

# Software Requirement Specification

for

**RIT Placement Management System**

Prepared by

**Anupriya Suresh**

**Neha Elizabeth Thomas**

**Anjana K Santhosh**

**Asma PA**

**Department of Computer Science and Engineering**

**Rajiv Gandhi Institute of Technology, Kottayam**

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Problems with the Current Approach . . . . .	4
1.3	Solutions with the Placement Management Website . . . . .	4
1.4	Document Conventions . . . . .	5
1.5	Intended Audience and Reading Suggestions . . . . .	5
1.6	Project Scope . . . . .	6
1.7	References . . . . .	6
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product Perspective . . . . .	6
2.2	Product Features . . . . .	7
2.3	User Classes and Characteristics . . . . .	7
2.4	Operating Environment . . . . .	7
2.5	User Documentation . . . . .	8
2.6	Assumptions and Dependencies . . . . .	8
<b>3</b>	<b>System Features</b>	<b>8</b>
3.1	User Authentication . . . . .	8
3.1.1	Description and Priority . . . . .	8
3.1.2	Stimulus/Response Sequences . . . . .	8
3.1.3	Functional Requirements . . . . .	9
3.2	Placement Event Management . . . . .	9
3.2.1	Description and Priority . . . . .	9
3.2.2	Stimulus/Response Sequences . . . . .	9
3.2.3	Functional Requirements . . . . .	9
3.3	Student Registration for Placement Drives . . . . .	9
3.3.1	Description and Priority . . . . .	9
3.3.2	Stimulus/Response Sequences . . . . .	10
3.3.3	Functional Requirements . . . . .	10
3.4	Notifications and Alerts . . . . .	10
3.4.1	Description and Priority . . . . .	10
3.4.2	Stimulus/Response Sequences . . . . .	10
3.4.3	Functional Requirements . . . . .	10
3.5	Progress and Result Tracking . . . . .	10
3.5.1	Description and Priority . . . . .	10
3.5.2	Stimulus/Response Sequences . . . . .	11
3.5.3	Functional Requirements . . . . .	11
3.6	Resume Upload and Management . . . . .	11
3.6.1	Description and Priority . . . . .	11
3.6.2	Stimulus/Response Sequences . . . . .	11
3.6.3	Functional Requirements . . . . .	11
3.7	Interview Scheduling . . . . .	11
3.7.1	Description and Priority . . . . .	11
3.7.2	Stimulus/Response Sequences . . . . .	12
3.7.3	Functional Requirements . . . . .	12
3.8	Placement Statistics and Reports . . . . .	12
3.8.1	Description and Priority . . . . .	12
3.8.2	Stimulus/Response Sequences . . . . .	12
3.8.3	Functional Requirements . . . . .	12

<b>4</b>	<b>External Interface Requirements</b>	<b>12</b>
4.1	User Interfaces . . . . .	12
4.2	Hardware Interfaces . . . . .	13
4.3	Software Interfaces . . . . .	13
4.4	Communications Interfaces . . . . .	14
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>15</b>
5.1	Performance Requirements . . . . .	15
5.2	Safety Requirements . . . . .	15
5.3	Security Requirements . . . . .	15
5.4	Software Quality Attributes . . . . .	16
<b>6</b>	<b>Other Requirements</b>	<b>17</b>
<b>A</b>	<b>Appendix A: Glossary</b>	<b>17</b>
<b>B</b>	<b>Appendix B: Usecase Diagram</b>	<b>18</b>
<b>C</b>	<b>Appendix C: Issues List</b>	<b>19</b>

# 1 Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) document defines the requirements for the Placement Management System developed for Rajiv Gandhi Institute of Technology (RIT), Pampady. The product aims to automate and streamline placement-related activities, including event management, student registrations, notifications, and progress tracking.

## 1.2 Problems with the Current Approach

### 1. Dependency on Emails

- Important notifications may be missed if students fail to check their emails regularly.
- Email clutter can cause confusion, leading to missed deadlines or actions.

### 2. Manual Registration Process

- Payments are manually tracked, leading to errors or delays in confirming registrations.
- Coordination between payment and registration may require additional effort.

### 3. Eligibility Criteria Verification

- Tracking students' attendance for training sessions manually can be time-consuming and prone to errors.

### 4. Inefficient Communication

- Emails are one-way communication, and resolving queries might take longer without direct interaction or FAQs.

### 5. Limited Visibility for Students

- Students lack a centralized platform to view upcoming placement events, registered events, results, or progress.

### 6. Database Maintenance

- Manual database management by coordinators can lead to data inconsistencies, duplication, or loss.

### 7. Scalability Issues

- The current process might struggle with handling large-scale placement activities or growing numbers of students.

## 1.3 Solutions with the Placement Management Website

### 1. Centralized Platform for Notifications

- All placement-related announcements can be posted on the website, ensuring easy access and reducing dependency on emails.
- Students can set up reminders or receive notifications via the website.

### 2. Online Registration with Payment Integration

- A seamless online registration process, integrated with payment gateways, can automate fee collection and instantly confirm registrations.
- Generates receipts and updates the student's dashboard.

### 3. Automated Attendance Tracking

- The system can integrate with training session attendance records to automatically verify eligibility for placement drives.

#### 4. Interactive Communication

- A dedicated FAQ section, a query submission feature ensures quicker issue resolution.

#### 5. Student Dashboard

- Personalized dashboards allow students to view:
  - Placement events they are eligible for.
  - Registration status and payment history.
  - Results and feedback from previous events.
  - Progress in placement activities.

#### 6. Secure Database Management

- The website uses a database to securely manage student data, reducing errors and ensuring data consistency.
- Coordinators can access data in real time without manual updates.

#### 7. Scalable and Flexible System

- The website can handle increasing student data and placement events without performance issues.

#### 8. Enhanced Reporting for the Placement Cell

- Admin features for the placement officer and student coordinators:
  - Insights on student participation and progress.
  - Automated report generation for event performance and student placements.

### 1.4 Document Conventions

This document adheres to the following conventions:

- **Headings:** Bold and numbered for hierarchical clarity (e.g., 1. Introduction).
- **Text Formatting:**
  - **Bold text** highlights key terms or section headers.
  - Bulleted or numbered lists are used for organized information.
- **Functional Requirements:** Identified using tags such as FR-1, FR-2, etc., for unique identification.
- **Terminology:** Acronyms like RIT (Rajiv Gandhi Institute of Technology) and RBAC (Role-Based Access Control) are defined where first used.

### 1.5 Intended Audience and Reading Suggestions

This document is intended for the following audience:

- **Developers:** To understand the functional and nonfunctional requirements for implementation.
- **Project Managers:** To track scope and ensure deliverables align with institutional needs.
- **Testers:** To verify system functionality against specified requirements.
- **Placement Cell Members:** To review whether the system meets operational expectations.
- **Documentation Writers:** To prepare user manuals and help content based on the system's features.

#### Reading Suggestions

- **Overview:** Begin with Sections 1 and 2 to understand the product context and scope.
- **Developers and Testers:** Focus on Sections 3, 4, and 5 for detailed functional and nonfunctional requirements.
- **Project Managers:** Refer to all sections for complete coverage, but prioritize Sections 1, 2, and 8 (if present).

## 1.6 Project Scope

The Placement Management System is a web-based platform designed to replace manual placement processes with an automated, scalable solution.

### Objectives and Goals

- **Simplify Operations:** Enable the Placement Cell to manage placement drives, events, and student applications efficiently.
- **Improve Communication:** Provide automated notifications and updates for students and recruiters.
- **Enhance Transparency:** Allow students to track their placement progress and view results in real-time.
- **Generate Insights:** Offer reports and analytics for better decision-making.

### Benefits

- Saves time and reduces administrative workload for the Placement Cell.
- Enhances the user experience for students and recruiters.
- Ensures data accuracy and availability for all stakeholders.

## 1.7 References

This document refers to the following resources:

- **SRS Format Document:**
  - **Title:** Software Requirements Specification Template
  - **Author:** Unknown
  - **Date:** December 29, 2024
- **Web Development Guidelines:**
  - **Title:** React.js Official Documentation
  - **Source:** <https://reactjs.org/docs>
- **Database Design Guidelines:**
  - **Title:** PostgreSQL Official Documentation
  - **Source:** <https://www.postgresql.org/docs>
- **Institution Policies:** Placement guidelines document provided by Rajiv Gandhi Institute of Technology (RIT).

## 2 Overall Description

### 2.1 Product Perspective

The Placement Management System is a new, self-contained product designed specifically for Rajiv Gandhi Institute of Technology, Pampady, to automate and streamline all placement-related activities managed by the Placement Cell. It aims to replace the current manual processes, such as event organization, student registrations, and communication, with a web-based solution.

This system leverages modern technologies (PERN stack: PostgreSQL, Express.js, React.js, Node.js) to provide a scalable and efficient solution that integrates seamlessly with the existing academic infrastructure. It is designed as a standalone application but can be extended for integration with future institutional systems if required.

## 2.2 Product Features

The Placement Management System offers the following high-level features:

- **User Authentication:** Secure login and role-based access control for students, recruiters, and Placement Cell members.
- **Placement Event Management:** Tools for creating, updating, and managing placement drives and internship opportunities.
- **Student Registration:** A seamless registration process for eligible placement drives or internships.
- **Progress and Result Tracking:** Allow students to track their progress in placement stages and view results.
- **Notifications and Alerts:** Automated notifications for new opportunities, event updates, and results.
- **Reports and Statistics:** Generate comprehensive reports on placement activities, student participation, and recruiter engagement.

## 2.3 User Classes and Characteristics

### 1. Students

- **Role:**
  - Basic technical knowledge of web applications.
  - Primarily use the system to register for events, track results, and receive notifications.
- **Usage Frequency:** Moderate to high during placement seasons.

### 2. Placement Cell Members (Admins)

Consist mainly of Placement officer and 2 student coordinators.

- **Role:**
  - Moderate technical expertise.
  - Use the system for managing placement events, monitoring student progress, and generating reports.
- **Usage Frequency:** High throughout the year, especially during placement seasons.

## 2.4 Operating Environment

The system is designed to operate in the following environment:

### Hardware Requirements

- **Client Devices:**
  - Laptops, desktops, tablets, or smartphones with modern web browsers.
  - Minimum hardware: 4 GB RAM, dual-core processor, 1280x720 resolution display.
- **Server Requirements:**
  - Cloud-hosted server (AWS or equivalent).
  - Minimum hardware: 16 GB RAM, SSD storage, multi-core processors.

### Software Requirements

- **Operating System:**
  - Server: Ubuntu 20.04 LTS or similar Linux-based distributions.
  - Client: Any OS (Windows, macOS, Linux) with browser support.
- **Browsers Supported:** Latest versions of Chrome, Firefox, Safari, and Edge.
- **Protocols and Technologies:**
  - HTTPS for secure communications.
  - RESTful APIs for client-server interaction.

## 2.5 User Documentation

The following user documentation components will be provided with the system:

- **User Manuals:** Comprehensive guides for students, recruiters, and Placement Cell members.
- **Online Help:** Integrated help sections accessible within the application.
- **FAQs:** A collection of frequently asked questions for troubleshooting common issues.

All documentation will be delivered in PDF format and accessible from the system's help menu.

## 2.6 Assumptions and Dependencies

### Assumptions:

- Students will regularly update their profiles with accurate information.
- Recruiters will provide clear eligibility criteria and timely feedback.
- The internet connection at the institution will be stable and reliable.
- Users will have basic familiarity with web-based platforms.

### Dependencies:

- Integration with third-party APIs for email/SMS notifications (e.g., Twilio, SendGrid).
- Cloud hosting providers (e.g., AWS or equivalent) for reliable uptime and performance.
- PostgreSQL as the primary database for secure and efficient data storage.
- The availability of a technical team for maintenance and updates.

# 3 System Features

## 3.1 User Authentication

### 3.1.1 Description and Priority

This feature provides secure login and role-based access for all users, including students, recruiters, and Placement Cell members.

**Priority:** High

**Benefit:** 9

**Penalty:** 8

**Cost:** 5

**Risk:** 4

### 3.1.2 Stimulus/Response Sequences

- **Stimulus:** A user navigates to the login page and enters their credentials.
- **Response:**
  - System validates the credentials against the database.
  - If valid, the user is redirected to their dashboard based on their role.
  - If invalid, an error message is displayed, and the user is prompted to re-enter credentials.



### 3.1.3 Functional Requirements

- FR-1: The system must support user registration and login.
- FR-2: Passwords must be stored securely using hashing algorithms.
- FR-3: Provide role-based dashboards for students, recruiters, and Placement Cell members.
- FR-4: Implement a password recovery mechanism using registered email addresses.
- FR-5: Display clear error messages for invalid login attempts.

## 3.2 Placement Event Management

### 3.2.1 Description and Priority

This feature allows the Placement Cell to create, update, delete, and manage placement drives and internship events.

**Priority:** High

**Benefit:** 9

**Penalty:** 8

**Cost:** 6

**Risk:** 3

### 3.2.2 Stimulus/Response Sequences

- **Stimulus:** A Placement Cell member enters details to create a new placement drive.
- **Response:**
  - The system saves the event and notifies eligible students via email/SMS.
  - If an event is updated, the system updates the database and re-notifies the students.
  - If an event is deleted, it is removed from the system, and a notification is sent to all registered users.

### 3.2.3 Functional Requirements

- FR-6: Enable the Placement Cell to create, update, delete, and manage placement events.
- FR-7: Allow setting eligibility criteria for events.
- FR-8: Notify students automatically about eligible events.
- FR-9: Display all active placement events on the student dashboard.
- FR-10: Provide error handling for invalid inputs (e.g., missing required fields).

## 3.3 Student Registration for Placement Drives

### 3.3.1 Description and Priority

This feature allows students to register for placement drives or internships they are eligible for.

**Priority:** High

**Benefit:** 8

**Penalty:** 7

**Cost:** 5

**Risk:** 4

### 3.3.2 Stimulus/Response Sequences

- **Stimulus:** A student selects a placement event from the list of available opportunities.
- **Response:**
  - The system verifies eligibility and records the student’s registration.
  - If ineligible, the system displays an error message.
  - A confirmation email is sent upon successful registration.

### 3.3.3 Functional Requirements

- FR-11: Display a list of available placement events with eligibility indicators.
- FR-12: Enable eligible students to register for events.
- FR-13: Record and store student registrations in the database.
- FR-14: Send confirmation emails for successful registrations.
- FR-15: Provide error messages for ineligible students attempting to register.

## 3.4 Notifications and Alerts

### 3.4.1 Description and Priority

This feature sends automated notifications to students about placement events, updates, and results.

**Priority:** Medium

**Benefit:** 7

**Penalty:** 5

**Cost:** 4

**Risk:** 3

### 3.4.2 Stimulus/Response Sequences

**Stimulus:** A Placement Cell member updates event details or publishes results.

**Response:**

- The system sends email/SMS notifications to students.
- Updates the dashboard notification section.

### 3.4.3 Functional Requirements

- FR-16: Integrate with email and SMS APIs to deliver notifications.
- FR-17: Display notifications on the dashboard for easy access.
- FR-18: Allow students to view notification history.
- FR-19: Notify students about placement results as soon as they are updated.

## 3.5 Progress and Result Tracking

### 3.5.1 Description and Priority

This feature tracks and displays students’ progress through placement stages and provides final results.

**Priority:** High

**Benefit:** 9

**Penalty:** 7

**Cost:** 5

**Risk:** 3

### 3.5.2 Stimulus/Response Sequences

**Stimulus:** Placement cell officer updates shortlist or result data for a placement drive.

**Response:**

- The system updates the student's profile with the new status.
- Notifies the student.
- Students can view their progress and final result on their dashboard.

### 3.5.3 Functional Requirements

- FR-20: Maintain logs of student progress through placement stages.
- FR-21: Allow Placement Officer to upload shortlists and results.
- FR-22: Display progress and results on the student dashboard.
- FR-23: Notify students when their progress is updated.

## 3.6 Resume Upload and Management

### 3.6.1 Description and Priority

This feature allows students to upload and manage their resumes for placement drives.

**Priority:** High

**Benefit:** 9

**Penalty:** 6

**Cost:** 3

**Risk:** 2

### 3.6.2 Stimulus/Response Sequences

**Stimulus:** A student uploads or updates their resume.

**Response:**

- The system validates the uploaded file format (PDF preferred).
- If valid, the resume is stored and linked to the student's profile.

### 3.6.3 Functional Requirements

- FR-28: Enable students to upload or replace their resumes.
- FR-29: Validate file format and size during upload.
- FR-30: Allow students to download their uploaded resume.
- FR-31: Notify students if their resume is missing or invalid.

## 3.7 Interview Scheduling

### 3.7.1 Description and Priority

This feature manages the scheduling of interviews between recruiters and students.

**Priority:** Medium

**Benefit:** 7

**Penalty:** 6

**Cost:** 4

**Risk:** 4

### 3.7.2 Stimulus/Response Sequences

**Stimulus:** A recruiter provides interview slots and invites shortlisted students.

**Response:**

- The system notifies students of their interview time and location (or link for virtual interviews).
- Students confirm or request a reschedule if allowed.

### 3.7.3 Functional Requirements

- FR-32: Allow Placement Cell to schedule interviews and assign slots to students.
- FR-33: Notify students of scheduled interviews via email/SMS and dashboard updates.
- FR-34: Enable students to confirm or request rescheduling (if supported).
- FR-35: Maintain a centralized calendar view for the Placement Cell.

## 3.8 Placement Statistics and Reports

### 3.8.1 Description and Priority

This feature generates placement statistics and reports for analysis.

**Priority:** Low

**Benefit:** 7

**Penalty:** 5

**Cost:** 3

**Risk:** 2

### 3.8.2 Stimulus/Response Sequences

**Stimulus:** A Placement Cell member requests a placement report for a specific time period.

**Response:**

- The system generates the report and presents it in a downloadable format.

### 3.8.3 Functional Requirements

- FR-36: Generate reports for placement events (e.g., participation, offers).
- FR-37: Provide visualization tools (charts, graphs) for easier interpretation.
- FR-38: Export reports as PDF or Excel files.
- FR-39: Include filters for generating custom reports (e.g., department-wise).

## 4 External Interface Requirements

### 4.1 User Interfaces

The Placement Management System will feature intuitive and user-friendly interfaces to cater to different user roles (students, recruiters, Placement Cell members).

#### Logical Characteristics

##### Web Interface:

- Developed using React.js for dynamic and responsive design.
- Consistent UI/UX styling following Material Design guidelines.
- Clear navigation menus for role-specific features.

- Screen layout includes:
  - **Student Dashboard:** List of placement events, notifications, progress tracking.
  - **Placement Cell Dashboard:** Event management tools, student performance data, and reports.
- **Error Messages:** Displayed prominently in red with meaningful messages (e.g., "Invalid credentials. Please try again.").

#### **GUI Standards:**

- Consistent fonts, button styles, and colors throughout the application.
- Mobile-first design for accessibility across devices.
- Drop-down menus, modals, and forms for easy data input.

#### **Sample Components Needing a User Interface:**

- Login Page
- Event Registration Form
- Notifications Center
- Placement Results Display

Detailed designs will be documented in a separate User Interface Specification Document.

## **4.2 Hardware Interfaces**

The system interacts with standard client and server hardware components:

#### **Logical Characteristics**

##### **Client Devices:**

- Compatible with desktops, laptops, tablets, and smartphones with modern web browsers.
- Minimum screen resolution: 1280x720 (responsive design supports smaller resolutions).

##### **Server:**

- Hosted on cloud infrastructure (AWS or equivalent).
- Supports high availability and load balancing during placement drives.

#### **Physical Characteristics**

##### **Client Requirements:**

- Minimum 4 GB RAM and a 2 GHz processor for optimal performance.
- Internet connection: 1 Mbps or higher.

##### **Server Requirements:**

- 16 GB RAM, SSD storage, and multi-core processors.
- Communication protocols: HTTPS for secure data transfer.

## **4.3 Software Interfaces**

The system integrates with several software components for efficient operation:

**Databases:**

- PostgreSQL (version 13 or higher) for storing and managing user data, event details, and results.
- Data sharing mechanisms include SQL queries and API calls for seamless integration.

**Operating Systems:**

- Server: Ubuntu 20.04 or similar Linux-based distributions.
- Client: Compatible with Windows, macOS, and Linux.

**Tools and Libraries:**

- React.js (frontend), Express.js (backend), Node.js runtime environment.
- Authentication: JSON Web Tokens (JWT) for secure sessions.

**APIs:**

- Email/SMS: Twilio or SendGrid for notifications.
- Cloud Storage: AWS S3 for resume and document uploads.

**Messages and Data Exchange:**

- Input Data: User credentials, placement registration data, event updates.
- Output Data: Notifications, student results, and reports.
- Communication Protocols: RESTful APIs for interaction between frontend and backend.

## 4.4 Communications Interfaces

The system requires communication functionalities to support notifications, browser access, and server interaction:

**Web Communication:**

- **Protocols:** HTTPS for secure browser-server communication, WebSocket (optional) for real-time updates.
- **Browser Compatibility:** Supports latest versions of Chrome, Firefox, Safari, and Edge.

**Notifications:**

- Email and SMS alerts for event updates and results.
- **Message Formatting:**
  - Emails: HTML-based templates for readability.
  - SMS: Text-only notifications with concise information.

**Security:**

- Encrypted communication using TLS/SSL.
- Secure storage of sensitive data (e.g., credentials, results).

**Data Transfer and Synchronization:**

- Asynchronous communication to reduce user wait times.
- JSON used as the standard format for API responses.

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

The Placement Management System must perform efficiently under varying conditions to ensure a seamless user experience.

**System Throughput:**

- The system must support at least 500 concurrent users during peak periods, such as placement drives.

**Response Time:**

- Average response time for key operations (e.g., login, event registration) must be under 2 seconds.
- Notifications should be sent within 30 seconds of an event update.

**Data Retrieval:**

- Query execution for reports and dashboards should take less than 3 seconds, even for large datasets.

**Scalability:**

- The system must scale horizontally to support future increases in users and events, accommodating up to 2000 concurrent users.

**Rationale:** Efficient performance is critical to ensure user satisfaction and avoid disruptions during time-sensitive placement activities.

### 5.2 Safety Requirements

The system must safeguard against potential harm caused by misuse, errors, or malicious activities.

**Data Integrity:**

- Prevent data loss during server crashes by implementing automatic backups every 24 hours.
- Transactions, such as registrations or result uploads, must be atomic to avoid partial updates.

**Error Prevention:**

- Input validation must prevent invalid or harmful data (e.g., malicious scripts in form fields).

**Compliance:**

- The system must comply with relevant IT and data safety standards (e.g., GDPR for data privacy, if applicable).

**Rationale:** Ensuring data integrity and safety protects users and the institution from potential legal and operational issues.

### 5.3 Security Requirements

The system must ensure the confidentiality, integrity, and availability of user data.

**Authentication and Authorization:**

- Use role-based access control (RBAC) to restrict access to specific functionalities based on user roles.
- Implement multi-factor authentication (MFA) for Placement Cell members.

**Data Encryption:**

- Encrypt sensitive data (e.g., passwords, results, personal information) both in transit (TLS/SSL) and at rest.

**Auditing and Monitoring:**

- Maintain logs of user activities (e.g., logins, data changes) for 90 days for audit purposes.
- Detect and respond to unauthorized access attempts in real-time.

**Secure Communication:**

- All interactions between the client and server must use HTTPS.

**Compliance:**

- Adhere to FERPA or equivalent student privacy regulations if applicable.

**Rationale:** Maintaining a secure environment is essential to protect sensitive data and ensure user trust.

## 5.4 Software Quality Attributes

- **Adaptability:** The system should support updates to add new features or modify existing ones without disrupting current operations.
- **Availability:** The system must ensure 99% uptime during placement seasons, with scheduled maintenance during off-peak hours.
- **Correctness:** Data displayed on dashboards (e.g., student progress, placement statistics) must always reflect the latest updates.
- **Flexibility:** The system should allow customization of event settings (e.g., eligibility criteria, job descriptions) without requiring code changes.
- **Interoperability:** The system should integrate seamlessly with third-party APIs for notifications, file storage, and authentication.
- **Maintainability:** Modular architecture must allow developers to make updates or fix bugs within 3 business days.
- **Portability:** The application should run on any modern web browser (e.g., Chrome, Firefox, Safari, Edge) without requiring plugins.
- **Reliability:** Data consistency must be ensured even during unexpected server failures, with automated recovery mechanisms in place.
- **Reusability:** Components like the notification system and authentication module must be reusable for other college applications.
- **Robustness:** The system should gracefully handle unexpected errors and notify users with meaningful error messages (e.g., "Service unavailable. Please try again later.").
- **Testability:** Each feature must be testable independently, with automated tests covering 80% of the codebase.
- **Usability:** The interface should be intuitive, requiring no more than 5 minutes of exploration for a first-time user to understand the dashboard.



## 6 Other Requirements

### Database Requirements:

- Store user data, event details, and logs in a relational database (PostgreSQL).
- The database schema must support normalization up to 3NF to prevent data redundancy.

### Legal Requirements:

- Ensure compliance with data protection laws applicable to student and recruiter information.

### Reuse Objectives:

- Develop the system's modules (e.g., authentication, notifications) with reusable code for future institutional applications.

## A Appendix A: Glossary

**API** Application Programming Interface: A set of routines, protocols, and tools for building software applications.

**GUI** Graphical User Interface: A user interface that includes graphical elements like windows, icons, and buttons.

**JWT** JSON Web Token: A compact, URL-safe method for representing claims to be transferred between two parties.

**PERN Stack** A technology stack consisting of PostgreSQL (database), Express.js (backend), React.js (frontend), and Node.js (runtime environment).

**Placement Cell** The administrative body responsible for organizing and managing placement drives and internship opportunities in an academic institution.

**RIT** Rajiv Gandhi Institute of Technology, Pampady: The institution for which the Placement Management System is being developed.

**RBAC** Role-Based Access Control: A security mechanism that restricts system access based on user roles.

**SRS** Software Requirements Specification: A document that outlines the functionalities and constraints of a software product.

**TBD** To Be Decided/Determined: Indicates pending decisions or information that needs to be clarified.

## B Appendix B: Usecase Diagram

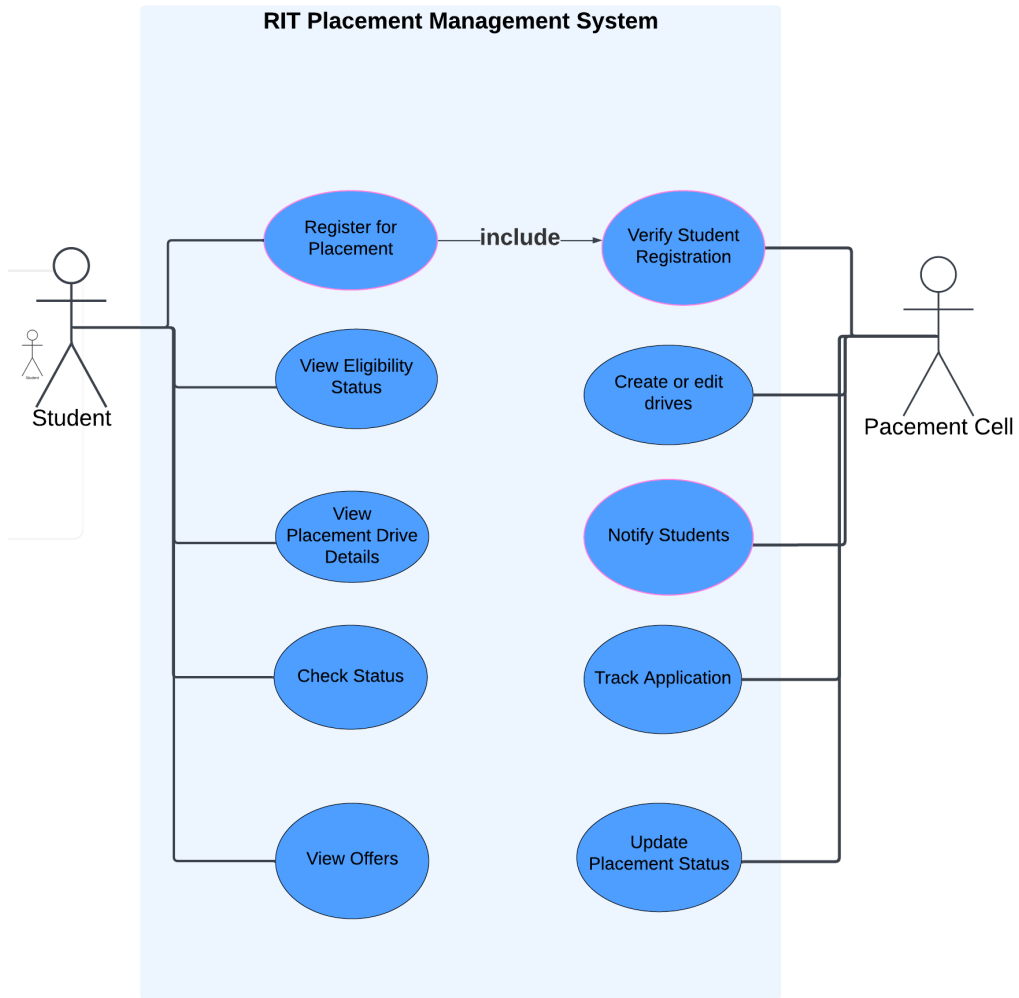


Figure 1: Usecase Diagram for the Placement Management System

## C Appendix C: Issues List

1. **Integration with Third-Party APIs:** The final decision on whether to integrate with specific third-party APIs (e.g., for notifications or analytics) is still pending.
2. **Recruiter Access Scope:** The exact level of access and permissions for recruiters within the system (e.g., data visibility) has not been finalized.
3. **Data Backup Frequency:** The system's backup schedule (e.g., daily or weekly) needs to be determined based on institutional policies.
4. **Compliance Requirements:** Detailed compliance requirements for data privacy laws like GDPR or FERPA need to be reviewed and finalized.
5. **Mobile Optimization:** Whether to prioritize mobile optimization during the initial release or in subsequent updates is to be decided.
6. **Customization Features for Placement Cell:** The extent of event customization options (e.g., eligibility criteria, interview formats) is under discussion.
7. **Performance Benchmarks:** Specific metrics for scalability and performance (e.g., number of concurrent users) need further refinement.
8. **Finalization of Reporting Formats:** The exact format and visualization tools for placement statistics reports are yet to be agreed upon.