SOFTWARE ENGINEERING CONCEPTS

Assignment 1

SOTWARE REQUIREMENTS SPECIFICATIONS

TITLE:

UNIVERSITY EVENTS MANAGEMENT SYSTEM

Submitted By:

Haram Najam Khan

Registration Number:

FA22-BCE-041

Section:

BCE-6B

Submitted To:

Dr. Ghulam Mujtaba

Date of Submission:

29/04/2025

Department of Electrical and Computer Engineering, COMSATS University Islamabad, Abbottabad Campus

Project Title: University Event Management Web Application

1. Project Selection

The University Event Management web application will specifically serve universities, allowing departments, student societies, and administration to manage academic and extra-curricular events efficiently. It will handle event creation, participant registration, event announcements, and real-time updates.

2. Stake Holders

Stake-holder	Responsibility
Students	Primary users who browse, register, and participate in university events.
Student Societies and Clubs	Create and manage event listings (like seminars, workshops, competitions).
Faculty Advisors	Supervise student societies' activities and approve academic events.
Department Coordinators	Manage department-level events and oversee registration and participation.
University Administration	Approve or reject events, monitor overall system activities, manage official university events.
Event Organizers	Design event details, manage registrations, and communicate with participants.
IT Support Team	Maintain, update and troubleshoot technical issues with the system
Developers	Build and maintain the web application.

3. Requirements Gathering

To collect the requirements for the University Event Management Web Application, several simple methods can be used to understand what users wanted and needed. The methods used are:

I. Interviews with Stakeholders:

- Talk to event organizers, students, teachers, and IT staff.
- Ask them about the problems they face when managing events and what features they would like to see in a new system.

II. Surveys and Questionnaires:

- Design surveys and share them with students and teachers.
- This will help gather ideas from many people about their experiences and what they would like in an event management system.

III. **Focus Group Discussions:**

- Arrange small group meetings with students, teachers, and organizers.
- Discuss different ideas and confirm what features will be most important for the application.

IV. **Observation:**

- Attend some university events and watch how they are organized.
- This will help see the problems in the current system, like manual registrations and difficulty in booking venues.

V. **Research on Other Platforms:**

- Study event management websites used by other universities.
- This will give good ideas about useful features like online event calendars and mobile-friendly designs.

VI. **Prototyping and Feedback:**

- Make basic designs of the web application.
- Share these early designs with users to get their feedback and make improvements based on their suggestions.

SOFTWARE REQUIREMENTS SPECIFICATIONS

1. INTRODUCTION

1.1. Purpose:

The purpose of this Software Requirements Specification (SRS) document is to present a detailed description of the ***University Event Management Web Application***. It will provide a platform where students, faculty members, and university administrators can create, promote, and manage events effectively. The goal is to streamline the event organization and participation process within the university environment by introducing a centralized, efficient, and user-friendly web application.

This document serves as a communication tool between stakeholders (students, faculty, administrators) and developers, ensuring that all parties have a mutual understanding of the application requirements.

1.2. Scope:

This web-based system will enable:

- Event organizers (faculty or authorized student bodies) to create, edit, manage, and cancel events.
- Students to view, register for, and receive notifications about university events.
- The administration to monitor all events, manage users, and oversee system integrity.

The system will handle various types of events including seminars, competitions, cultural programs, and workshops.

It will offer:

- Secure login and registration.
- Real-time event tracking and updates.
- Automated email reminders.
- Feedback collection for event improvement.

Initially, the platform will be deployed for internal university use only.

1.3. Definitions, acronyms, abbreviations:

Term	Meaning
SRS	Software Requirement Specifications
UI	User Interface
UX	User Experience
Admin	Administration
SMTP	Simple Mail Transfer Protocol
API	Application Programming Interface
HTTPS	Hypertext Transfer Protocol Secure

Database	Organized collection of structured information
Front-end	The visible part of application that users see.
Back-end	The hidden server-side part that processes logic and stores data

1.4. References:

- **Course Materials:** Software Engineering Concepts – BCE Lectures (COMSATS University Islamabad, Abbottabad Campus).

- **Slides:** Requirements Engineering I & II (Shared in class by Dr. Ghulam Mujtaba).

2. OVERALL DESCRIPTION

2.1. Product Perspective:

The University Event Management Web Application is an independent system but will be developed in such a way that it can later integrate with the university's main portal.

The product will use a three-tier architecture:

- Frontend
- Backend
- Database

2.2. Product Functions:

The main functionalities will include:

- User authentication and role-based access.
- Event creation, updating, viewing, and deletion.
- User event registration and attendance tracking.
- Automatic email notifications and reminders.
- Feedback form generation and submission post-event.
- Administrative control for managing users and events.
- Event category-based filtering and search options.

2.3. Assumptions and Dependencies:

Assumptions

- ✓ **Internet Access:** It is assumed that all users (students, faculty, and admin) will have a working internet connection to use the web application.
- ✓ **Modern Web Browsers:** It is assumed that users will use updated web browsers like Chrome or Firefox to access the system properly.
- ✓ **Valid University Accounts:** It is assumed that users will have valid university email IDs for registration and login.
- ✓ **Device Availability:** It is assumed that users will have devices like laptops, mobiles, or tablets to use the web application.
- ✓ **Correct Information:** It is assumed that users will provide correct information when they create accounts or events.
- ✓ **Technical Support:** It is assumed that the university's IT team will help maintain the servers, fix technical problems, and provide backups if needed.
- ✓ **User Load:** It is assumed that the system will handle up to 10,000 users in the starting phase without major problems.
- ✓ **Email Service:** It is assumed that the email system used by the university will work properly to send event confirmations and reminders.
- ✓ **Admin Responsibility:** It is assumed that admins will monitor events and user activity regularly to prevent misuse.

Dependencies

- ✓ **Website Hosting:** The web application depends on university-provided servers or an external hosting service to keep the website running.
- ✓ **Email System:** The system depends on an email server to send important emails like registration confirmations and event updates.
- ✓ **Login Systems:** If Single Sign-On is used, the system will depend on those services for user login.
- ✓ **Database System:** The application depends on a database like MySQL to save users, events, registrations, and feedback data.
- ✓ **Security Certificates:** The web application depends on SSL certificates to make sure the website is safe (HTTPS secure).
- ✓ **University Rules:** The system must follow university policies related to data protection and user privacy.

3. SPECIFIC REQUIREMENTS

3.1. Functional Requirements:

These are the specific actions the system must perform:

- ***User Management:***
  - The system shall allow users to register new accounts.
  - The system shall allow users to recover/reset forgotten passwords.
  - The system shall allow users to update their personal profile information.
- ***Authentication and Authorization:***
  - The system shall ensure users must log in to create, register for, or manage events.
  - The system shall differentiate between normal users, event organizers, and admins.
- ***Event Management:***
  - The system shall allow event organizers to create new events by entering event title, description, location, date, time, and category.
  - The system shall allow organizers to upload event posters or banners.
  - The system shall allow event organizers to edit the details of their own events.
  - The system shall allow event organizers to cancel an event if needed.
- ***Event Registration:***
  - The system shall allow users to register for events through a "Register Now" button.
  - The system shall prevent users from registering for events that are already full.
- ***Event Notifications:***
  - The system shall send registration confirmation emails to users.
  - The system shall send automatic reminder emails 24 hours before the event.
  - The system shall notify users about any changes or cancellations of events they registered for.
- ***Search and Filter Events:***
  - The system shall allow users to search events by title, date, category, or organizer.
  - The system shall allow users to filter events by type (seminar, workshop, sports, cultural event, etc.).
- ***Feedback and Ratings:***
  - The system shall allow users to submit feedback after attending an event.
  - The system shall allow users to rate events on a scale (e.g., 1 to 5 stars).
- ***Admin Controls:***
  - The admin shall be able to view, approve, edit, or delete any event.
  - The admin shall view reports showing event registrations, user activity, and system usage statistics.

3.2. Non-Functional Requirements

These describe the qualities of the system:

- ***Usability:***
    - The system shall have a user-friendly interface with clear navigation menus.
    - The system shall provide instructions on forms for easy understanding.
- ***Performance:***
    - The system shall be able to handle 100 simultaneous event registrations without slowing down.
    - The homepage shall load within 3 seconds under normal network conditions.
- ***Reliability:***
    - The system shall have an uptime of at least 99% annually, with planned maintenance during low-usage hours.
- ***Security:***
    - All user passwords shall be stored using encryption.
    - The system shall use HTTPS for secure communication.
- ***Privacy:***
    - Personal user information (email, phone) shall not be shared without consent.
    - Feedback submissions shall be anonymous unless users choose to disclose their identity.
- ***Mobile Responsiveness:***
    - The system shall be fully responsive, adjusting layout automatically for desktops, tablets, and smartphones.
- ***Scalability:***
    - The system shall be designed to easily add more servers if user numbers increase.
- ***Maintainability:***
    - The code shall be modular to allow easy updates and bug fixes.

3.3. External Interface Requirements:

These define how the system interacts with users and other systems:

- ***User Interface (UI):***
    - The application shall support multiple screen resolutions (mobile, tablet, desktop).
    - Buttons, forms, and text fields shall be clearly visible and easily clickable.
- ***Hardware Interface:***
    - The web application will run on standard university or cloud servers (Linux or Windows servers).
    - Users require basic internet-enabled devices like smartphones, laptops, or tablets.
- ***Software Interface:***
    - The system will use MySQL for the database.
    - The frontend shall be built using HTML, CSS, JavaScript.

    - The backend shall use PHP, Node.js frameworks.
    - Integration with SMTP servers for email sending.
- ***Communication Interface:***
    - The system shall use APIs to communicate between frontend and backend modules.

3.4. Constraints:

These are **limitations** or **rules** that affect how the system is built.

- ***Hosting Limitations:***
    - The university will provide limited server space (initially 50GB), so file sizes and attachments must be controlled.
- ***Project Timeline:***
    - The full system must be designed, developed, and deployed within six months.
- ***Security Policies:***
    - The system must follow university IT security rules and protect user data according to data privacy laws.
- ***User Authentication:***
    - Only university-verified students, faculty, and staff are allowed to use the system.
- ***Email Limitations:***
    - The system depends on the university's email services (SMTP); if email service fails, users might not get confirmations or reminders.
- ***Browser Support:***
    - The system will support only the latest two versions of Chrome, Firefox, and Safari browsers.

4. APPENDICES

4.1. Glossary

- ***Event:*** Any official university gathering, including academic, cultural, or extracurricular.
- ***Organizer:*** A faculty member or authorized student organization that creates events.
- ***Participant:*** A user who registers and attends events.

4.2. Supporting Documents

- ***Wireframes:*** Basic UI mock-ups of homepage, event listing page, and registration form.
- **Database Schema:**
    - ***Users Table:*** User_ID, Name, Email, Password, Role
    - ***Events Table:*** Event_ID, Title, Description, Date, Time, Venue, Organizer_ID
    - ***Registrations Table:*** Reg_ID, Event_ID, User_ID, Timestamp
    - ***Feedback Table:*** Feedback_ID, Event_ID, User_ID, Comments, Rating