Software Requirements Specification

for

NaSCon Management System

Version 1.0 Approved

Prepared by

Huzaifa Rabbani (21i-2496)
Asif Ali (21i-0515)
Faizan Ali (21i-0422)

Date: 7th April, 2024

National University of Computer & Emerging Sciences
Islamabad, Pakistan

Table of Contents

1. Introduction

1.1 Purpose

This document outlines the software requirements for the NaSCon Management System. The system aims to efficiently manage the NaSCon event organized annually by FAST National University, Islamabad campus. It covers various aspects of event organization, including participant registration, event scheduling, venue management, and sponsorship coordination. It also specifies various functional as well as non-functional requirements of the product.

1.2 Document Conventions

NaSCon (National Solutions Conventions)
Other than this, the document follows standard conventions for clarity and readability.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and stakeholders involved in the development and implementation of the NaSCon Management System. Developers and testers should focus on the detailed requirements sections, while project managers and stakeholders may find the overview sections more relevant.

1.4 Product Scope

The NaSCon Management System is a user-friendly application designed to address challenges faced in efficiently managing the NaSCon event. It aims to streamline event categories, registrations, participant information, event logistics, and sponsorship activities. The system aligns with the objectives of FAST National University, Islamabad campus, to enhance event organization and participation experience.

1.5 Product Perspective

The NaSCon Management System is a standalone software product developed specifically for managing the NaSCon event organized annually by FAST National University, Islamabad campus. It is not part of a product family but serves as a crucial component for efficiently organizing and coordinating various aspects of the NaSCon event. The system interfaces with external entities such as participants, sponsors, and faculty mentors.

1.6 Product Functions

- Streamlined platform for managing event categories, registrations, and participant information.
- User roles with varying privileges to ensure secure access and efficient administration.
- Features for handling event logistics, including scheduling, venue management, and team formation.
- Facilitation of sponsor interaction and financial tracking, offering different sponsorship packages.

1.7 User Classes and Characteristics

- Administrators: Responsible for overall system management and administration.
- Faculty Mentors: Oversee specific event categories and provide guidance to student executives.
- Student Executives: Manage event-specific tasks and coordinate with participants and sponsors.
- Sponsors: Provide financial support and expect visibility and branding opportunities.
- Participants: Individuals or teams registering to participate in NaSCon competitions.

1.8 Operating Environment

The NaSCon Management System will operate in a web-based environment. It is compatible with modern web browsers such as Google Chrome, Mozilla Firefox, and Safari. The system is platform-independent and will run on any device with internet access, including desktops, laptops, tablets, and smartphones.

1.9 Design and Implementation Constraints

- Compliance with corporate and regulatory policies.
- Integration with third-party payment gateways for secure online transactions.
- Adherence to user interface design standards for consistency and usability.

1.10 User Documentation

User documentation components to be delivered along with the software include:

- User Manuals
- Tutorials

The documentation will be provided in digital formats accessible through the system interface.

1.11 Assumptions and Dependencies

- Assumption: The availability of stable internet connectivity for system operation.
- Assumption: Third-party services, such as payment gateways, will function as expected.
- Dependency: Availability of resources and support for system maintenance and updates from the development team

2. External Interface Requirements

2.1 User Interfaces

The NaSCon Management System will feature a web-based user interface accessible through standard web browsers. The interface will include the following logical characteristics:

- Intuitive navigation menus and buttons for easy access to system functionalities.
- Consistent layout and design across all screens for enhanced user experience.
- Sample screen images will be provided in a separate User Interface Specification document.

2.2 Hardware Interfaces

The NaSCon Management System does not have direct hardware dependencies. It is platform-independent and will operate on any device with internet access, including desktops, laptops, tablets, and smartphones.

2.3 Software Interfaces

The system will interface with the following software components:

- Operating System: Platform-independent, compatible with any modern operating system supporting standard web browsers.
- Payment Gateways: Integration with third-party payment gateways for secure online transactions.
- Email Services: Integration with email services for sending notifications and digital tickets to users.

3. System Features

3.1 Participant Registration

3.1.1 Description and Priority

Allow users to register for individual events or register teams for group events. (Priority = 9)

3.1.2 Stimulus/Response Sequences

- User selects the desired event(s) and provides personal information.
- System validates the information and registers the participant(s) or team.
- Confirmation email is sent upon successful registration.

3.1.3 Functional Requirements

REQ-1: Provide a registration interface for individual events and team events.

REQ-2: Validate user input to ensure accuracy and completeness of registration data.

REQ-3: Generate and send confirmation emails containing event details and unique IDs to registered participants

3.2 User Role Management

3.2.1 Description and Priority

Implement user roles with different access modifiers to ensure secure access and efficient administration. (Priority = 6)

3.2.2 Stimulus/Response Sequences

- A User logs in with assigned credentials.
- System verifies user role and permissions.
- Access to specific features and functionalities is granted based on user role.

3.2.3 Functional Requirements

REQ-1: Implement user authentication and authorization mechanisms.

REQ-2: Define user roles such as admin, faculty mentors, student executives, sponsors, and participants.

REQ-3: Assign appropriate privileges to each user role to control access to system features.

3.3 Events Registration

3.3.1 Description and Priority

Enable participants to register for individual and group events efficiently. (Priority = 9)

3.3.2 Stimulus/Response Sequences

- User selects desired event(s) and provides personal information.
- System validates the information and registers the participant(s) or team.
- Confirmation email is sent upon successful registration.

3.3.3 Functional Requirements

REQ-1: Provide registration interfaces for individual and group events.

REQ-2: Validate user input to ensure accuracy and completeness of registration data.

REQ-3: Generate and send confirmation emails containing event details and unique IDs to registered participants.

3.4 Event Category Management

3.4.1 Description and Priority

Enable administrators to manage event categories effectively. (Priority = 6)

3.4.2 Stimulus/Response Sequences

- Administrator accesses the event category management interface.
- Administrator adds, edits, or deletes event categories as needed.
- System reflects the changes in the available event categories for participant registration.

3.4.3 Functional Requirements

REQ-1: Provide an interface for administrators to add, edit, or delete event categories.

REQ-2: Ensure proper validation of category information to maintain data integrity.

3.5 Event Scheduling and Venue Management

3.5.1 Description and Priority

Allow administrators to schedule events and manage venue details effectively. (Priority = 5)

3.5.2 Stimulus/Response Sequences

- Administrator accesses the event scheduling and venue management interface.
- Administrator adds or modifies event timings, venues, or other relevant details.
- Changes made are reflected in real-time and communicated to registered participants.

3.5.3 Functional Requirements

REQ-1: Provide an interface for administrators to schedule events and manage venue details.

REQ-2: Notify registered participants of any changes made to event timings or venues.

3.6 Tickets Generation and Distribution

3.6.1 Description and Priority

Automatically generate digital tickets upon successful registration and distribute them to attendees. (Priority = 9)

3.6.2 Stimulus/Response Sequences

- User completes the registration process.
- System generates a digital ticket containing event details.
- Ticket is sent to the user via email for convenient access.

3.6.3 Functional Requirements

REQ-1: Automatically generate digital tickets upon successful registration.

REQ-8: Include event details and a unique identifier in the digital ticket.

REQ-9: Distribute the digital ticket to the user via email for easy access.

3.7 Food Registration and Deals

3.7.1 Description and Priority

Allow participants to register for food deals during event registration. (Priority = 7)

3.7.2 Stimulus/Response Sequences

- User proceeds to the food registration section during event registration.
- User selects preferred food deals from available options.
- System confirms food deal selection and includes it in the registration details.

3.7.3 Functional Requirements

REQ-1: Provide options for different food deals during event registration.

REQ-2: Include selected food deals in the registration confirmation sent to participants.

REQ-3: Ensure seamless integration of food registration with event registration process.

3.8 Administration Portal for Admin Tasks

3.8.1 Description and Priority

Provide administrators with a centralized portal to manage various tasks related to account, faculty mentor, student executives, event, and sponsor management. (Priority = 4)

3.8.2 Stimulus/Response Sequences

- Administrator logs into the administration portal using valid credentials.
- Administrator accesses different modules for account, faculty mentor, student executives, event, and sponsor management.
- System displays relevant options and functionalities based on the user's role and permissions.

3.8.3 Functional Requirements

REQ-1: Implement user authentication and authorization mechanisms for the administration portal.

REQ-2: Provide separate modules for account management, faculty mentor management, student executives management, event management, and sponsor management.

REQ-3: Ensure role-based access control to restrict access to sensitive functionalities.

3.9 Search for Specific Participants

3.9.1 Description and Priority

Allow administrators to search for specific participants based on various criteria. (Priority = 3)

3.9.2 Stimulus/Response Sequences

- Administrator accesses the participant search feature.
- Administrator enters search criteria such as name, email, or registration ID.
- System retrieves matching participant records and displays them to the administrator.

3.9.3 Functional Requirements

REQ-1: Implement a search feature for specific participants based on name, email, or registration ID.

REQ-2: Ensure fast and accurate retrieval of participant records.

REQ-3: Provide options to filter search results based on additional criteria if needed.

3.10 Logistics Management

3.10.1 Description and Priority

Allow Enable logistics heads to manage logistics effectively for events, including food arrangements based on participant registrations. (Priority = 5)

3.10.2 Stimulus/Response Sequences

- Logistics head accesses the logistics management interface.
- Logistics head generates reports listing participants who have registered for food deals.
- System provides necessary information for planning food arrangements accordingly

3.10.3 Functional Requirements

REQ-1: Provide logistics management interface with features to generate reports on food registration.

REQ-2: Include participant details and selected food deals in the generated reports.

REQ-3: Ensure easy export of reports in downloadable formats for convenience.

3.11 Search Functionality for Participants

3.11.1 Description and Priority

Allow participants to search for specific events based on name or category for easy registration. (Priority = 8)

3.11.2 Stimulus/Response Sequences

- Participant accesses the event search feature.
- Participant enters keywords or selects event categories for filtering.
- System displays relevant events matching the search criteria.

3.11.3 Functional Requirements

REQ-1: Implement a search feature for participants to find specific events based on name or category.

REQ-2: Ensure fast and accurate retrieval of event listings based on search criteria.

REQ-3: Provide options to view detailed event information and proceed with registration from the search results page.

3.12 Payment Gateways

3.12.1 Description and Priority

Integrate secure payment gateways to facilitate online transactions during event registration. (Priority = 8)

3.12.2 Stimulus/Response Sequences

- Participant completes the registration process and proceeds to payment.
- System presents secure payment options through integrated payment gateways.
- Participant selects a preferred payment method and completes the transaction.
- System confirms successful payment and updates registration status accordingly.

3.12.3 Functional Requirements

REQ-1: Integrate popular and secure payment gateways such as MasterCard, mobile apps like (easypaisa, jazzcash), or others.

REQ-2: Ensure encryption and secure transmission of payment data to protect user privacy.

REQ-3: Provide seamless payment processing with real-time confirmation of successful transactions.

REQ-4: Update registration status upon successful payment to grant access to event participation.

3.13 Event Feedback Form

3.13.1 Description and Priority

Provide participants with a feedback form to submit their opinions and suggestions about attended events. (Priority = 3)

3.13.2 Stimulus/Response Sequences

- Participant accesses the event feedback form after attending an event.

- Participant fills out the form with ratings and comments on various aspects of the event.
- System records the submitted feedback and updates event organizers accordingly.

3.13.3 Functional Requirements

REQ-1: Implement an event feedback form accessible to participants after event attendance.

REQ-2: Include rating scales and open-text fields for participants to provide comprehensive feedback.

REQ-3: Store submitted feedback securely and provide access to event organizers for analysis and improvement purposes.

3.14 Reports Generation

3.14.1 Description and Priority

Enable administrators to generate various reports for tracking and analysis purposes. (Priority = 4)

3.14.2 Stimulus/Response Sequences

- Administrator accesses the reporting tool from the administration portal.
- Administrator selects parameters such as date range, event category, or participant criteria for report generation.
- System generates the requested report based on the specified parameters.
- Administrator reviews and exports the generated report for further analysis or distribution.

3.14.3 Functional Requirements

REQ-1: Provide a reporting tool with options to generate reports based on predefined criteria.

REQ-2: Include parameters such as event category, date range, participant details, or financial transactions for report filtering.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

The system should respond to user actions within 2 seconds under normal load conditions to ensure a smooth user experience. The system should be able to handle a minimum of 1000 concurrent users without significant degradation in performance. Database queries should execute within 500 milliseconds to retrieve event information and participant details promptly.

4.2 Safety Requirements

This system has separate interfaces for Users. So, every user will be provided with a fixed area for viewing, updating, deleting their info regarding events etc.

4.3 Security Requirements

System would provide user the option for registration and login, backed up by their username and password. So every user will be authenticated before entering the system.

4.4 Software Quality Attributes

The system would allow multiple users to register, login and use different features depending on their roles. It would have different user interfaces for different roles. It also has some more attributes like reusability, testability and flexibility.

4.5 Reliability

The system should have a high level of uptime to ensure availability during critical event registration periods and throughout the duration of NaSCon.

4.6 Compatibility

The system should be compatible with a wide range of web browsers and devices to ensure accessibility for users regardless of their preferred platform.

4.7 Data Backup

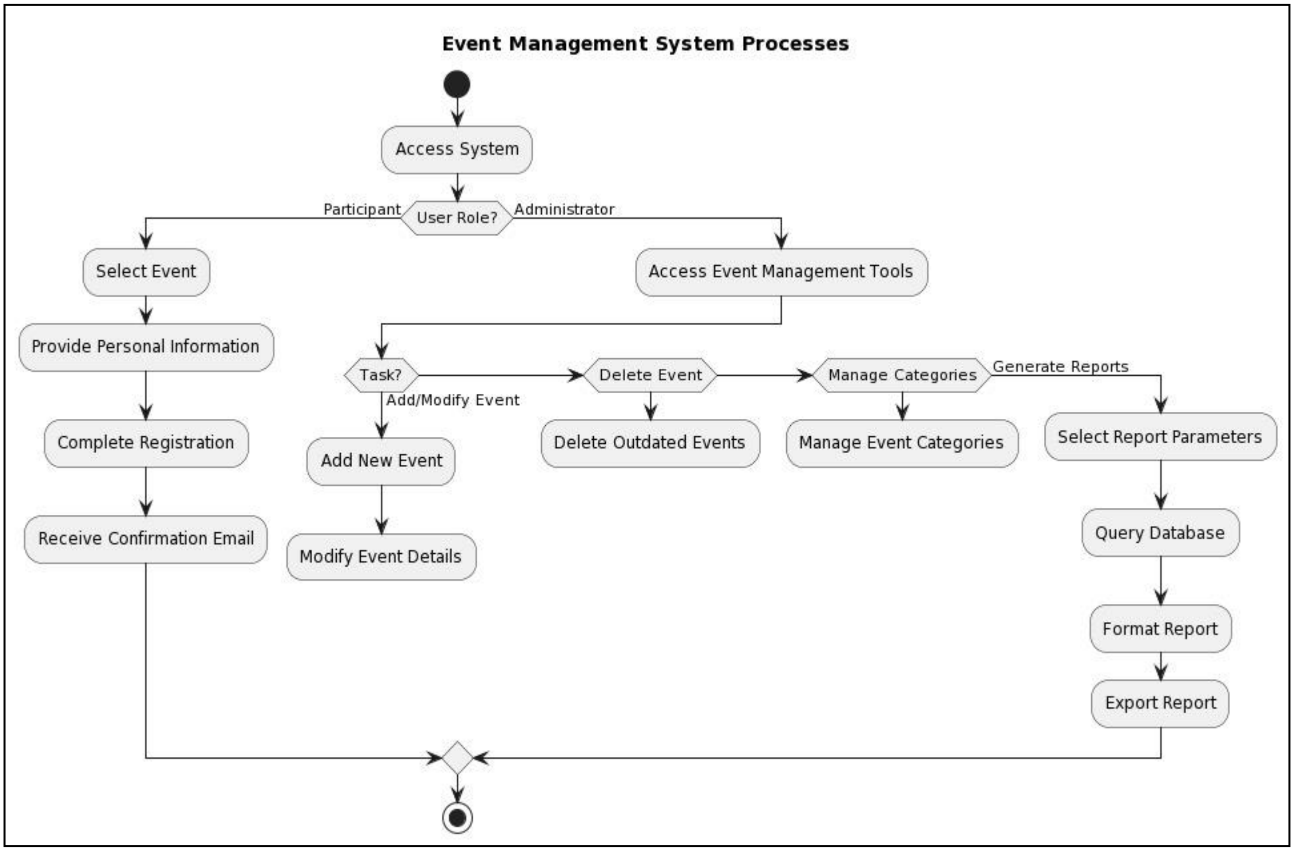
Regular backups of system data should be performed to prevent data loss in the event of hardware failures, system errors, or other unforeseen circumstances.

4.8 Business Rules

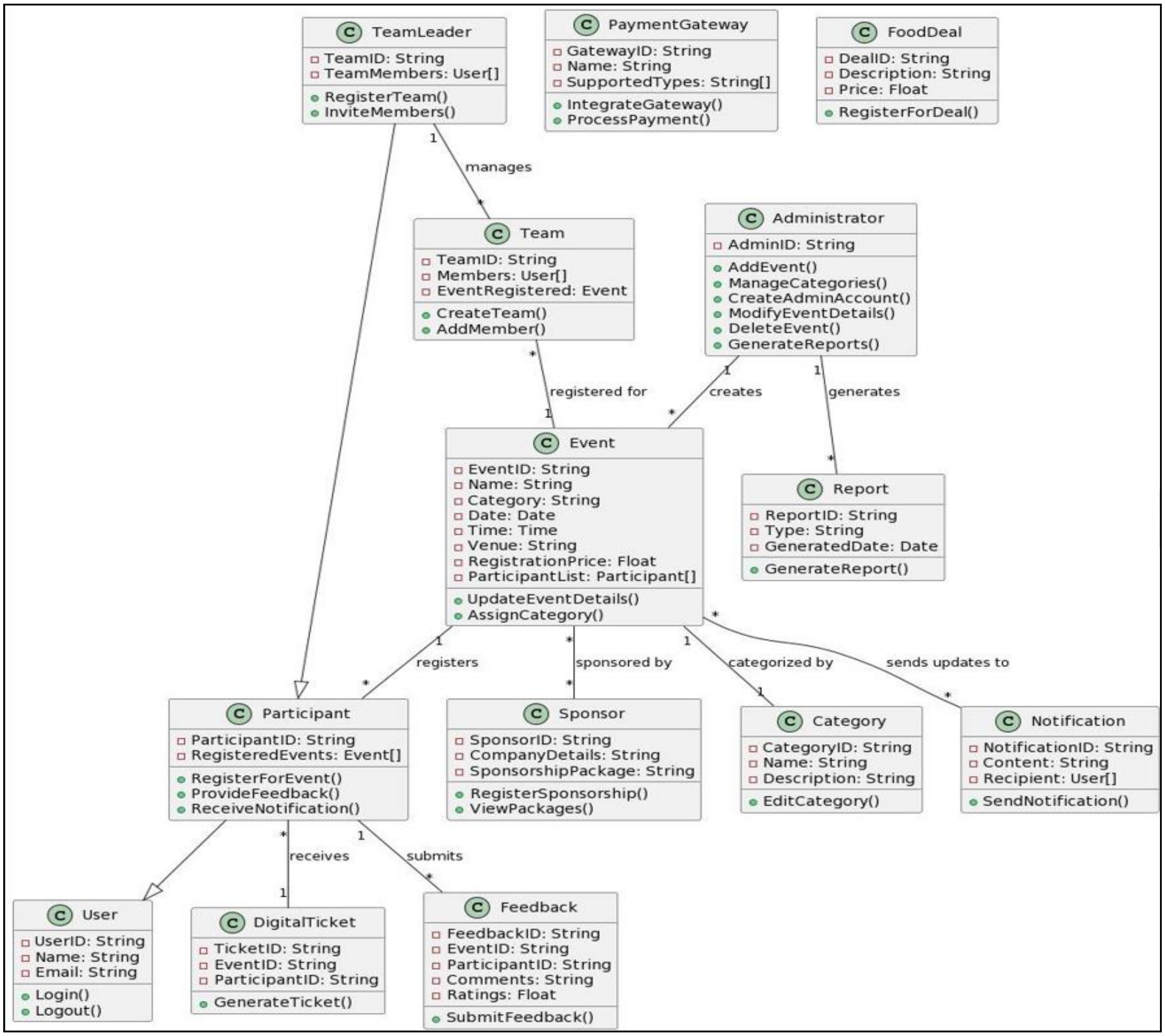
- **Role-Based Access Control**: Only authorized administrators should have access to administrative functions such as event management and user account management.
- **Event Registration Restrictions:** Participants should be allowed to register for events only within their specified category or eligibility criteria to ensure fair competition and resource allocation.
- **Feedback Submission Deadline**: Participants should be required to submit event feedback within a specified timeframe after the event's conclusion to gather timely insights for improvement.

5. Diagrams

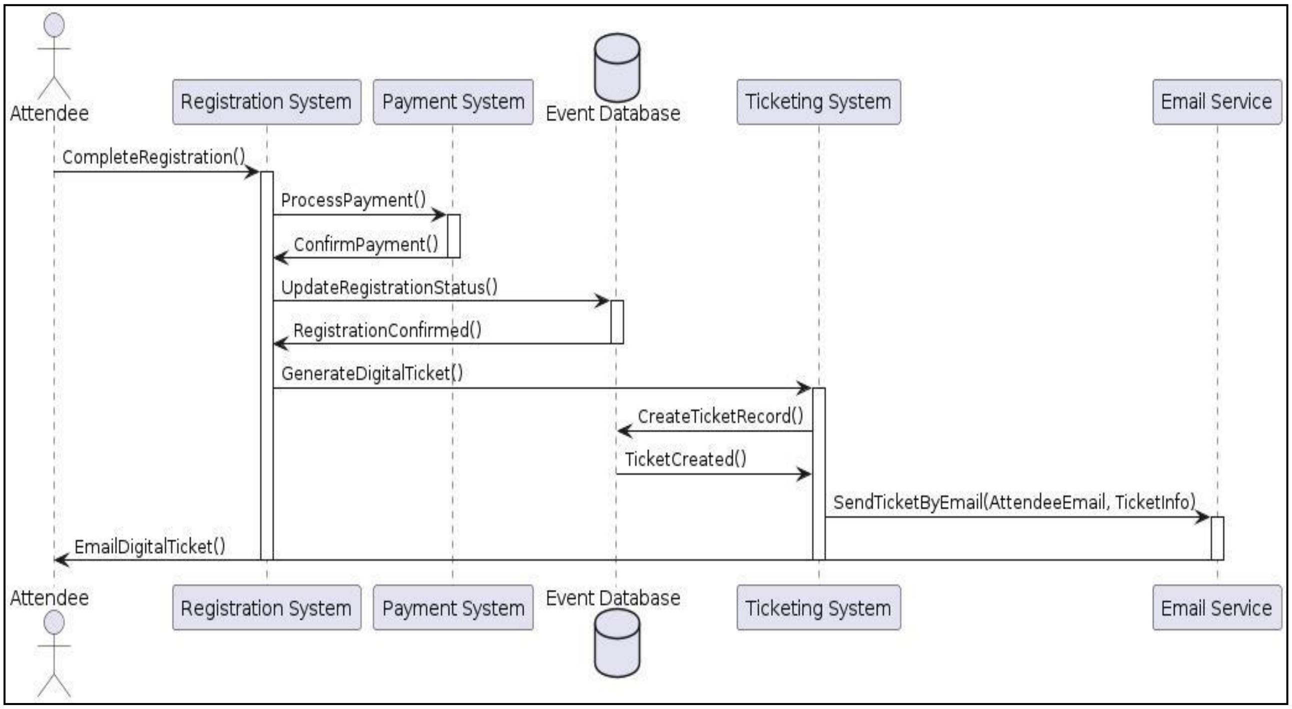
5.1 Activity Diagram

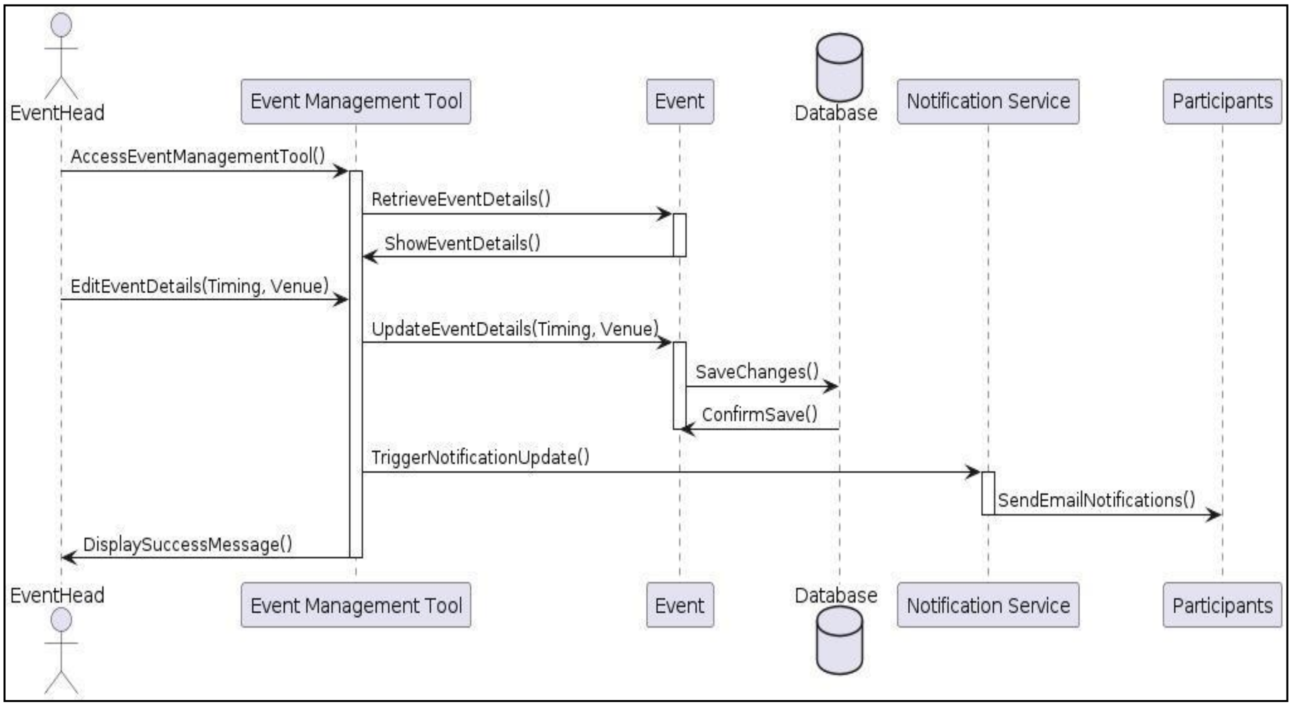


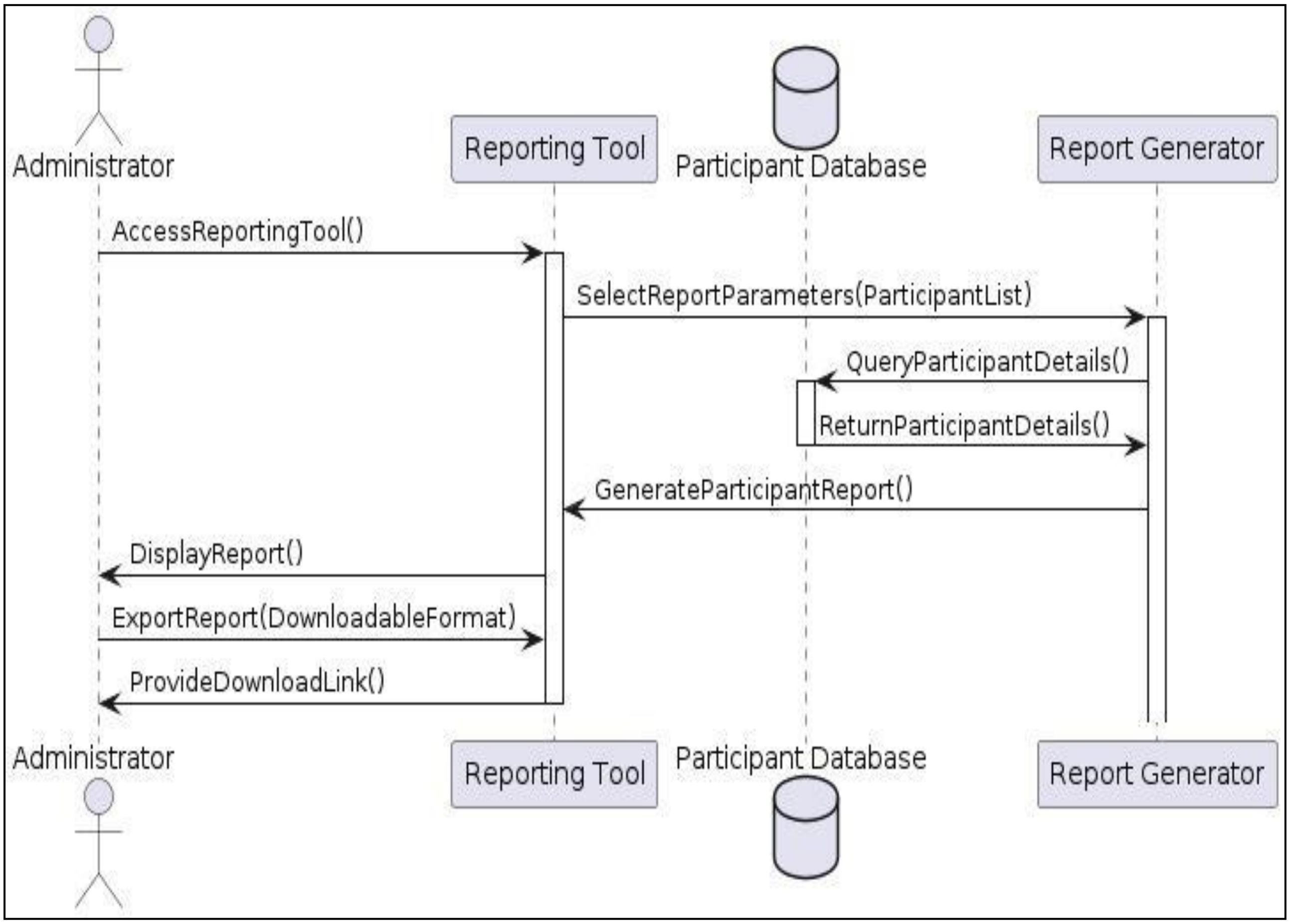
5.2 Class Diagram



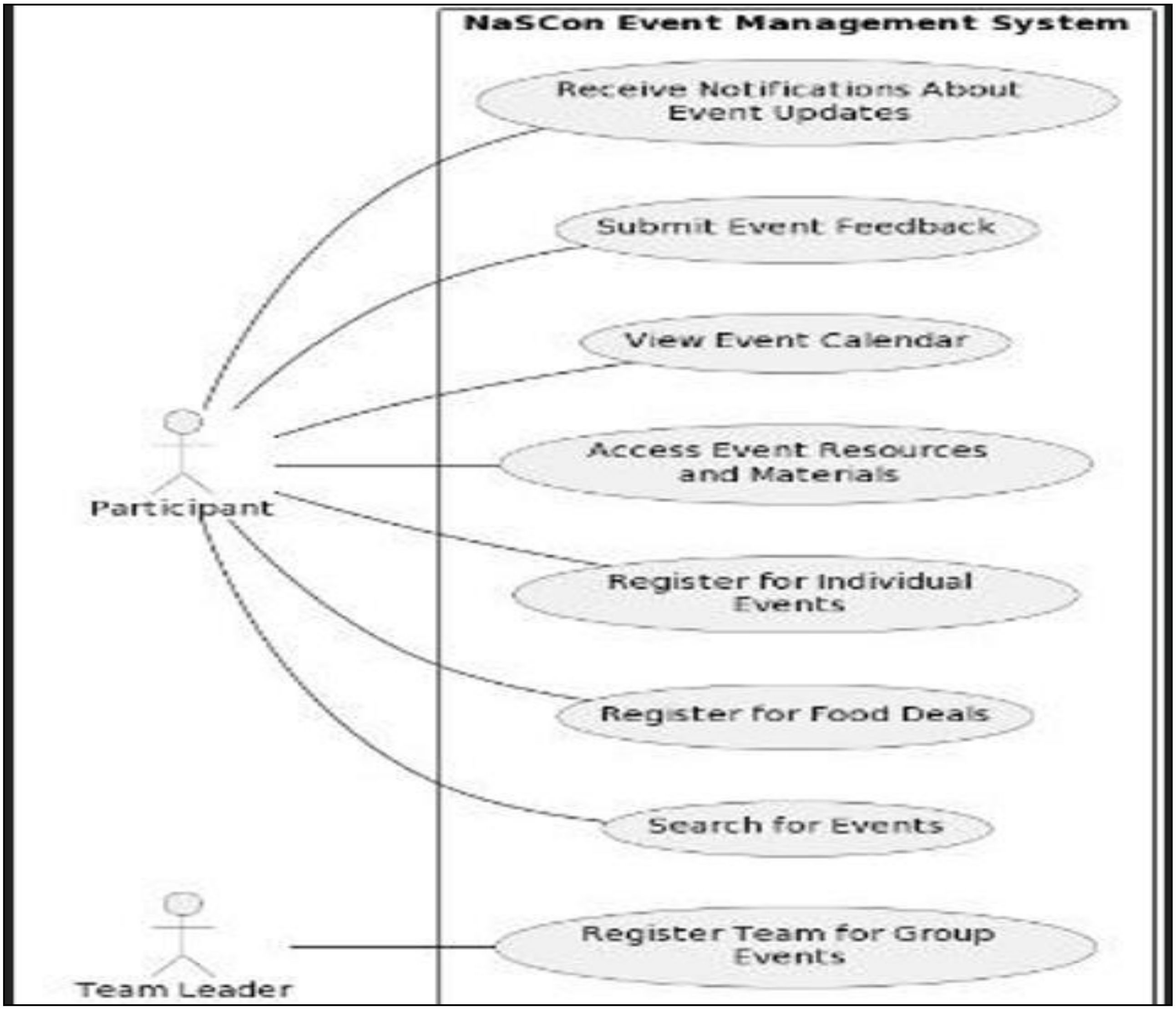
5.3 Sequence Diagrams

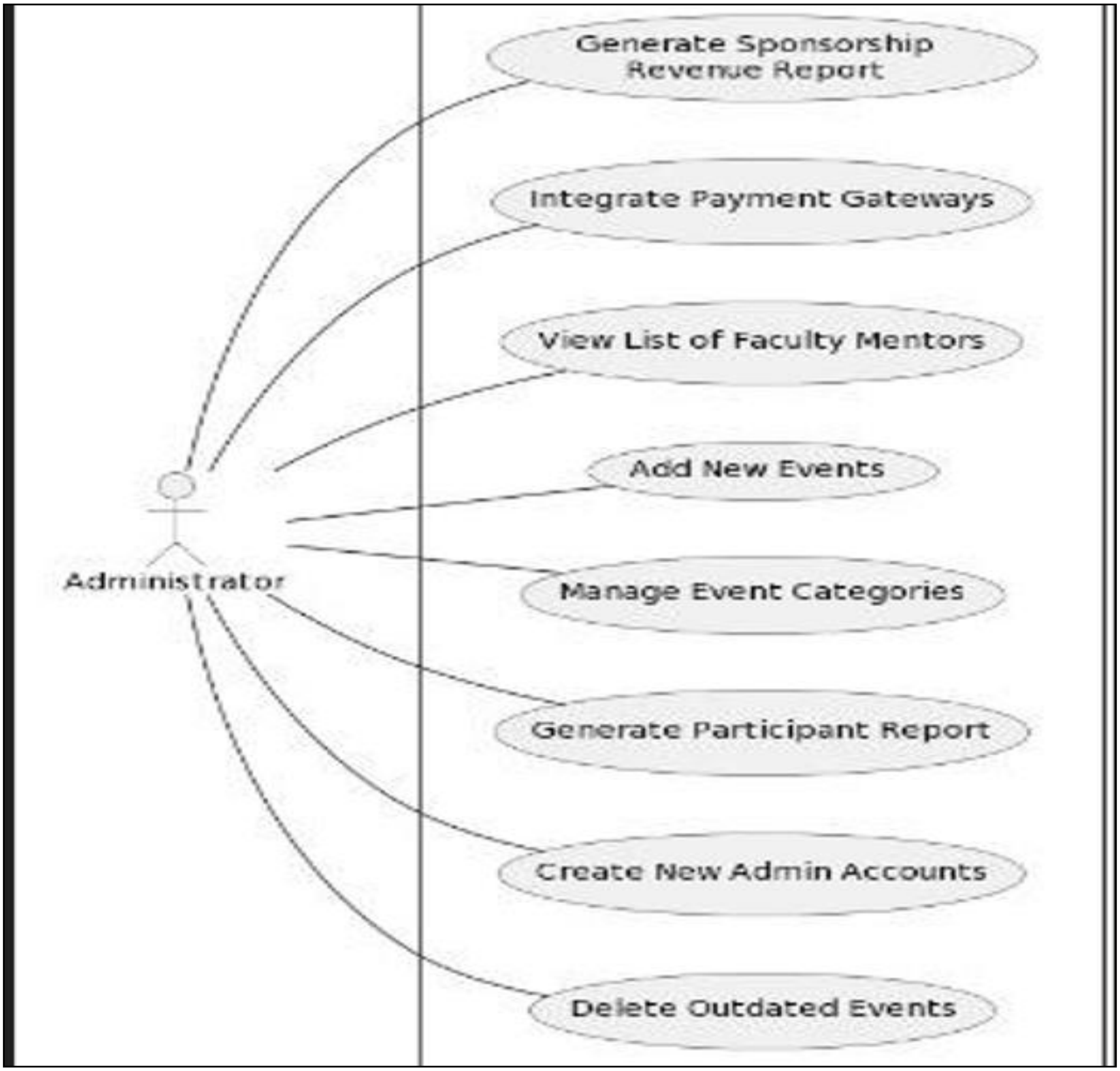


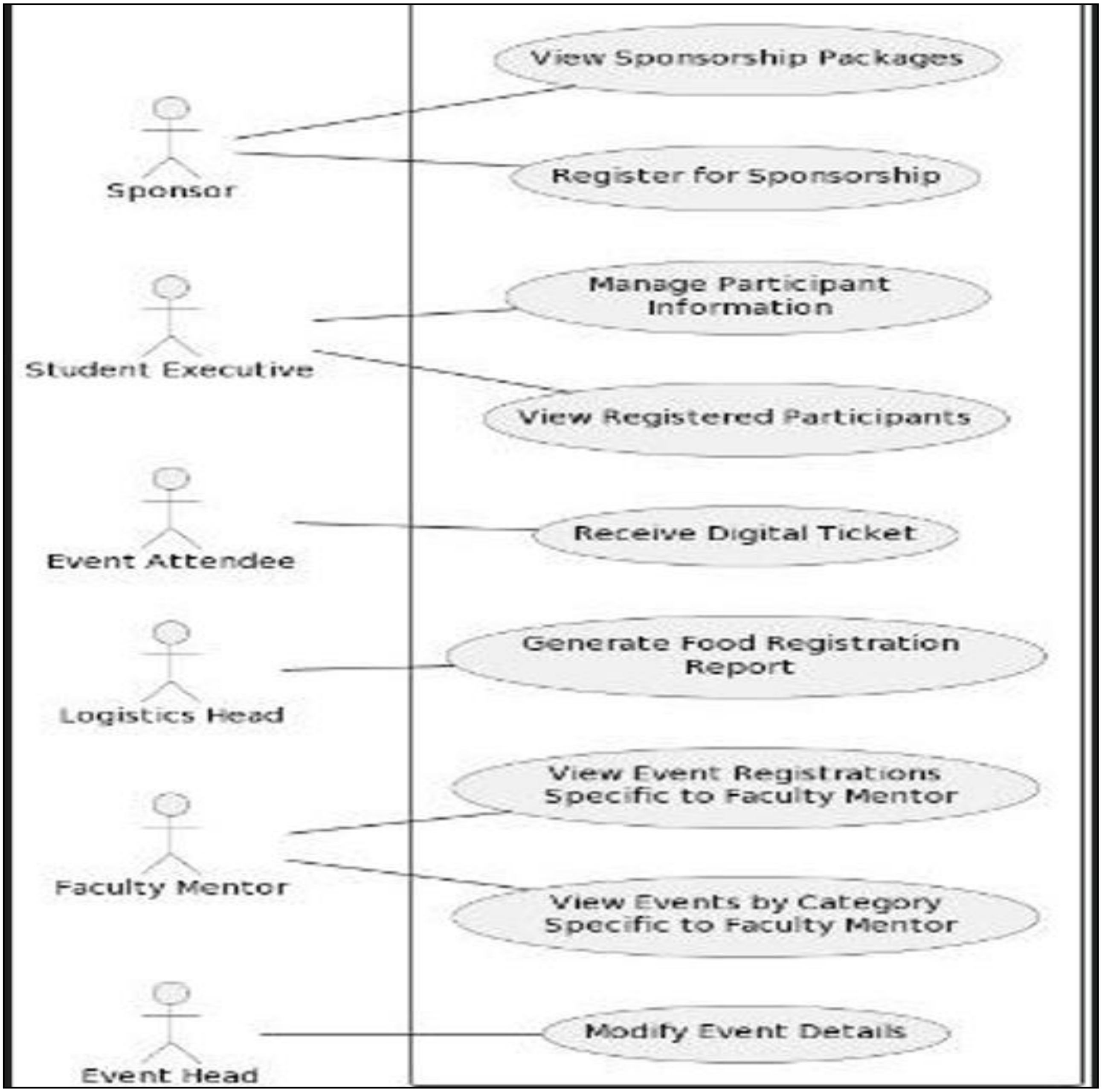



Administrator
Reporting Tool
Participant Database
Report Generator
AccessReportingTool()
SelectReportParameters(ParticipantList)
QueryParticipantDetails()
ReturnParticipantDetails()
GenerateParticipantReport()
DisplayReport()
ExportReport(DownloadableFormat)
ProvideDownloadLink()
Administrator
Reporting Tool
Participant Database
Report Generator

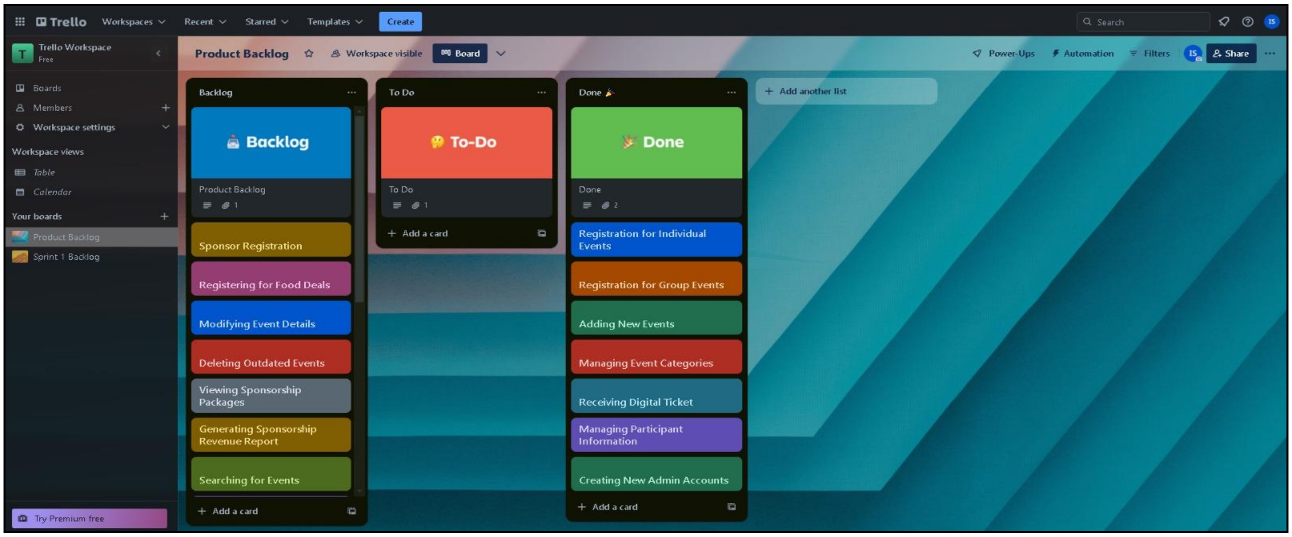
5.4 Use Case Diagrams


NaSCon Event Management System
Receive Notifications About Event Updates
Submit Event Feedback
View Event Calendar
Access Event Resources and Materials
Register for Individual Events
Register for Food Deals
Search for Events
Register Team for Group Events
Participant
Team Leader


Administrator
Generate Sponsorship Revenue Report
Integrate Payment Gateways
View List of Faculty Mentors
Add New Events
Manage Event Categories
Generate Participant Report
Create New Admin Accounts
Delete Outdated Events


Sponsor
Student Executive
Event Attendee
Logistics Head
Faculty Mentor
Event Head
View Sponsorship Packages
Register for Sponsorship
Manage Participant Information
View Registered Participants
Receive Digital Ticket
Generate Food Registration Report
View Event Registrations Specific to Faculty Mentor
View Events by Category Specific to Faculty Mentor
Modify Event Details

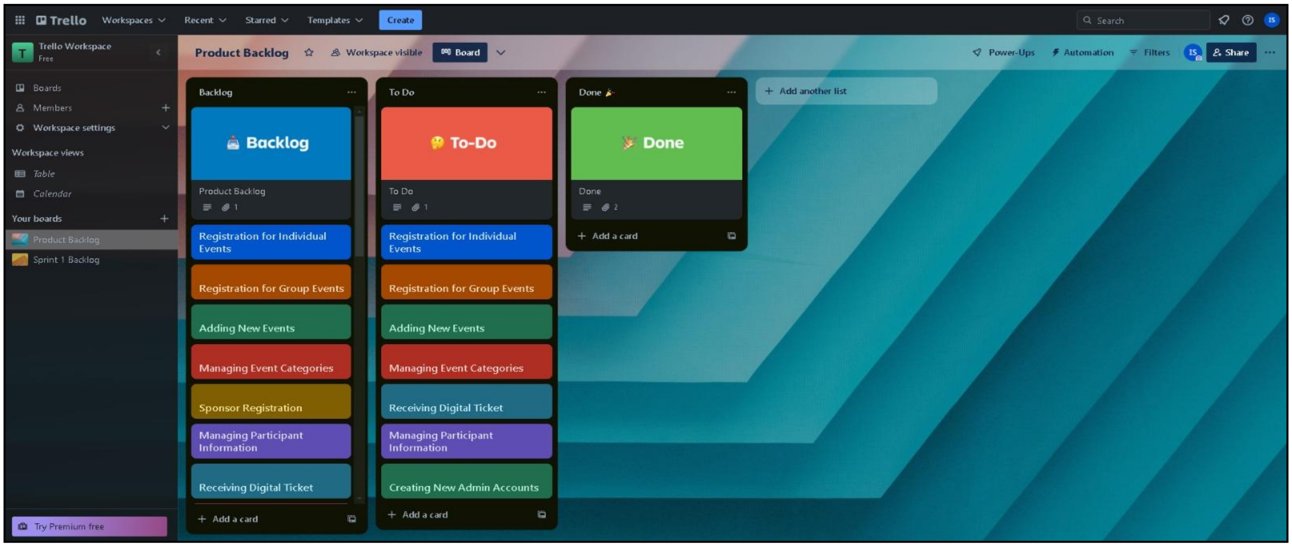
6. Snapshots of Trello Board



Backlog
- Backlog — Product Backlog
- Sponsor Registration
- Registering for Food Deals
- Modifying Event Details
- Deleting Outdated Events
- Viewing Sponsorship Packages
- Generating Sponsorship Revenue Report
- Searching for Events

To Do
- To-Do — To Do

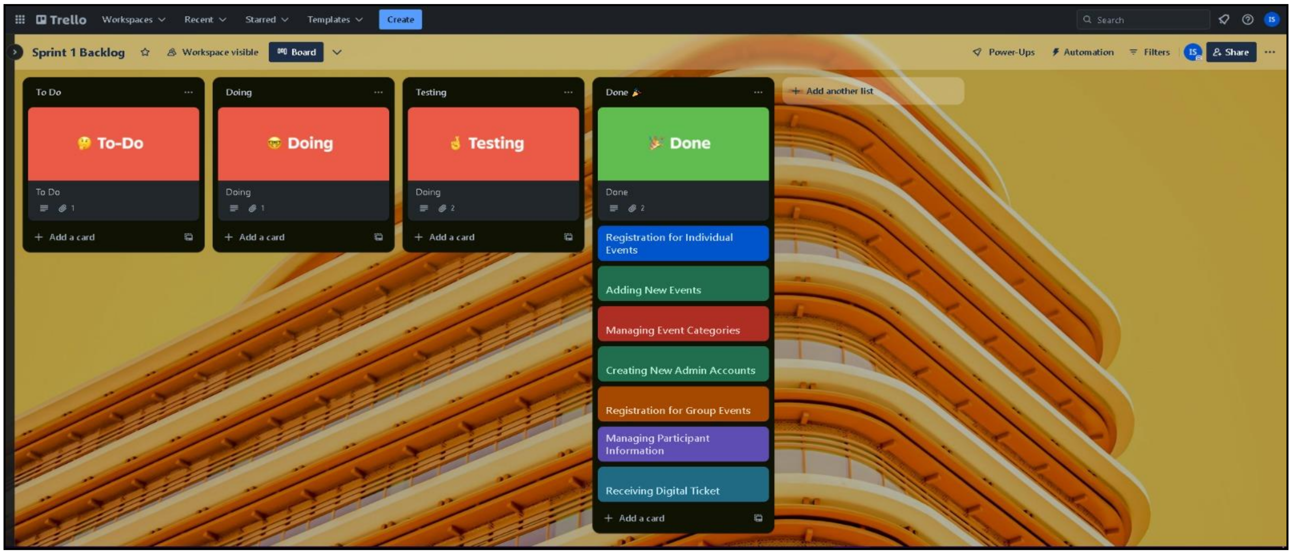
Done
- Done — Done
- Registration for Individual Events
- Registration for Group Events
- Adding New Events
- Managing Event Categories
- Receiving Digital Ticket
- Managing Participant Information
- Creating New Admin Accounts



Backlog
- Backlog — Product Backlog
- Registration for Individual Events
- Registration for Group Events
- Adding New Events
- Managing Event Categories
- Sponsor Registration
- Managing Participant Information
- Receiving Digital Ticket

To Do
- To-Do — To Do
- Registration for Individual Events
- Registration for Group Events
- Adding New Events
- Managing Event Categories
- Receiving Digital Ticket
- Managing Participant Information
- Creating New Admin Accounts

Done
- Done — Done

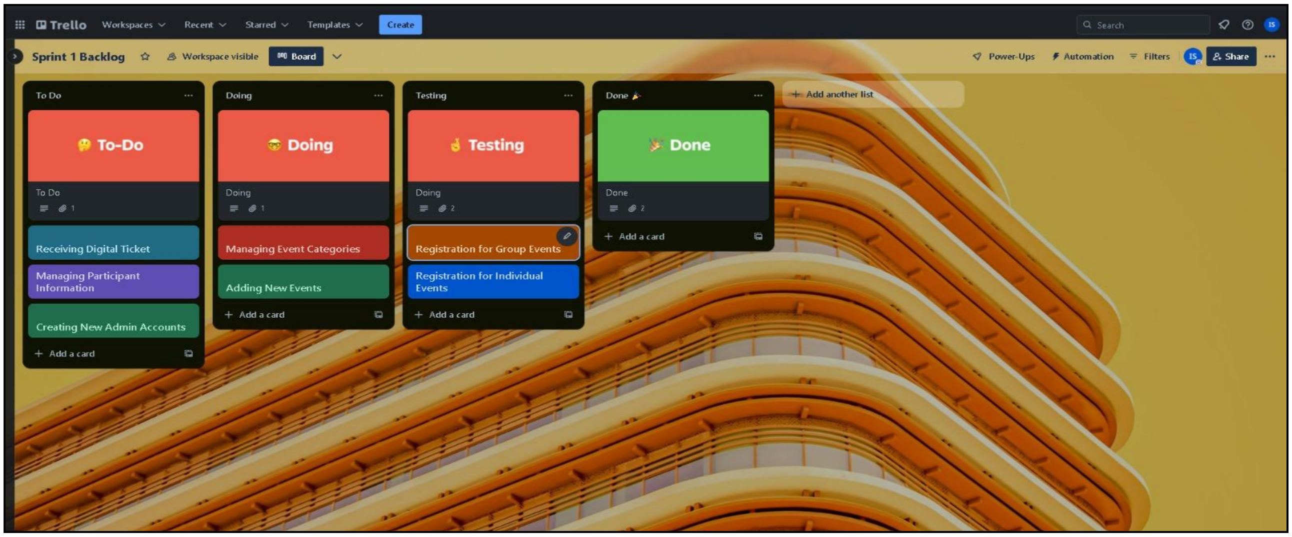


To Do
- To-Do — To Do

Doing
- Doing — Doing

Testing
- Testing — Doing

Done
- Done — Done
- Registration for Individual Events
- Adding New Events
- Managing Event Categories
- Creating New Admin Accounts
- Registration for Group Events
- Managing Participant Information
- Receiving Digital Ticket

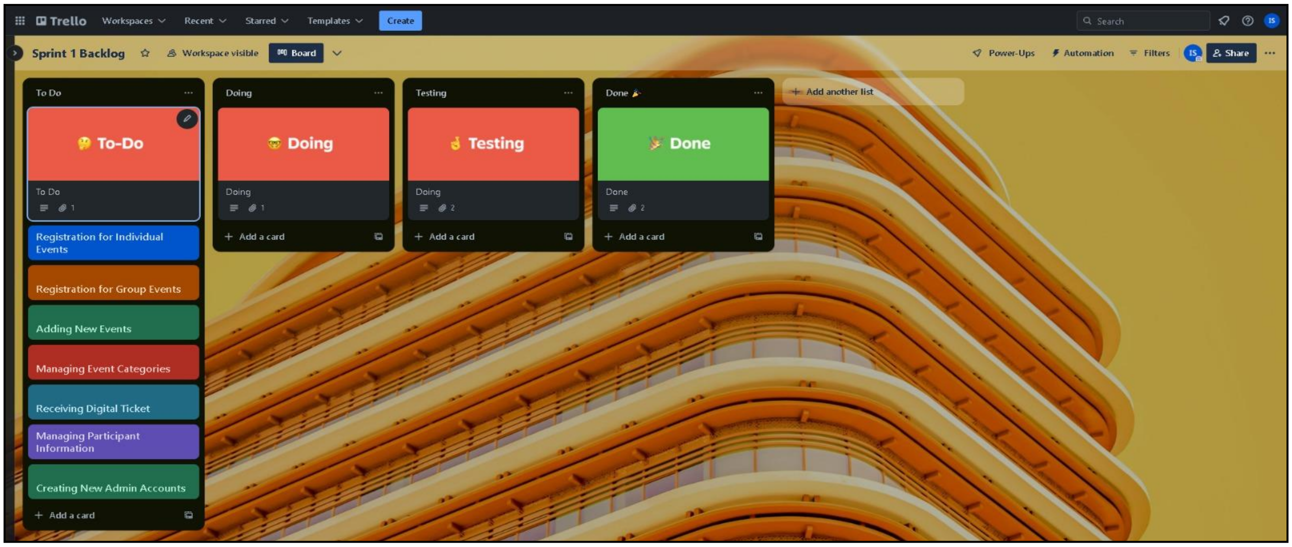


To Do
- To-Do
- Receiving Digital Ticket
- Managing Participant Information
- Creating New Admin Accounts
- + Add a card

Doing
- Doing
- Managing Event Categories
- Adding New Events
- + Add a card

Testing
- Testing
- Registration for Group Events
- Registration for Individual Events
- + Add a card

Done
- Done
- + Add a card

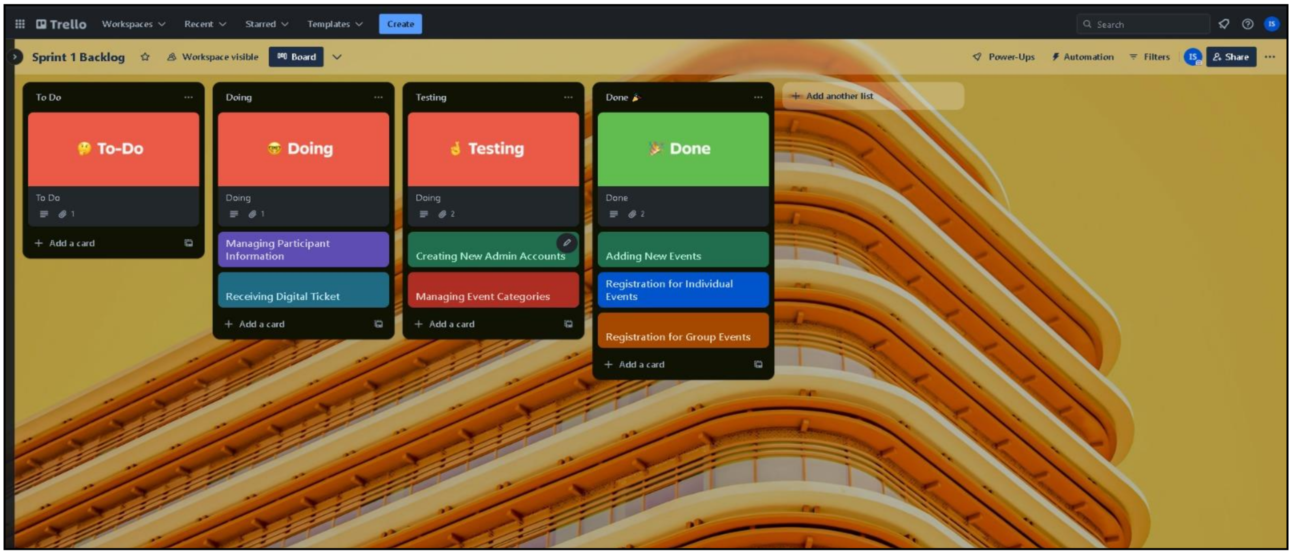


To Do
- To-Do
- Registration for Individual Events
- Registration for Group Events
- Adding New Events
- Managing Event Categories
- Receiving Digital Ticket
- Managing Participant Information
- Creating New Admin Accounts
- + Add a card

Doing
- Doing
- + Add a card

Testing
- Testing
- + Add a card

Done
- Done
- + Add a card



To Do
- To-Do
- + Add a card

Doing
- Doing
- Managing Participant Information
- Receiving Digital Ticket
- + Add a card

Testing
- Testing
- Creating New Admin Accounts
- Managing Event Categories
- + Add a card

Done
- Done
- Adding New Events
- Registration for Individual Events
- Registration for Group Events
- + Add a card

7. Sprint Backlog

7.1 User Story 1: Registration for Individual Events

- Task 1.1: Design and implement the event registration page.
- Task 1.2: Implement functionality to select individual events.
- Task 1.3: Implement functionality to input personal information.
- Task 1.4: Implement functionality to send confirmation email with event details and unique participant ID.

7.2 User Story 2: Registration for Group Events

- Task 2.1: Implement functionality for team leaders to log in.
- Task 2.2: Design and implement the team registration page.
- Task 2.3: Implement functionality to create a team and invite members.
- Task 2.4: Implement functionality to select group events and provide necessary team information.
- Task 2.5: Implement functionality to send confirmation emails to all team members with event details and unique team ID.

7.3 User Story 3: Adding New Events

- Task 3.1: Design and implement the event management interface.
- Task 3.2: Implement functionality to fill out the event details form and submit it.
- Task 3.3: Implement functionality to display the new event in the appropriate category for participants to register.
- Task 3.4: Implement functionality to send a notification confirming successful addition.

7.4 User Story 4: Managing Events Categories

- Task 4.1: Implement functionality to select an existing event category.
- Task 4.2: Implement functionality to edit the category name and description.

7.5 User Story 5: Managing Participant Information

- Task 5.1: Implement functionality for student executives to log in.
- Task 5.2: Design and implement the event dashboard.
- Task 5.3: Implement functionality to view and edit participant details for assigned events.

7.6 User Story 6: Receiving Digital Ticket

- Task 6.1: Implement functionality to complete the registration process.
- Task 6.2: Implement functionality to send an email containing a digital ticket with event information.

7.7 User Story 7: Creating New Admin Accounts

- Task 7.1: Implement functionality to fill out the new admin's details form and submit it.
- Task 7.2: Implement functionality to send login credentials and access permissions to the new admin.

7.8 User Story 8: Modifying Event Details

- Task 8.1: Implement functionality to edit event timings, venue information, or any other relevant details.
- Task 8.2: Implement functionality to reflect changes in real-time and communicate them to registered participants via email or notifications.

8. GitHub Repository