PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

B. Tech (CSE) – 5th Semester – Aug-Dec 2023

SOFTWARE REQUIREMENTS SPECIFICATION

EVENTRA - College Events Management Platform

VERSION 1 APPROVED

Bhuvan Vijay Kumar	*PES1UG21CS140*
Chandana S M	*PES1UG21CS149*

Deeksha Krishnappa	*PES1UG21CS169*
Dhiruram B	*PES1UG21CS918*

Organization : PES University

Date Created : 8-10-2023

Table of Contents

Revision History

Name	Date	Reason for Change	Version

1. Introduction

1.1 Purpose

The purpose of EVENTRA - a college event management platform, is to facilitate the planning, organization, promotion, and execution of events within a college or university campus. Such a platform serves several important functions to enhance the overall college experience and campus community.

1.2 Intended Audience

The project documentation is intended for the various stakeholders such as:

- **Developers**: This group needs detailed technical information to understand the software's architecture, features, and functionality.
- **Project Managers**: They require a comprehensive overview of the project's scope, objectives, and requirements to effectively plan and oversee the development process.
- **Marketing staff**: Marketing staff should understand the software's features and benefits to develop marketing strategies and materials that effectively promote EVENTRA.
- **Users**: Users, including college students and staff, need to grasp the system's functionalities and how it will improve their event-related experiences on campus
- **Testers**: Testers must have a clear understanding of the software's requirements to create test cases and perform comprehensive testing.
- **Documentation writers**: Writers need to know the software's functionality and features in-depth to produce user manuals, help guides, and other documentation.

1.3 Product Scope

The College Event Management System encompasses a comprehensive set of features and functionalities to meet the specific needs of colleges and universities in planning, organizing, and managing events. The product scope includes:

1. **Efficient Event Planning**
2. **Event Creation and Scheduling**
3. **Simplified Registration**
4. **Communication and Promotion**
5. **Resource Management**
6. **Ticketing and Payment Processing**
7. **Data Analytics and Reporting**
8. **User Roles and Permissions**
9. **Mobile Accessibility**
10. **Integration and Compatibility**

1.4 References

There are no specific external documents or web references referenced in this SRS. The document itself serves as a standalone reference for the specified software requirements of the sports app. However, any relevant internal documents or guidelines from the development team or organization should be considered during the development process.

2. Overall Description

2.1 Product Perspective

The product specified is "EVENTRA," a college event management platform. Colleges and universities require a tailored event management platform due to the dynamic nature of events they host, spanning academic conferences, sports, cultural festivals, and workshops. Efficient event management is vital for student and staff engagement and a vibrant campus community.

EVENTRA is a new, self-contained product.It is not a follow-on member of a product family, nor is it intended as a replacement for existing systems. Instead, it is developed to address the specific challenges and requirements associated with event management in an academic setting. This platform serves as a comprehensive solution to create a more engaging and inclusive campus community.

2.2 Product Functions

- **User Registration and Profiles**
- **Event Registration and Management**
- **Event Discovery**
- **Participant Registration and Ticketing**
- **Event Approvals**
- **Feedback and Reviews**

2.3 User Classes and Characteristics

- **Club Heads:**
  - **Frequency of Use:** Frequent users, as they are responsible for planning and managing events.
  - **Subset of Functions:** They require access to the full suite of event planning and management features.
  - **Privilege Levels:** Typically have administrative privileges.
- **Participants:**
  - **Frequency of Use:** Occasional users, as they interact with the system primarily to register for events.
  - **Subset of Functions:** Primarily involved in event registration, payment, and access to event-specific information

- **Privilege Levels:** Limited access to event-related features.

➢ **Club Mentors:**

- **Frequency of Use:** Occasional users, as they interact with the system primarily to approve and overlook events.
- **Subset of Functions:** Primarily involved in event approvals, feedback, and access to event-specific information
- **Privilege Levels:** Administrative privileges

2.4 Operating Environment

OE1 - Eventra will operate on Windows 10 and above on Google Chrome version 117.0.5938.149

OE2 - Eventra will operate on Ubuntu 23.04 and above on Firefox version 118.0.1.

OE3 - Eventra will operate on macOS 11 (BigSur) and above on Safari version 17.0

2.5 Design and Implementation Constraints

Interoperability: It should be capable of integrating with existing college systems, such as student information systems or academic calendars. Ensuring compatibility and data exchange with these systems might pose technical challenges.

Hardware Limitations: The system should consider potential hardware limitations, such as server capacity, network bandwidth, and memory requirements, to ensure optimal performance during peak event periods.

User Authentication and Authorization: Strict user authentication and authorization mechanisms are required to ensure that only authorized personnel have access to specific features and data, especially for sensitive information.

Scalability: The system needs to be designed to accommodate a potentially large number of events, users, and data over time. Scalability constraints may arise when the system experiences rapid growth in terms of usage.

2.6 Assumptions and Dependencies

Assumptions:

1. Third-Party Services: The project assumes the availability and reliability of third-party services such as email service providers for notifications and payment gateways for financial transactions.
2. Stability of APIs: Assumption that the APIs used for integration with external systems (e.g., student databases, calendar applications) will remain stable.
3. Data Accuracy: Assumption that event data provided by organizers is accurate. Inaccurate data could impact event planning.
4. User Connectivity: The project assumes users will have internet access and suitable devices. Network issues for users could affect system usability.

Dependencies:

1. Data Input: Timely and accurate data input by organizers, administrators, and users is a project dependency.
2. Third-Party Services: Dependency on services like email and payment gateways for various functionalities.
3. API Integrations: Integration depends on external API availability and reliability.
4. Server Infrastructure: The project depends on procuring and setting up server infrastructure, possibly involving coordination with hosting providers.

3. External Interface Requirements

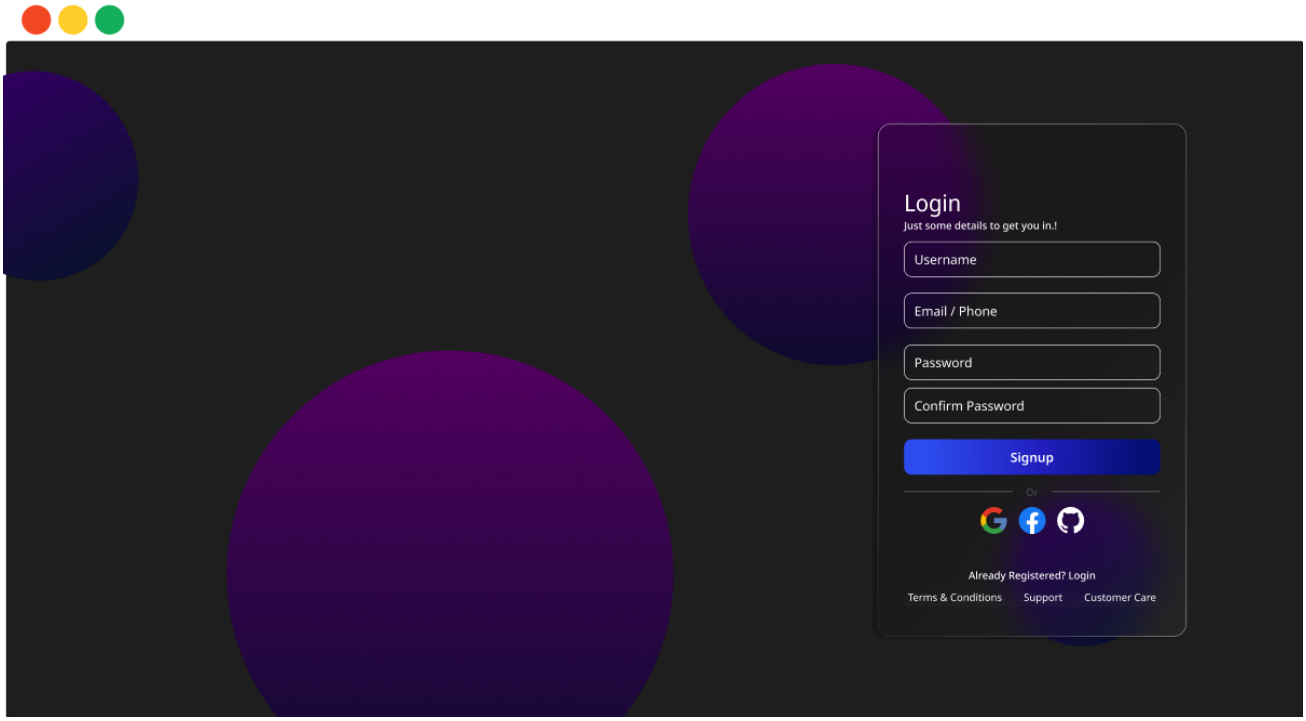
3.1 User Interfaces

An overview of the logical characteristics for some common user interfaces in such a platform:

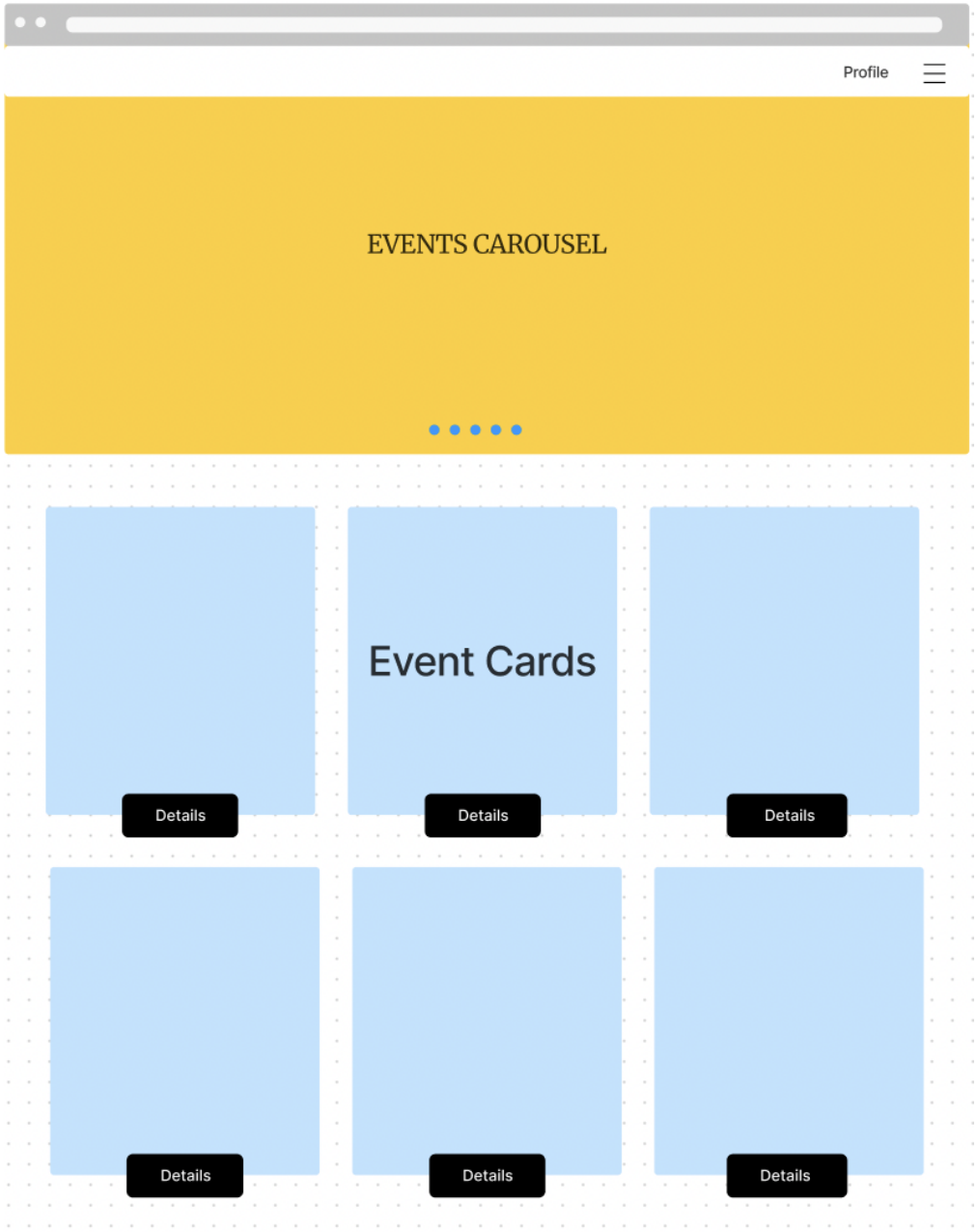
1. **User Registration and Login:** A typical login screen with fields for credentials and buttons for registration and login.
2. **Dashboard:** A dashboard providing an overview of upcoming events, user information, and options to navigate to various parts of the platform.
3. **Event Creation and Management:** A form for creating and managing events, including options to edit, delete, or publish.

4. **Event Listings:** A page displaying event listings in a grid or list format, allowing users to browse and access more event details.
5. **Event Registration:** A page where users can register for events, choose the number of tickets, and complete the registration process.
6. **Event Approvals:** A page which displays the approval status from the Faculty Advisor and the Dean.
7. **Search and Filters:** A search and filtering interface to help users find specific events.

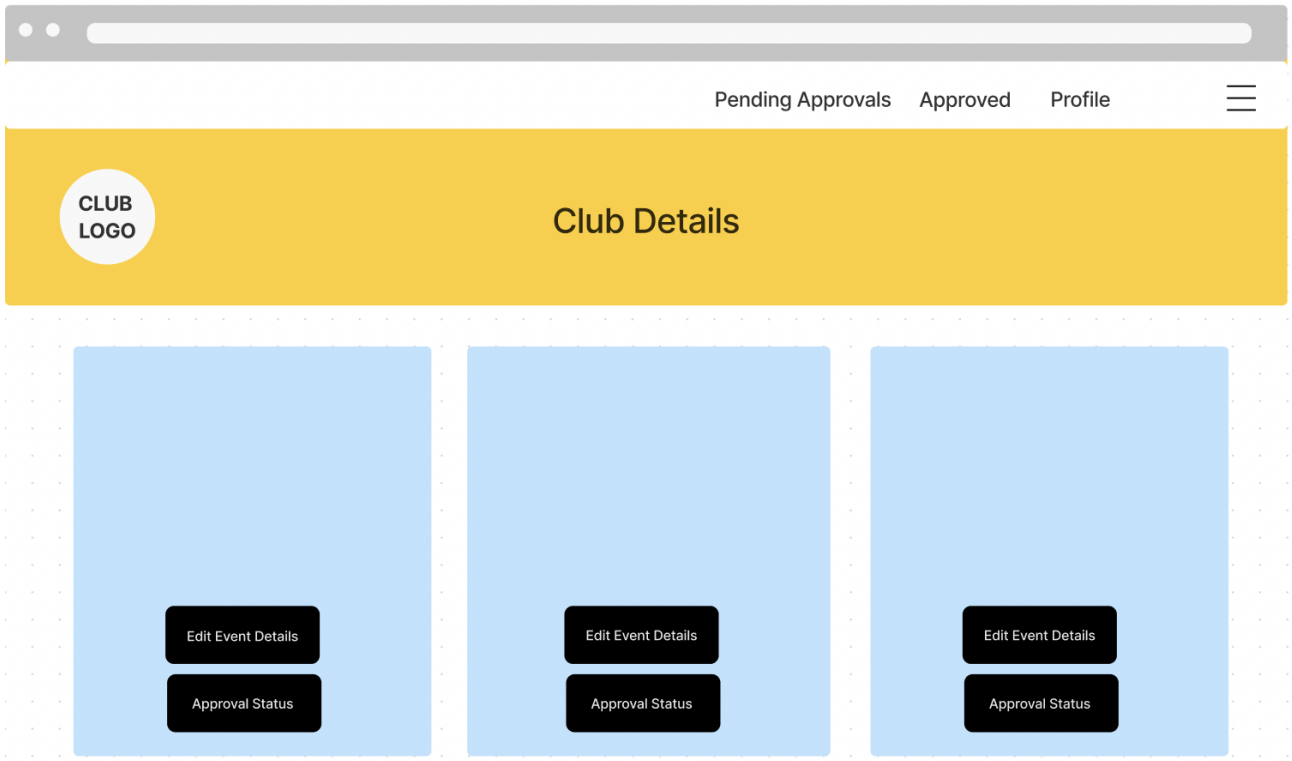
Login Page:



Participants Interface:



Club Head Interface:



3.2 Software Interfaces

3.2.1 System Connections:

- **Database**: Utilizes MySQL as the relational database management system for storing and retrieving event details, user profiles, registrations, and feedback.
- **Operating Systems**: The platform is designed to be cross-platform and is compatible with various operating systems, including Windows (Windows 7 and later), macOS, Linux distributions, Android, and iOS.
- **Third-Party Tools and Libraries**: The software integrates with third-party services for specific functionalities, such as email notifications, payment processing, and authentication services.

3.2.2 Data Flow:

3.2.2.1 Incoming Data and Messages:

- **User Input**: Users provide data input, including event details, registration information, feedback, and profile updates.
- **Third-Party Integrations**: Data flows from third-party services, such as payment gateways and email providers, for payment processing and communication.
- **Event Organizers**: Event organizers input event details and updates through administrative interfaces.
- **Approvals**: Club mentors and dean set the status of approvals.
- **User Registrations**: Users submit registration data for events, including attendee information and ticket purchases.

3.2.2.1 Outgoing Data and Messages:

- **Event Updates**: Messages about event details, registration confirmations, and event changes are sent to users and administrators. These updates keep users informed of event-related changes.
- **Payment Confirmations:** Messages about payment processing, transaction confirmations, and invoice receipts are sent to users and administrators.
- **Approvals**: Messages about approval status from club mentor and dean.
- **Feedback**: Messages about post-event feedback and surveys are sent to attendees and event organizers.

3.2.3 Services and Nature of Communication:

User Registration and Authentication Service:

a. Allows users to register, log in, and manage their profiles.

b. Communication includes user data, login credentials, and profile updates.

Event Management Service:

a. Enables event organizers to create, update, and manage events.

b. Communication includes event details, registrations, and updates.

Payment Processing Service:

a. Handles financial transactions and payment confirmations.

b. Communication includes payment details and transaction status.

Database Service (MySQL):

a. Used to store event data, user profiles, and transaction records.

b. Communication is bidirectional, involving CRUD (Create, Read, Update, Delete) operations over the SQL protocol.

3.3 Communications Interfaces

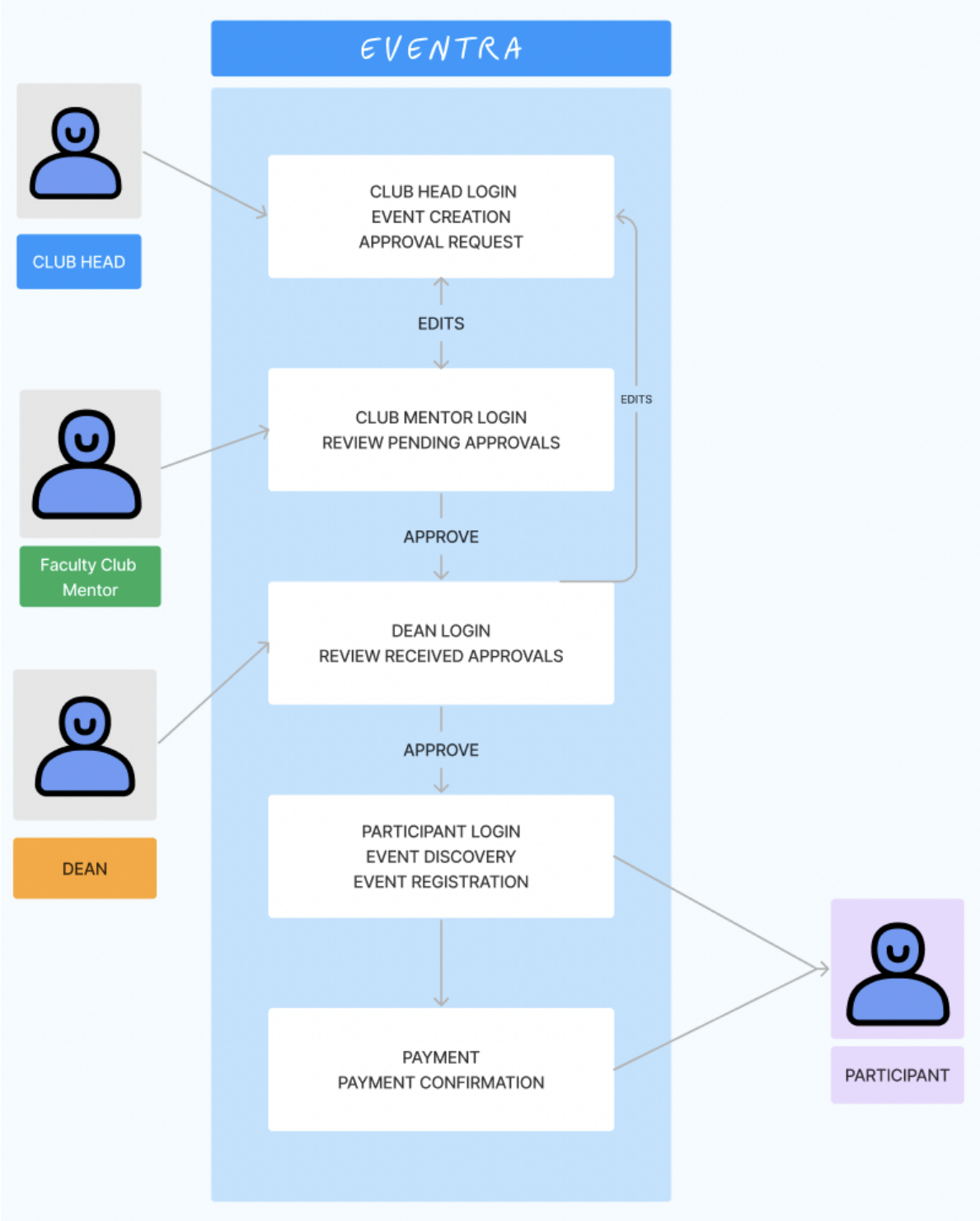
The EVENTRA college event management system relies on various communication interfaces to facilitate its functionality, including but not limited to:

1. **Web Browsing:** EVENTRA operates through standard web browsers (e.g., Chrome, Firefox, Safari) for user access. The system's user interface must be compatible with the latest versions of these browsers.
2. **HTTP/HTTPS:** Communication between the user's browser and the EVENTRA server is conducted over HTTP and secured via HTTPS to protect data transmission.
3. **Email:** The system uses email for various communication purposes, including user registration confirmation, event notifications, and password reset requests. Event-related notifications may contain event details, including name, date, time, and location.
4. **APIs:** EVENTRA may integrate with third-party services or external systems via Application Programming Interfaces (APIs). These APIs could be used for purposes such as event calendar synchronization, user authentication, or payment processing.
5. **Database Connectivity:** The system communicates with the database server using standard database communication protocols, such as ODBC or JDBC, to perform data storage and retrieval operations.
6. **Internal Messaging:** EVENTRA may include an internal messaging system for communication between event organizers, faculty club representatives, and students. Messages should be formatted to include sender, receiver, content, and timestamp.

7. **Notification Services:** The system may use push notification services for real-time event updates. These services should follow appropriate messaging standards and ensure the security of notifications.
8. **Authentication Services:** To verify user identities, the system may communicate with the college's existing authentication service using standard protocols, such as OAuth 2.0 or SAML.
9. **User Forms:** The system may employ electronic forms for user registration and event creation, which should follow web form standards, including data validation and security measures.
10. **Web Services:** For data exchange with external systems, EVENTRA may use web services, such as RESTful APIs, with appropriate message formats (e.g., JSON or XML).
11. **Communication Security:** All communication should be secured with industry-standard encryption protocols (e.g., TLS) to protect data during transmission. Passwords and sensitive data should be hashed and salted for storage.
12. **Data Transfer Rates:** The system should be optimized for efficient data transfer, striving for minimal latency in data retrieval and transmission.
13. **Synchronization Mechanisms:** In cases where data synchronization is necessary (e.g., with the academic calendar), appropriate synchronization mechanisms should be employed to ensure data consistency and accuracy.

4. Analysis Models

Use Case Diagram:



5. System Features

System Feature 1 : Club Registration

5.1.1 Description and Priority

Description: Allow students to register for an EVENTRA account.
Priority: High

5.1.2 Stimulus/Response Sequences

- User navigates to the registration page.
- Users enter their personal information.
- User submits the registration form.
- System validates the information and creates a new student account.
- System sends a confirmation email to the student.

5.1.3 Functional Requirements

REQ-1: The system must provide a user-friendly registration form that includes fields for the student's name, email, password, and other required information.
REQ-2: Upon submission, the system must validate the email format and ensure that the email is not already associated with an existing account.
REQ-3: The system must securely store student account information, including hashed passwords.
REQ-4: After successful registration, the system must send a confirmation email to the registered student's email address.
REQ-5: In case of a registration error (e.g., duplicate email or invalid password), the system must provide appropriate error messages to the user.

System Feature 2 : Event Creation

5.2.1 Description and Priority

Description: Allow event organizers to create new events.

Priority: High

5.2.2 Stimulus/Response Sequences:

- Event organizer logs in and selects the "Create Event" option.
- Organizer provides event details such as name, date, location, and description.
- System validates and saves the event information.
- Organizer receives a confirmation message.

5.2.3 Functional Requirements:

REQ-1: The system must offer an event creation form with fields for event name, date, location, description, and other relevant details.
REQ-2: The system must validate event details to ensure that required information is provided and that the date is in the future.
REQ-3: After successful event creation, the system must associate the event with the logged-in organizer.
REQ-4: The system must send a confirmation message to the event organizer upon successful event creation.
REQ-5: If there are issues with event creation (e.g., missing information or conflicting events), the system must provide appropriate error feedback.

System Feature 3 : Approval from faculty mentor of club

5.3.1 Description and Priority:

Description: Allow faculty member club representatives to approve or reject event creation requests from students.
Priority: High

5.3.2 Stimulus/Response Sequences:

- Faculty club representative logs in and navigates to the "Event Approval" section.
- Representative views a list of pending event creation requests.
- Representative selects a request and reviews event details.
- Representative approves or rejects the event creation request.

- System updates the event status and notifies the student organizer.

5.3.3 Functional Requirements:

REQ-1: The system must provide a dedicated "Event Approval" section for faculty club representatives with appropriate access rights.
REQ-2: Faculty club representatives should be able to view a list of pending event creation requests, including event details and organizer information.
REQ-3: Representatives should have the ability to approve or reject event creation requests.
REQ-4: Upon approval or rejection, the system should update the event status accordingly and notify the student organizer of the decision.
REQ-5: If a representative tries to approve or reject a request that is already processed or no longer exists, the system should provide an appropriate error message.

System Feature 4: Event Discovery

5.4.1 Description and Priority:

Description: Enable students to discover and explore events available in the system.
Priority: High

5.4.2 Stimulus/Response Sequences:

- Student logs in and navigates to the "Event Discovery" section.
- Students can search for events based on filters (e.g., date, category, location).
- Students view event details, including name, date, location, and description.
- Students can register for events directly from the discovery interface.

5.4.3 Functional Requirements:

REQ-1: The system must provide an intuitive and user-friendly "Event Discovery" section where students can explore and search for events.
REQ-2: Students should be able to search for events based on various filters, such as event date, category, location, and keywords.

REQ-3: Event details, including the event name, date, location, description, and organizer information, should be accessible to students.
REQ-4: Students should have the ability to register for events directly from the event discovery interface.
REQ-5: In case of event registration errors (e.g., event already full), the system should provide clear error messages to the user.

System Feature 5: Event Registration and Ticketing

5.5.1 Description and Priority:

Description: Allow students to register for events, generate tickets, and manage their event registrations.
Priority: High

5.5.2 Stimulus/Response Sequences:

- Students login and navigate to the "Event Registration" section.
- Students search for events they wish to attend.
- Student selects an event and registers.
- The system generates an electronic ticket for the registered event.
- Students can view, edit, or cancel their event registrations.

5.5.3 Functional Requirements:

REQ-1: The system must offer a user-friendly "Event Registration" section where students can search for and register for events.
REQ-2: Registered students should be provided with an electronic ticket for each event they register for.
REQ-3: The electronic ticket should contain essential event details, including the event name, date, time, location, and a QR code for event check-in.
REQ-4: Students should be able to view, edit, or cancel their event registrations.
REQ-5: When a student cancels an event registration, the system should release the event slot for other potential attendees.
REQ-6: The system should have a seamless event check-in process that allows organizers to scan electronic tickets at the event venue.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Response Time: The system must respond to user interactions within 2 seconds, ensuring a smooth user experience.

Scalability: The system should be able to handle an increasing number of events and users without a significant decrease in performance. It should support at least 10,000 simultaneous users during peak usage periods.

Database Performance: Database queries and transactions should have a response time of no more than 1 second under standard load.

6.2 Safety Requirements

Data Security: The system must encrypt all sensitive user data (e.g., passwords) using industry-standard encryption methods to prevent unauthorized access.

User Privacy: EVENTRA should comply with all relevant data protection and privacy regulations to safeguard user privacy.

Event Information: Users' personal information, such as email addresses, should not be shared with event organizers without explicit user consent.

6.3 Security Requirements

Authentication: Users should be required to authenticate their identity through secure means, such as email verification, to access their accounts.

Authorization: The system should employ role-based access control, ensuring that users only have access to features and data relevant to their roles.

Secure Data Transmission: Data transmitted between the user and the system should be encrypted using HTTPS to prevent eavesdropping.

Security Auditing: The system should log and monitor security-related events and provide audit trails for user activities.

6.4 Software Quality Attributes

Usability: The system should have a user-friendly interface with clear navigation and well-organized information.

Maintainability: Code should be well-documented and follow coding best practices to facilitate future maintenance and updates.

Availability: The system should aim for 99.9% availability, with planned maintenance communicated to users in advance.

Interoperability: The system should support common web browsers (e.g., Chrome, Firefox, Safari) and mobile platforms (iOS and Android).

Robustness: The system should handle unexpected errors gracefully and provide informative error messages to users.

6.5 Business Rules

Event Approval: Only authorized administrators can approve event creations before they are visible to students.

User Account Roles: Event organizers have different privileges compared to regular students. For example, they can create events, while students can only register for events.

6.6 Domain Requirements:

Academic Calendar Integration: EVENTRA should be synchronized with the college's academic calendar to avoid scheduling conflicts.

Event Categories: The system should have categories for different types of events (e.g., academic, social, sports).

User Authentication: Use the college's existing authentication system for student verification when registering for EVENTRA.

7. Other Requirements:

7.1 Database Requirements:

- **Database Management System:** The system should use a relational database management system (e.g., MySQL, PostgreSQL) for data storage.
- **Data Backup:** The system should automatically back up the database daily, and backups should be stored securely for at least 30 days.
- **Data Retention:** Event data should be retained for at least 2 years, while user data should be retained as per relevant privacy regulations.

7.2 User Support and Training:

- **User Support Channels:** Define the channels (e.g., email, chat, support tickets) through which users can seek assistance.
- **Training Materials:** If applicable, specify the availability of training materials or resources for users.

7.3 Reuse Objectives:

- **Code Reusability:** Code components should be organized and documented in a way that encourages code reusability in future projects or system enhancements.
- **Third-Party Libraries:** Utilize open-source libraries and frameworks that have appropriate licensing for easy integration and future development.

7.4 Documentation Requirements:

- **User Manual:** Provide a comprehensive user manual that explains how to use the system effectively, including user guides for students, event organizers, and administrators.
- **Code Documentation:** All code should be documented, with clear comments and explanations to aid in maintenance and future development.

Appendix A: Glossary

SRS: Software Requirements Specification- A document that defines the functional and non-functional requirements of a software system.

API: Application Programming Interface-A set of rules and protocols that allow different software applications to communicate with each other.

DBMS: Database Management System- Software that manages storage retrieval, and organization of data in a database.

HTTP: HyperText Transfer Protocol-A protocol used for transferring data over the internet, commonly used for web browsing.

HTTPS: HyperText Transfer Protocol Secure- A secure version of HTTP that encrypts data transferred between a user's browser and a web server.

Bug: A defect or error in the software that causes it to behave unexpectedly or incorrectly.

Encryption: The process of converting data into a code to protect it from unauthorized access.

Backups: copies of data that are stored separately to prevent data loss in case of system failures or disasters.

User Manual: Documentation that provides instructions and guidance on how to use a software application.

Appendix B: Field Layouts

Field	Length (bytes)	Data type	Description	Is Mandatory
Event ID	10	Numeric	Unique identifier for the event	Y
Event Name	60	String	Name of the event	Y
Event Date	8	Date	Date of the event	Y
Event Time	8	Time	Time of the event	Y
Event Location	60	String	Location of the event	Y
Event Organizer	60	String	Name of the event organizer	Y
Event Description	255	String	Description of the event	Y
Event Capacity	10	Numeric	Maximum number of attendees for the event	N
Event Registration Fee	10	Numeric	Fee for registering for the event	N
Registration Deadline	8	Date	Deadline for registering for the event	Y

Sample Report Requirements: Include the fields to be included in the report

Registration Report
Event Name
Event Date
Event Time
Event Location
Event Organizer
Registrant Name
Registrant Email Address
Registrant Phone Number
Registration Date
Payment Status

Transaction Report
Event Name
Registrant Name
Registration Fee
Payment Method
Payment Date
Payment Status

Additional Report Requirements
Attendance Report
Payment Summary Report
Event Feedback Report

Appendix C: Requirement Traceability matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ-01	The system shall allow users to create, edit, and manage events.	event_management.md	event_management_design.pdf	EventManager.py	TC-01	TC-02
2	REQ-02	The system shall allow users to register for events.	registration.md	registration_design.pdf	RegistrationManager.py	TC-03	TC-04
3	REQ-03	The system shall allow users to pay for event registration fees online.	payment_processing.md	payment_processing_design.pdf	PaymentProcessor.py	TC-05	TC-06
4	REQ-04	The system shall generate reports on attendance, registration, and payments.	reporting.md	reporting_design.pdf	ReportGenerator.py	TC-07	TC-08
5	REQ-05	The system shall provide status of event approvals.	approval.md	approval_design.pdf	ApprovalStatus.py	TC-09	TC-10