Software Requirements Specification

for

CAMS

(Campus Activity Management System)

Prepared by

Gabriel M S - NIE22CS023
Joseph Mohan - NIE22CS029
Akshaykumar V S – LNIE22CS054

Nirmala College Of Engineering, Technology & Management

January 2025

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Campus Activity Management System. The system is designed to streamline the management and participation process of campus activities, ensuring efficient communication between faculty and students.

1.2 Scope

The Campus Activity Management System is a web-based application built using Python with the Django framework and MySQL as the database.

The system aims to:

- Allow faculty to publish information about various activities, such as sports, arts, and other extracurricular events.
- Enable students to view, apply for participation, and receive notifications and reminders for events.
- Provide participation certificates generated by the system, verified by faculty.
- Generate activity reports at the end of each semester.
- Track activity points for students, as mandated by APJ Abdul Kalam Technical University.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification

- Django: A high-level Python web framework
- MySQL: A relational database management system
- APJAKTU: APJ Abdul Kalam Technical University
- Faculty: Teaching staff responsible for publishing and managing activity information
- Students: Users who participate in campus activities

1.4 References

- APJ Abdul Kalam Technical University guidelines for activity points.
- IEEE Standard for Software Requirements Specifications (IEEE 830-1998).

2. Overall Description

2.1 Product Perspective

The Campus Activity Management System is a standalone application, designed to replace manual processes for managing campus activities. The system integrates with existing university infrastructure and operates on modern web technologies.

2.2 Product Functions

The system offers the following functionalities:

2.2.1 Faculty Side:

- Publish activity information (e.g., sports, arts, and other events).
- Verify participation certificates.
- Generate activity reports at the end of each semester.

2.2.2 Student Side:

- View activity details.
- Apply for participation in activities.
- Receive notifications and reminders 1 day prior to the event.
- Download participation certificates.

2.2.3 Administrative Features:

- Maintain a database of activities, students, and points.
- Track and calculate activity points for students.

2.3 User Classes and Characteristics

2.3.1 Faculty:

- Moderate technical proficiency.
- Responsible for creating and verifying activity data.

2.3.2 Students:

- Basic technical proficiency.
- End-users who interact with the system to view and participate in activities.

2.3.3 Administrators:

- High technical proficiency.
- Manage and maintain the system’s backend and database.

2.4 Operating Environment

- Frontend: Web-based, accessible via modern web browsers (e.g., Chrome, Firefox).
- Backend: Python with Django framework.
- Database: MySQL.
- Platform: Hosted on a Linux-based server.

2.5 Design and Implementation Constraints

- The system must adhere to APJAKTU's guidelines for activity points.
- User data must be securely stored and transmitted using encryption protocols (e.g., HTTPS).
- The system should support scalability to handle a large number of users.

2.6 Assumptions and Dependencies

- Users have access to the internet and a web browser.
- Faculty and students have university-provided credentials for authentication.
- The university will provide the necessary resources for hosting and maintenance.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication:

- Students and faculty must log in using provided credentials.

3.1.2 Activity Management (Faculty Side):

- Faculty can create, update, and delete activity details.
- Faculty can view participation lists.

3.1.3 Student Participation:

- Students can view available activities and apply for participation.

3.1.4 Notifications and Reminders:

- Students receive notifications about event updates and reminders 1 day prior to the event.

3.1.5 Certificate Generation and Verification:

- The system generates participation certificates.
- Faculty verify the certificates.

3.1.6 Activity Points Tracking:

- The system calculates and tracks activity points for each student.

3.1.7 End-of-Semester Reporting:

- Generate a comprehensive report summarizing student activities and points.

3.2 Non-Functional Requirements

3.2.1 Performance:

- The system should support up to 500 concurrent users.

3.2.2 Usability:

- The interface should be intuitive and user-friendly.

3.2.3 Security:

- Implement secure authentication mechanisms.
- Protect sensitive user data with encryption.

3.2.4 Availability:

- The system should have 99.9% uptime.

3.3 Interface Requirements

3.3.1 User Interface:

- Dashboard for faculty and students.
- Mobile-friendly design.

4. Appendices

4.1 Appendix A: Glossary

4.1.1Activity Points:

- Points awarded to students for participation in extracurricular activities.

4.1.2 Participation Certificate

- A document generated by the system to certify student involvement in activities.

4.2 Appendix B: References

- **APJAKTU** Guidelines: https://aktu.ac.in/guidelines
- Django Documentation: https://docs.djangoproject.com/
- MySQL Documentation: https://dev.mysql.com/doc/