

Student Details

FN 5

Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this project is to develop a **Student Details Management System** that maintains a database of students, recording their extracurricular activities. The system will facilitate easy access to student records, assist in accreditation processes, and provide features such as event tracking, announcements, and placement records.

1.2 Scope

The system will allow students to record their participation in events (which include sports, arts, drama, music, dance), internships, professional societies as well as the programs held outside of the college for the students. It will provide administrative features such as importing data, managing announcements, and maintaining extracurricular activities. Security measures using **Cloudflare** will be implemented to prevent cyber threats.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS** - Software Requirements Specification
- **DDoS** - Distributed Denial of Service
- **API** - Application Programming Interface
- **UI/UX** - User Interface/User Experience

1.4 References

No references required.

1.5 Overview

This document describes the functional, non-functional, and external requirements of the system.

2. General Description

2.1 Product Perspective

This system will function as a **centralized platform** for student record management, replacing manual or fragmented databases. It will **integrate with existing university databases using the migrations of the database that can be run on our system or a script which can be used directly to fetch all the data from the database directly.**

2.2 Product Functions

- **Student Activity Management:** Store records of technical events like tathva, cultural events like ragam, club activities, competitions that happen in these events and we maintain the achievements of the students which later helps in the data collection in the later stages.

- **Placement Tracking:** Maintain details of placements of an individual student as well as the total students placed or who got internships, and allow offer letter uploads by the students which will be cross verified by the faculty advisor whenever the letter reaches him/her when the upload happens and notifications reaches.
- **Publication Integration:** Suggest paper titles from faculty publication databases.
- **Announcements:** Notify students of important updates. A separate announcements page will be maintained in the website by the admins which shows all the users the most important announcements regarding the upcoming events.
- **Security Measures:** Implement **Cloudflare Workers** for DDoS protection, rate limiting on the requests sent by the user, two factor authentication (may or may not be included) , encrypted sensitive data (like passwords) in the database.

2.3 User Characteristics

- **Students:** Can add/edit their activities, upload offer letters, and receive announcements.
- **Faculty:** Can manage student records.
- **Admin:** Post announcements, manage website.

2.4 Constraints

- Good Internet connectivity is required for real-time updates and fast internet is needed for operations like offer letter uploads.
- File upload size restrictions for documents (e.g., max 5MB per file).
- Compliance with **university data policies** will be checked for any data added for the user or any file uploaded by the respective faculties.

2.5 Assumptions and Dependencies

- The system will be hosted as follows => AWS for the code logic and Object stores like S3 for the file uploads.
- Integration with **Cloudflare Workers for security.**

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Student Module

- Students should be able to **log in** securely via google auth or normal email-password login.
- Students can **add, edit and view their details** of their participation in events which can be internal as well as external, societies in the college, all the club activities the student has been a part of and placements information.
- Option to **upload offer letters if the student secures a job or an internship which will then be validated by the faculty advisor as well as any achievement of the student in other domains.**

- Search and select publications from **faculty research database**.

3.1.2 Faculty Module

- Faculty can **manage student records** and verify submitted data which include the offer letters as well as the certificates of achievements or participation by the student.

3.1.3 Admin Module

- Ability to **import student data** from external databases (CSV or script through which the migrations can be imported and then another script to import the entire data).
- **Post announcements** that students can view on their dashboard.
- **They also maintain the website for its proper functioning.**

3.1.5 Security Features

- Implement **Cloudflare** for **DDoS protection**, rate limiting and **firewall rules**.
- **Multi-factor authentication (MFA)** for admin and student accounts.
- **Role-based access control (RBAC)** to restrict unauthorized access of the website. The admin has the role of maintaining the website as well as uploading any announcements on the website for the upcoming events.

3.1.6 Event Calender

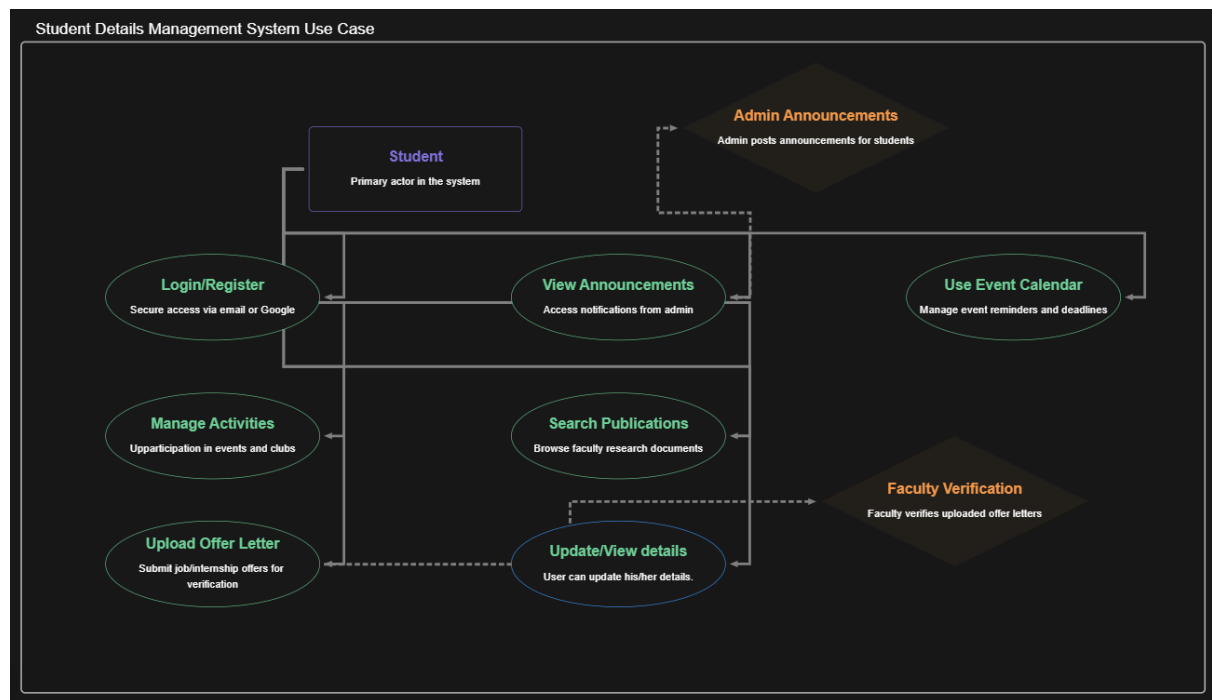
- Integrated **calendar** for students and admins to add events and deadlines.
- Option to **send reminders** and notifications for upcoming events.

3.2 Non-Functional Requirements

- **Performance:** System should load pages within **2 seconds**.
- **Scalability:** Support up to **10,000 student records**.
- **Availability:** 99.9% uptime guarantee.
- **Usability:** Responsive UI for desktop and mobile compatibility.

3.3 External Interface Requirements

- **User Interface:** Developed using **React.js** for frontend and **Java/Springboot** for backend. AWS for deployment. Object stores like S3 to stores files uploaded by the student.
- **Database:** Uses **MySQL** for storing student records. May use Prisma and other ORMs to simplify the complexities of the SQL queries.
- **Security Integration:** Uses **Cloudflare API** for security and analytics.



End of the document. Thank you.