Software Requirements Specification

for

WEBSITE TO ASSIGN PROJECTS FOR FINAL YEAR STUDENTS.

Version <1.0>

Prepared by

N.MANOJ KUMAR	**SE22UCSE178**	**Se22ucse178@mahindrauniversity.edu.in**
A.SAIPRIYATHAM	**SE22UCSE024**	**Se22ucse024@mahindrauniversity.edu.in**
G.DEEKSHITHA	**SE22UCSE097**	**Se22ucse097@mahindrauniversity.edu.in**
G.SNEHA REDDY	**SE22UCSE095**	**Se22ucse095@mahindrauniversity.edu.in**
P.YESHWANTH	**SE22UCSE208**	**Se22ucse208@mahindrauniversity.edu.in**
Shaik Naveed Ahmed	SE22UCSE252	**Se22ucse252@mahindrauniversity.edu.in**

Instructor: *D.VIJAY RAO*

Course: **Software Engineering(CS-3201)**

Teaching Assistant: *A.Sravanthi*

Date: **12-03-25**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	N.MANOJ KUMAR A.SAIPRIYATHAM G.DEEKSHITHA G.SNEHA REDDY P.YESHWANTH SHAIK NAVEED AHMED	Initial draft of SRS for Library Management System	12/03/25

1 Introduction

1.1 Purpose of this Document

This document explains the requirements for the **Finalyear Project Website**, a platform that helps final-year students at **Mahindra University** apply for projects posted by faculty members. It describes what the website will do, how it will work, and the technical details needed to build it.

1.2 Product Scope

The website will allow faculty members to post projects and students can browse and apply for them. It will include features like:

- User login and authentication.
- A system for posting and managing projects.
- A way for students to apply for projects and track their application status.
- Notifications and messaging between students and faculty.
- An admin panel for monitoring system activities.

This system aims to **reduce manual work** and **ensure a smooth, transparent process** for project allocation.

1.3 Intended Audience and Document Overview

This document is for:

- **Developers**: To understand how to build and implement the system.
- **University Administration**: To monitor and manage project assignments.
- **Students and Faculty**: To understand how they will use the platform.

This document is structured as follows:

1. **Introduction** - Overview of the website and its purpose.
2. **Website Description** - Details about features, design constraints, and assumptions.
3. **Detailed Requirements** - Functional and non-functional requirements.
4. **Technical & Security Requirements** - Performance and security details.

5. **Technology Stack** - The tools and frameworks used.
6. **Future Enhancements** - Possible improvements in the future.

1.4 Definitions, Acronyms and Abbreviations

- **Admin**: Administrator responsible for managing the platform.
- **API**: Application Programming Interface, used for system integration.
- **DBMS**: Database Management System, used for storing and retrieving data.
- **GUI**: Graphical User Interface, the visual part of the website users interact with.
- **HTTPS**: HyperText Transfer Protocol Secure, ensures secure communication.
- **SRS**: Software Requirements Specification, this document.
- **UI**: User Interface, how the website looks and interacts with users.
- **UX**: User Experience, overall experience and ease of use of the website.

1.5 Document Conventions

- This document follows IEEE formatting standards.

- Requirements labeled FR refer to Functional Requirements, while NFR refer to Non-functional Requirements.

- All headings are in plain text, and standard font sizes (11 or 12 pt) are used throughout.

1.5 References and Acknowledgments

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications.

- College Library Policy Documents (if applicable).

- Acknowledgments to Instructor, Teaching Assistant, and any external sources consulted during requirements gathering

2 Overall Description

2.1 Product Overview

The **Project Assignment Website** is a **new, stand-alone system** designed to replace the manual way of assigning projects to students. It provides a **simple and organized online platform** where faculty members can post projects, and students can find and apply for them. This makes the process **faster and more efficient**.

The website does not replace any existing system but can be connected to university login services for secure access.

The system has **three types of users**:

- **Students** – Can browse and apply for projects.
- **Faculty Members** – Can post and manage projects.
- **Administrators** – Can oversee platform activities and manage users.

Below is a simple explanation of how the system works:

1. **User Login** – Users log in using their university email.
2. **Project Listings** – Faculty create projects, and students browse them.
3. **Application Process** – Students apply, and faculty review applications.
4. **Notifications** – Users get updates about project status.
5. **Admin Control** – Admins manage users and monitor activities.

This system **helps students find projects easily** and **helps faculty manage applications smoothly**.

2.2 Product Functionality

The website will provide the following major functions:

- **User Registration and Login**
  - Students and faculty can create accounts and log in securely.
  - Only university-verified users can access the system.
- **Project Management**
  - Faculty can add, edit, and delete project listings.
  - Each project will have details like title, description, required skills, and deadlines.
- **Project Browsing and Search**
  - Students can view a list of available projects.
  - Projects can be filtered by faculty, category, and deadline.
- **Notifications and Messaging**

  - Students and faculty receive email and in-app notifications about project status.
  - A messaging system allows communication between faculty and students.
- **Admin Controls**
  - Admins can manage users, projects, and system settings.
  - They can generate reports on student applications and project assignments.

This system **ensures a smooth and structured way for students to apply for projects and for faculty to manage them efficiently**.

2.3 Design and Implementation Constraints

The system has certain limitations and requirements that must be followed:

- **Hardware Limitations**
  - The website should run on any standard desktop, laptop, tablet, or smartphone with an internet connection.
  - The system should handle at least **500 users at the same time** without slowing down.
- **Software Constraints**
  - The system design must follow **UML (Unified Modeling Language) standards** for diagrams and documentation.
- **Technology Constraints**
  - The backend must be built using **Express.js**.
  - The frontend should be developed using **HTML,CSS,JAVASCRIPT**.
  - The database should be **MySQL or MongoDB**.
  - The system must be hosted on **AWS, Google Cloud, or DigitalOcean**.
- **Security Constraints**
  - The website must use **HTTPS** for secure data transfer.

  - User authentication must be done using **university email-based login**.
  - All sensitive data should be **encrypted** to prevent unauthorized access.

- **Design Standards**
  - The website must follow **responsive design principles** to work on all screen sizes.
  - The interface should be **user-friendly and accessible** to all users.

These constraints ensure that the system is **secure, efficient, and scalable** for future use.

2.4 Assumptions and Dependencies

The project has the following assumptions and dependencies:

- **Users will have internet access** – The website is online, so students and faculty need a stable internet connection.
- **University authentication system will be available** – The system relies on university email-based login.
- **Supported web browsers** – The website will work on modern browsers like Chrome, Firefox, and Edge.
- **Hosting on cloud services** – The platform will be deployed on **AWS, Google Cloud, or DigitalOcean**.
- **Database availability** – The system depends on **MySQL** for data storage.
- **Notifications** – The notifications will be displayed in website itself.

If any of these assumptions change, the system may need modifications to function properly.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 Main Features of the User Interface:

- **Login Page:**
  - Users will enter their university email and password to log in.
  - Forgot password option for password recovery.
- **Dashboard:**
  - Students: View available projects, apply, and track application status.
  - Faculty: Create, edit, and manage projects.
  - Admins: Manage users, monitor platform activity, and generate reports.
- **Project Listings:**
  - A list of available projects with search and filter options.
  - Clicking on a project shows detailed information, including faculty details, requirements, and application deadline.
- **Application System:**
  - Students can apply for projects and upload ‘CGPA’.
  - Faculty can review cgpa and approve/reject them.
- **Notifications & Messaging:**
  - Users will receive notifications about application status and new projects if and only if you open the website.

  - A built-in messaging system for communication between faculty and students.

- **Admin Panel:**
  - Admins can view system reports, manage users, and monitor system activity.

User Interaction:

- **Navigation:** The website will have a top navigation bar with links to Dashboard, Projects, Applications, and Notifications.
- **Forms & Buttons:** Users will interact with forms for logging in, applying for projects, and managing project details.
- **Responsive Design:** The website will be fully responsive and accessible on desktops, tablets, and mobile devices.

This user-friendly design ensures that students can easily find and apply for projects while faculty can manage applications efficiently.

3.1.2 Hardware Interfaces

The **Project Assignment Website** will be accessible from various devices. The following hardware interfaces will be supported:

- **Desktop Computers & Laptops**
  - The system will work on Windows, macOS, and Linux-based computers.
  - Requires a modern web browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari.
- **Tablets & Mobile Devices**
  - The website will be optimized for Android and iOS devices.
  - Users can access the system through mobile web browsers.
- **Server Hardware**
  - The website will be hosted on cloud-based servers (AWS, Google Cloud, or DigitalOcean) to ensure high availability.
  - Requires a database server (MySQL, PostgreSQL, or MongoDB) to store project and user data securely.
- **Networking Equipment**
  - The system relies on internet connectivity to function.

  - Users must have access to Wi-Fi or mobile data to use the platform.

This hardware setup ensures that users can access the platform from any device with an internet connection, providing flexibility and ease of use.

3.1.3 Software Interfaces

- **Web Browsers**
  - The website will be accessible via modern browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- **Database Management System (DBMS)**
  - The system will use **MySQL, PostgreSQL, or MongoDB** to store project details, user accounts, and application data.
- **University Authentication System**
  - The website will integrate with the university's authentication system to verify student and faculty credentials.

❑ **Website Notification System**

- Notifications about project updates and application status will be displayed directly on the website.
- Users will see alerts on their dashboard without requiring third-party email services.
- **Cloud Hosting Services**
  - The website will be hosted on **AWS, Google Cloud, or DigitalOcean** to ensure high availability and scalability.

These software interfaces ensure that the system functions efficiently and provides a seamless experience for users.

3.2 Functional Requirements

Functional requirements define how the system should operate and the key features it must provide.

F1: User Authentication

- The system shall allow students, faculty, and admins to log in using their university email credentials.

- The system shall provide a password recovery option.

F2: Project Management

- Faculty shall be able to create, edit, and delete project listings.
- Each project shall include details such as title, description, required skills, and deadline.

F3: Student Applications

- Students shall be able to browse and search for projects.
- Students shall be able to apply for projects by submitting an application form and uploading necessary documents.

F4: Faculty Review and Selection

- Faculty shall be able to view student cgpa's for their projects.
- Faculty shall be able to accept or reject based on their cgpa's.
- Students shall receive notifications about the status of their projects update.

F5: Notifications and Messaging

- The system shall provide in-website notifications for project updates .
- The system shall allow faculty and students to communicate through a built-in messaging feature in website itself.

F6: Admin Controls

- Admins shall be able to manage user accounts.
- Admins shall be able to monitor system activity and generate reports.

These functional requirements ensure that the **Project Assignment Website** operates smoothly and efficiently for students, faculty, and administrators.

…

3.3 Use Case Model

Use Case #1: Post Project (U1)

Author: Development Team
Purpose: To allow faculty members to create and publish new project listings for students to apply.

Requirements Traceability: Links to project management requirements (F2, F4).
Priority: High - Ensures that students have access to available projects.

Preconditions:

- The faculty member must be logged into the system.
- The faculty member must have the required permissions to post a project.

Postconditions:

- The project is published and visible to students.
- The faculty member can edit or delete the project later.

Actors:

- Faculty

Basic Flow:

1. The faculty logs into their dashboard.
2. The faculty clicks on **Post Project** in the project management section.
3. The system prompts the faculty to enter project details (title, description, requirements, deadline).
4. The faculty submits the project, and the system saves it in the database.
5. The project is now visible in the project listings for students.

Alternative Flow:

- If the faculty does not enter all required fields, the system will prompt them to complete the missing information before submission.

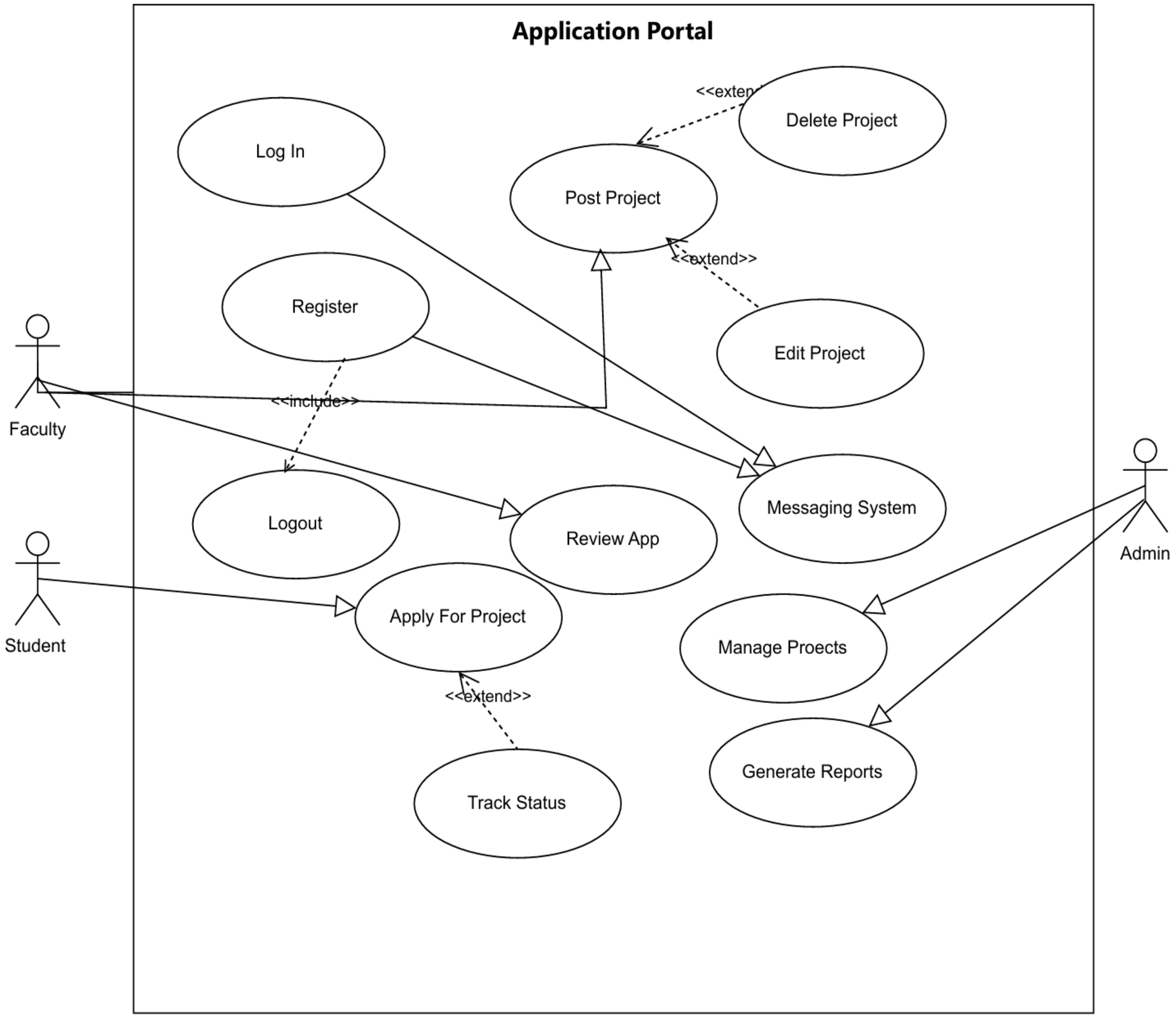
Exceptions:

- If there is a system error while saving the project, the faculty will be notified to try again later.

Includes: Notification system module (to notify students about newly posted projects).

Notes/Issues:

- In a future update, faculty may be able to upload supporting documents for projects.


Application Portal
<<extend>>
Delete Project
Log In
Post Project
<<extend>>
Register
Edit Project
Faculty
<<include>>
Messaging System
Logout
Review App
Admin
Student
Apply For Project
Manage Proects
<<extend>>
Generate Reports
Track Status

4 Other Non-functional Requirements

4.1 Performance Requirements

o ensure a seamless experience for both students and faculty members, the system has been designed with the following performance standards:

- **P1: System Response Time**
  - The website will load within **3 seconds** under normal network conditions.
  - The project listing page will display results within **2 seconds** after a search query to allow for quick access to available projects.
- **P2: Concurrent Users**
  - The system will support **at least 500 users** accessing the platform at the same time without slowdowns, ensuring smooth navigation for all users.
- **P3: Database Performance**
  - Queries for project listings and application statuses will execute within **1 second**, ensuring that faculty can efficiently manage project details.
  - The system is built to handle at least **1000 active project listings** efficiently without delays.
- **P4: Notification Updates**
  - Faculty and students will receive notifications on their dashboard within **2 seconds** of an update, ensuring timely communication.
- **P5: Application Processing**
  - Faculty will be able to view and process student applications within **3 seconds** of clicking on an application, allowing for quick decision-making.
- **P6: Uptime and Availability**
  - The system is designed to have **99.5% uptime**, ensuring that it remains accessible to faculty and students at all times.

These performance parameters have been set to provide an efficient and user-friendly platform for managing project assignments. Your feedback and suggestions for further improvements are always welcome.

4.2 Safety and Security Requirements

- **S1: User Authentication and Access Control**
    - All users (students, faculty, and admins) must log in using their **university email credentials**.
    - Role-based access ensures that students, faculty, and admins only have access to relevant system functionalities.
- **S2: Data Encryption and Protection**
    - All sensitive data, including project details and user information, will be **encrypted** to prevent unauthorized access.
    - The system will use **SSL (Secure Sockets Layer)** encryption to ensure safe data transmission.
- **S3: Secure Application Process**
    - Student applications will be protected from unauthorized modifications.
    - Only faculty members assigned to a project will be able to view and process applications.
-

4.3 Software Quality Attributes

o ensure that the **Project Assignment Website** provides a high-quality experience, the system has been designed with the following software quality attributes:

4.3.1 Reliability

- The system will have **99.5% uptime**, ensuring that faculty and students can access it at any time.
- Automatic **error detection and logging** will be implemented to quickly identify and resolve issues.

4.3.2 Usability

- The interface will be **simple and intuitive**, allowing users to navigate easily without requiring training.
- The platform will be **mobile-friendly**, ensuring accessibility across desktops, tablets, and smartphones.

4.3.3 Maintainability

- The system will be built using **modular code**, making it easy to update and add new features in the future.
- Regular software updates will be scheduled to improve performance and security.

4.3.4 Security and Privacy

- All user data will be **encrypted** and securely stored to protect against unauthorized access.
- Strict **role-based access controls** will be enforced to ensure that only authorized users can access specific system features.

4.3.1.1 4.3.5 Scalability

- The platform will be able to **support future growth**, allowing more faculty and students to use it as demand increases.
- The system will be designed to **handle increased project postings and user activity** without performance issues.

5 Other Requirements

❑ **Internationalization**

- The system will initially be developed in **English** but can be adapted to support multiple languages if required in the future.

❑ **Legal and Compliance Requirements**

- The system will comply with **university data privacy policies** to protect user information.

Appendix A – Data Dictionary

Name	Type	Description	Related Operations
userRole	String	Defines whether the user is a Student, Faculty, or Admin	AssignRole, ViewDashboard
email	String	University email address for login and communication	RegisterUser, Login, Notify
password	String	Encrypted password for secure login	Login, ChangePassword

projectID	Integer	Unique identifier for each project	CreateProject, EditProject
projectTitle	String	Title of the project visible to students	CreateProject, ViewProject
projectStatus	String	Status of the project (Open, Closed)	UpdateProject, CloseProject
applicationID	Integer	Unique ID assigned to each application submitted	SubmitApplication, ReviewApplication
applicationStatus	String	Indicates whether an application is Pending, Accepted, or Rejected	UpdateStatus, NotifyStudent
messageID	Integer	Unique ID for messages exchanged between users	SendMessage, ReadMessage
notificationID	Integer	Unique ID for notifications generated in the system	GenerateNotification, ViewNotification

Appendix B - Group Log

Date	*Attendees*	*Topics Discussed*	*Action Items*
2025-02-20	*g.Deekshitha, G.Sneha*	*Project outline, initial requirements*	*Each member to gather required data*
2025-03-04	*A.Saipriyatham, N.Manoj*	*Draft functional specifications, use cases*	*Draft initial SRS sections*
2025-03-12	*P.Yeshwanth, Shaik Naveed Ahmed*	*SRS review, finalizing details*	*Finalize SRS, prepare for submission*