

Design Documentation

InternHub - Students&Companies (S&C) Platform

Version 1.0

Prepared by: Shreesh Kumar Jha, Samarth Bhatia and Satvik Sharma

3 January 2025

Table of Contents

1. Introduction
 2. Architectural Design
 3. User Interface Design
 4. Requirements Traceability
 5. Implementation, Integration and Test Plan
 6. Effort Spent
 7. References
-

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 needs of all stakeholders—students, companies, and universities—thereby ensuring a more efficient, productive, and rewarding internship ecosystem.

1.2 Scope

By connecting students, businesses, and academic institutions, InternHub - Students & Companies (S&C) 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.

1.3 Definitions, 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

1.4 Revision History

Version 1.0 - 03/01/2025

1.5 Reference Documents

Specification Document Assignment

1.6 Document Structure

Introduction: In the first section, the importance of the Design Document is established, and acronyms and abbreviations are defined and explained in detail. The InternHub - 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.

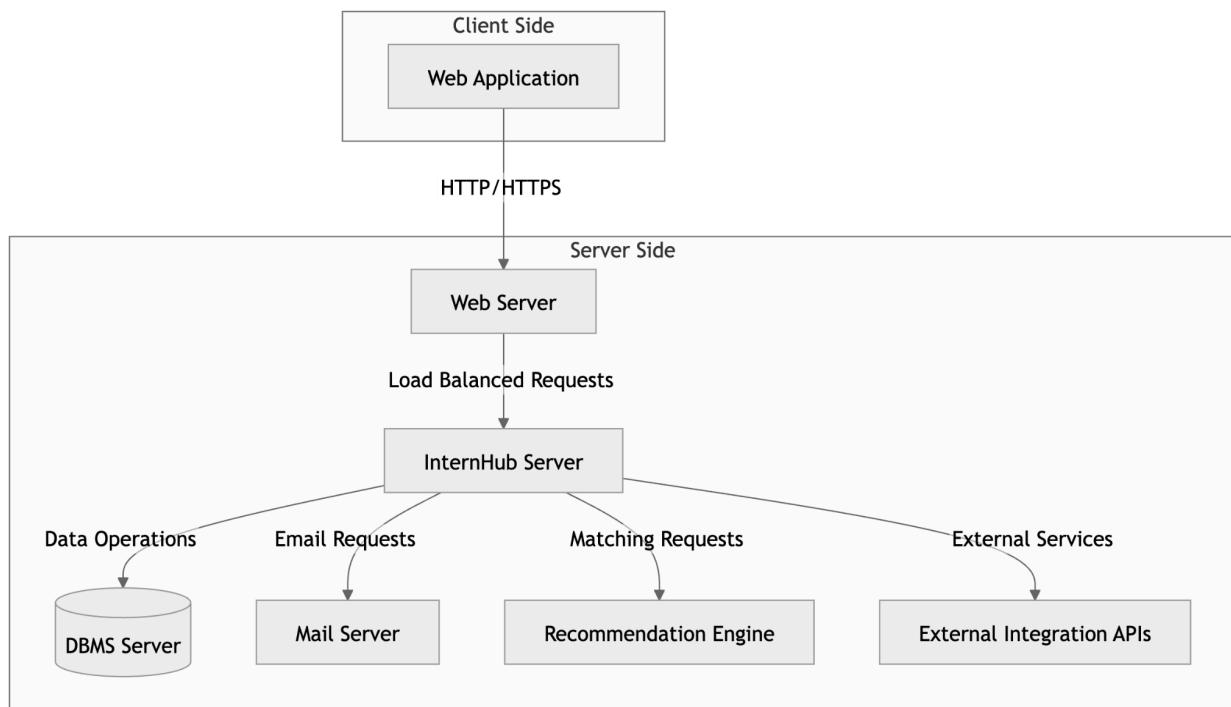
Effort Spent. In the sixth section are included information about the number of hours each group member has worked for this document.

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:



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.

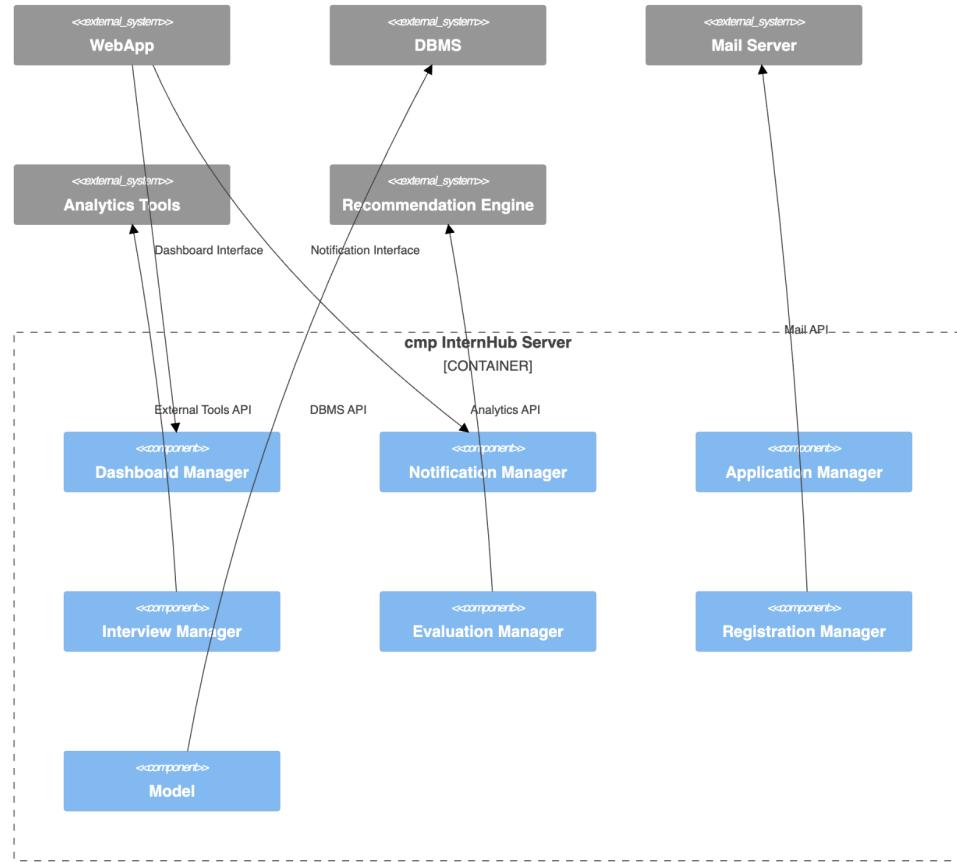
Server Side:

- **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

High Level Diagram

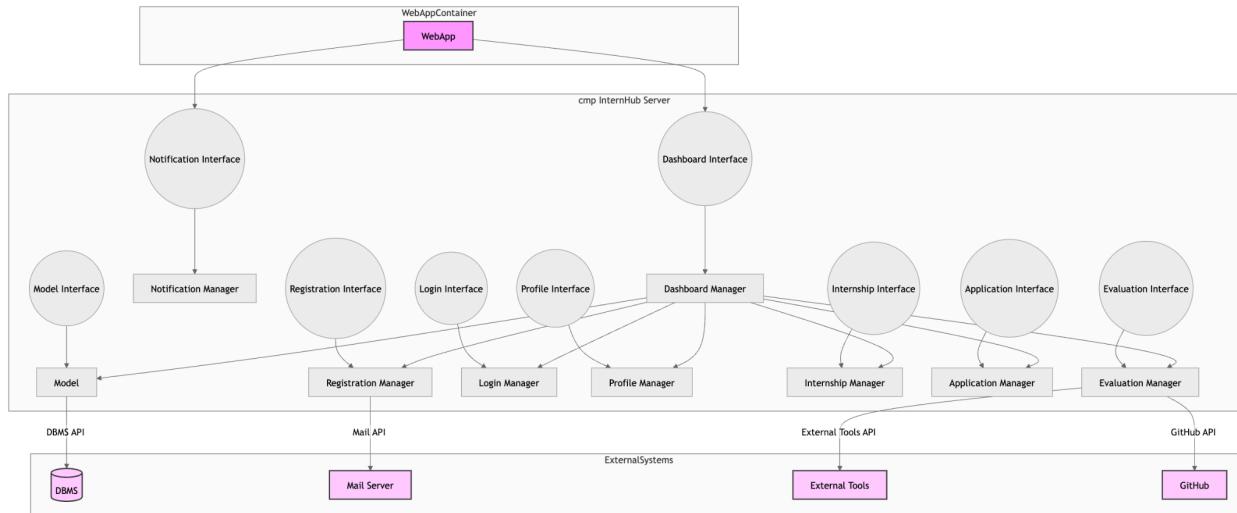


The external components of S&C are depicted in the high level component diagram of S&C above, along with their methods of communication with the S&C server.

- 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.

3. **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.
 4. **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.
 5. **Analytics Engine:** 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.
 6. **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.
- 7. External Systems:**
- a. WebApp connects to the Dashboard Manager and Notification Manager using respective interfaces.
 - b. DBMS connects to the Model using the DBMS API.
 - c. Mail Server communicates with the Registration Manager through the Mail API.
 - d. Recommendation Engine and Analytics Tools interact with the Notification Manager.
- 8. InternHub Server:**
- Consists of core components like:
- i. Dashboard Manager
 - ii. Notification Manager
 - iii. Application Manager
 - iv. Registration Manager
 - v. Evaluation Manager
 - vi. Interview Manager
 - vii. Model for centralized data access.

2.2.2 Low Level Diagram

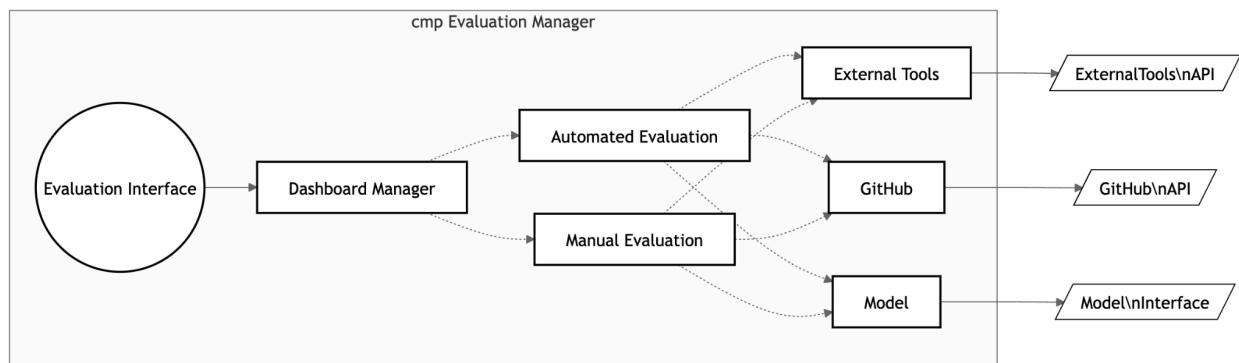


The figure above represents the detailed architecture of the **InternHub – Students & Companies (S&C)** platform, showing the key components within the **InternHub Server** and their interactions.

- 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. The Internship Manager interacts with the Model Component for database functions, and the Internship Interface makes user interactions easier.

7. **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.
8. **Evaluation Manager:** Sees how students' success during internships is evaluated. The Evaluation Manager uses the other Tools API to communicate with other tools (such analytics systems) and the Model Component to update the evaluation results in the database. Requests are handled by the Evaluation Interface.
9. **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



The InternHub-Students & Companies (S&C) platform's Evaluation Manager is made to evaluate and oversee students' performance reviews while they are interning. 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.

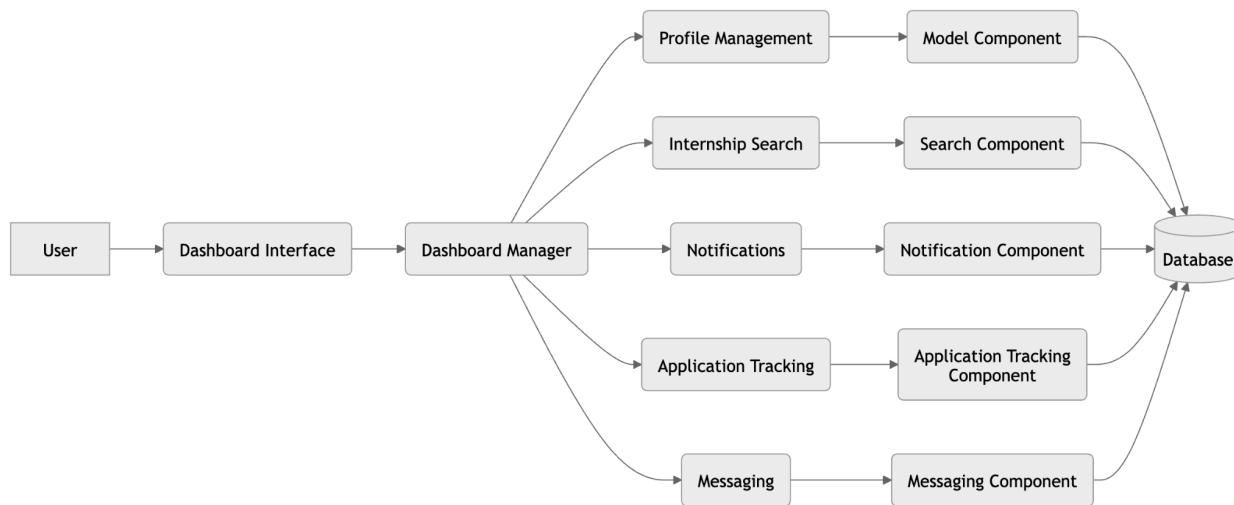
- In turn, the Model Component uses the DBMS API to update the student's score in the database, guaranteeing that the evaluation results are appropriately recorded in the appropriate database area.

Manual Evaluation Component

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

- The evaluator (company or administrator) initiates a request via the WebApp, which connects to the Dashboard Interface.
- Through the Evaluation Interface, the request is sent to the Manual Evaluation Component.
- The GitHub API or other integrated technologies are used by the Manual Evaluation component to retrieve the student's submitted work or internship data.
- 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.
- 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



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, with this subcomponent.
- 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 visitors to search and filter internships according on geography, preferences, and abilities.
- 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.

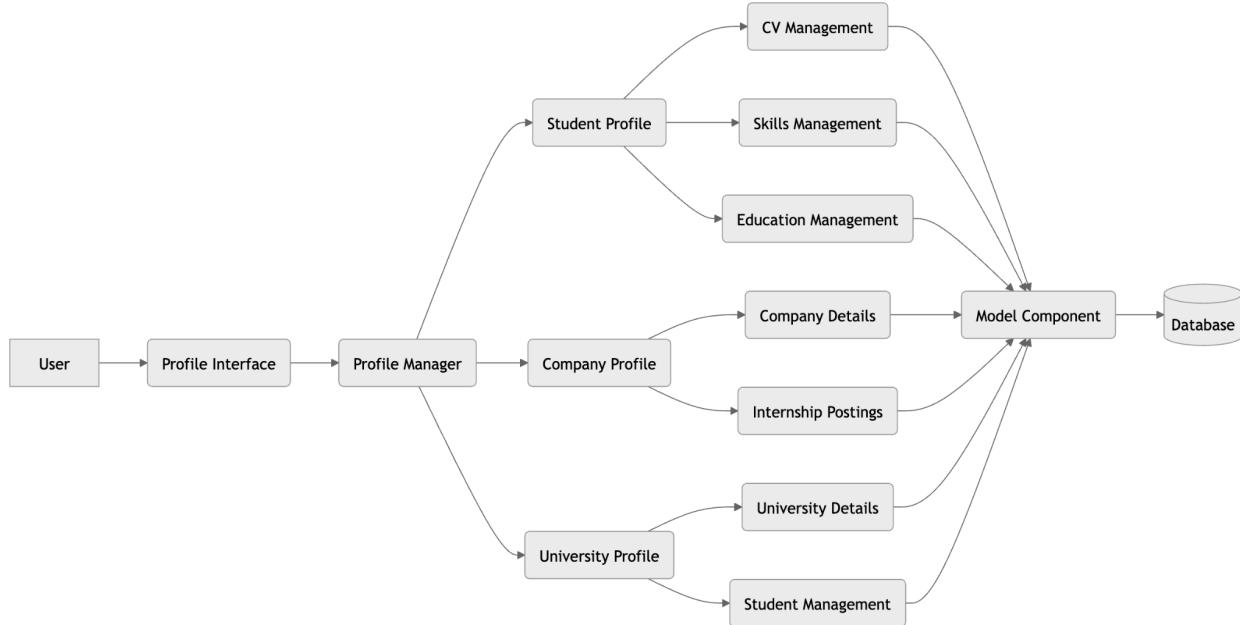
Application Tracking

- Allows users to keep track of the acceptance, rejection, and company changes of their internship applications.
- In 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 talk about internships.
- Communicates with the Messaging Component, which guarantees that every conversation is recorded and safely kept in the database.

2.2.5 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 makes sure that the Model Component communicates with the Database in a smooth manner. These are the main subcomponents and what they are responsible for:

Student Profile

- Manages student-specific details, including academic achievements, skills, and internship preferences.
- Subcomponents:
 - **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

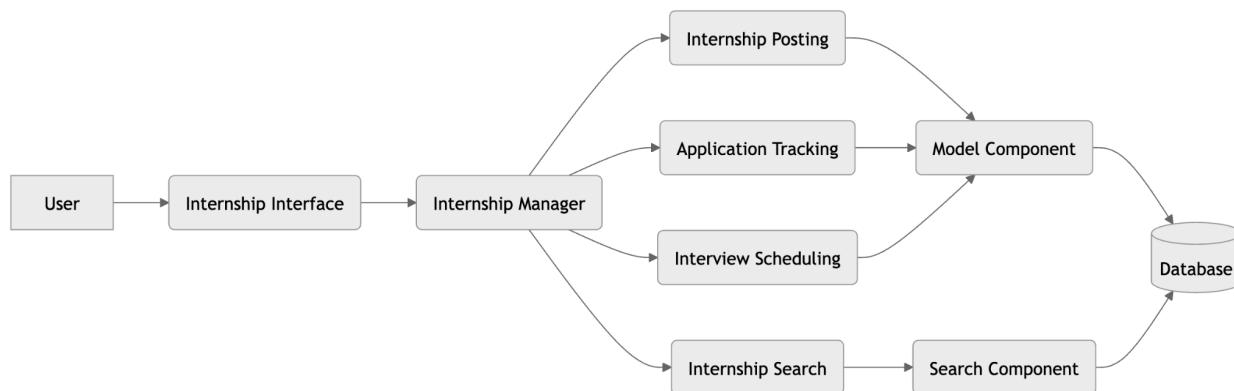
- Responsible for maintaining company-related data, such as organizational details, internship postings, and company preferences.
- Subcomponents:
 - **Company Details:** Stores and updates company information, such as name, size, industry, and contact details.

- **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.
- Subcomponents:
 - **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



Managing the internship lifecycle is the primary responsibility of the InternHub - Students & Companies (S&C) platform's Internship Manager. It makes it easier for users to communicate with the platform so that internship-related features can be handled effectively. A summary of its constituents and interrelations may be found below:

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.

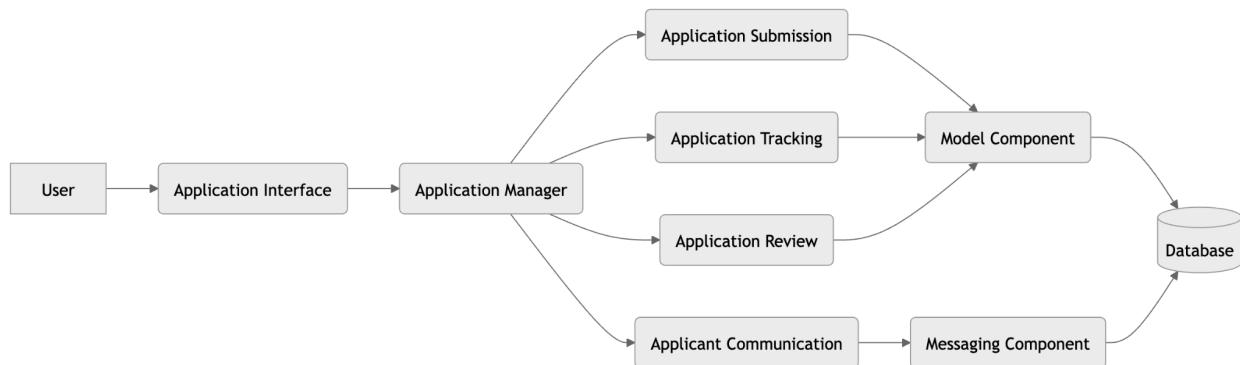
Interview Scheduling

- 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



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.

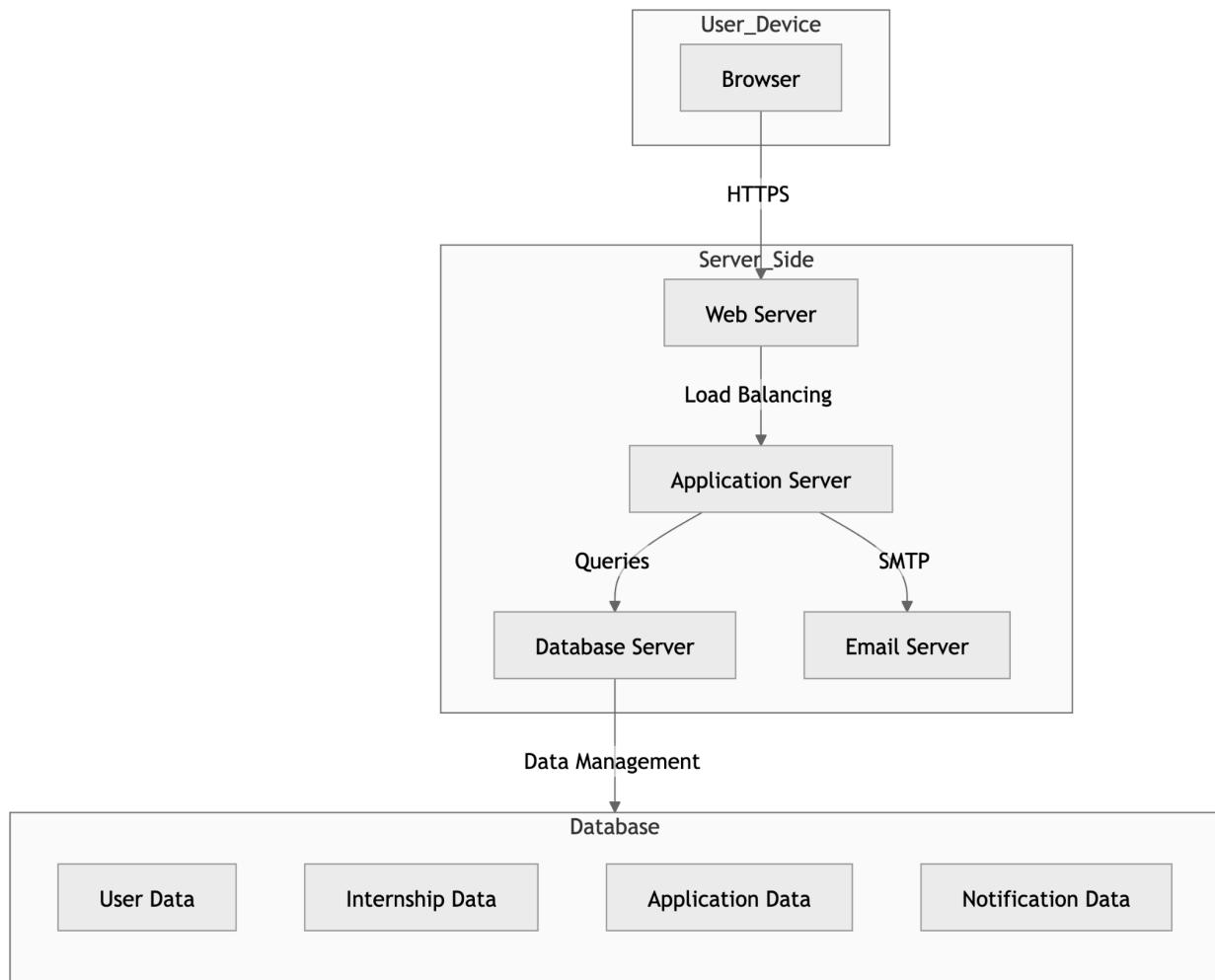
Application Review

- 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



The platform's **Client-side** is made up of user devices, which are mostly accessed via web browsers. These gadgets stand in for the platform's users, which include businesses, academic institutions, and students. Users engage with the platform's services, including applications, notifications, internship posts, profile management, registration, and login, using their browsers.

Secure HTTPS connections are used for user-platform communication, protecting and protecting user data.

On the **Server-Side**, the **Web Server** acts as the entry point for all client requests. It manages static content delivery, such as HTML, CSS, and JavaScript files, ensuring a responsive user experience. The Web Server also establishes secure connections using HTTPS and performs load balancing, distributing incoming requests to multiple application server replicas to maintain reliability and handle high traffic loads. Additionally, the Web Server routes these requests to the appropriate application server endpoints based on user operations.

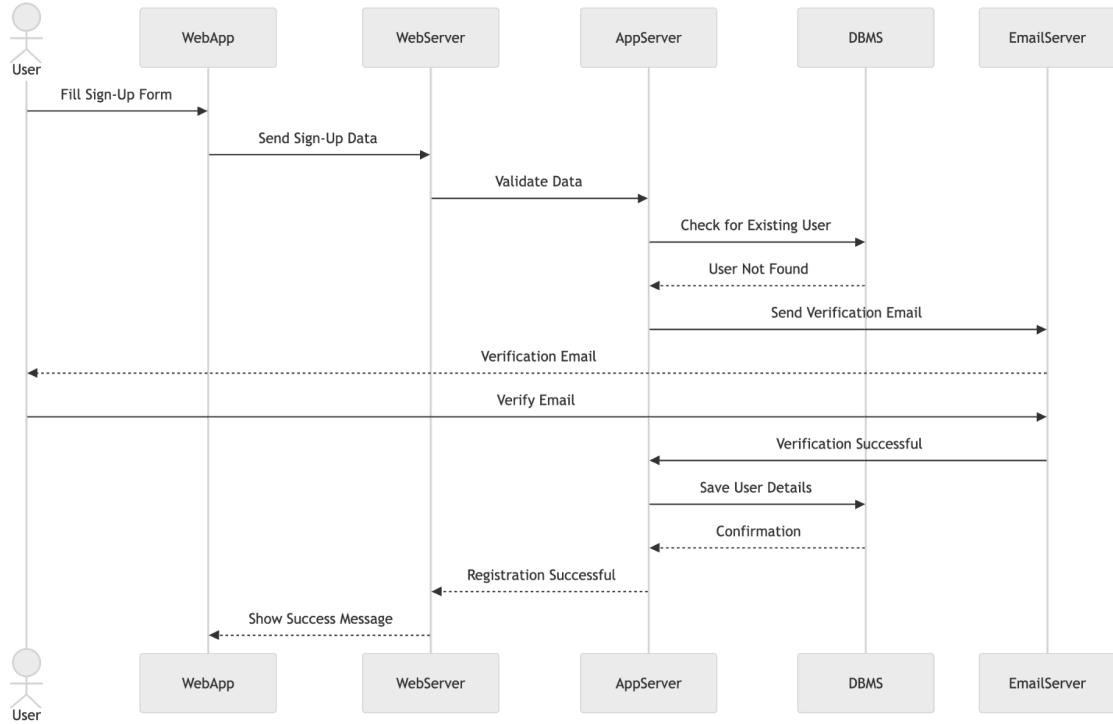
The **Application Server** is the core backend of the platform, responsible for processing business logic and coordinating with other components. It handles critical functions such as user management (registration, login, and profile updates), internship management (postings, applications, and tracking), and notification delivery (application updates, reminders, and alerts). The Application Server also serves as the intermediary between the Web Server and the DBMS Server, processing data retrieval and updates. Furthermore, it integrates with the Email Server to send automated communications such as registration confirmations and application status updates. The Application Server ensures seamless coordination between all components of the platform to deliver a consistent user experience.

The **DBMS Server** is the central repository for all data managed by the platform. It stores user data, including profiles and credentials for students, companies, and universities. It also maintains internship-related data, such as postings, descriptions, and requirements, as well as application data tracking the status of student applications. Additionally, the DBMS Server logs notification data to ensure proper tracking of all communications sent to users. The Application Server interacts with the DBMS Server through a secure API, enabling efficient data storage, retrieval, and updates.

Lastly, the **Email Server** is responsible for managing outgoing email communications. It handles system-generated emails such as registration confirmations, application updates, and reminders. Communication between the Application Server and the Email Server is facilitated through the SMTP protocol, ensuring reliable and secure delivery of emails to users. The Email Server plays a crucial role in keeping users informed and engaged with the platform.

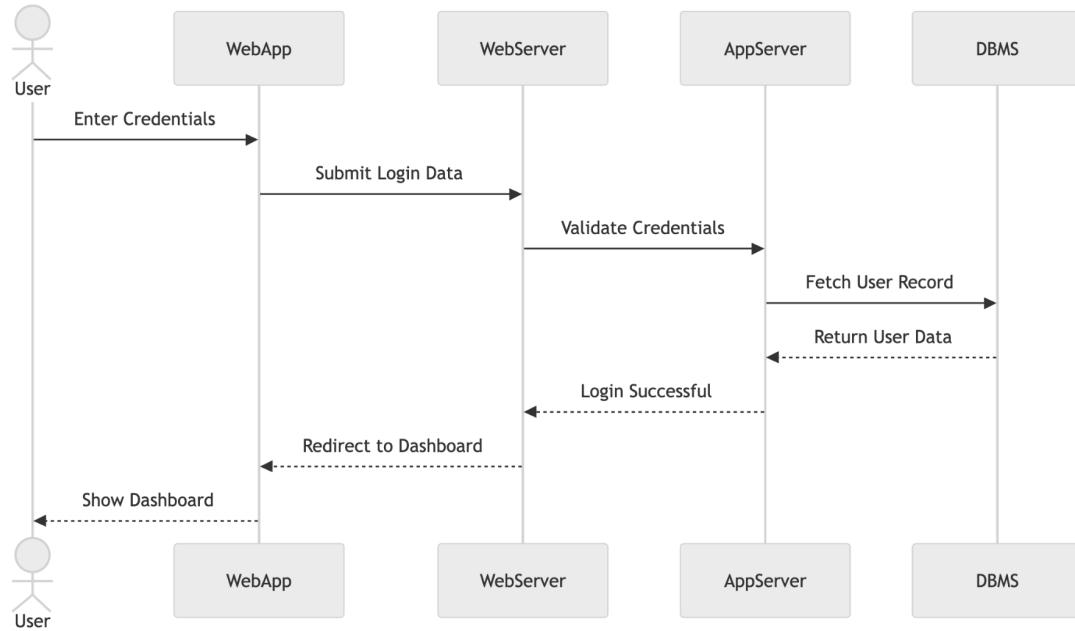
2.4 Runtime View

2.4.1 SignUp



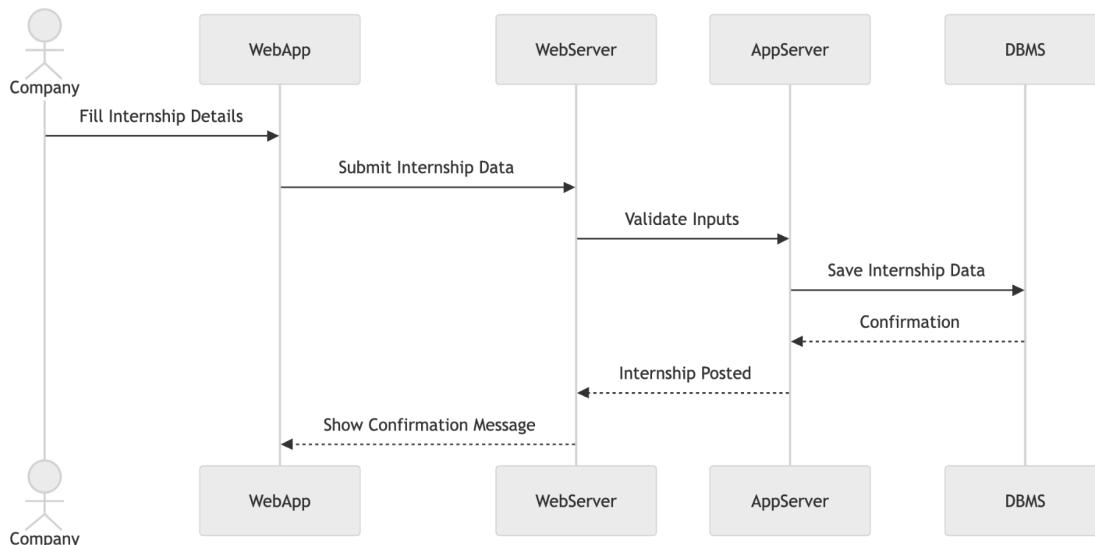
The platform's sign-up procedure serves as the gateway for new users, such as businesses, university administration, or students. The user starts by entering their information using the WebApp's sign-up screen, including their name, email address, password, and user role. Before being forwarded to the Application Server, the Web Server verifies the inputs after receiving this data securely. The Application Server's Registration Manager component uses the DBMS Server to search the database for duplicate records. A verification email is issued via the Email Server if the user is unique. Following the user's email verification, the Application Server completes the sign-up process by storing the user's data in the database. The user is then notified of a successful registration, ensuring their credentials are securely recorded and ready for login.

2.4.2 Login



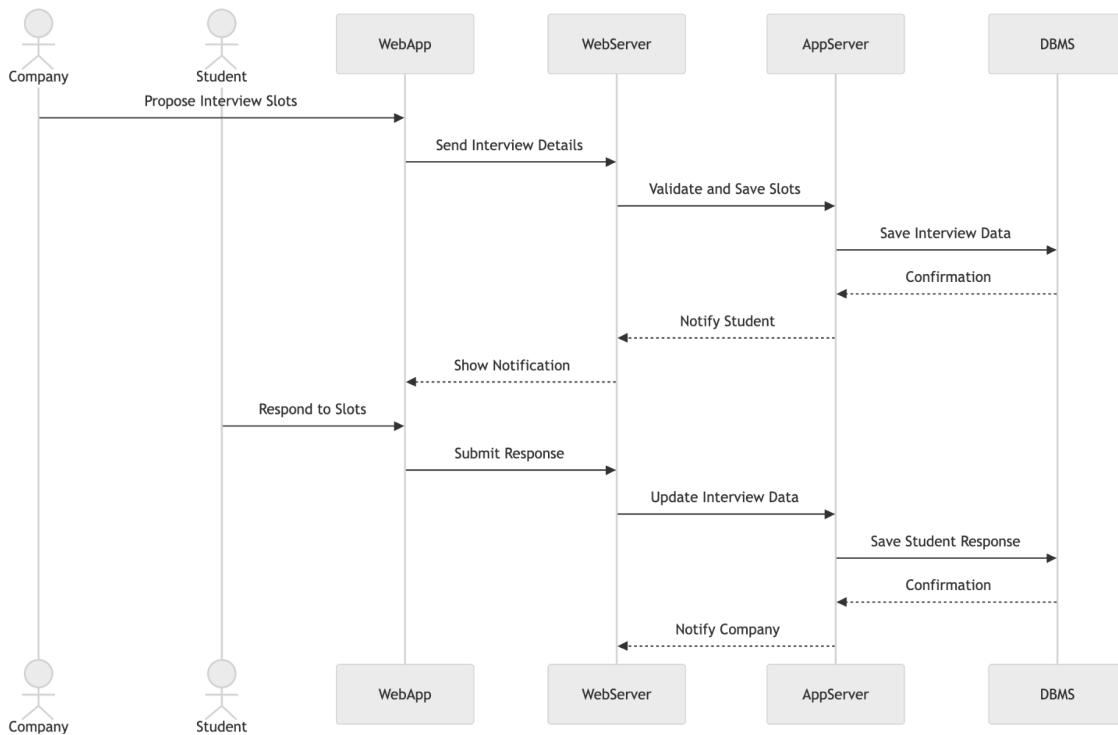
For registered users, secure access is guaranteed through the login procedure. A secure HTTPS connection is used by the WebApp to send the data to the Web Server after a user inputs their login information on the login window. The Login Manager next uses the DBMS Server to verify the credentials against database entries after the Web Server and Application Server have communicated. The user is logged into the platform when the system creates a session token if the credentials match. The WebApp then reroutes the user to their dashboard after receiving the response via the Web Server. While keeping the platform user-friendly, this smooth procedure guarantees that only authorized users can access it.

2.4.3 Post Internship



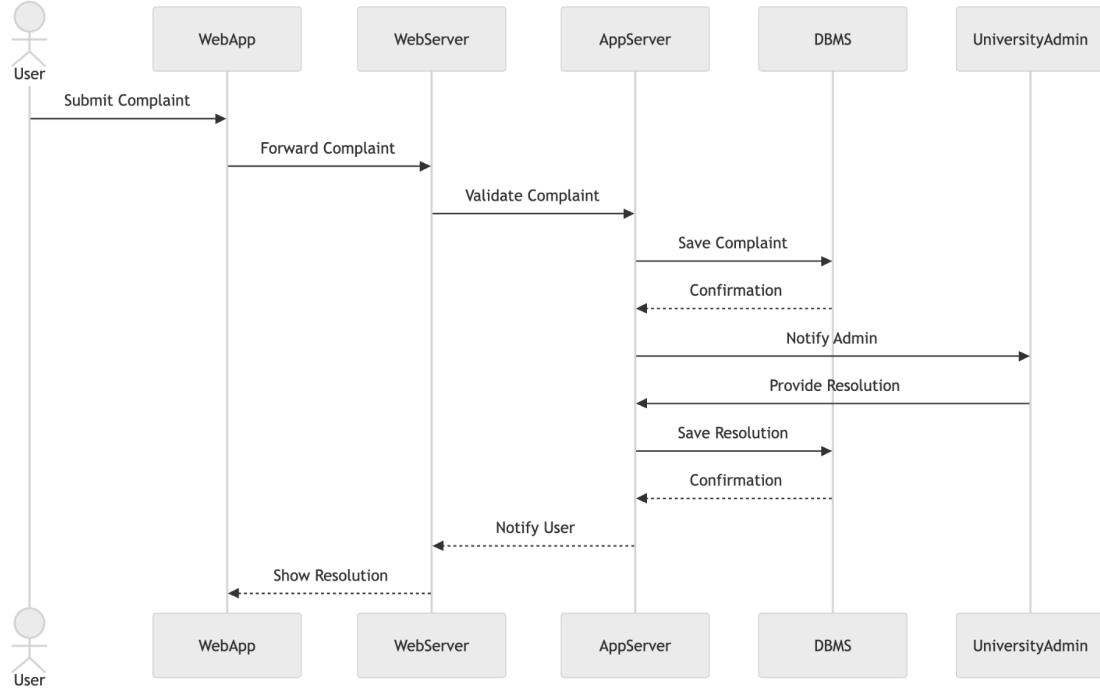
For users of the company, posting an internship is an essential function. A company starts the process by using the WebApp to fill out the internship details, which include the job title, description, prerequisites, and duration. After that, the Web Server receives the data and sends it to the Application Server for verification. To save the new internship posting in the database, the Internship Manager component processes the data and communicates with the DBMS Server. The company user receives a confirmation message when the data has been correctly stored. This procedure guarantees that qualified students looking for opportunities will see the posted internship right away.

2.4.4 Interview Management



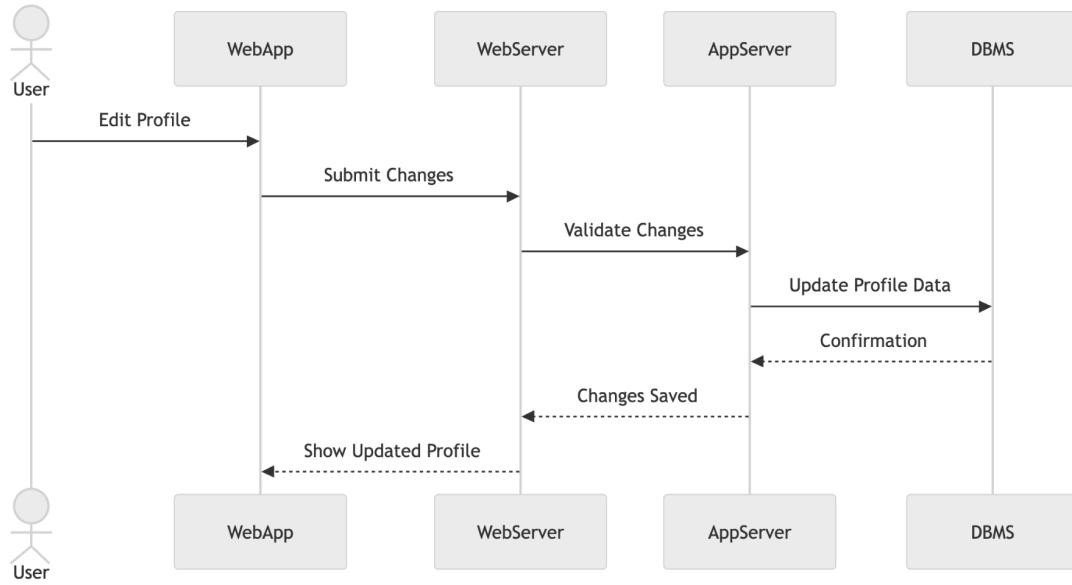
Students and businesses can schedule and coordinate more easily thanks to the interview management procedure. Through the WebApp, which sends the data to the Web Server and then to the Application Server, companies offer interview times. Through the DBMS Server, the Interview Manager component verifies and stores the suggested database slots. Students receive notifications about the suggested timeslots through the Email Server and Notification Manager. After that, students can reply to the suggested timeslots using the WebApp, and the Interview Manager will update their answers in the database. Additionally, the company receives notifications verifying the student's selection. Both sides benefit from seamless scheduling and communication thanks to this effective procedure.

2.4.5 Complaint Handling



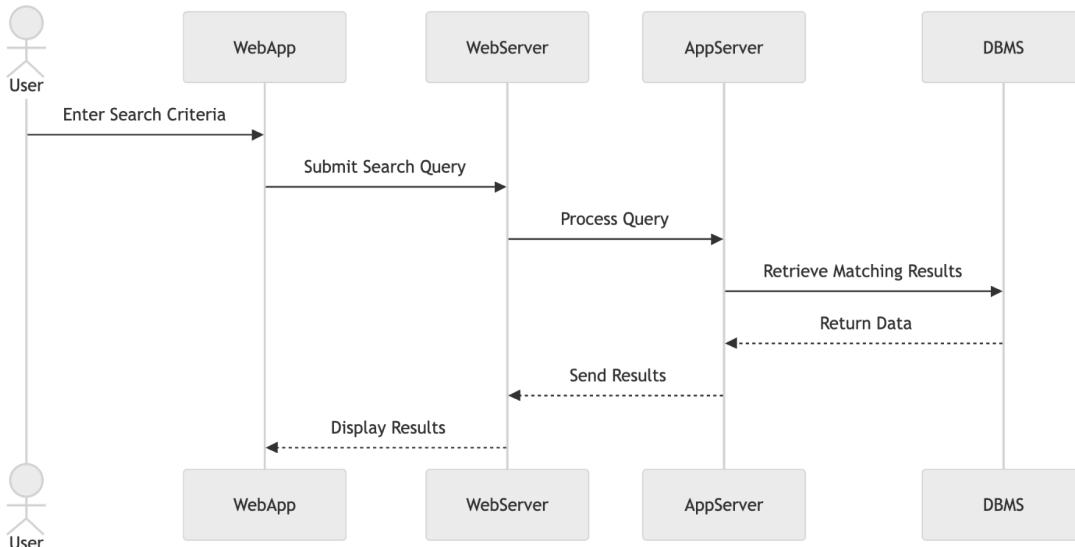
The complaint-handling procedure is made to effectively handle user complaints. Through the WebApp, users can file complaints, which are then sent to the Web Server and then the Application Server. Through the DBMS Server, the Complaint Manager component verifies the complaint and stores it in the database. After receiving notification from the system, the appropriate university official examines the complaint and offers a remedy. The user is informed of the outcome using the Notification Manager and Email Server, and the resolution is recorded in the database. This open procedure guarantees that every complaint is handled quickly and efficiently.

2.4.6 Profile Management



Users can update and maintain their personal information through profile management. Through the WebApp's profile page, users can initiate updates by entering information about themselves, including contact data, skills, and educational history. These updates are sent from the Web Server to the Application Server, where the Profile Manager verifies the modifications. The DBMS Server is then used to store the changed data in the database. The user sees a success message after the changes have been saved. With the help of this function, user profiles are kept up to date and pertinent for the matching and application processes for internships.

2.4.7 Search and Filter



Users can find pertinent information fast thanks to the search and filter process. Students apply filters like location, necessary skills, and duration when searching for internships through the WebApp. The Web Server receives these filters before forwarding them to the Application Server. In order to retrieve relevant results, the Search Component interacts with the DBMS Server to process the query. The Web Server sends the results back to the WebApp, where the user can see them. With the help of this function, users may quickly search through the platform's data and identify the most pertinent 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):** Allows recruiters to add a new internship after validating permissions.
- **deleteInternship(String internshipID):** Deletes a specified internship after validating recruiter permissions.
- **fetchAllInternships(String recruiterID):** Fetches all internships posted by a specific recruiter.
- **fetchInternshipById(String internshipID):** Retrieves details of a specific internship using its ID.

Application Manager

- **SubmitApplication(String internshipID, String studentNickname, String resume):** Allows students to submit their application for a specific internship.

- **ViewApplications(String internshipID):** Retrieves all applications for a particular internship.
- **UpdateApplicationStatus(String applicationID, String status):** Updates the status of a submitted application.

Interview Manager

- **ProposeInterview(String internshipID, List<Date> availableDates):** Allows companies to propose interview slots for a specific internship.
- **RespondToInterview(String interviewID, String response):** Allows students to accept or decline an interview invitation.
- **GetInterviewDetails(String interviewID):** Retrieves details about a specific interview.

Complaint Manager

- **SubmitComplaint(String userID, String description):** Allows users to file a complaint about an issue.
- **ViewComplaint(String complaintID):** Retrieves details about a specific complaint.
- **ResolveComplaint(String complaintID, String resolution):** Updates the status and resolution of a complaint.

Profile Manager

- **EditProfile(String nickname, Map<String, String> updatedDetails):** Updates the user's profile information.
- **ViewProfile(String nickname):** Retrieves the profile details of a user.
- **SearchProfiles(String keyword):** Searches for profiles based on the provided keyword.

Notification Manager

- **SendNotification(String userID, String message):** Sends a notification to a user.
- **ViewNotifications(String userID):** Retrieves all notifications for a specific user.

OTP Manager

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

Token Manager

- **generateAccessToken(String email, String userType):** Generates a secure access token for the user based on their email and type.

Dashboard Manager

- **GetDashboardData(String userID):** Retrieves an overview of the dashboard for the logged-in user.

- **ViewStatistics(String userID):** Fetches statistical insights relevant to the user's activity.

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 internship search, application management, and profile maintenance
 - Company Portal: Handles posting internships, reviewing applications, and managing candidates
 - Admin Portal: Provides oversight tools, complaint management, and system monitoring
- Uses modern web technologies with responsive design principles
- Implements client-side validation and state management

Services Layer (Business Logic)

- User Management Service
 - Handles authentication and authorization
 - Manages user profiles and permissions
 - Processes registration and account updates
- Matching Engine Service
 - Implements advanced algorithms for internship-student matching
 - Processes search queries with multiple parameters
 - Handles filtering and sorting of results
- Recommendation System
 - Analyzes student profiles and internship requirements
 - Generates personalized recommendations
 - Updates suggestions based on user interactions
- Interview Management Service
 - Coordinates interview scheduling
 - Manages interview status updates
 - Handles feedback collection

Data Layer

- Database Management System
 - Stores structured data like user profiles, applications, internships
 - Maintains relationship mappings between entities

- Handles data integrity and consistency
- File Storage System
 - Manages document storage (CVs, certificates)
 - Handles secure file uploads/downloads
 - Implements file versioning

2.6.2 Client-Server Architecture

Client Side Implementation

- Browser-based responsive UI
 - Adapts to different screen sizes and devices
 - Provides consistent user experience
 - Implements client-side validations
- Real-time Updates
 - Uses WebSocket connections for instant notifications
 - Updates UI states without page refresh
 - Maintains synchronized views

Server Side Implementation

- RESTful API Services
 - Implements standardized HTTP endpoints
 - Handles request/response processing
 - Manages API versioning
- Backend Processing
 - Executes business logic operations
 - Processes data transformations
 - Manages system state

2.6.3 Event-Driven Architecture

Event Publishers

- Application System
 - Generates events for status changes
 - Triggers notification workflows
 - Updates related entities
- Interview System
 - Creates scheduling events
 - Sends reminder notifications
 - Updates calendar integrations
- Complaint System
 - Triggers escalation events

- Notifies relevant administrators
- Updates tracking status

Event Subscribers

- Notification Service
 - Processes event notifications
 - Sends emails/alerts
 - Updates user dashboards
- Analytics Service
 - Tracks system events
 - Generates usage statistics
 - Updates reporting metrics

2.6.4 Design Patterns

1. Model-View-Controller (MVC)

Models

- User Model
 - Base user properties and behaviors
 - Specialized student/company/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

This architecture and pattern implementation ensures:

- 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 ensure continuous, round-the-clock platform operation
- Failover Mechanism enables automatic switching to backup systems during disruptions
- Error handling mechanisms ensure data integrity
- System recovery procedures after failures
- Transaction integrity for critical processes like application submissions and interview scheduling

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
- Database and storage systems designed for growing data volumes
- API-first approach enabling future mobile app development

2.7.3. Security

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

2.7.4. Notification Handling

- Real-time notifications for application status changes
- Interview scheduling alerts
- Complaint status updates
- Email notifications for critical events

- Calendar integration for interview scheduling
- Platform-wide announcements capability

2.7.5. Document Management

- Secure storage for CVs, certificates, and official documents
- Version control for user documents
- Document validation and verification
- Format standardization
- Access control based on user roles
- Automated document processing capabilities

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

2.7.8. 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)

These design decisions align with the platform's core requirements while ensuring:

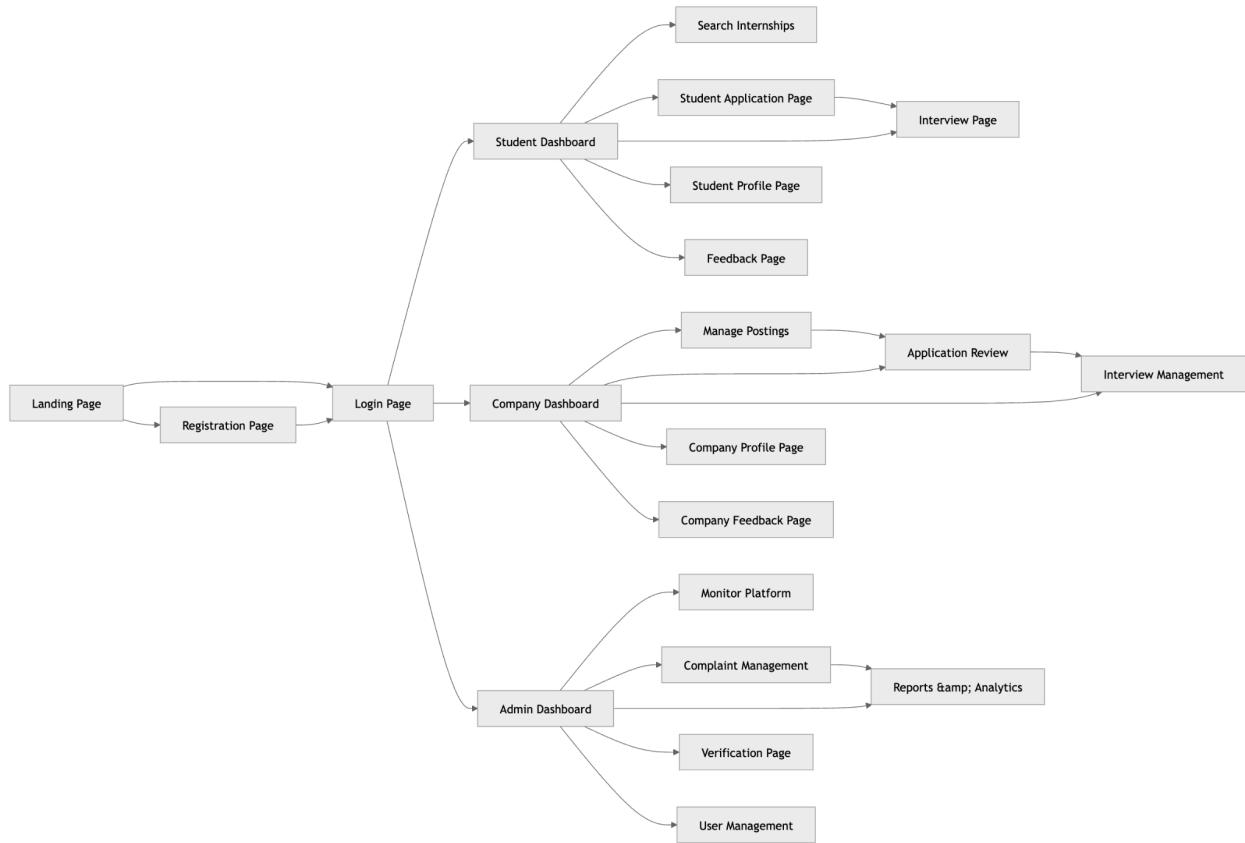
- Reliable service delivery
- Secure data handling
- Efficient performance
- User satisfaction
- System maintainability
- Regulatory compliance
- Future scalability

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 flawless 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



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. Beginning with the Landing Page, the system directs all users to

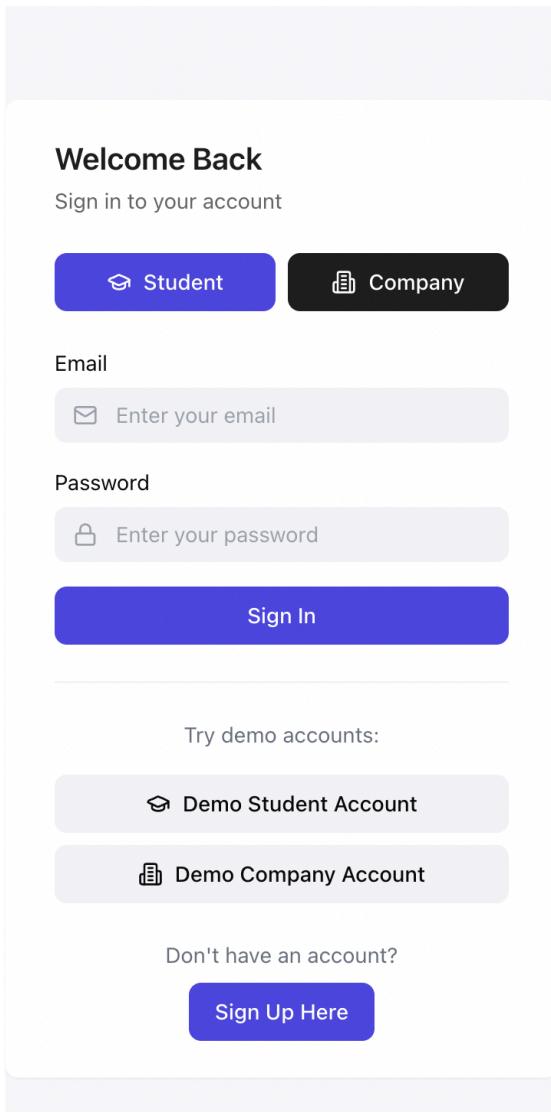
either the Registration Page for new users or the Login Page for existing users. The Student Dashboard, Company Dashboard, or Admin Dashboard are role-specific dashboards that users are taken to when logging in. These dashboards offer unique features that are tailored to the needs of their respective user types.

With additional links to crucial procedures like Interview Management, the Student Dashboard offers access to key functions like Search Internships, Student Application Page, Feedback Page, and Student Profile Page. Similarly, the Company Dashboard supports efficient recruitment workflows by giving recruiters access to the Company Profile Page, Company Feedback Page, and the ability to manage postings and review applications. With features like User Management, Verification Page, Complaint Management, and Reports & Analytics, the Admin Dashboard makes administrative oversight easier while guaranteeing effective platform administration and monitoring.

With its emphasis on role-specific features and easy navigation, this structured user interface design guarantees a smooth and effective user experience for all stakeholder groups. The diagram's links show how the platform's interface is dynamic and scalable, catered to the various requirements of administrators, businesses, and students.

3.2 User Interfaces

UI1. Login



The InternHub – Students & Companies (S&C) platform may be accessed easily through the login page, which is made to support the two main user roles of companies and students. A toggle option at the top of the interface makes it simple for users to choose their responsibilities, guaranteeing a customized login experience. Users are asked to enter their password and email address below, with user-friendly placeholders helping them along the way.

The login process is made easier with a noticeable and unambiguous "Sign In" button, which guarantees usability and accessibility. Demo accounts are offered to both students and businesses so they may test out important features if they would like to use the platform without enrolling. Furthermore, there is a clear "Sign Up Here" link at the bottom that makes it simple for new users to register. Simplicity, clarity, and efficiency are given first priority in the clean design, which guarantees an intuitive experience for all users while upholding a polished appearance.

UI2. SignUp

The screenshot shows a registration form titled "Create Account" with the subtitle "Join S&C to start your journey". At the top, there are two buttons: "Student" (blue) and "Company" (black). Below these are four input fields: "Enter first name", "Enter last name", "Enter your university", and "Enter your email". There is also a "Create password" field. A large blue "Create Account" button is positioned at the bottom. Below the button, a link says "Already have an account?" followed by a black "Sign in" button.

New users can create an account on the InternHub – Students & Companies (S&C) platform with ease thanks to the registration page's simplified and user-friendly UI. To ensure a customized experience for both user roles, users can switch between the Student and Company options at the top. To ensure that all required data is gathered effectively, the form contains fields for vital information including First Name, Last Name, University (for students), Email, and Password.

A noticeable "Create Account" button at the bottom makes it easier to submit the registration form and leads users through the account creation procedure with ease. Those who already have an account are redirected to the login page via a prominent "Sign in" link at the bottom. Simplicity and usability are given top priority in the simple design, guaranteeing that users can register swiftly and easily. This design guarantees new users entering the site a professional and easy-to-use experience.

UI3. Company Dashboard

The dashboard features a header with the title "TechCorp Dashboard" and a subtitle "Manage your internship programs". It includes a blue button "+ Post New Internship", a notification bell icon, and a user profile icon. Below the header are four summary cards: "Active Positions 8", "Total Applicants 156", "Scheduled Interviews 12", and "Hired This Month 5". A large button "View" with a document icon is positioned below these cards. The main content area is divided into two sections: "Recent Applications" and "Today's Interviews". The "Recent Applications" section shows three entries for "Sarah Parker" applying for a "Frontend Developer Position". Each entry includes a profile icon, the candidate's name, position, and experience ("React 3 Years Exp."), along with "Schedule Interview" and "Review" buttons. The "Today's Interviews" section shows three scheduled interviews for "John Doe" as a "Backend Developer" from 2:00 PM - 3:00 PM, each with a profile icon and a calendar icon.

Recruiters may successfully manage their internship programs with the help of the corporate dashboard, which offers a well-organized and useful interface. Important indicators that provide a clear picture of ongoing recruitment efforts are shown at the top, including the number of Active Positions, Total Applicants, Scheduled Interviews, and Hired This Month. A button on the dashboard allows employers to post new internships fast.

Recruiters can monitor and review applicant submissions in the Recent Applications area. Along with instructions to schedule an interview or review the application, each application contains pertinent information about the candidate, including their name, the position they are applying for, their abilities, and their experience. The Today's Interviews section, located on the right, offers a condensed interview schedule for the day along with the names, positions, and times of the candidates. Recruiters can navigate and complete their work more effectively thanks to the layout, which guarantees that all necessary information is easily available.

UI4. Post Internship

Please fill in this field.

Post New Internship

Title

Description

Location

Duration (in months)

Stipend (in USD)

Cancel

Post Internship

Recruiters can post internship opportunities to the site using a structured interface by using the Post New Internship form. Important information like the Title, Description, Location, Duration (in months), and Stipend can be included in the form's fields. Recruiters can give thorough information about the internship because these fields are defined and spaced appropriately to ensure clarity and simplicity of use.

The Post Internship button at the bottom allows you to submit the internship details, and the Cancel button allows you to end the process. Recruiters can effectively finish or stop the action according to their preferences thanks to this arrangement. Functionality and simplicity are given first priority in the design, which makes it easy for businesses to communicate openings to prospective candidates.

UI5. Post Internship

The screenshot shows a mobile-style student dashboard. At the top, a welcome message "Welcome, John!" is displayed, followed by the student's major "Computer Science Student". On the right side of the header are three buttons: "Build CV" (blue), a notification bell icon (black), and a user profile icon (blue).

Below the header, there are three main engagement metrics displayed in rounded boxes:

- "Active Applications" (2) with a briefcase icon.
- "Upcoming Interviews" (2) with a calendar icon.
- "New Messages" (3) with a speech bubble icon.

The main content area is divided into two sections:

Recent Internship Matches (Based on your profile and preferences)

- Frontend Developer** at **TechCorp** in **San Francisco**. Includes an "Apply Now" button.
- Backend Developer** at **WebSolutions** in **New York**. Includes an "Apply Now" button.

Upcoming Interviews

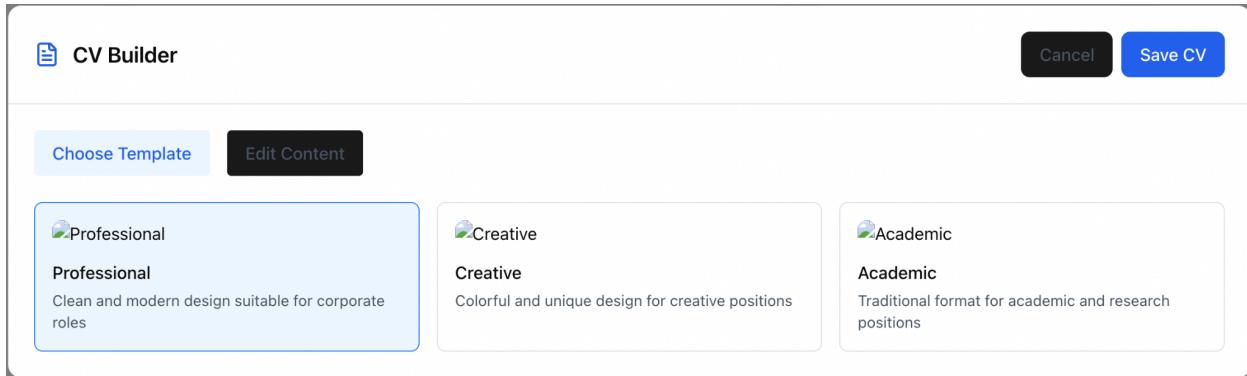
- Frontend Developer** at **WebSolutions Ltd** on **Tomorrow at 10:00 AM**.
- Backend Developer** at **TechCorp** on **Friday at 3:00 PM**.

A customized and simplified interface is offered by the student dashboard to assist students in effectively managing their internship application and search process. An overview of the student's platform engagement is provided by the prominent display of key data including Active Applications, Upcoming Interviews, and New Messages. To make resume-building resources easily accessible, there is also a "Build CV" button.

With their unique firm names, locations, and an Apply Now button for instant action, the Recent Internship Matches area displays internship opportunities that are customized based on the student's profile and choices. These opportunities include roles like Frontend Developer and Backend Developer. To help students stay organized and ready for their commitments, the Upcoming Interviews area on the right-hand side shows scheduled interviews along with information about the position, employer, and time.

With an emphasis on accessibility and relevancy, this layout makes sure that students can easily access the most important information and steps to improve their internship experience.

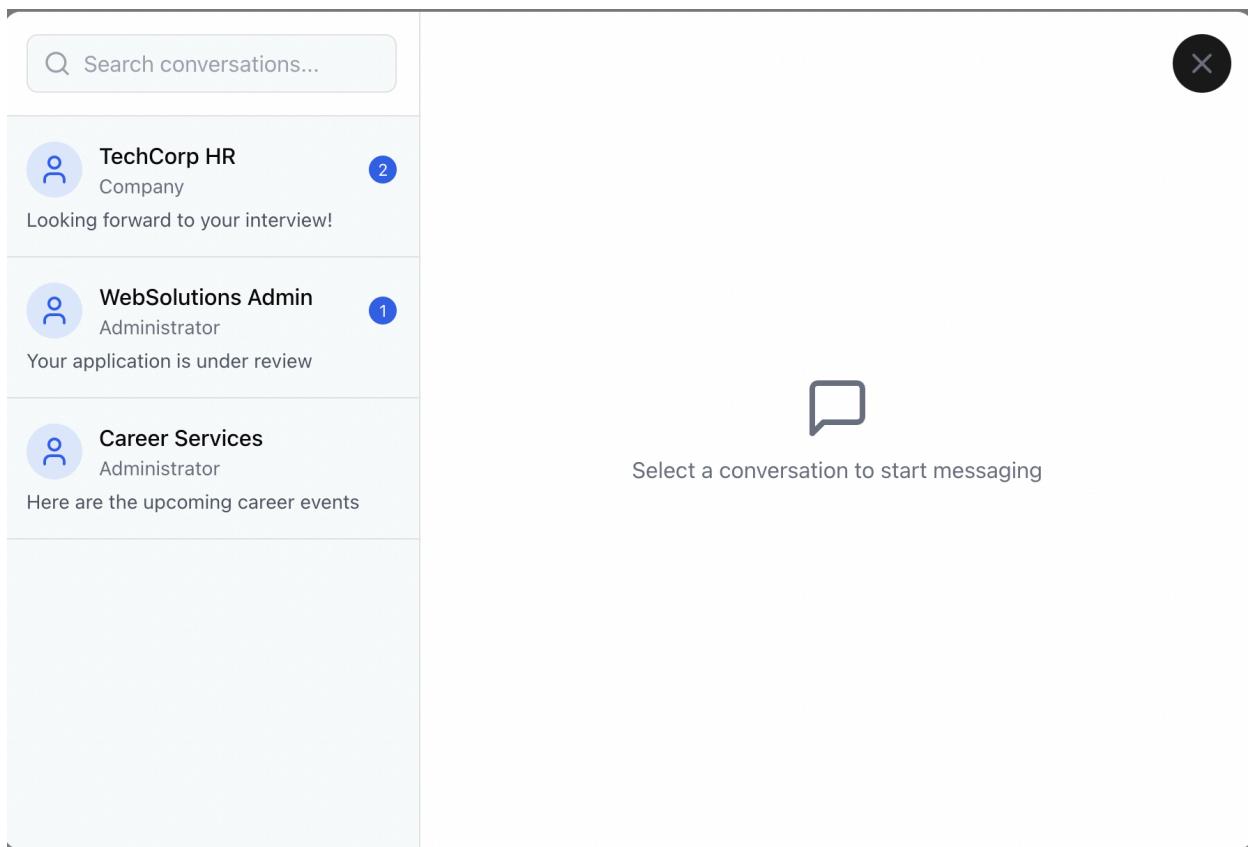
UI6. Build CV



The CV Builder interface offers a simplified and well-structured layout that enables users to generate and modify their resumes. Users can choose a design and then edit it with their personal information thanks to the interface's two primary steps: Choose Template and Edit Content. Three options—Professional, Creative, and Academic—are offered in the template selection area, each with a description to assist users in selecting the best format for their professional objectives and the kind of position they are aiming for.

Users can choose to Cancel to remove changes or Save CV to complete their resume in the upper-right corner. This design guarantees that users can effectively create and customize their resumes, meeting a range of professional needs and preferences while preserving simplicity and clarity all along the way.

UI7. Messaging System



Users can have conversations with different stakeholders on the InternHub – Students & Companies (S&C) platform thanks to the messaging interface's clear and practical architecture. The contact's name, role (e.g., Company, Administrator), and a synopsis of the most recent message are used to arrange the list of ongoing discussions on the left side. Blue badges, which serve as notifications, show how many unread messages there are in each chat, making it simple for users to see pending exchanges.

To begin texting, users are prompted to choose a conversation with an empty placeholder in the right panel. Users may concentrate on their ongoing chats without interruptions thanks to the split-pane design, which maintains the interface simple to use. Users may easily find particular discussions thanks to the search box at the top of the conversation list. This design facilitates smooth communication, allows for real-time changes, and encourages cooperation between administrators, businesses, and students.

4. Requirements Traceability

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

That way, the implementation will start with basic, independent modules that don't 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 finally integrated into its entirety, smaller functional subsystems can be created using the Bottom-Up approach. With this incremental approach, testing is done on smaller parts of the system initially and continues for each module as it becomes ready, making debugging and error tracking easier. This approach facilitates parallel development as well, allowing separate teams to work on various elements at the same time.

5.2. Features Identification

[F1] Login and Registration Features

These are the core features required for students, companies, and administrators to access the platform. They include user registration, login, and secure authentication. As foundational features, they will be implemented first to support the functioning of subsequent features.

[F2] Profile Management Features

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

[F3] Internship Management Features

This includes the ability for companies to create, update, and delete internship postings. It also involves managing applications received for these internships. These features require the proper implementation of profile management (F2) and will be developed subsequently.

[F4] Search and Filter Features

This includes advanced search and filter functionalities for students to find relevant internships and for companies to search through applicants. These features depend on the successful implementation of profile and internship management (F2 and F3).

[F5] Application and Interview Features

This includes submitting applications, reviewing them, scheduling interviews, and notifying users about interview updates. These features depend on internship management (F3) and profile management (F2) functionalities.

[F6] Complaint Handling Features

This set of features allows students and companies to lodge complaints and administrators to review and resolve them. These features ensure user satisfaction and platform reliability. They rely on the proper implementation of profile management (F2) and dashboard functionalities.

[F7] Notification Features

These 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.

[F8] Dashboard Features

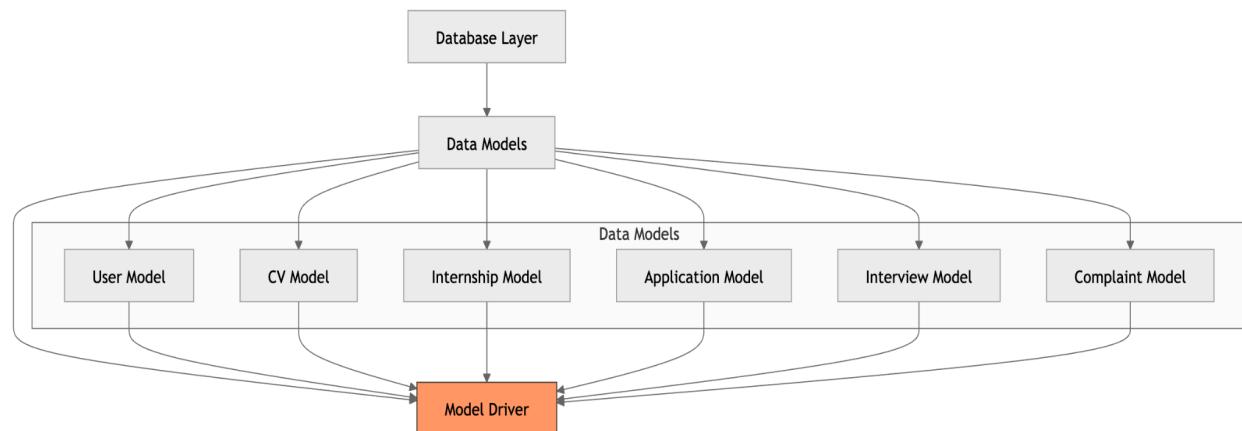
The dashboard provides an overview of active internships, applications, interviews, and platform activities for students, companies, and administrators. It consolidates data from various modules and is critical for system usability.

5.3. Implementation Strategy

5.3.1 Overview and Integration Plan

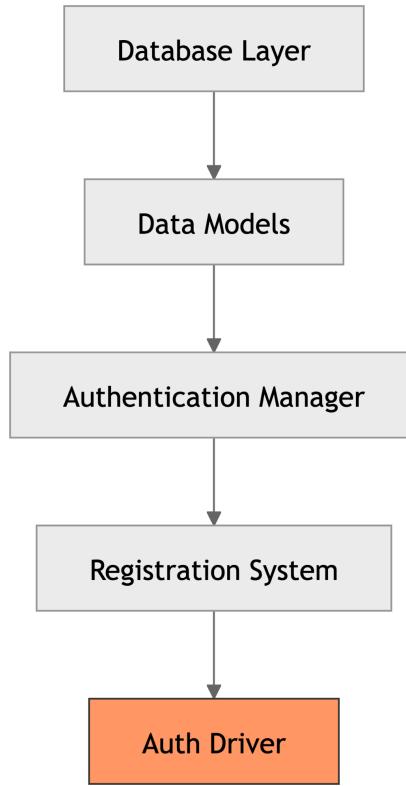
A systematic bottom-up method is used to integrate the system's components. The emphasis will be on developing solid fundamental modules that can be gradually combined, starting with the essential elements. Before each module is integrated into the larger system, it will be tested using the relevant drivers. This approach enables parallel programming, effective debugging, and incremental functional validation. The following crucial areas will be included in the integration process:

Core Model Integration:



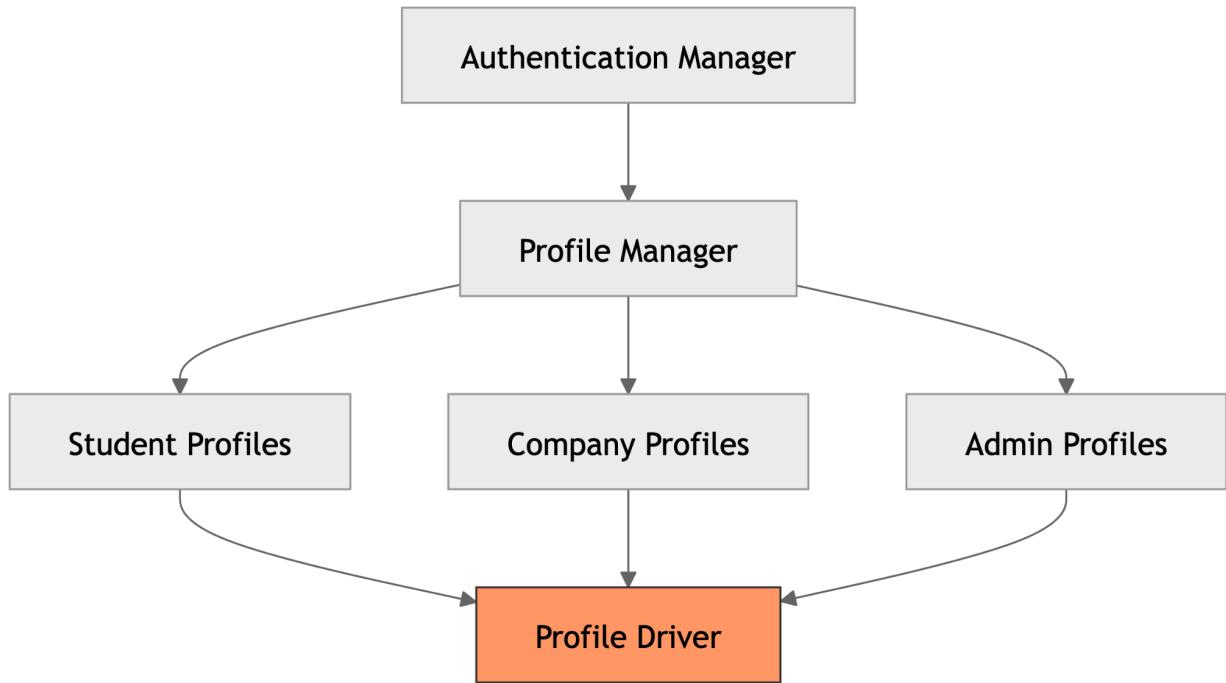
The core model integration, which includes the data models necessary for the platform's operation, serves as the system's cornerstone. The user model, resume model, internship model, application model, interview model, and complaint model are some examples of these models. In order to carry out CRUD 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:



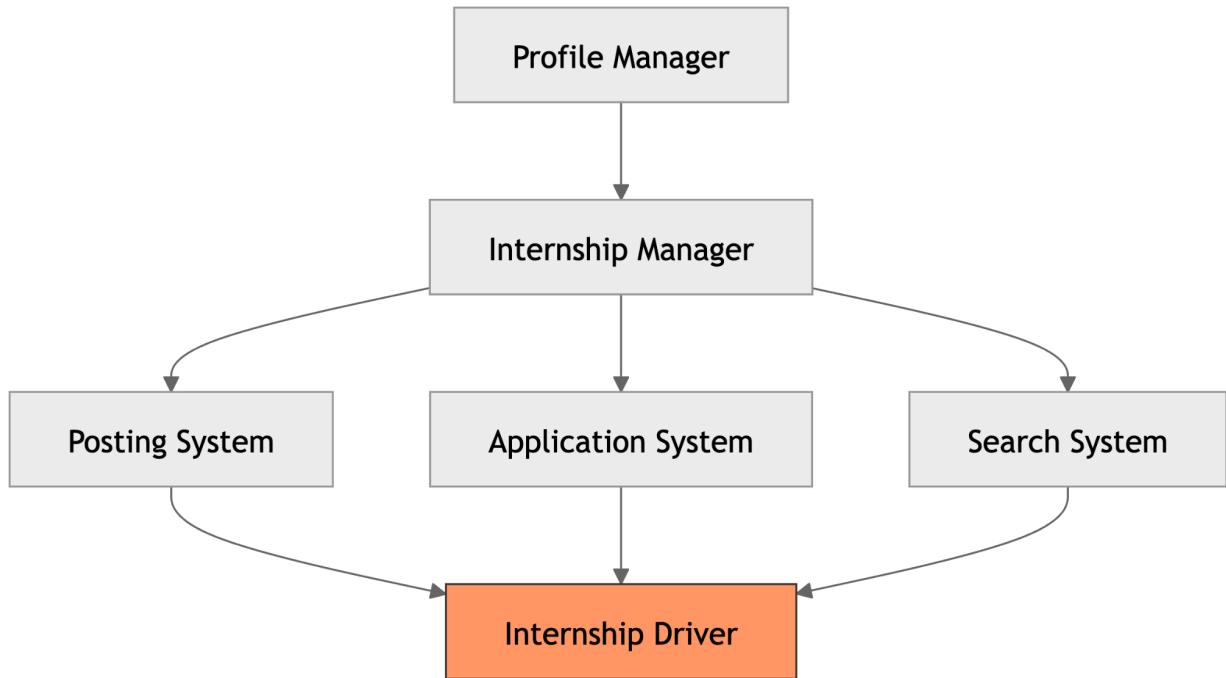
The procedures involved in user registration and login are the main focus of the authentication integration. Interactions between the data models and the user interfaces for registration and login will be managed by the Authentication Manager. It guarantees safe communication with the Registration System and data validation. 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:



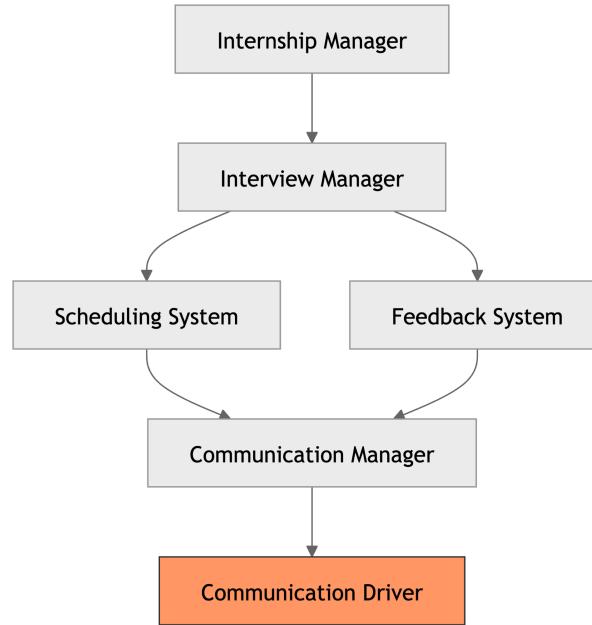
The Profile Manager, which manages the creation and administration of admin, company, and student profiles, is implemented as part of the Profile Management Integration. To guarantee that only authorized users are able to manage their profiles, this module communicates directly with the Authentication Manager. Features including creating, editing, and retrieving profiles will be tested using the Profile Driver. In order to enable individualized user experiences and create user identification across the platform, this integration is essential.

Internship Management Integration:



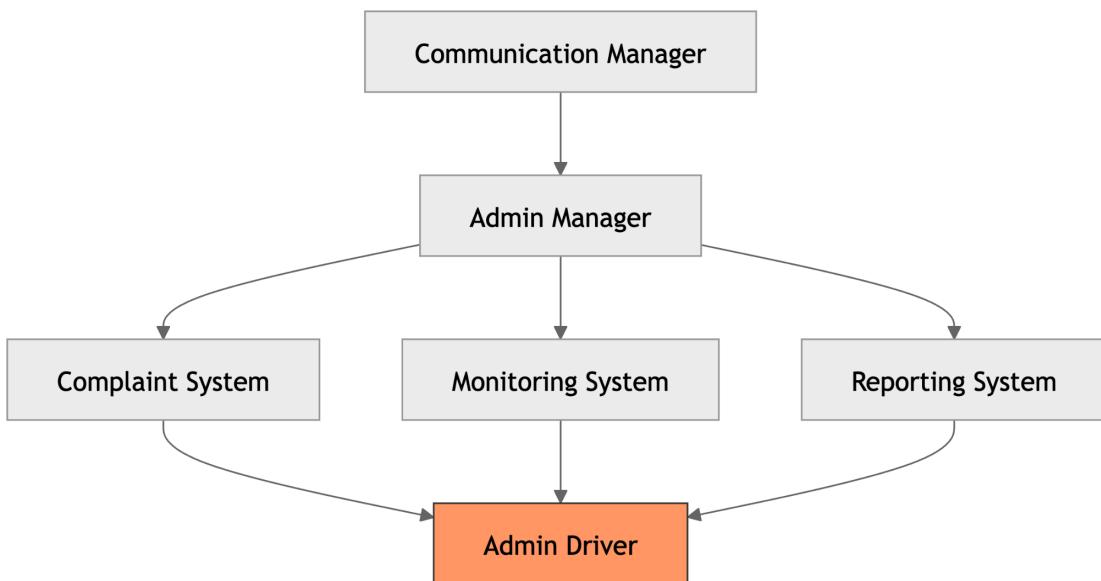
The Internship Manager is at the center of the Internship Management Integration, coordinating the posting and administration of internship opportunities. It communicates with the Profile Manager to confirm the responsibilities and permissions of recruiters. In order to manage student applications and to facilitate filtering and internship searches, it also interfaces with the Application System. Posting, removing, and searching for internships are among the features of the internship management system that a driver will try. This element guarantees safe and effective internship administration.

Interview and Communication Integration:



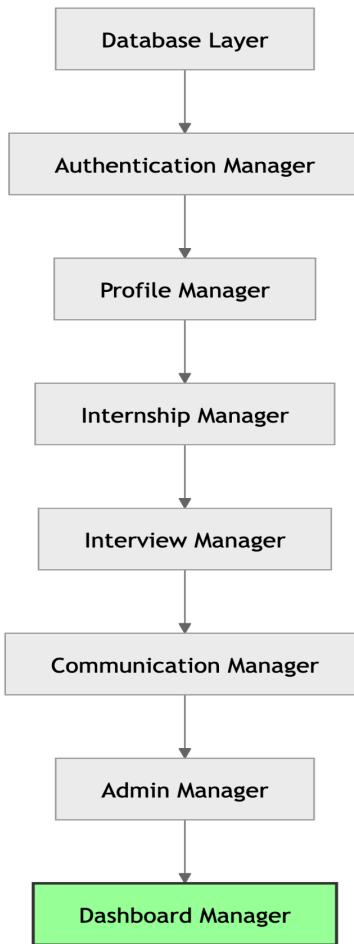
The Interview Manager and the Communication Manager, who are in charge of interview scheduling and stakeholder communication, are connected via the Interview and Communication Integration. Setting up interviews is handled by the Scheduling System, and post-interview feedback is handled by the Feedback System. 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:



In this step, the Admin Manager is integrated with systems such as the Monitoring System, the Reporting System, and the Complaint System. 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 guarantees that every element functions in unison to provide a seamless user experience. 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:

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 including applications, internship posts, profile management, login, interview scheduling, and complaint handling will be the main focus.

Load Testing

The system's behavior under various workloads will be evaluated through load testing in order to find memory leaks, buffer overflows, and ineffective 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.

Performance Testing

Performance testing will discover bottlenecks and assess how quickly the system responds to demanding workloads. By doing this, the platform is guaranteed to support many users at once with minimal latency. To improve overall system performance, optimization opportunities in the underlying algorithms will also be found.

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.

User Interface Testing

To confirm the platform's usability and accessibility across a variety of devices and browsers, user interface testing will be conducted. This testing will guarantee that all user types—students, businesses, and administrators—have a smooth and uniform experience with the system. Additionally, compatibility with various screen sizes and resolutions will be verified.

The system's functionality, dependability, and scalability will be fully verified by combining various testing techniques, guaranteeing a flawless user experience and compliance with the specifications listed in the RASD document.

6. Effort Spent

Team Member Contributions

Team Member	Task	Hours Spent
Shreesh Kumar Jha	<ol style="list-style-type: none">1. Introduction2. Architectural Design3. User Interface Design4. Requirements Traceability5. Implementation, Integration and Test Plan6. Effort Spent7. References	35
Samarth Bhatia	<ol style="list-style-type: none">1. Introduction2. Architectural Design3. User Interface Design4. Requirements Traceability5. Implementation, Integration and Test Plan6. Effort Spent7. References	35
Satvik Sharma	-	-

7. References

1. IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications
2. Software Engineering 2 Course Materials, A.Y. 2024-2025
3. Daniel Jackson, Software Abstractions: Logic, Language, and Analysis
4. Ian Sommerville, Software Engineering (10th Edition)
5. Assignment RDD AY 2024-2025.pdf
6. GDPR Documentation (<https://gdpr.eu/>)
7. Web Content Accessibility Guidelines (WCAG) 2.1
8. ISO/IEC 27001 Information Security Standards