

INTERNSHIP MANAGEMENT SYSTEM

The Open University Sri Lanka



March 1, 2025

Contents

[1. Introduction 3](#_Toc199622567)

[1.1 Purpose of the Document 3](#_Toc199622568)

[1.2 Scope of the Project 3](#_Toc199622569)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc199622570)

[1.4 References 3](#_Toc199622571)

[2. Overall Description 4](#_Toc199622572)

[2.1 Product Perspective 4](#_Toc199622573)

[2.2 Product Functions 4](#_Toc199622574)

[2.3 User Classes and Characteristics 5](#_Toc199622575)

[2.4 Operating Environment 5](#_Toc199622576)

[2.5 Design and Implementation Constraints 5](#_Toc199622577)

[2.6 User Documentation 6](#_Toc199622578)

[2.7 Assumptions and Dependencies 6](#_Toc199622579)

[3. Specific Requirements 7](#_Toc199622580)

[3.1 Functional Requirements 7](#_Toc199622581)

[3.1.1 Student Requirements 7](#_Toc199622582)

[3.1.2 Company Requirements 8](#_Toc199622583)

[3.1.3 Admin Requirements 9](#_Toc199622584)

[3.2 Non-Functional Requirements 10](#_Toc199622585)

[3.2.1 Performance Requirements 10](#_Toc199622586)

[3.2.3 Usability Requirements 11](#_Toc199622587)

[3.2.5 Maintainability Requirements 12](#_Toc199622588)

[3.2.6 Portability Requirements 12](#_Toc199622589)

[3.2.7 Data Integrity 12](#_Toc199622590)

[3.3 External Interface Requirements 12](#_Toc199622591)

[3.3.1 User Interfaces 12](#_Toc199622592)

[3.3.2 Hardware Interfaces 13](#_Toc199622593)

[3.3.3 Software Interfaces 13](#_Toc199622594)

[3.3.4 Communications Interfaces 13](#_Toc199622595)

[4. Use Case Models 14](#_Toc199622596)

[4.1 Use Case Diagrams 14](#_Toc199622597)

[4.2 Use Case Description 14](#_Toc199622598)

[4.2.1 Student Registration and Profile Management 14](#_Toc199622599)

[4.2.2 Search and Apply for Internships 15](#_Toc199622600)

[4.2.3 Internship Posting Management 15](#_Toc199622601)

[4.3.4 Application Review by Company 16](#_Toc199622602)

[4.2.5 User Verification 16](#_Toc199622603)

[4.2.6 Use Case: Generate System Reports 16](#_Toc199622604)

[4.2.7 Manage Internship Listings 17](#_Toc199622605)

[4.2.8 Manage Industry visit Listings 17](#_Toc199622606)

[5. Data Requirements 18](#_Toc199622607)

[5.1 Data Dictionary 18](#_Toc199622608)

[5.2 Entity-Relationship Diagrams (ERD) 21](#_Toc199622609)

[6. Interfaces 22](#_Toc199622610)

[7. Future Enhancements 28](#_Toc199622611)

# 1. Introduction

## 1.1 Purpose of the Document

This Software Requirements Specification (SRS) document describes the functionality and requirements for the Internship Registration System for the Open University of Sri Lanka. The system will enable students to register for internships, companies to offer internships, and university administrators to manage the entire process.

## 1.2 Scope of the Project

The system will include the following key functionalities:

* Student Registration and Profile Management: Students can register for the internship program, create and manage their profiles, and apply to internships.
* Company Registration and Internship Offerings: Companies can register on the platform, post internship opportunities, and manage their internship applications.
* Admin Management: University administrators can manage student registrations, company registrations, internship postings, and monitor the overall process.

The system will be accessible via a web-based interface.

## 1.3 Definitions, Acronyms, and Abbreviations

* SRS: Software Requirements Specification
* Internship System: The web-based platform being developed.
* Admin: University Administrator.
* Student: Registered student from the Open University of Sri Lanka.
* Company: A business entity offering internship opportunities.
* OUSL: Open University of Sri Lanka

## 1.4 References

* Open University of Sri Lanka Student Handbook
* IEEE Standard for Software Requirements Specifications

# 2. Overall Description

## 2.1 Product Perspective

The Internship Registration System will be a web-based platform where students, companies, and university administrators can interact and manage internship applications and offerings. The system will integrate with the Open University of Sri Lanka’s existing student database and will be user-friendly, secure, and scalable. The system aims to streamline the internship process, reducing the manual effort required by all parties involved.

## 2.2 Product Functions

The system will provide the following functions, grouped by user role:

* **Students:**
  + Register and manage profiles.
  + Search for internships based on various criteria.
  + Apply for internships.
  + Track application status.
  + View accepted, pending, or rejected offers.
* **Companies:**
  + Register and manage company profiles.
  + Post internship opportunities with detailed descriptions.
  + Review student applications.
  + Select students for internships.
  + Communicate with students (optional).
* **University Administrators:**
  + Manage student and company registrations (approve/reject).
  + Manage internship listings.
  + Monitor internship applications and the assignment process.
  + Generate reports on system usage, student placements, and company participation.
  + Maintain system data and ensure proper functioning.
  + Handle user support and inquiries.

## 2.3 User Classes and Characteristics

* Students: Individuals who are enrolled in the Open University of Sri Lanka and are seeking internships.
  + Requirements: Ability to register, search, and apply for internships.
  + Skills: Basic web browsing.
  + Experience: May have limited prior experience with online application systems.
* Companies: External organizations offering internship opportunities to students.
  + Requirements: Ability to register, post internship listings, and review student applications.
  + Skills: Basic web browsing.
  + Experience: May vary in experience with online recruitment platforms.
* University Admin: Open University staff who manage the system, verify student and company registrations, and monitor internship offers and applications.
  + Requirements: Manage system data and ensure proper functioning.
  + Skills: Advanced system knowledge, administrative skills, reporting, and data analysis.
  + Experience: Familiarity with student information systems and university procedures.

## 2.4 Operating Environment

The system will be web-based, accessible via any modern browser (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge). The backend will run on a cloud-based server (e.g., AWS, Azure, Google Cloud) with scalable architecture. The system will be designed to be responsive, ensuring usability on various devices, including desktops, laptops, tablets, and smartphones.

## 2.5 Design and Implementation Constraints

* Technology limitations: The system should be developed using open-source technologies (e.g., Python, Node.js) to minimize licensing costs.
* Security requirements: The system must comply with OUSL security policies, including secure authentication, data encryption (using HTTPS), and protection against common web vulnerabilities (e.g., SQL injection, XSS).
* Performance requirements: The system should be able to handle a large number of concurrent users (e.g., 500+ students, 100+ companies) with minimal response time (e.g., less than 2 seconds for page loads).
* Data Privacy: The system must comply with relevant data privacy regulations (e.g., GDPR) to protect student and company information.
* Integration: The system must integrate with the OUSL student database for authentication and data synchronization.

## 2.6 User Documentation

The following user documentation will be provided:

* Student User Manual: A comprehensive guide for students on how to register, search for internships, apply, and manage their profiles.
* Company User Manual: A guide for companies on how to register, post internships, review applications, and select students.
* Admin User Manual: A detailed manual for university administrators on system management, user verification, report generation, and system maintenance.
* Online Help System: Context-sensitive help within the application to assist users with specific tasks.
* FAQs: A list of frequently asked questions and their answers for all user roles.

## 2.7 Assumptions and Dependencies

* The university will provide access to the student database for authentication and profile management.
* The system will be hosted on a secure, scalable web server.
* Companies will provide accurate and complete information about their internship programs.
* Students will have access to computers and internet connectivity.
* Regular backups of the system data will be performed by the university's IT department.

# 3. Specific Requirements

## 3.1 Functional Requirements

### 3.1.1 Student Requirements

* Student Registration and Profile Management
  + Description: Students will be able to register on the platform, create profiles, and manage their internship applications.
  + Functional Requirements:
    - Student Registration:
      * Students must be able to register with their details.
      * The system will validate student information through the university database to ensure accuracy and prevent duplicate accounts.
      * Upon successful registration, the system will send a confirmation email to the student.
    - Profile Management:
      * Students can update their personal information (e.g., address, contact details), upload resumes/CVs, and set their internship preferences (e.g., field of study, location, desired skills).
      * The system will allow students to specify their availability (e.g., start date, duration).
      * Students can manage their privacy settings to control the visibility of their profile information to companies.
    - Search and Apply for Internships:
      * Students can search for internship opportunities based on criteria such as location, field, company, and duration.
      * The system will provide filtering and sorting options to refine search results.
      * Students can view detailed information about each internship, including job description, requirements, and company profile.
      * Students can apply to internships and track the status of their applications (e.g., pending, reviewed, shortlisted, offered, rejected).
      * The system will notify students of updates to their application status (via email or in-app notifications).
    - Application History:
      * Students can view a history of their applications, including accepted, pending, or rejected offers.
      * The system will allow students to withdraw applications that are still pending.

### 3.1.2 Company Requirements

* Company Registration and Profile Management
  + Description: Companies will be able to register on the platform, create profiles, and post internship opportunities.
  + Functional Requirements:
    - Company Registration:
      * Companies must register by providing their organization details (e.g., name, address, industry, company registration number) and contact information for the internship coordinator.
      * The system will verify company information (e.g., through a business registry API, manual admin review).
      * Upon successful registration, the system will send a confirmation email to the company.
    - Company Profile Management:
      * Companies can create and update their company profiles, including a description of their organization, industry, size, and culture.
      * Companies can upload their logo and other relevant information to enhance their profile.
    - Internship Posting Management:
      * Companies can post internship opportunities with detailed information, including:
        + Job title and description
        + Required skills and qualifications
        + Location and duration
        + Start date and application deadline
        + Compensation (if applicable)
        + Number of available positions
      * The system will allow companies to specify the type of internship (e.g., full-time, part-time).
      * Companies can edit, update, or delete their internship postings.
    - Application Review and Management:
      * Companies can view student applications for their internship postings.
      * The system will provide tools for companies to filter, sort, and shortlist applicants based on criteria such as GPA, skills, and experience.
      * Companies can communicate with applicants (e.g., send messages, schedule interviews) through the system (optional, depending on design).
      * Companies can select students for internships and notify them of their decision.
      * Companies can track the status of their internship offers (e.g., pending, accepted, declined).
      * Companies can provide feedback to students on their applications (optional).

### 3.1.3 Admin Requirements

* Admin Management
  + Description: University administrators will have full control over the system, enabling them to monitor, manage, and update the entire internship process.
  + Functional Requirements:
    - Student and Company Verification:
      * Admins can review and approve or reject student and company registrations to ensure data accuracy and system integrity.
      * The system will provide admins with the necessary information to verify user credentials and company legitimacy.
    - Manage Internship Listings:
      * Admins can view and manage all internship postings from companies.
      * Admins can edit or remove inappropriate or expired internship listings.
      * Admins can ensure that internship postings comply with university policies and guidelines.
    - Monitor Internship Applications:
      * Admins can view student applications and track the progress of the internship assignment process.
      * Admins can intervene in the application process if necessary (e.g., resolve disputes, facilitate communication).
    - Generate Reports:
      * Admins can generate reports for internal and external auditing, including statistics on:
        + Registered students (e.g., by faculty, department)
        + Registered companies (e.g., by industry)
        + Internship postings (e.g., by field, location)
        + Student applications and placements
        + System usage and activity
      * Admins can customize reports based on specific criteria and timeframes.
      * Admins can export reports in various formats (e.g., PDF, Excel).
    - System Configuration and Maintenance:
      * Admins can configure system settings, such as user roles, permissions, and notifications.
      * Admins can manage system data, including database backups and recovery.
      * Admins can monitor system performance and troubleshoot issues.
      * Admins can manage user accounts and access privileges.
      * Admins can update system content and announcements.

## 3.2 Non-Functional Requirements

### 3.2.1 Performance Requirements

* The system should be responsive, with page load times of less than 2 seconds under normal operating conditions.
* The system should be able to handle a peak load of 500 concurrent student users and 100 concurrent company users without significant performance degradation.
* The system should be scalable to accommodate future growth in the number of users and data volume.
* Database queries should be optimized for efficient data retrieval.
* The system should utilize caching mechanisms to improve performance.

3.2.2 Security Requirements

* Authentication:
  + The system will use secure password storage (e.g., hashing with salt).
  + Students will authenticate using their university credentials, verified against the OUSL database.
  + Companies will authenticate using a secure login and password.
  + Administrators will authenticate using a separate, highly secure login.
  + The system will support password recovery and reset functionality.
  + The system will enforce strong password policies (e.g., minimum length, complexity).
  + Two-factor authentication (2FA) will be implemented for administrators. Optional for Students and Companies.
* Authorization:
  + The system will implement role-based access control (RBAC) to restrict access to sensitive data and functionality based on user roles (e.g., student, company, admin).
  + Students can only access their own profile information and application data.
  + Companies can only access information related to their own internship postings and applicants.
  + Administrators have full access to all system data and functionality.
* Data Encryption:
  + All sensitive data, including passwords, personal information, and application data, will be encrypted both in transit (using HTTPS) and at rest (in the database).
  + The system will use industry-standard encryption algorithms (e.g., AES-256).
* Vulnerability Protection:
  + The system will be protected against common web vulnerabilities, such as:
    - SQL injection
    - Cross-site scripting (XSS)
    - Cross-site request forgery (CSRF)
    - Session hijacking
  + Regular security audits and penetration testing will be conducted to identify and address potential vulnerabilities.
* Audit Logging:
  + The system will maintain a detailed audit log of all security-related events, such as login attempts, failed login attempts, and data access.

### 3.2.3 Usability Requirements

* Ease of Use:
  + The system will have a user-friendly interface that is intuitive and easy to navigate for all user roles.
  + The system will provide clear and concise instructions and feedback to users.
  + The system will minimize the number of steps required to complete common tasks.
* User Interface Design:
  + The system will adhere to modern UI/UX principles and guidelines.
  + The system will have a consistent design and layout across all pages.
  + The system will use clear and consistent terminology.
  + The system will be visually appealing and engaging.
* Accessibility:
  + The system will support keyboard navigation.
  + The system will provide alternative text for images.
  + The system will use sufficient color contrast.
* Responsiveness:
  + The system will be responsive and accessible on various devices, including desktops, laptops, tablets, and smartphones.

3.2.4 Reliability Requirements

* Availability:
  + The system will be available 24/7, with a target uptime of 99.9%.
  + The system will be hosted on a reliable cloud platform with redundancy and failover capabilities.
  + Regular backups of system data will be performed to minimize data loss in case of system failures.
* Fault Tolerance:
  + The system will be designed to handle errors gracefully and prevent system crashes.
  + The system will implement error logging and monitoring to identify and address issues promptly.

### 3.2.5 Maintainability Requirements

* Code Maintainability:
  + The system will be developed using a modular and well-structured architecture.
  + The codebase will be well-documented and follow coding standards.
  + The system will use version control (e.g., Git) to manage code changes.
* Ease of Updates:
  + The system will be designed to allow for easy updates and modifications without requiring significant downtime.
  + The system will use a configuration management system for easy deployment.

### 3.2.6 Portability Requirements

* Browser Compatibility: The system will be compatible with the latest versions of major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
* Cross-Platform Compatibility: The system should be deployable on different operating systems (e.g., Linux, Windows).

### 3.2.7 Data Integrity

* Data Validation:
  + The system will validate all user inputs to ensure data accuracy and consistency.
  + The system will use server-side validation to prevent data corruption.
  + The system will provide clear error messages to users for invalid data entries.
* Data Consistency:
  + The system will maintain data consistency across all modules and functions.
  + The system will use database transactions to ensure that data is updated correctly.

## 3.3 External Interface Requirements

### 3.3.1 User Interfaces

* The user interface will be web-based and accessible through modern web browsers.
* The UI will follow a consistent design language and branding guidelines.
* The UI will be responsive and adapt to different screen sizes and devices.
* The UI will be intuitive and easy to navigate, with clear calls to action.
* The UI will provide feedback to users for their actions (e.g., success messages, error messages).
* The UI will adhere to accessibility standards (e.g., WCAG).
* The system will use a modern UI framework (e.g., React, Angular, Vue.js).

### 3.3.2 Hardware Interfaces

* The system will not require any specific hardware interfaces beyond a standard web browser and internet connection.
* The server will be hosted on cloud infrastructure.

### 3.3.3 Software Interfaces

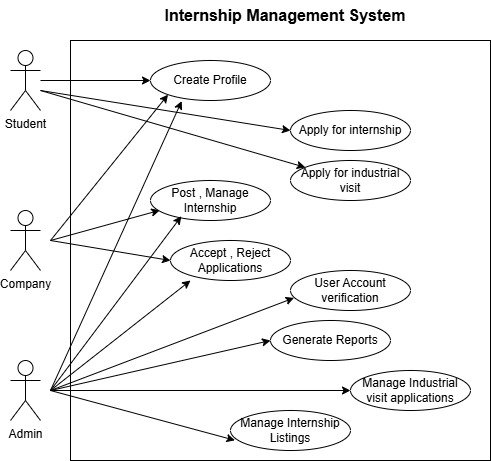
* The system will integrate with the Open University of Sri Lanka's student database for student authentication and data retrieval. APIs will be used for this integration.
* The system may integrate with third-party services, such as:
  + Email services for sending notifications (e.g., SendGrid, Mailgun).
  + Mapping services for location-based search (e.g., Google Maps API).

### 3.3.4 Communications Interfaces

* The system will use standard web protocols (e.g., HTTP, HTTPS) for communication between the client and server.
* The system will send email notifications to students and companies for important events, such as application status updates and new internship postings.

# 4. Use Case Models

## 4.1 Use Case Diagrams



## 4.2 Use Case Description

### 4.2.1 Student Registration and Profile Management

Use Case ID: UC001

Actor: Student

Description: Allows students to register, create, and manage their profiles.

Preconditions: The student is not already registered.

Post conditions: A new student account is created and active.

Basic Flow:

Student accesses the registration form.

Student enters required information.

System validates the data with the university database.

Upon success, an email confirmation is sent.

Student logs in and creates their profile including resume, skills, preferences.

Alternate Flows:

Validation fails → System shows an error.

Missing mandatory data → System prompts for completion.

Exceptions:

Network issues.

Invalid university records.

### 4.2.2 Search and Apply for Internships

Use Case ID: UC002

Actor: Student

Description: Students can search for internships and apply based on preferences.

Preconditions: Student is logged in and profile is completed.

Post conditions: Application is recorded and tracked.

Basic Flow:

Student logs in.

Navigates to the internship listing.

Applies filters (location, skills, field).

Views detailed internship post.

Clicks apply.

System records application and updates the status.

Alternate Flows:

No matching results → System shows a message.

Already applied → Show status.

Exceptions:

Internship deadline passed.

Internship removed by admin.

### 4.2.3 Internship Posting Management

Use Case ID: UC003

Actor: Company

Description: Allows companies to create and manage internship listings.

Preconditions: Company account is verified and logged in.

Post conditions: Internship posting is live.

Basic Flow:

Company logs in.

Navigates to “Post Internship”.

Fills in internship details.

Submits posting.

System publishes and displays the listing.

Alternate Flows:

Incomplete information → System prompts to complete fields.

Exceptions:

Posting deadline date format error.

Company under review.

### 4.3.4 Application Review by Company

Use Case ID: UC004

Actor: Company

Description: View and manage applications submitted by students.

Preconditions: Internship is active and has applications.

Post conditions: Applications are filtered, shortlisted, or responded to.

Basic Flow:

Company logs in and navigates to received applications.

Filters by qualifications.

Reviews each applicant.

Shortlists, schedules interviews, or sends decisions.

Alternate Flows:

No applications yet → System shows message.

### 4.2.5 User Verification

Use Case ID: UC005

Actor: Admin

Description: Admin verifies student and company registrations.

Preconditions: New user registration is pending verification.

Post conditions: User is approved or rejected.

Basic Flow:

Admin logs into the dashboard.

Reviews pending registrations.

Verifies details (student via university, company via registry).

Approves or rejects.

System sends notification to the user.

### 4.2.6 Use Case: Generate System Reports

Use Case ID: UC006

Actor: Admin

Description: Generate reports for audits and tracking.

Preconditions: Admin logged in.

Post conditions: Reports generated in desired format.

Basic Flow:

Admin navigates to “Reports”.

Selects type and criteria (timeframe, department, etc.).

System compiles and generates reports.

Admin downloads or views report.

### 4.2.7 Manage Internship Listings

Use Case ID: UC007

Actor: Admin

Description: Admin monitors and moderates internship posts.

Preconditions: Company postings exist.

Post conditions: Listings remain compliant and updated.

Basic Flow:

Admin views all internship listings.

Filters for flagged or expired posts.

Edits or deletes inappropriate entries.

### 4.2.8 Manage Industry visit Listings

Use Case ID: UC008

Actor: Admin

Description: Admin monitors and moderate industry visit requests.

Preconditions: Student internship request exist.

Post conditions: Listings remain compliant and updated.

Basic Flow:

Admin views all visit request listings.

Filters for based on the status.

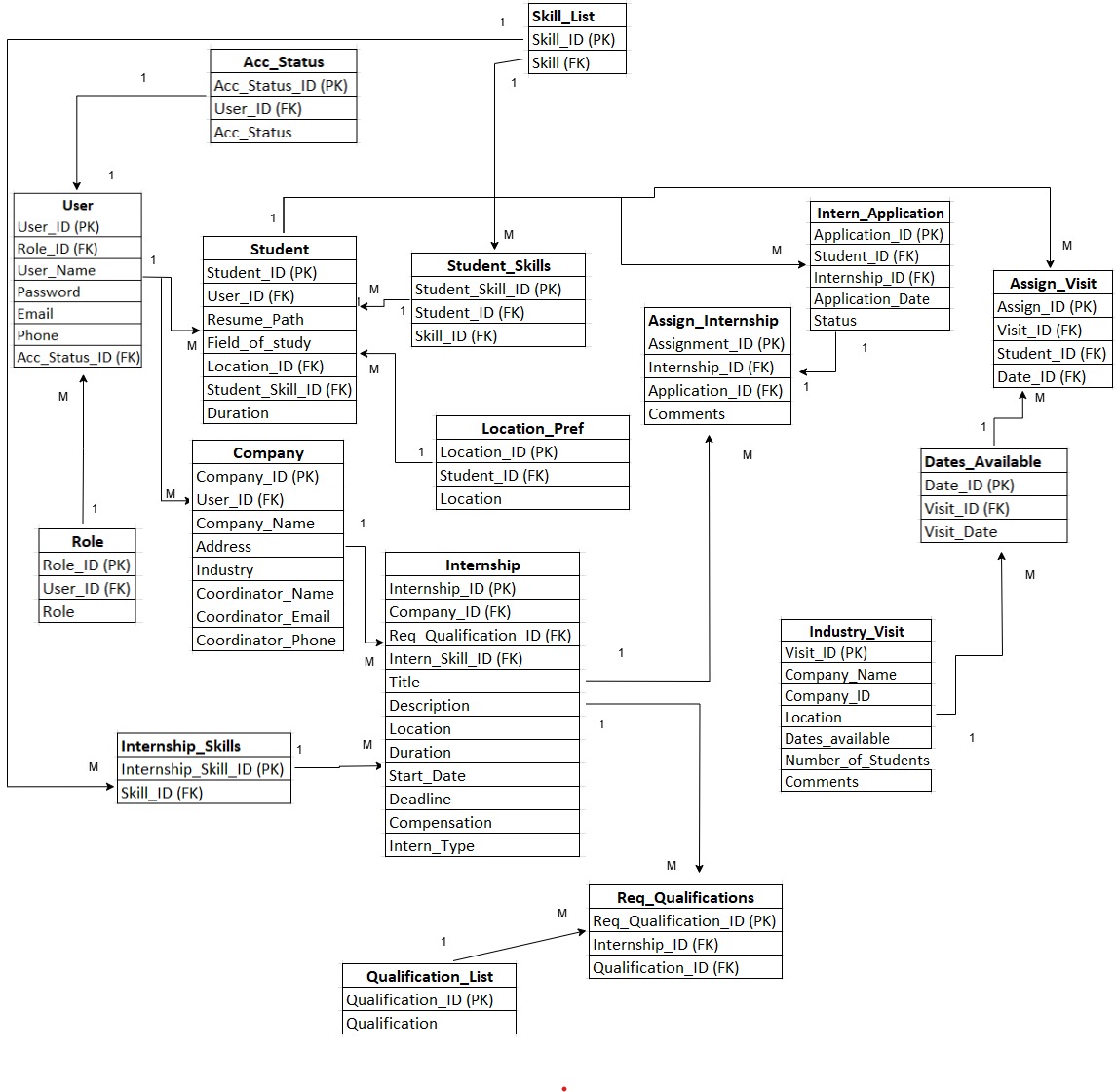
Accept , reject, assign visit details.

# 5. Data Requirements

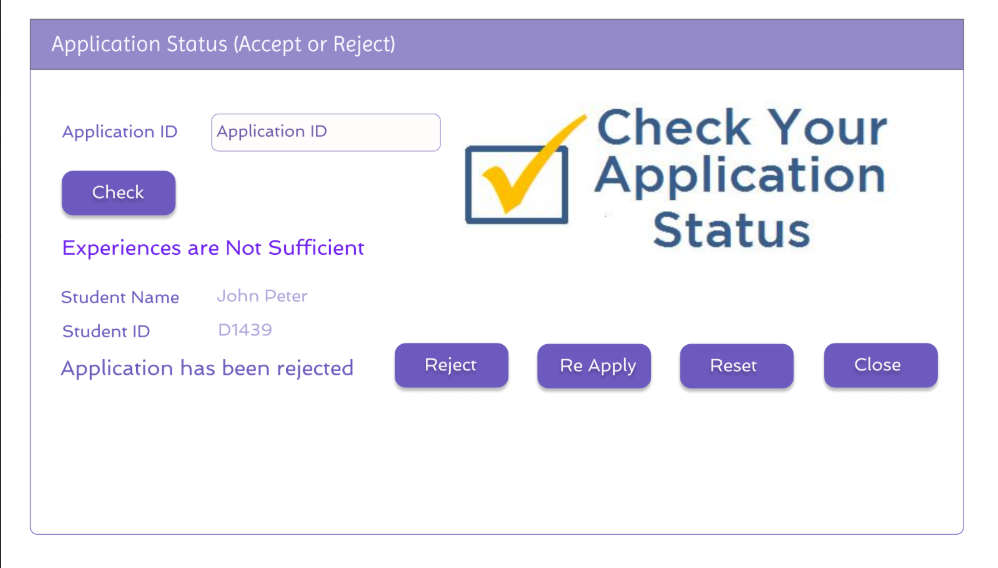
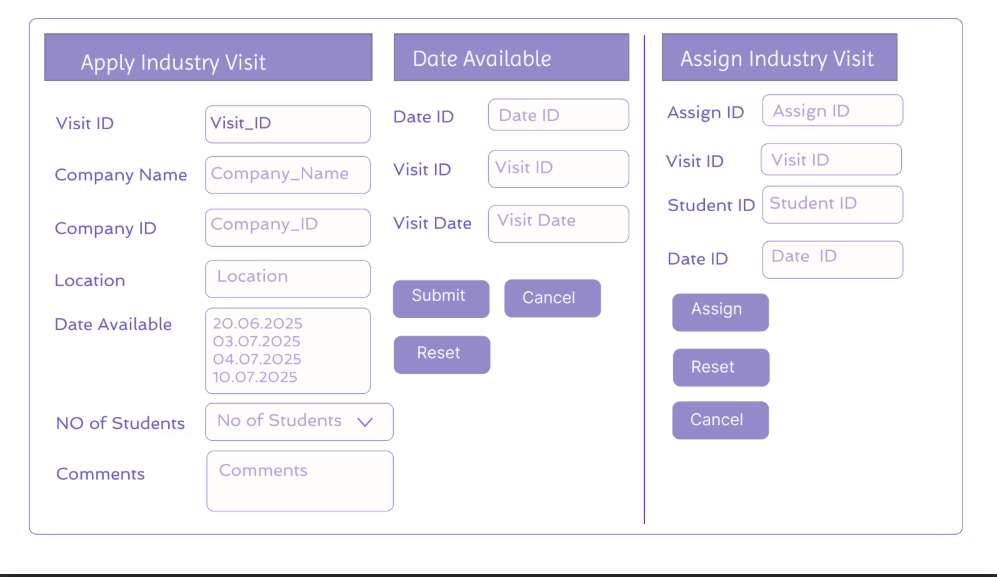
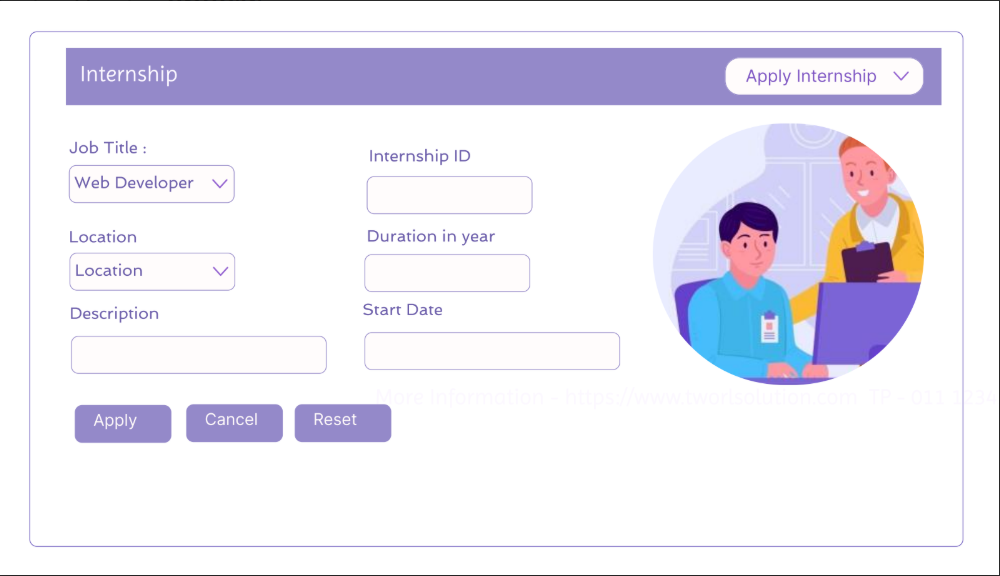
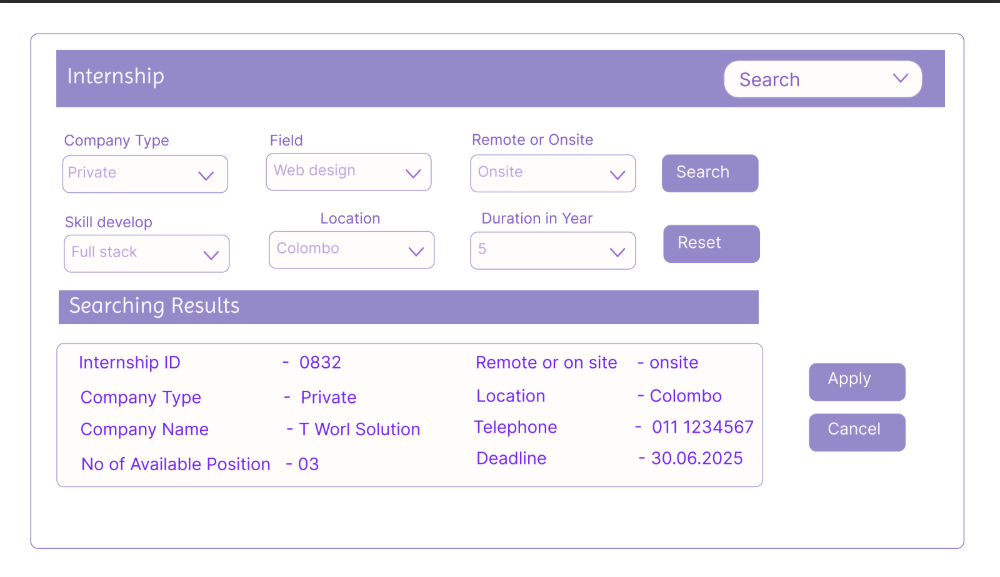
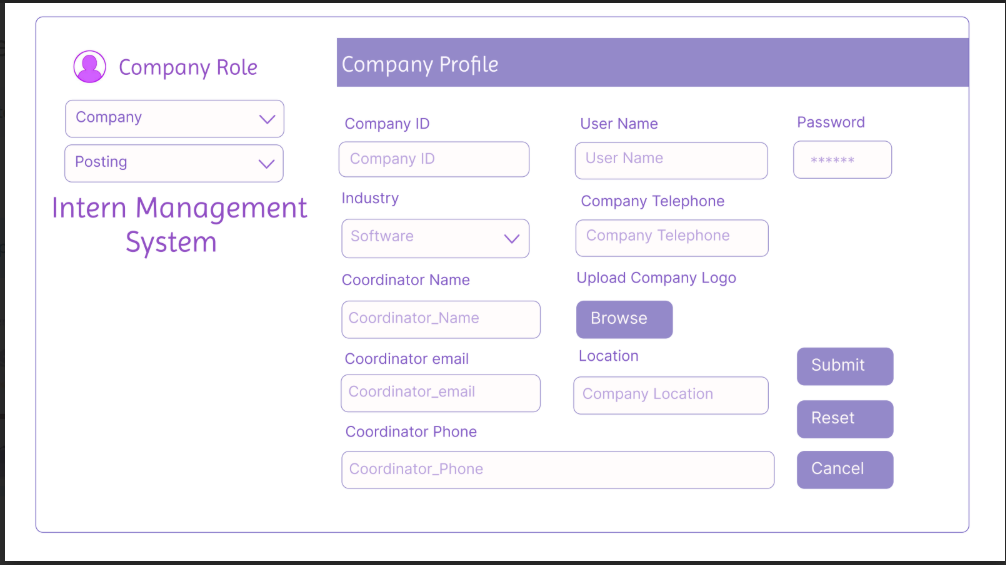
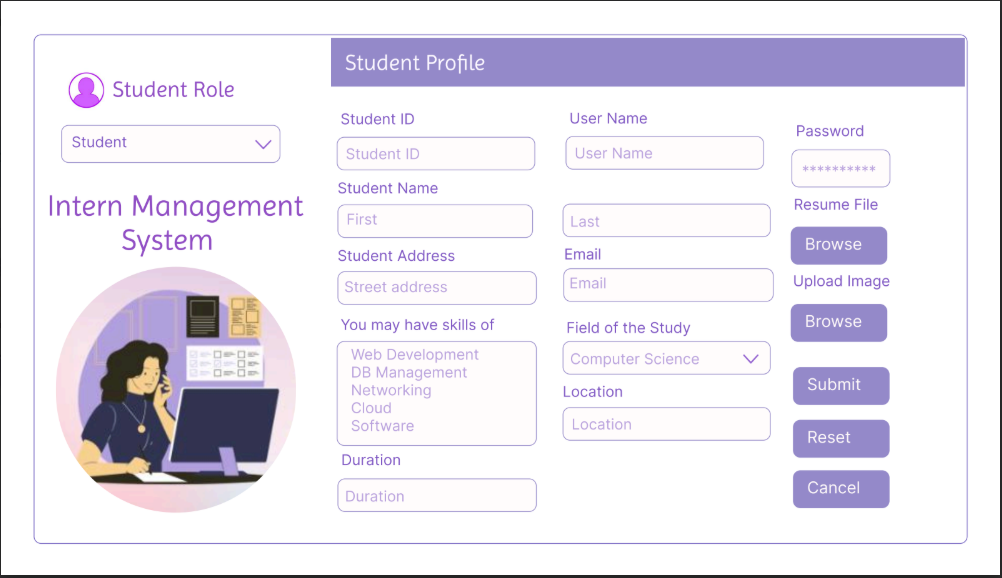
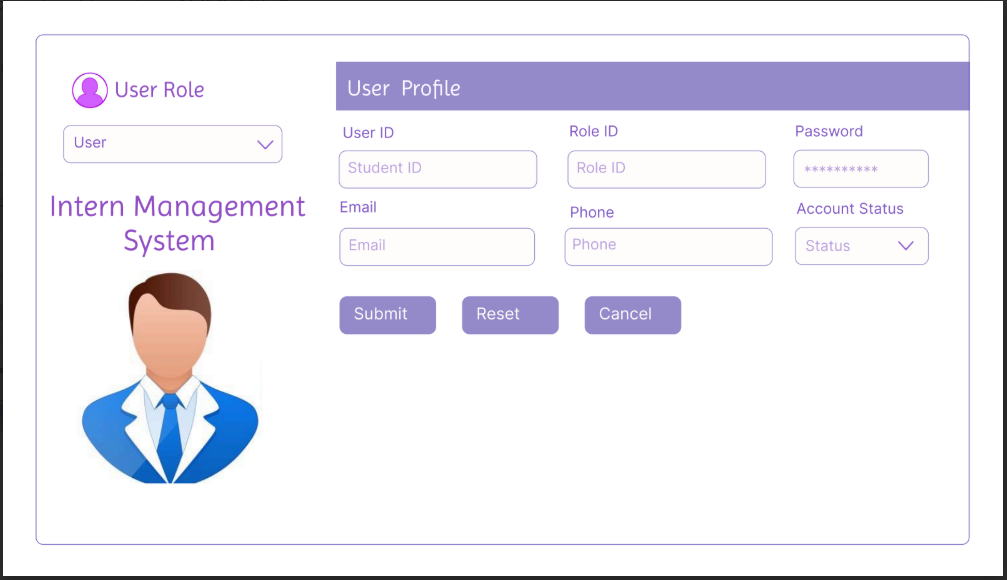
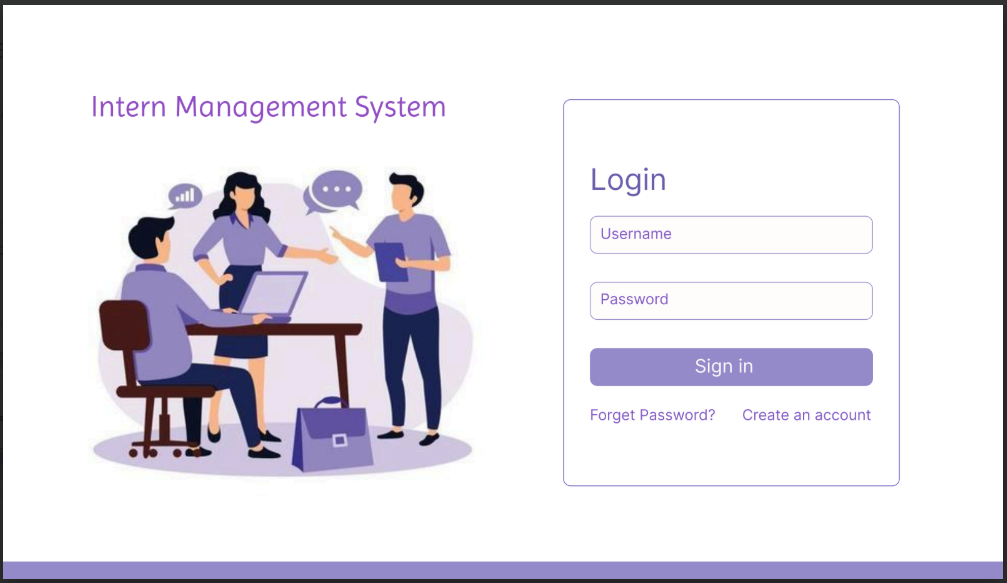
## 5.1 Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element Name** | **Description** | **Data Type** | **Size** | **Constraints** |
| User Name | Display name of the user | Varchar | 100 | Not Null |
| User ID | Unique identifier for a user | Integer | - | Primary Key, Unique, Not Null |
| User Role | Role of the user (Student, Admin,Company) | Varchar | 20 | Not Null, ('Student', 'Admin','Company') |
| Student ID | Unique identifier for a student | Integer | - | System-generated, Foreign Key (User ID) |
| First Name | First name of the user | Varchar | 50 | Optional for User, Not Null for Admin |
| Last Name | Last name of the user | Varchar | 50 | Optional for User, Not Null for Admin |
| Email | Email address of the user | Varchar | 100 | Optional for User, Not Null for Admin, Unique, Valid email format |
| Phone Number | Phone number of the user | Varchar | 20 | Optional |
| Address | Address of the user | Varchar | 255 | Optional |
| Resume/CV | File path to the student's resume/CV | Varchar | 255 | Optional, Stores path/URL |
| Internship Preferences (Field of Study) | Student's preferred field of study for internships | Varchar | 100 | Optional |
| Internship Preferences (Location) | Student's preferred location for internships | Varchar | 100 | Optional |
| Internship Preferences (Desired Skills) | Student's desired skills for internships | Varchar | 255 | Optional |
| Internship Preferences (Availability - Start Date) | Student's preferred internship start date | Date | - | Optional |
| Internship Preferences (Availability - Duration) | Student's preferred internship duration | Varchar | 50 | Optional |
| Admin ID | Unique identifier for an Admin | Integer | - | System-generated, Foreign Key (User ID) |
| Company ID | Unique identifier for a company | Integer | - | Primary Key, System-generated, Unique, Not Null |
| Company Name | Name of the company | Varchar | 255 | Not Null, Unique |
| Company Address | Physical address of the company | Varchar | 255 | Not Null |
| Industry | Industry sector of the company | Varchar | 100 | Not Null |
| Internship Coordinator Name | Name of the company's internship coordinator | Varchar | 100 | Not Null |
| Internship Coordinator Email | Email of the company's internship coordinator | Varchar | 100 | Not Null, Unique, Valid email format |
| Internship Coordinator Phone Number | Phone number of the company's internship coordinator | Varchar | 20 | Optional |
| Internship ID | Unique identifier for an internship listing | Integer | - | Primary Key, System-generated, Unique, Not Null |
| Job Title | Title of the internship position | Varchar | 100 | Not Null |
| Description | Detailed description of the internship | Text | - | Not Null |
| Required Skills | Skills required for the internship | Varchar | 255 | Not Null, Comma-separated list |
| Qualifications | Educational or other qualifications | Varchar | 255 | Not Null |
| Location | Location of the internship | Varchar | 100 | Not Null |
| Duration | Duration of the internship | Varchar | 50 | Not Null |
| Start Date | Start date of the internship | Date | - | Not Null |
| Application Deadline | Deadline for submitting applications | Date | - | Not Null |
| Compensation | Details about compensation (e.g., stipend) | Varchar | 100 | Optional |
| Number of Available Positions | Number of open positions for this internship | Integer | - | Not Null, Greater than 0 |
| Internship Type | Type of internship (Full-time/Part-time) | Varchar | 20 | Not Null, ('Full-time', 'Part-time') |
| Posting Date | Date when the internship was posted | Date | - | Not Null, Automatically set to current date |
| Company Registration Number (FK) | Foreign Key to Company ID | Integer | - | Foreign Key to Company.Company ID, Not Null |
| Application ID | Unique identifier for an application | Integer | - | Primary Key, System-generated, Unique, Not Null |
| Application Date | Date when the application was submitted | Date | - | Not Null, Automatically set to current date |
| Application Status | Current status of the application | Varchar | 50 | Not Null, Enum ('Pending', 'Reviewed', 'Shortlisted', 'Offered', 'Rejected', 'Withdrawn') |

## 5.2 Entity-Relationship Diagrams (ERD)



# 6. Interfaces



# 7. Future Enhancements

* Implement a messaging system for communication between students and companies.
* Add a feature for students to rate and review their internship experiences.
* Incorporate a recommendation system to suggest relevant internships to students.
* Develop a mobile app for students and companies.
* Integrate with social media platforms for sharing internship opportunities.
* Implement a calendar feature to track important dates.
* Add functionality for companies to manage the intern evaluation process.
* Implement functionality for alumni to post internships.