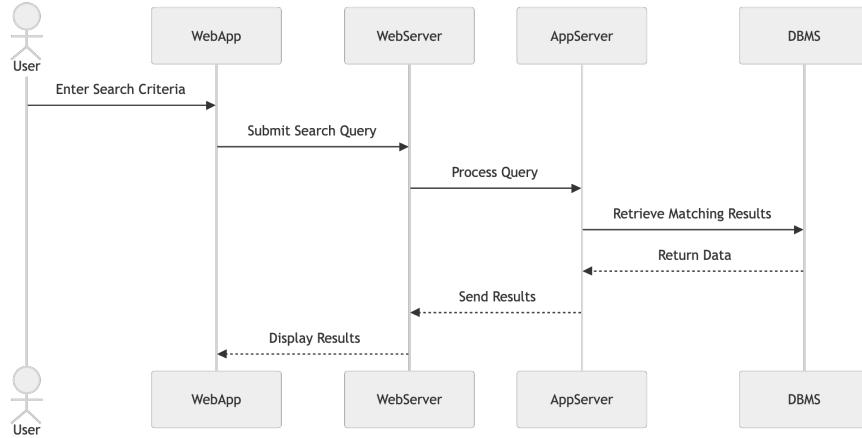


Figure 2.16: Search and Filter

The search and filter process enables users to quickly find relevant information. The workflow is as follows:

1. Using the WebApp, users may search for internships by using filters like location, needed skills, and duration.
2. These filters are sent to the Application Server by the Web Server.
3. In order to process the query and obtain relevant results, the Search Component communicates with the DBMS Server.
4. The WebApp receives the results from the Web Server and displays them for users to see.

This functionality helps users efficiently navigate the platform's data and find the most relevant opportunities.

## 2.5. Component Interfaces

### Health Manager

- **getHealthStatus():** Verifies the health of the server and ensures it is running properly. Returns a status code and a message indicating the server status.

### Login Manager

- **Login(String nickname, String password):** Validates user credentials and initiates a session for the user.

### Registration Manager

- **CreateAnAccount():** Initiates the account creation process.
- **registerStudent(Map<String, String> studentDetails):** Registers a new student with details like name, email, and profile information.
- **registerRecruiter(Map<String, String> recruiterDetails):** Registers a new recruiter with details like company name, email, and profile information.

### Internship Manager

- **addInternship(Map<String, String> internshipDetails):** enables recruiters to add a new internship following authorization validation.
- **deleteInternship(String internshipID):** After verifying recruiter rights, a specified internship is deleted.
- **fetchAllInternships(String recruiterID):** retrieves every internship that a particular recruiter has posted.
- **fetchInternshipById(String internshipID):** uses its ID to retrieve information on a particular internship.

### Application Manager

- **SubmitApplication(String internshipID, String studentNickname, String resume):** gives students the opportunity to apply for a particular internship.
- **ViewApplications(String internshipID):** retrieves every applicant submitted for a specific internship.
- **UpdateApplicationStatus(String applicationID, String status):** updates a submitted application's status.

### Interview Manager

- **ProposeInterview(String internshipID, List<Date> availableDates):** enables companies to suggest times for interviews for a particular internship.

- **RespondToInterview(String interviewID, String response):** gives students the option to accept or reject an invitation to an interview.
- **GetInterviewDetails(String interviewID):** Retrieves details about a specific interview.

### Complaint Manager

- **SubmitComplaint(String userID, String description):** enables people to complain about a problem.
- **ViewComplaint(String complaintID):** brings up information on a certain complaint.
- **ResolveComplaint(String complaintID, String resolution):** provides an update on the complaint's status and outcome.

### Profile Manager

- **EditProfile(String nickname, Map<String, String> updatedDetails):** Updates the user's profile information.
- **ViewProfile(String nickname):** retrieves a user's profile information.
- **SearchProfiles(String keyword):** uses the supplied keyword to search for profiles.

### Notification Manager

- **SendNotification(String userID, String message):** gives a user a notification.
- **ViewNotifications(String userID):** brings up every notification for a particular user.

### OTP Manager

- **requestOTP(String email):** Requests an OTP for email-based authentication.
- **verifyOTP(String email, String otp):** confirms the OTP supplied for email authentication.

### Token Manager

- **generateAccessToken(String email, String userType):** confirms the OTP supplied for email authentication.

## Dashboard Manager

- **GetDashboardData(String userID):** gives the user who is logged in a summary of the dashboard.
- **ViewStatistics(String userID):** retrieves statistics information pertinent to the user's actions.

## 2.6. Architectural Styles and Patterns

### 2.6.1. Layered Architecture

#### Front-End Layer

- Implements responsive web interfaces for three distinct user types:
  - **Student Portal:** focuses on managing applications, searching for internships, and maintaining profiles.
  - **Company Portal:** manages prospects, reviews applications, and posts internships.
  - **Admin Portal:** offers system monitoring, complaint handling, and oversight tools.
- Uses modern web technologies with responsive design principles.
- carries out state management and client-side validation.

#### Services Layer (Business Logic)

- **User Management Service:** manages user profiles, rights, authorization, and authentication.
- **Matching Engine Service:** carries out the filtering and sorting of results, processes search queries, and applies sophisticated algorithms for matching internship students.
- **Recommendation System:** evaluates internship requirements and student profiles to provide tailored recommendations that are updated in response to user interactions.
- **Interview Management Service:** oversees the scheduling of interviews, keeps track of interview status updates, and collects feedback.

## Data Layer

- **Database Management System:** keeps organized information such as applications, internships, and user profiles. preserves relationship mappings and guarantees the consistency and integrity of data.
- **File Storage System:** oversees the safe uploading and downloading of files, manages the storage of documents (certificates, resumes), and applies file versioning.

### 2.6.2. Client-Server Architecture

#### Client Side Implementation

- uses a browser-based responsive user interface that can be adjusted to fit various screen sizes and devices.
- applies client-side validations and offers a consistent user experience.

#### Real-Time Updates

- Uses WebSocket connections for instant notifications.
- Updates UI states that preserve synchronized views without requiring a page refresh.

#### Server Side Implementation

- **RESTful API Services:** handles request/response processing, API versioning, and the implementation of defined HTTP endpoints.
- **Backend Processing:** Executes business logic operations, processes data transformations, and manages system state.

### 2.6.3. Event-Driven Architecture

#### Event Publishers

- **Application System:** activates notification workflows, creates events for status changes, and updates relevant entities.
- **Interview System:** creates events, notifies users of reminders, and modifies calendar integrations.
- **Complaint System:** Notifies the appropriate administrators, initiates escalation events, and modifies the tracking status.

### Event Subscribers

- **Notification Service:** sends emails and alerts, handles event notifications, and modifies user dashboards.
- **Analytics Service:** maintains reporting metrics, creates usage statistics, and keeps track of system occurrences.

#### 2.6.4. Design Patterns

##### 1. Model-View-Controller (MVC)

- **Models:**

- **User Model:**

- \* Base user properties and behaviors.
    - \* Specialized student, company, and admin classes.
    - \* Data validation rules.

- **Application Model:**

- \* Application states and transitions.
    - \* Document attachments.
    - \* Status tracking.

- **Internship Model:**

- \* Position details and requirements.
    - \* Application management.
    - \* Scheduling information.

- **Views:**

- **User Interfaces:**

- \* Role-specific dashboards.
    - \* Form components.
    - \* Interactive elements.

- **Data Presentation:**

- \* Lists and tables.
  - \* Search results.
  - \* Statistical reports.
- **Controllers:**
    - **Authentication Controller:**
      - \* Login/logout handling.
      - \* Session management.
      - \* Access control.
    - **Application Controller:**
      - \* Process submissions.
      - \* Status updates.
      - \* Document handling.
- ## 2. Observer Pattern
- **Subject Components:**
    - **Internship Posting System:**
      - \* Notifies matched students.
      - \* Updates search results.
      - \* Triggers recommendations.
    - **Application Processing:**
      - \* Updates stakeholders.
      - \* Triggers next steps.
      - \* Maintains status logs.
- **Observer Components:**
    - **Student Notifications:**
      - \* New matching internships.
      - \* Application updates.

- \* Interview schedules.

- **Company Notifications:**

- \* New applications.
- \* Interview confirmations.
- \* Status changes.

### 3. Factory Pattern

- **User Factory:**

- Creates appropriate user types.
- Initializes role-specific properties.
- Sets up required relationships.

- **Document Factory:**

- Generates different document types.
- Handles format conversions.
- Creates appropriate storage entries.

### 4. State Pattern

- **Application States:**

- Submitted.
- Under Review.
- Interview Scheduled.
- Accepted/Rejected.
- Completed.

- **Complaint States:**

- Filed.
- Under Investigation.
- Resolved.
- Closed.

- Escalated.

## 5. Strategy Pattern

- Matching Strategies:

- Skills-based matching.
- Location-based matching.
- Experience-level matching.
- Industry-specific matching.

- Search Strategies:

- Keyword search.
- Filter-based search.
- Category-based search.
- Combined search approaches.

### Benefits of Implementing Design Patterns:

- Clear separation of concerns.
- Maintainable and scalable codebase.
- Robust error handling.
- Secure data management.
- Efficient performance.
- Real-time responsiveness.
- Consistent user experience.

## 2.7. Other Design Decisions

### 2.7.1. Availability

- Redundant systems guarantee uninterrupted, 24/7 platform operation.
- During interruptions, a failover mechanism allows for automatic shift to backup systems.

- Data integrity is guaranteed by error handling procedures.
- System recovery procedures after failures.
- Transaction integrity for important procedures such as scheduling interviews and submitting applications.

### 2.7.2. Scalability

- Front-end layer scalability for multiple user portals (Student, Company, Admin).
- Services layer that can scale independently based on demand:
  - User Management Service.
  - Matching Engine.
  - Recommendation System.
  - Interview Management Service.
- Systems for databases and storage that can handle increasing amounts of data.
- API-first approach enabling future mobile app development.

### 2.7.3. Security

- Robust authentication systems for user verification.
- authorization for sensitive operations based on roles.
- End-to-end data encryption.
- GDPR compliance measures.
- Secure document storage and handling.
- routine security monitoring and audits.

### 2.7.4. Notification Handling

- Notifications in real time when the status of an application changes.
- Interview scheduling alerts.
- Complaint status updates.
- Notifications about important occurrences via email.

- Integration of a calendar to schedule interviews.
- Platform-wide announcements capability.

### 2.7.5. Document Management

- Certificates, formal documents, and resumes should all be stored securely.
- Version control for user documents.
- Document validation and verification.
- Format standardization.
- Access control based on user roles.
- Automated capacity for processing documents.

### 2.7.6. Data Persistence

- Centralized database for core functionality:
  - User profiles.
  - Internship listings.
  - Applications.
  - Interview records.
  - Complaints.
- File storage system for documents.
- Data backup and recovery systems.
- Audit logging for critical operations.
- Data retention policies compliance.

### 2.7.7. Performance Optimization

- Page load optimization (<5 seconds).
- Search results delivery (<2 seconds).
- File upload handling (<10 seconds for 10MB).
- Real-time updates (<1 second).

- Concurrent user support (up to 50 users).
- Database transaction optimization (10 transactions per second).

**Alignment with Platform Goals:** These design decisions align with the platform's core requirements while ensuring:

- Reliable service delivery.
- Secure data handling.
- Efficient performance.
- User satisfaction.
- System maintainability.
- Regulatory compliance.
- Future scalability.



# 3 | User Interface Design

This section outlines the **InternHub – Students & Companies (S&C)** platform's user interface, including a summary of the main pages that make up the system. Since changes to the design may be made throughout the testing phase, the design mockups displayed here focus more on the interface's usability and interaction dynamics than on its final visual aesthetics. Equivalent pages will be made for mobile devices, even if the desktop browser version is the main focus due to its suitability for thorough profile management and internship-related procedures. By adapting and scaling the interface to fit smaller screens, this guarantees a seamless user experience.

The design prototypes shown here serve as a fundamental depiction of the platform's user interface, as specified in the **RASD**. As the system develops, these designs are subject to optimization and improvement, taking into account input from user interactions and testing.

### 3.1. Overview

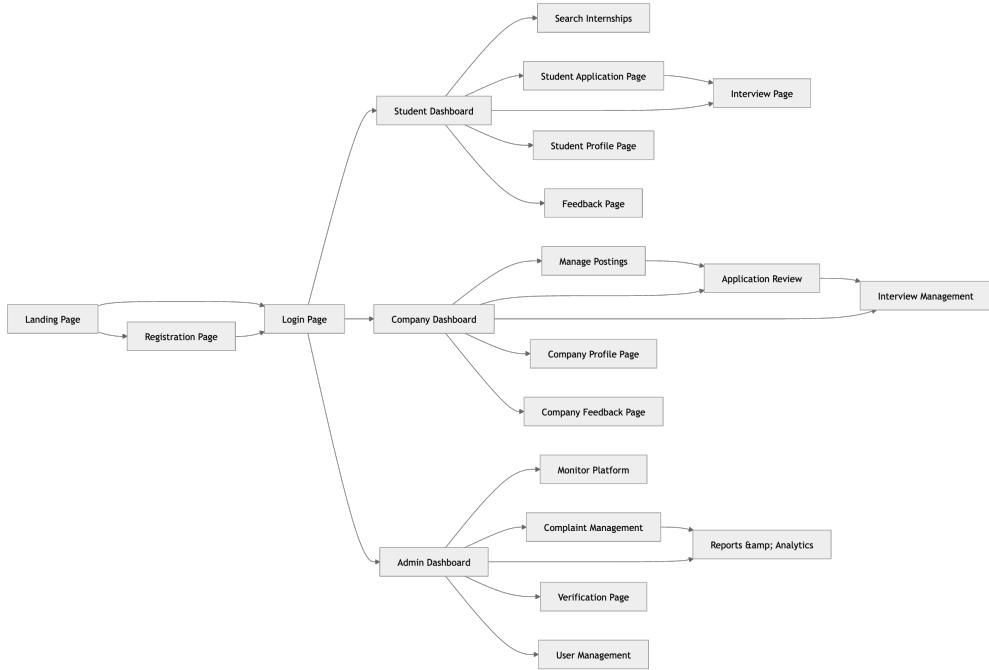


Figure 3.1: UI Overview

This graphic shows the flow and relationships between distinct pages designed for different user roles, giving a thorough overview of the **InternHub – Students & Companies (S&C)** platform's user interface (UI) architecture.

1. **Landing Page:** Points users to the Login Page for current users or the Registration Page for new users.
  2. **Role-Specific Dashboards:**
    - **Student Dashboard:** Access to the Student Application Page, Feedback Page, Student Profile Page, Interview Management, and Search Internships.
    - **Company Dashboard:** supports hiring processes by giving users access to the Feedback Page, Company Profile Page, and tools for managing job postings and applicant reviews.
    - **Admin Dashboard:** Features like User Management, Complaint Management, Verification Page, and Reports & Analytics allow for administrative monitoring.

This structured design emphasizes role-specific features and smooth navigation, ensuring

an effective user experience for all stakeholders. The diagram's links illustrate the dynamic and scalable nature of the interface, tailored to the diverse needs of administrators, businesses, and students.

## 3.2. User Interfaces

### 3.2.1. UI1. Login

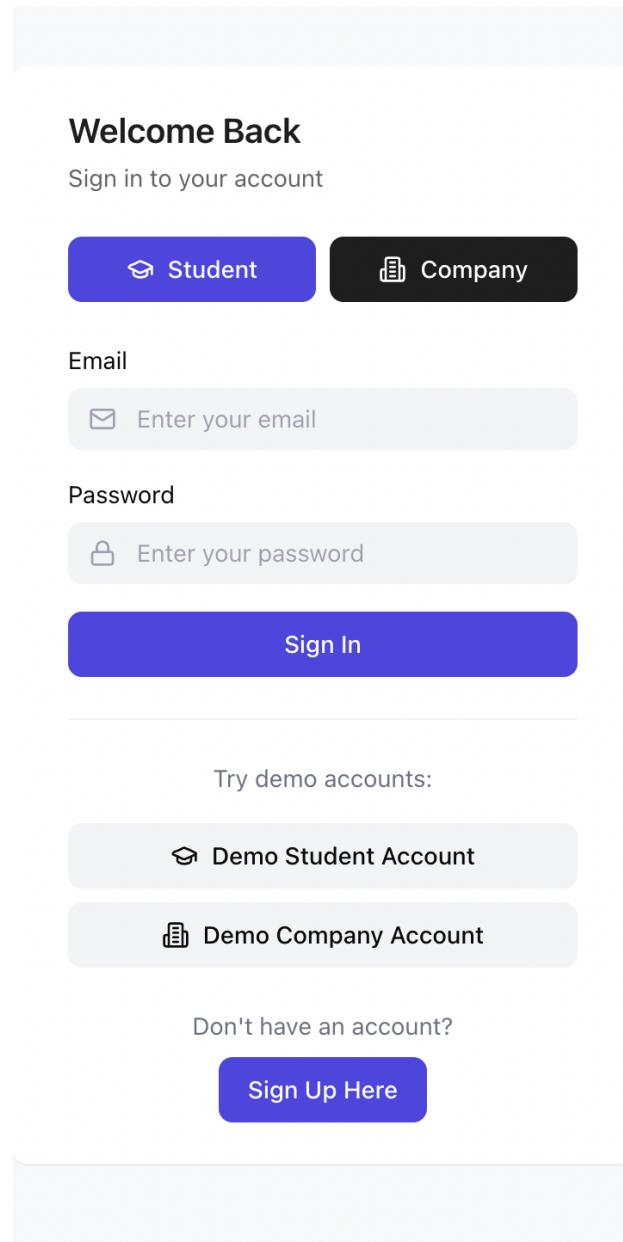


Figure 3.2: Login Interface Mockup

The login page is designed to provide a seamless access experience for the two main user roles: companies and students.

1. **Role Toggle:** Users can choose their role using a toggle option at the top, guaranteeing a customized login experience.
2. **User Input Fields:** With the help of user-friendly placeholders, users enter their email address and password.
3. **"Sign In" Button:** Prominently displayed for easy access.
4. **Demo Accounts:** Allows users to test features without signing up.
5. **"Sign Up Here" Link:** Redirects new users to the registration page.

The simple, clear design ensures an intuitive experience for all users while maintaining a professional appearance.

### 3.2.2. UI2. SignUp

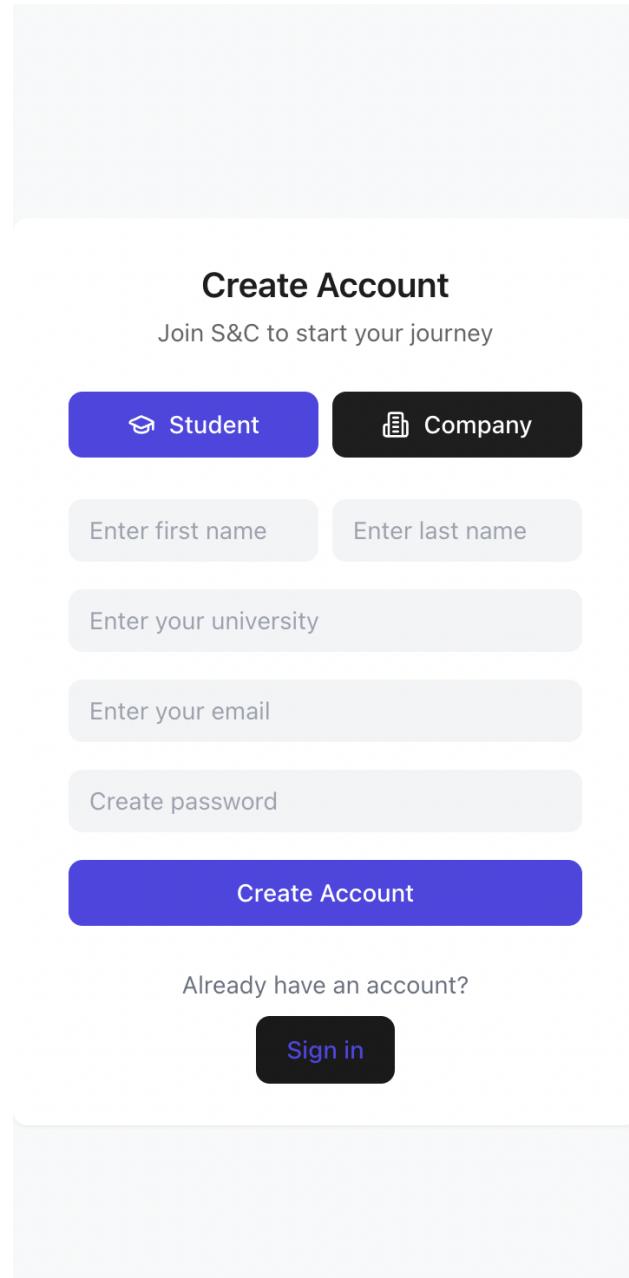


Figure 3.3: SignUp Interface Mockup

The registration page provides a streamlined and user-friendly interface for new users to create accounts.

1. **Role Selection:** At the top of the website, users have the option to switch between the Student and Company roles.
2. **Input Fields:** Consists of the student's university, email, password, and first and

last names.

3. "Create Account" Button: Simplifies the submission process.
4. "Sign In" Link: Redirects existing users to the login page.

This design guarantees a professional and user-friendly experience for new users entering the platform.

### 3.2.3. UI3. Company Dashboard

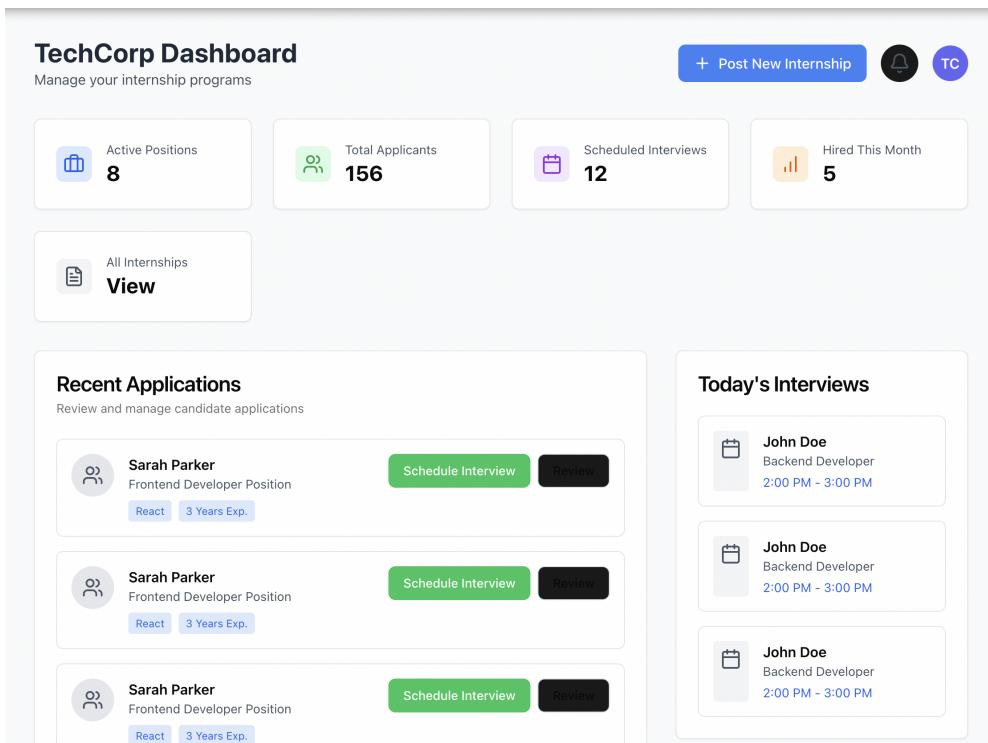


Figure 3.4: Company Dashboard

The Company Dashboard offers a well-structured interface for recruiters to efficiently manage their internship programs.

1. **Dashboard Overview:** shows important information like Hired This Month, Total Applicants, Scheduled Interviews, and Active Positions.
2. **Post Internship:** An easy-to-use button for publishing new internship opportunities.
3. **Recent Applications:** shows applicant submissions together with pertinent information such as name, experience, talents, and position applied for. includes choices

for reviewing applications or setting up interviews.

4. **Today's Interviews:** Displays a brief interview schedule with the names, positions, and times of the candidates.

This layout ensures all essential information is easily accessible, enabling recruiters to navigate and complete their tasks effectively.

### 3.2.4. UI4. Post Internship

The image shows a modal dialog box titled "Post New Internship". Inside the dialog, there are five input fields stacked vertically: "Title", "Description" (which includes a scroll bar), "Location", "Duration (in months)", and "Stipend (in USD)". At the bottom of the dialog are two buttons: "Cancel" on the left and "Post Internship" on the right, both in white text on their respective colored backgrounds.

Figure 3.5: Post Internship

Recruiters can post internship opportunities to the platform using a structured interface by utilizing the **Post New Internship** form. The form includes fields for essential details such as:

1. **Title:** The name of the internship position.
2. **Description:** Comprehensive details regarding the position and duties.
3. **Location:** The physical or remote location of the internship.
4. **Duration:** Length of the internship in months.
5. **Stipend:** Details regarding the compensation offered.

The form is designed with clearly defined and spaced fields to ensure clarity and ease of use. At the bottom, a **Post Internship** button enables submission of the internship details, while a **Cancel** button allows recruiters to terminate the process. This arrangement prioritizes simplicity and functionality, enabling businesses to efficiently communicate openings to prospective candidates.

### 3.2.5. UI5. Student Dashboard

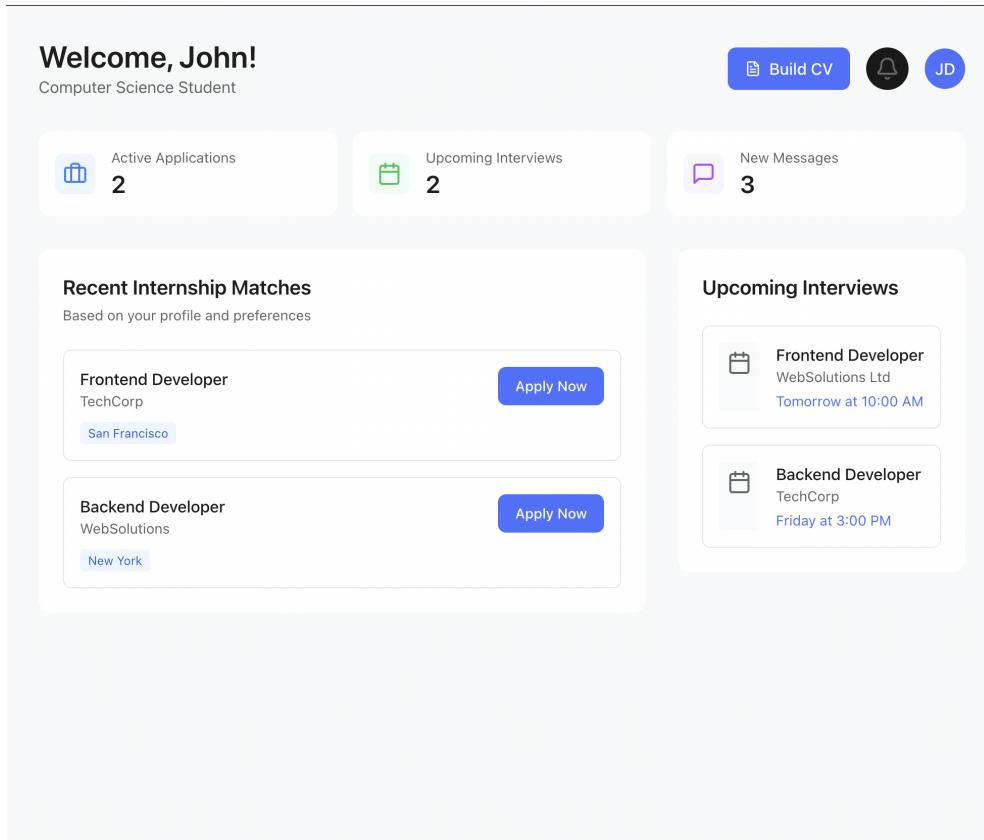


Figure 3.6: Student Dashboard

The **Student Dashboard** provides a customized and streamlined interface to help students effectively manage their internship applications and search processes. Key features include:

1. **Overview:** Shows important indicators like New Messages, Upcoming Interviews, and Active Applications.
2. **Build CV:** A button to access resume-building resources.
3. **Recent Internship Matches:** Shows internship opportunities tailored to the student's profile and preferences, including roles like Frontend Developer and Backend Developer, with company names, locations, and an **Apply Now** button for quick action.
4. **Upcoming Interviews:** Shows the time, employer, and position details for the planned interviews.

The layout emphasizes accessibility and relevance, ensuring students can quickly find and act on the most important information to enhance their internship experience.

### 3.2.6. UI6. Build CV

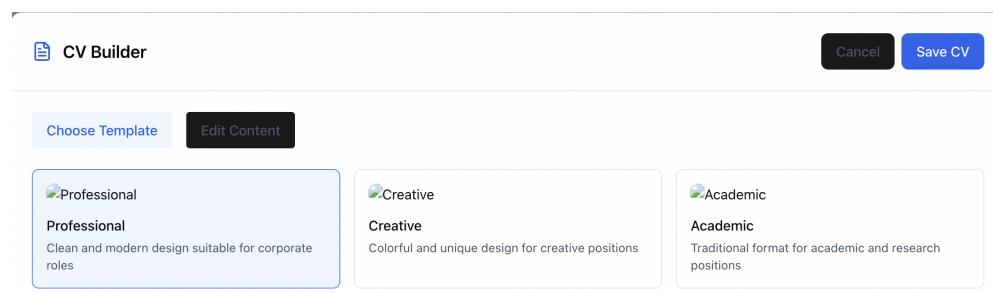


Figure 3.7: Build CV

The **CV Builder** interface provides a simplified and well-structured layout that enables users to create and modify resumes. The interface includes two primary steps:

1. **Choose Template:** To help users choose the ideal format for their career goals and the kind of position they are aiming for, users can choose from three options: Professional, Creative, and Academic. Each choice has an explanation.
2. **Edit Content:** Users can personalize their resumes by adding their education, employment experience, talents, and personal details.

Users can **Cancel** changes or **Save CV** to finalize their resume using buttons in the upper-right corner. This design ensures users can efficiently create resumes tailored to their professional needs while maintaining simplicity and clarity throughout the process.

### 3.2.7. UI7. Messaging System

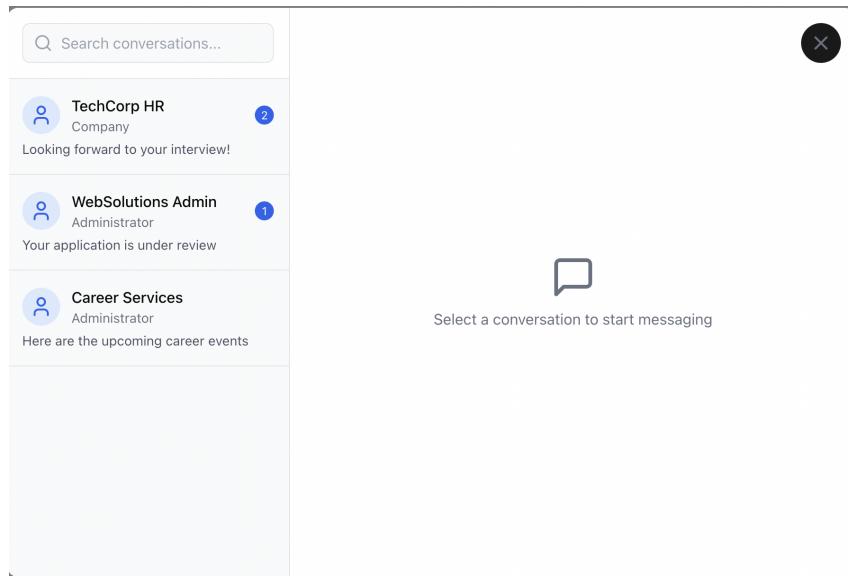


Figure 3.8: Messaging System

The **Messaging System** interface enables users to have seamless conversations with various stakeholders on the **InternHub – Students & Companies (S&C)** platform. Key features include:

1. **Conversation List:** Arranged on the left, displaying ongoing conversations together with information such as the contact's name, role (e.g., Administrator, Company), and a synopsis of the most recent message. The quantity of unread messages is indicated by blue badges.
2. **Chat Panel:** The chosen chat is shown in the right panel. Here, users can send and view messages. A placeholder asks the user to select a discussion if none is selected.
3. **Search Bar:** Located at the top of the conversation list, it allows users to look for particular conversations.

This split-pane design ensures users can easily navigate their conversations and focus on ongoing chats without interruptions. The interface supports real-time updates, facilitating efficient communication and collaboration between administrators, businesses, and students.

# 4 | Requirements Traceability

## 4.1. System Requirements

This section outlines the functional requirements of the InternHub – Students & Companies (S&C) platform, categorized by the major system components.

### 4.1.1. Login Manager

- R1: The system allows registered students to log in.
- R2: The system allows registered companies to log in.
- R3: The system ensures secure access to accounts through credential verification.
- R4: The system validates user input during login.
- R5: The system provides error messages for invalid credentials.

### 4.1.2. Registration Manager

- R6: The system allows unregistered users (students and companies) to sign up.
- R7: The system verifies user details before creating accounts.
- R8: The system communicates with the mailing system to send verification emails during registration.
- R9: The system ensures unique email addresses during registration.
- R10: The system allows recruiters to register company profiles with required information fields.

### 4.1.3. Internship Manager

- R11: The system allows companies to create new internship postings.

- R12: The system allows companies to update existing internship details.
- R13: The system enables recruiters to delete internships.
- R14: The system fetches all internship postings associated with a specific recruiter.
- R15: The system provides students with the ability to search and filter internships based on preferences.
- R16: The system supports adding multiple job roles under a single internship posting.
- R17: The system tracks the total number of internships posted by a company.

#### **4.1.4. Application Manager**

- R18: The system allows students to submit applications for internships.
- R19: The system enables recruiters to review student applications.
- R20: The system tracks the status of submitted applications.
- R21: The system supports updating the status of an application (e.g., accepted, rejected).
- R22: The system provides recruiters with filters to search through applications.
- R23: The system notifies students of changes in their application status.

#### **4.1.5. Interview Manager**

- R24: The system allows recruiters to schedule interviews for shortlisted candidates.
- R25: The system notifies students of upcoming interviews.
- R26: The system tracks interview details and schedules.
- R27: The system sends reminders to students and recruiters for scheduled interviews.
- R28: The system allows students to view interview details, including time and interviewer.
- R29: The system supports rescheduling of interviews by recruiters.

#### **4.1.6. Complaint Manager**

- R30: The system allows users (students and companies) to file complaints regarding issues.

- R31: The system ensures proper logging of all complaints for administrative review.
- R32: The system enables administrators to resolve complaints and update their status.
- R33: The system notifies users about updates to their complaints.
- R34: The system maintains a history of resolved complaints for audit purposes.

#### **4.1.7. Profile Manager**

- R35: The system allows students to manage their profiles, including uploading resumes and updating details.
- R36: The system allows companies to manage their profiles and update organization details.
- R37: The system supports the ability to view profiles of other users (students or companies).
- R38: The system provides recommendations for internships based on student profiles.
- R39: The system highlights incomplete profiles for users and prompts them to complete missing details.

#### **4.1.8. Search and Filter**

- R40: The system provides students and companies with advanced search and filter options for internships, candidates, or postings.
- R41: The system ensures search results are relevant and aligned with user preferences.
- R42: The system supports search by location, duration, stipend, and skills.
- R43: The system allows recruiters to filter student applications by skills and experience.
- R44: The system enables sorting of search results by relevance or other criteria.

#### **4.1.9. Dashboard Manager**

- R45: The system provides students with an overview of their active applications, upcoming interviews, and recent internship matches.
- R46: The system provides companies with an overview of active postings, total applicants,

and scheduled interviews.

R47: The system allows administrators to monitor platform activities, including user engagement and complaint handling.

R48: The system displays key metrics for students, such as total applications and new messages.

R49: The system allows companies to see statistics on applicants and hires.

#### 4.1.10. Notification Manager

R50: The system sends notifications to students when their application status is updated.

R51: The system notifies students and companies of scheduled interviews.

R52: The system sends reminders for upcoming deadlines, including applications and interviews.

R53: The system notifies users about updates to their complaints.

R54: The system alerts students about new internship matches based on their profiles.

#### 4.1.11. Model

R55: The system securely stores user data, including profiles, applications, and complaints.

R56: The system ensures the integrity and security of all data through encryption and access controls.

R57: The system logs all user actions for security and audit purposes.

R58: The system supports scalable storage for managing increasing user data.

R59: The system ensures efficient data retrieval for dashboard and search functionalities.

# 5 | Implementation, Integration and Test Plan

## 5.1. Overview and Implementation Plan

This chapter describes the InternHub – Students & Companies (S&C) platform’s integration strategy, test plan, and implementation procedure. A systematic and effective development process will be ensured by using the **Bottom-Up approach**.

The implementation will start with basic, independent modules that do not need additional modules to work. Drivers for testing each module separately will be created. Modules will gradually be added to the system, taking the place of their associated drivers as they are put into place and tested. For further testing, each integrated module will need its own driver.

Before a system is fully integrated, smaller functional subsystems can be created using the Bottom-Up approach. With this incremental approach:

- Testing is performed on smaller parts of the system initially and continues for each module as it becomes ready, making debugging and error tracking easier.
- Parallel development is facilitated, allowing separate teams to work on various elements at the same time.

## 5.2. Features Identification

The features of the platform are prioritized based on their dependencies and importance, as outlined below:

### 5.2.1. [F1] Login and Registration Features

- These are the core features required for students, companies, and administrators to access the platform.

- Include user registration, login, and secure authentication.
- As foundational features, they will be implemented first to support the functioning of subsequent features.

### 5.2.2. [F2] Profile Management Features

- This set of features includes creating and managing profiles for students, companies, and administrators.
- Students can update personal details, upload CVs, and showcase skills.
- Companies can maintain organization profiles.
- These features serve as the foundation for the search and application functionalities.

### 5.2.3. [F3] Internship Management Features

- Includes the ability for companies to create, update, and delete internship postings.
- Involves managing applications received for these internships.
- Requires proper implementation of profile management ([F2]).
- Will be developed subsequently after [F1] and [F2].

### 5.2.4. [F4] Search and Filter Features

- This includes submitting applications, reviewing them, scheduling interviews, and notifying users about interview updates.
- These features depend on the successful implementation of profile and internship management ([F2] and [F3]).

### 5.2.5. [F5] Application and Interview Features

- Includes submitting applications, reviewing them, scheduling interviews, and notifying users about interview updates.
- These features depend on internship management ([F3]) and profile management ([F2]).

### 5.2.6. [F6] Complaint Handling Features

- Allows students and companies to lodge complaints and administrators to review and resolve them.
- Ensures user satisfaction and platform reliability.
- Relies on the proper implementation of profile management ([F2]) and dashboard functionalities ([F8]).

### 5.2.7. [F7] Notification Features

- Ensure that students, companies, and administrators are notified about critical events such as interview schedules, application updates, and complaint resolutions.
- These features will be developed last as they depend on the correct functioning of all other features.

### 5.2.8. [F8] Dashboard Features

- The dashboard provides an overview of active internships, applications, interviews, and platform activities for students, companies, and administrators.
- Consolidates data from various modules.
- Critical for system usability.

### 5.2.9. Development Dependencies

The dependencies between the features ensure a structured and incremental implementation:

F1 Login and Registration Features serve as the foundation for all other functionalities.

F2 Profile Management Features are prerequisites for Internship Management ([F3]) and Search and Filter ([F4]).

F3 Internship Management Features depend on Profile Management ([F2]) and support Application and Interview Features ([F5]).

F4 Search and Filter Features depend on both Profile Management ([F2]) and Internship Management ([F3]).

F5 Application and Interview Features depend on Profile Management ([F2]), Intern-

ship Management ([F3]), and Search and Filter ([F4]).

F6 Complaint Handling Features rely on Profile Management ([F2]) and Dashboard Features ([F8]).

F7 Notification Features depend on the correct functioning of all other features.

F8 Dashboard Features consolidate data from all modules and rely on their successful implementation.

This dependency-based plan ensures that features are developed systematically, reducing errors and facilitating incremental testing.

## 5.3. Implementation Strategy

### 5.3.1. Overview and Integration Plan

A systematic **bottom-up method** is used to integrate the system's components. The emphasis will be on developing solid foundational modules that can be gradually combined, starting with the essential elements. Before each module is integrated into the larger system, it will be tested using the relevant drivers. This approach enables:

- Parallel programming.
- Effective debugging.
- Incremental functional validation.

The following crucial areas will be included in the integration process:

**Core Model Integration** The core model integration, which includes the data models necessary for the platform's operation, serves as the system's cornerstone. These include:

- **User Model:** Manages user-related information.
- **Resume Model:** Handles CVs and profile details.
- **Internship Model:** Stores internship data.
- **Application Model:** Tracks internship applications.
- **Interview Model:** Manages interview scheduling and feedback.
- **Complaint Model:** Logs and tracks complaints.

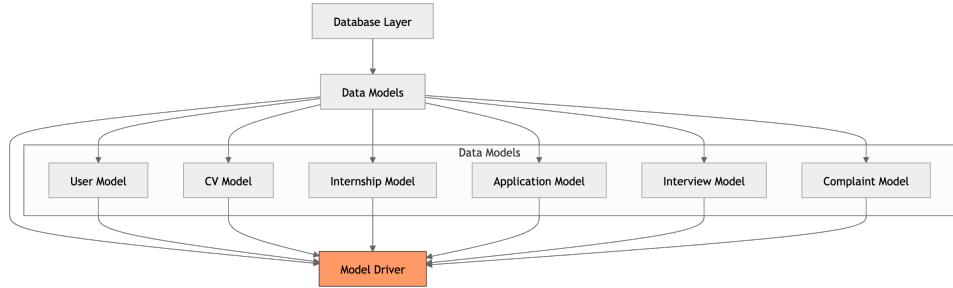


Figure 5.1: Core Model Integration

In order to carry out CRUD activities and guarantee data consistency and integrity, each of these data models communicates with the database layer. To test each of these models separately and confirm that they can manage database layer interactions efficiently, a Model Driver will be put in place. These models will be incorporated into the corresponding manager components after they have been validated.

**Authentication Integration** The procedures involved in user registration and login are the main focus of the authentication integration. Interactions between the data models and the user interfaces for registration and login will be managed by the **Authentication Manager**. Key functionalities include:

- Secure communication with the Registration System.
- Data validation.
- Access token generation.
- Credential validation workflows.

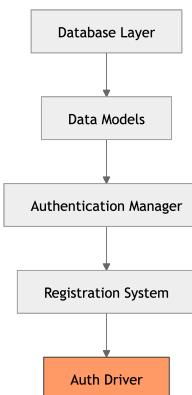


Figure 5.2: Authentication Integration

Workflows for authentication, such as the creation of access tokens and credential validation, will be tested using a driver. The Authentication Manager will be connected with the larger system to facilitate user authentication and smooth login once it is stable.

**Profile Management Integration** The **Profile Manager**, which manages the creation and administration of admin, company, and student profiles, is implemented as part of the Profile Management Integration. To guarantee that only authorized users are able to manage their profiles, this module communicates directly with the Authentication Manager. Key features include:

- Profile creation.
- Profile editing.
- Profile retrieval.

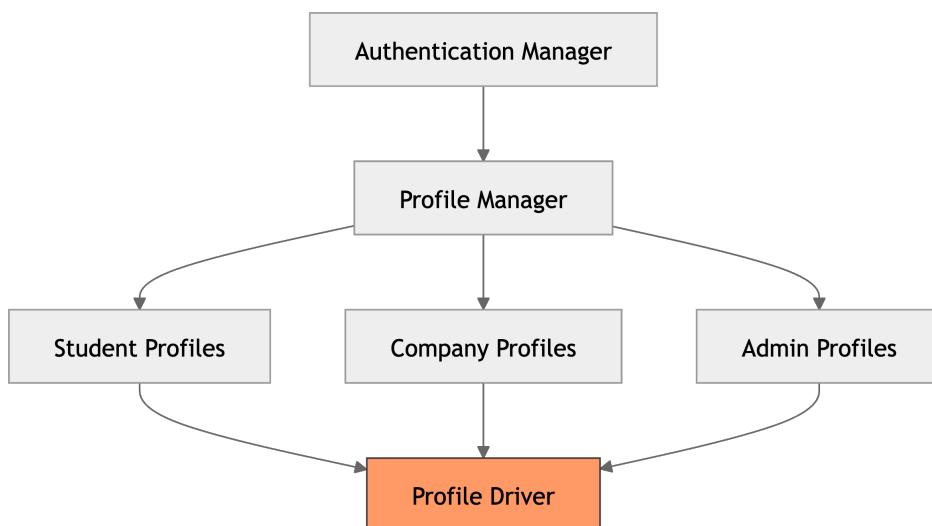


Figure 5.3: Profile Management Integration

The **Profile Driver** will test these features to validate functionality. This integration is critical for enabling personalized user experiences and creating user identification across the platform.

**Internship Management Integration** The **Internship Manager** is at the center of the Internship Management Integration, coordinating the posting and administration of internship opportunities. It integrates with:

- The Profile Manager to verify recruiter responsibilities and permissions.

- The Application System to manage student applications and facilitate filtering and internship searches.

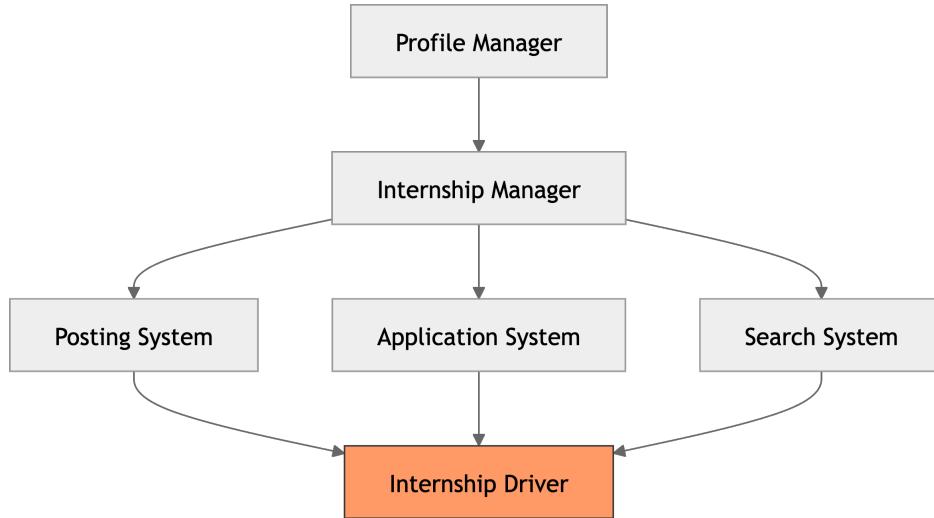
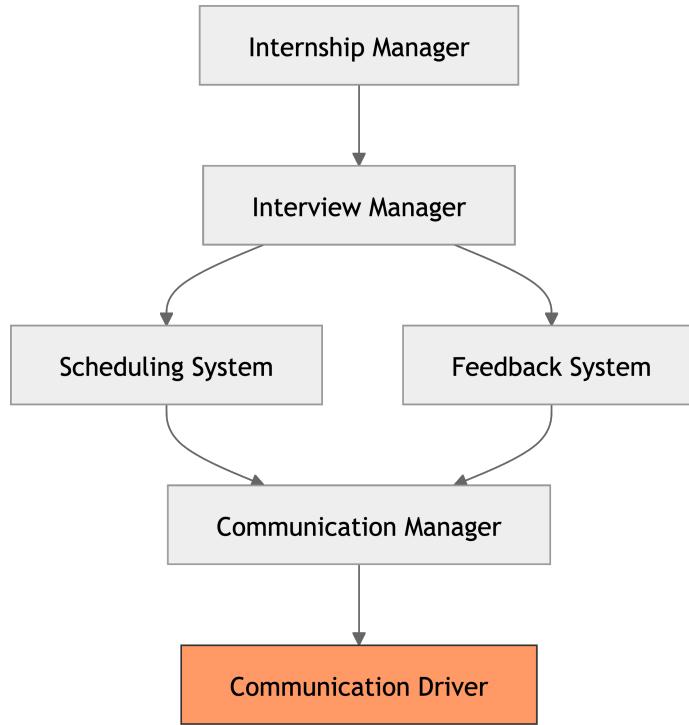


Figure 5.4: Internship Management Integration

A driver will test functionalities such as posting, deleting, and searching for internships to ensure safe and effective internship management.

**Interview and Communication Integration** The **Interview Manager** and **Communication Manager** are responsible for interview scheduling and stakeholder communication. The integration includes:

- The Scheduling System to handle interview setups.
- The Feedback System to collect post-interview feedback.



**Figure 5.5:** Internship and Communication Integration

The Communication Manager oversees the coordination of these systems, guaranteeing smooth communications and notifications. The integration of these elements will be tested by a Communication Driver, which will confirm that information is flowing between the user interface and interview management.

**Administrative Features Integration** The **Admin Manager** is integrated with the following systems:

- Monitoring System.
- Reporting System.
- Complaint System.

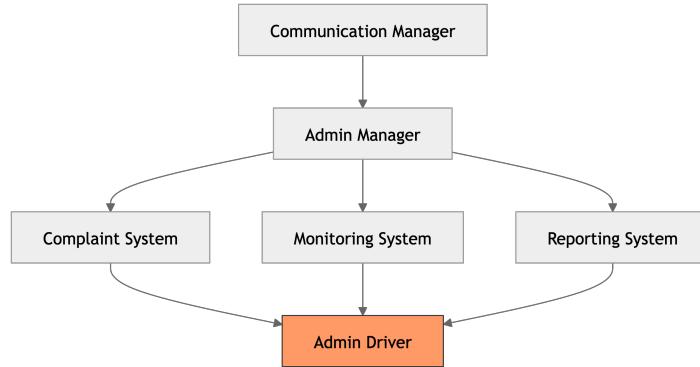


Figure 5.6: Administrative Features Integration

This integration also includes the Communication Manager to provide efficient communication for administrative duties. Functionalities like managing complaints, keeping an eye on platform activity, and producing reports are tested using the Admin Driver. This guarantees the stability and effectiveness of the platform's administrative functions.

**Final System Integration** All of the primary managers—Authentication Manager, Profile Manager, Internship Manager, Interview Manager, Communication Manager, and Admin Manager—must be connected to the **Dashboard Manager** as part of the last integration. This integration ensures:

- Seamless coordination of data and processes across the platform.
- A unified user experience.

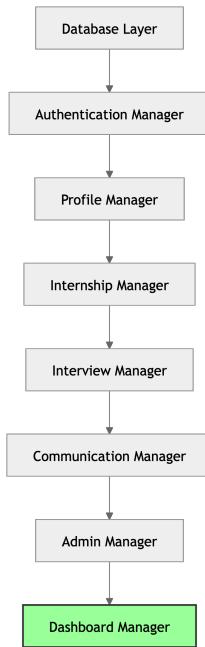


Figure 5.7: Final System Integration

Coordinating data and system processes, the Dashboard Manager acts as the main interface. With this phase, the integration process is complete, and the platform is completely functional and easy to use.

## 5.4. System Testing Strategy

Every newly created component will go through extensive testing before being incorporated into the system in order to guarantee the platform's accuracy and dependability. To verify each component's unique functionality, drivers will be used. To make sure that module properties are maintained and the workflow as a whole is unaffected, a new driver will be utilized to test the new component's compatibility with the current system after integration. Testing of the entire system will be done once all the components have been integrated to ensure correct operation and the lack of flaws. The testing methods listed below will be used:

### 5.4.1. Functional Testing

Functional testing will confirm that all objectives, specifications, and use cases are satisfied and that the platform complies with the capabilities described in the RASD document. This testing will check expected results by simulating user scenarios and confirming the system's proper workflow. Key elements that will be tested include:

- Applications
- Internship postings
- Profile management
- Login
- Interview scheduling
- Complaint handling

#### **5.4.2. Load Testing**

The system's behavior under various workloads will be evaluated through load testing to identify:

- Memory leaks
- Buffer overflows
- Inefficient memory management

This testing is necessary to confirm that the platform can efficiently manage several requests at once and remains stable during periods of high user activity.

#### **5.4.3. Performance Testing**

Performance testing will identify bottlenecks and assess how quickly the system responds to demanding workloads. This ensures that:

- The platform supports many users concurrently with minimal latency.
- Optimization opportunities in the underlying algorithms are identified to improve overall system performance.

#### **5.4.4. Stress Testing**

To make sure the system can bounce back from errors, stress testing will mimic harsh scenarios like a large number of people using the system at once or a reduction in processing power. Through this testing, the platform's robustness and resilience will be confirmed, guaranteeing that users will experience the least amount of disturbance possible in emergency situations.

### 5.4.5. User Interface Testing

User interface testing will validate the platform's usability and accessibility across a variety of devices and browsers. This testing will:

- Ensure a smooth and uniform experience for all user types—students, companies, and administrators.
- Verify compatibility with various screen sizes and resolutions.

### 5.4.6. Comprehensive Testing Approach

The system's functionality, reliability, and scalability will be fully verified by combining the following testing techniques:

- Functional Testing
- Load Testing
- Performance Testing
- Stress Testing
- User Interface Testing

This comprehensive approach guarantees:

- A flawless user experience.
- Compliance with the specifications outlined in the RASD document.

# 6 | Effort Spent

Team Member	Task	Hours Spent
Shreesh Kumar Jha	1. Introduction 2. Architectural Design 3. User Interface Design 4. Requirements Traceability 5. Implementation, Integration, and Test Plan	35
Samarth Bhatia	1. Introduction 2. Architectural Design 3. User Interface Design 4. Requirements Traceability 5. Implementation, Integration, and Test Plan	37
Satvik Sharma	1. Requirements Analysis 2. UML Diagrams 3. Backend Setup 4. Backend Implementation 5. API Refactor	32

Table 6.1: Effort spent by each member of the group.



# 7 | References

## 7.1. References

- Software Engineering 2 Course Materials, A.Y. 2024-2025.
- Daniel Jackson, *Software Abstractions: Logic, Language, and Analysis*.
- Assignment RDD AY 2024-2025.pdf.



# List of Figures

2.1	InternHub Overview	7
2.2	High Level Diagram	9
2.3	Low Level Diagram	11
2.4	Evaluation Manager	12
2.5	Dashboard Manager	14
2.6	Profile Manager	15
2.7	Internship Manager	16
2.8	Application Manager	17
2.9	Deployment View	19
2.10	Runtime SignUp	20
2.11	Runtime Login	21
2.12	Runtime Post Internship	22
2.13	Runtime Interview Management	23
2.14	Complaint Handling	24
2.15	Profile Management	25
2.16	Search and Filter	26
3.1	UI Overview	40
3.2	Login Interface Mockup	41
3.3	SignUp Interface Mockup	43
3.4	Company Dashboard	44
3.5	Post Internship	45
3.6	Student Dashboard	46
3.7	Build CV	47
3.8	Messaging System	48
5.1	Core Model Integration	57
5.2	Authentication Integration	57
5.3	Profile Management Integration	58
5.4	Internship Management Integration	59

5.5 Internship and Communication Integration . . . . .	60
5.6 Administrative Features Integration . . . . .	61
5.7 Final System Integration . . . . .	62

## List of Tables

1.1	Goals table.	3
1.2	Acronyms and terms used in the document.	3
1.3	Revision History	3
6.1	Effort spent by each member of the group.	65

