**Software Engineering**

**Software Requirements Specification**

**for**

FAST Events Portal

**Version 1.0 approved**

**Deliverable# 1**

**Group Members: Mohammad Attique (21i-0410), Taimur Amir (21i-0668), Shaheer Khan (21i-1358)**

**Submission Date: 13/04/2024**

**Table of Contents**

[**1. Introduction 1**](#_heading=h.3znysh7)

[1.1 Purpose 1](#_heading=h.2et92p0)

[1.2 Document Conventions 1](#_heading=)

[1.3 Intended Audience and Reading Suggestions 1](#_heading=)

[1.4 Product Scope 1](#_heading=)

[1.5 References 1](#_heading=)

[2. Overall Description 2](#_heading=)

[2.1 Product Perspective 2](#_heading=h.17dp8vu)

[2.2 Product Functions 2](#_heading=)

[User Registration and login: 2](#_heading=h.dg9ccc7qs85y)

[2.3 User Classes and Characteristics 3](#_heading=)

[2.4 Operating Environment 3](#_heading=)

[2.5 Design and Implementation Constraints 3](#_heading=)

[2.6 User Documentation 3](#_heading=)

[2.7 Assumptions and Dependencies 4](#_heading=)

[3. External Interface Requirements 4](#_heading=)

[3.1 User Interfaces 4](#_heading=h.z337ya)

[3.2 Hardware Interfaces 5](#_heading=)

[3.3 Software Interfaces 5](#_heading=)

[3.4 Communications Interfaces 6](#_heading=)

[4. System Features 7](#_heading=)

[4.1 System Feature 1 7](#_heading=)

[4.2 System Feature 2 7](#_heading=)

[4.3 System Feature 3 8](#_heading=h.az2f99cuwp1x)

[4.4 System Feature 4 8](#_heading=h.9ctzmvk4nmnf)

[4.5 System Feature 5 9](#_heading=h.ae93nsn92kzm)

[4.6 System Feature 6 9](#_heading=h.cjgquapf7k1i)

[4.7 System Feature 7 10](#_heading=h.xovg1dakde17)

[4.8 System Feature 8 10](#_heading=h.jd1rh3vs5bw5)

[4.9 System Feature 9 10](#_heading=h.iket02xds525)

[4.10 System Feature 10 11](#_heading=h.owwp4dupjeph)

[4.11 System Feature 11 11](#_heading=h.ix4reerk2pcz)

[4.12 System Feature 12 12](#_heading=h.5l89kzr7rxuj)

[4.13 System Feature 13 12](#_heading=h.ek5dnic6460m)

[4.14 System Feature 14 13](#_heading=h.debu3b5xuryk)

[4.15 System Feature 15 13](#_heading=h.gspa3bv1zqfl)

[4.16 System Feature 16 14](#_heading=h.tnwwupxd0l92)

[4.17 System Feature 17 14](#_heading=h.88ty7xvhg1sm)

[4.18 System Feature 18 15](#_heading=h.hvv1cxp3hnko)

[4.19 System Feature 19 15](#_heading=h.kbuny0biwmhh)

[4.20 System Feature 20 15](#_heading=h.r9j1808227lv)

[**5. Other Nonfunctional Requirements 16**](#_heading=h.2bn6wsx)

[5.1 Performance Requirements 16](#_heading=h.qsh70q)

[5.2 Safety Requirements 16](#_heading=)

[5.3 Security Requirements 16](#_heading=)

[5.4 Software Quality Attributes 16](#_heading=)

[5.5 Business Rules 16](#_heading=)

[6. Diagrams 17](#_heading=)

[6.1 Use Case Diagram 17](#_heading=)

[6.2 Activity Diagram 18](#_heading=)

[6.3 Sequence Diagram 21](#_heading=)

[**Appendix : Screenshots 23**](#_heading=h.n79feqshtij7)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

This document outlines what's needed to build the FAST Events Portal, an online platform to help manage events at FAST University. It details what the portal should do and how it should work.

## **Document Conventions**

This document follows standard rules for stating what the portal needs to do. Each requirement is labeled with how important it is.

## **Intended Audience and Reading Suggestions**

This document is for people involved in creating, deploying, and checking the FAST Events Portal, like developers, managers, university staff, and testers. It's set up to give a big picture first, then go into specific requirements. Start with the overview and then read the parts most relevant to you.

## **Product Scope**

The FAST Events Portal is meant to be a one-stop solution for managing events at FAST University. It lets students find events, sign up for them, and stay updated. For organizers, it provides tools to create event listings, track attendance, and get feedback. Overall, it's about making campus life more engaging and fun.

## **References**

Title: Event Portal GitHub Idea

Link: [*GitHub Style Idea*](https://github.com/HxnDev/Event-Management-System)

*Author:Sana Khan, Abeera Fatima, Azka Khurram*

*Date: 13/04/2024*

Title: Lums Event Portal Website

Link: [*Lums Event Portal*](https://lums.edu.pk/events)

*Author: LUMS UNIVERSITY*

*Date: 13/04/2024*

Title: FAST Event Portal Website

Link: [*Fast Event Portal*](https://www.nu.edu.pk/campus/Events/2)

*Author: FAST -NUCES*

*Date: 13/04/2024*

# **Overall Description**

## **Product Perspective**

The FAST event portal specified in this SRS is a product designed to meet the needs of a university community for managing and promoting events. It is a solution aimed at enhancing event organization, participation, and communication within the university ecosystem

The primary purpose of the university event portal is to provide a centralized platform for students, faculty, staff, and event organizers to create, discover, register for, and manage events across campus.

The idea was originated by FAST to fulfill the need to enhance event management processes and improve user engagement.

## **Product Functions**

## User Registration and login:

* Allow users to register an account and login securely

**Event Creation and Management:**

* Event Organizers can create new events, including specifying details like date, time, location, and description.
* Event Organizers can also manage events by updating or canceling an event.

**Event Tracking:**

* Allow users to register for events.
* Track attendance at events to record participation and manage capacity.

**Notifications and Reminders:**

* Send reminders to users about registered events.

**Profile Management:**

* Maintain profiles with personal information.

**Search and Filtering:**

* Users can find specific events based on criteria like date, category, location, or keywords.
* Filters will be present to refine search results and improve user experience.

**Feedback and Ratings:**

* Users can give Feedback and rating about events they attend, organizers, and overall experience.
* Ratings and reviews will be displayed to help users make informed decisions about attending events.

## **User Classes and Characteristics**

**Students:**

* **Frequency of Use:** Regularly access the portal to discover and register for events.
* **Technical Expertise:** Varied, ranging from basic to intermediate skills.
* **Privilege Levels:** Typically have standard user privileges, such as event registration and profile management.
* **Educational Level:** Undergraduate.
* **Importance:** most important user classes, as they constitute the primary audience for event participation.

**Administrators:**

* **Frequency of Use:** Regular use for event management, and overall portal administration.
* **Technical Expertise:** High, with proficiency in system administration and database management.
* **Privilege Levels:** Full administrative privileges, including user management, content moderation, and system configuration.
* **Educational Level:** Typically staff members responsible for IT or administrative functions.
* **Importance:** Critical user class for the portal's functionality, maintaining system integrity, security, and overall functionality of the portal.

## **Operating Environment**

* The software should be compatible with desktops used by students and administrators.
* The software will use a database management system to store and retrieve data.
* DBMS options may include MySQL, Microsoft SQL Server, or Oracle Database, depending upon preferences

## **Design and Implementation Constraints**

* Compliance with data protection rules will limit how user data is collected, stored, and processed.
* The software must perform well on standard desktops, considering factors like memory usage and processing power.
* Timing requirements for real-time updates or notifications may require optimization of code and server response times.
* coding standards, Git, and documentation requirements set by the development team and organization must be followed.
* Consistent UI/UX design.
* Use of specific programming languages or DBMS may be mandated by the organization,

## **User Documentation**

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## **Assumptions and Dependencies**

**Third-Party Components:**

* **Assumption:** Integration with a specific or paid payment gateway for events.
* **Impact:** If the chosen payment gateway is not compatible or experiences technical issues, it could affect the registration and payment process for events.

**Constraints:**

* **Assumption:** The project due date allows for sufficient testing and refinement of features before presenting.
* **Impact:** Insufficient time for testing could result in unresolved issues, impacting user satisfaction and adoption.

**Dependencies on External Factors:**

* **Assumption:** The portal relies on external APIs.
* **Impact:** Changes or disruptions in the APIs could affect the functionality of features within the portal.

**Reuse of Software Components:**

* **Assumption:** Reuse of authorized modules from a previous project.
* **Impact:** Compatibility issues of reused components could pose risks to the security and functionality of the portal.

# **External Interface Requirements**

## **User Interfaces**

**Software Components Requiring User Interface:**

1. User Registration and Login
2. Event Creation and Management
3. Event Registration Tracking
4. Notifications and Reminders
5. User Profiles and Preferences
6. Search and Filtering
7. Feedback and Ratings
8. Administration and Reporting

**GUI Standards and Style Guides:**

1. Follow design principles for consistency, usability, and aesthetics.
2. Use a clean and attractive layout with clear navigation menus, buttons, and interactive elements.
3. Follow Fast branding guidelines for color schemes, typography, and visual elements that reflect the university's identity.

**Standard Buttons and Functions:**

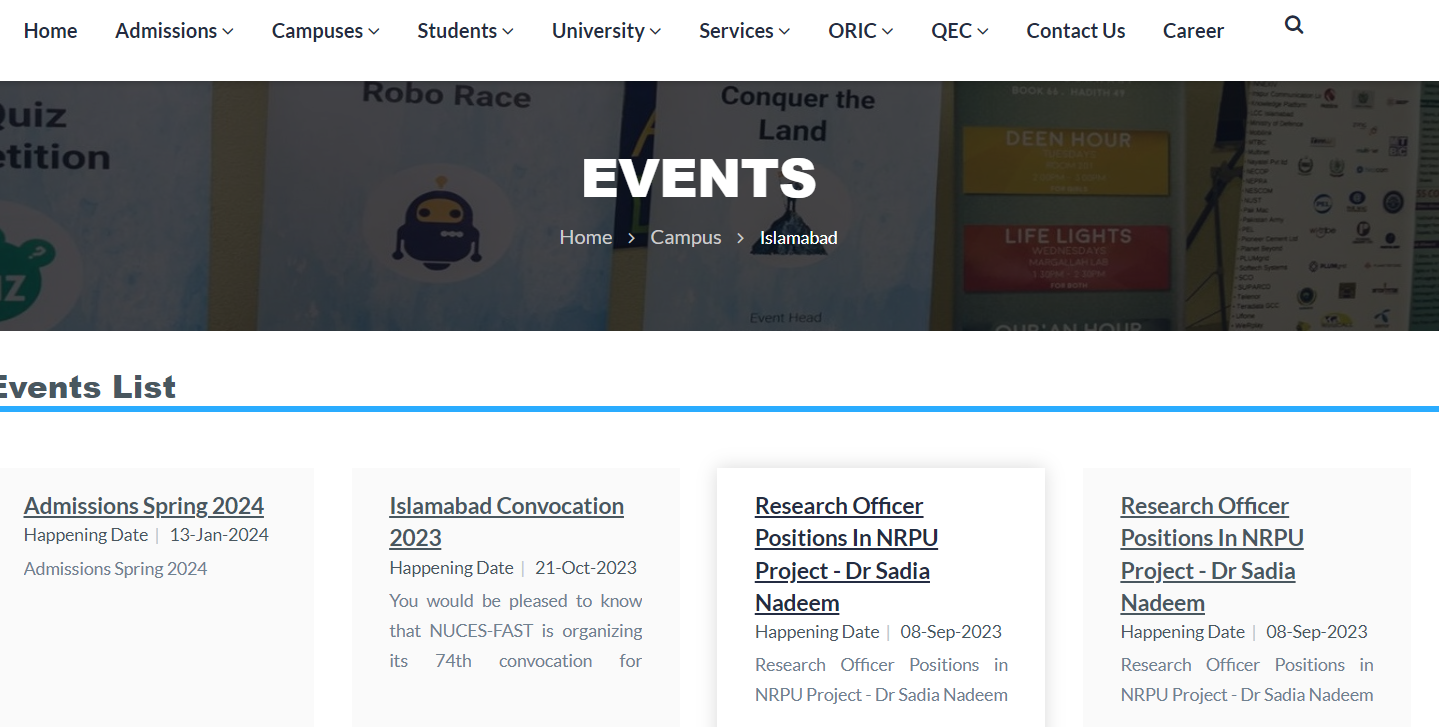
1. Include standard buttons such as "Register," "Login," "Create Event," "Edit Profile," "Search," "Filter," and "Submit."
2. Provide consistent functions like "Help," "Feedback," "Logout," and "Settings" accessible from every screen for user convenience.

**Error Message Display Standards:**

1. Display error messages in a clear and descriptive manner, highlighting the issue and providing guidance on resolving it.

**Sample Image:**

**NOTE: THIS IMAGE IS NOT FOR EVENT PORTAL ( UI/UX) will not be the same**

****

## **Hardware Interfaces**

**Supported Devices:** Desktops

**Interactions:** Receives user inputs and displays outputs

**Input Devices:** Mouse, keyboard.

**Output Devices:** Display screens, audio output.

**Security:** Secure communication, user authentication.

## **Software Interfaces**

**Database Management System (DBMS):**

* **Name:** MySrQL
* **Connection Purpose:** Store and retrieve event data, user profiles, and registration information.
* **Data Items:** Event details (date, time, location, description), user profiles (name, email, preferences), registration data (attendee information, event status).
* **Communication:** SQL queries and transactions through MySQL Connector.

**Operating System:**

* **Name and Version:** Windows
* **Connection Purpose:** Host the server-side components of the event portal.
* **Data Items:** System logs, configuration files, application execution.
* **Communication:** TCP/IP protocols for network communication, file system interactions.

**Frontend Framework:**

* **Name and Version:** ASP.NET
* **Connection Purpose:** Implement the user interface and client-side logic of the event portal.
* **Data Items:** User interface components, event listing, form inputs, interactive elements.
* **Communication:** RESTful APIs for server communication, state management using Redux.

**Backend Framework:**

* **Name and Version:** C#
* **Connection Purpose:** Implement server-side logic, data processing, and API endpoints.
* **Data Items:** User authentication data, event operations, database interactions.
* **Communication:** RESTful APIs for client-server communication, Object-Relational Mapping for database interactions.

## **Communications Interfaces**

**Email Communication:**

* **Requirement:** Sending notifications, reminders, and event updates to users via email.
* **Message Formatting:** HTML and plain text formats for email content.
* **Communication Standards:** SMTP (Simple Mail Transfer Protocol) for sending emails.

**Electronic Forms:**

* **Requirement:** User input via web forms for event registration, and profile updates.
* **Message Formatting:** Data validation, form submission formats.
* **Data Transfer Rates:** Optimized form submission for efficient data transfer.

**Data Synchronization:**

* **Requirement:** Ensuring data consistency across multiple devices and the server.
* **Synchronization:** Real-time updates.
* **Security:** Data encryption during synchronization to prevent unauthorized access.

# **System Features**

## **System Feature 1**

***User Account Registration***

4.1.1 Description and Priority:

Allows users to register for an account to access event portal services. Priority: High.

4.1.2 Stimulus/Response Sequences:

Stimulus: User accesses the registration page.

Response: System displays the registration form.

Stimulus: User submits registration information.

Response: System creates a new user account and displays a confirmation message.

4.1.3 Functional Requirements:

REQ-1: Display registration form on the registration page.

REQ-2: Validate user input for registration information.

REQ-3: Create a new user account upon successful registration.

REQ-4: Display a confirmation message for successful registration.

## **System Feature 2**

***Event Browsing and Filtering***

4.3.1 Description and Priority:

Allows users to browse and filter events by category and date. Priority: High.

4.3.2 Stimulus/Response Sequences:

Stimulus: User navigates to the home page.

Response: System displays all events sorted by departments.

Stimulus: User clicks on the Events button.

Response: System filters and displays events sorted by categories.

4.3.3 Functional Requirements:

REQ-1: Display all events sorted by departments on the home page.

REQ-2: Filter events by category when the user clicks on the Events button.

REQ-3: Sort events by date within each category.

## System Feature 3

***User Account Login***

4.2.1 Description and Priority:

Allows registered users to log in to their accounts. Priority: High.

4.2.2 Stimulus/Response Sequences:

Stimulus: User accesses the login page.

Response: System displays the login form.

Stimulus: User submits valid credentials.

Response: System redirects the user to the home page.

4.2.3 Functional Requirements:

REQ-1: Display login form on the login page.

REQ-2: Validate user credentials during login.

REQ-3: Redirect user to the home page upon successful login.

## System Feature 4

***Event Browsing and Filtering***

4.3.1 Description and Priority:

Allows users to browse and filter events by category and date. Priority: High.

4.3.2 Stimulus/Response Sequences:

Stimulus: User navigates to the home page.

Response: System displays all events sorted by departments.

Stimulus: User clicks on the Events button.

Response: System filters and displays events sorted by categories.

4.3.3 Functional Requirements:

REQ-1: Display all events sorted by departments on the home page.

REQ-2: Filter events by category when the user clicks on the Events button.

REQ-3: Sort events by date within each category.

## System Feature 5

***Event Registration***

4.5.1 Description and Priority:

Allows users to register for events they are interested in attending. Priority: High.

4.5.2 Stimulus/Response Sequences:

Stimulus: User clicks on the registration button for an event.

Response: System redirects the user to a registration page.

4.5.3 Functional Requirements:

REQ-1: Provide a clear registration procedure at the end of the event details page.

REQ-2: Redirect the user to a registration page upon clicking the registration button.

## System Feature 6

***Event Details***

4.4.1 Description and Priority:

Provides detailed information about events to users. Priority: High.

4.4.2 Stimulus/Response Sequences:

Stimulus: User clicks on an event.

Response: System displays detailed information about the selected event.

4.4.3 Functional Requirements:

REQ-1: Display detailed information including description, location, date, time, and speaker info.

REQ-2: Provide a link or button for users to register for the event if interested.

## System Feature 7

***Viewing Registered Events***

4.7.1 Description and Priority:

Allows users to view events they have already registered for. Priority: High.

4.7.2 Stimulus/Response Sequences:

Stimulus: User navigates to the registered events section.

Response: System displays a list of events the user has registered for.

4.7.3 Functional Requirements:

REQ-1: Provide a section for users to view their registered events.

REQ-2: Display detailed information about each registered event.

## System Feature 8

Closing Event Registration

4.8.1 Description and Priority:

Allows administrators to close event registration when maximum capacity is reached. Priority: High.

4.8.2 Stimulus/Response Sequences:

Stimulus: Administrator clicks on the close registration button for an event.

Response: System closes registration for the event and notifies the administrator.

4.8.3 Functional Requirements:

REQ-1: Display an option for administrators to close event registration.

REQ-2: Notify administrators when registration for an event is closed.

## System Feature 9

***Collecting Attendee Feedback***

4.9.1 Description and Priority:

Enables event organizers to collect feedback from attendees during and after events. Priority: High.

4.9.2 Stimulus/Response Sequences:

Stimulus: Event organizer activates the feedback survey feature.

Response: System sends real-time notifications to attendees prompting them to provide

feedback.

4.9.3 Functional Requirements:

REQ-1: Provide an option for event organizers to activate the feedback survey feature.

REQ-2: Send real-time notifications to attendees during events to collect feedback.

REQ-3: Store feedback data for analysis in a visual format.

## System Feature 10

***Providing Event Feedback***

4.10.1 Description and Priority:

Allows attendees to provide feedback on events they have attended. Priority: High.

4.10.2 Stimulus/Response Sequences:

Stimulus: Attendee receives a post-event feedback notification.

Response: System presents the attendee with a feedback form.

4.10.3 Functional Requirements:

REQ-1: Send post-event feedback notifications to attendees.

REQ-2: Present attendees with a feedback form containing rating options and space for comments.

REQ-3: Store attendee feedback for analysis by event organizers.

## System Feature 11

***Event Reminder Notifications***

4.11.1 Description and Priority:

Provides attendees with timely reminders for registered events. Priority: High.

4.11.2 Stimulus/Response Sequences:

Stimulus: Event date approaches for a registered attendee.

Response: System sends a notification with key event details.

4.11.3 Functional Requirements:

REQ-1: Send reminder notifications to registered attendees as the event date approaches.

REQ-2: Include key event details in the reminder notifications.

REQ-3: Ensure notifications are sent a day before the event.

## System Feature 12

***Automated Reminder Distribution***

4.12.1 Description and Priority:

Enables event organizers to send automated reminders to registered attendees. Priority: High.

4.12.2 Stimulus/Response Sequences:

Stimulus: Event organizer sets up automated reminders for a scheduled event.

Response: System sends automated reminders to registered attendees.

4.12.3 Functional Requirements:

REQ-1: Provide an option for event organizers to set up automated reminders.

REQ-2: Send automated reminders to registered attendees within the specified timeframe.

REQ-3: Provide confirmation or logs indicating successful reminder distribution to organizers.

## System Feature 13

***Bookmarking Events***

4.13.1 Description and Priority:

Allows users to bookmark events for future reference. Priority: Medium.

4.13.2 Stimulus/Response Sequences:

Stimulus: User clicks the Bookmark button next to an event.

Response: System adds the event to the Bookmarked section.

4.13.3 Functional Requirements:

REQ-1: Provide a Bookmark button next to each event.

REQ-2: Add bookmarked events to a designated Bookmarked section.

REQ-3: Ensure the Bookmarked section is easily accessible from the main navigation.

## System Feature 14

***Marking Events of Interest***

4.14.1 Description and Priority:

Allows users to mark events with a star button for detailed exploration. Priority: Medium.

4.14.2 Stimulus/Response Sequences:

Stimulus: User marks an event with a Star button.

Response: System adds the event to the Marked Events section.

4.14.3 Functional Requirements:

REQ-1: Provide a Star button for users to mark events.

REQ-2: Add marked events to the Marked Events section.

REQ-3: Provide the option for users to unmark events by toggling the Star button.

## System Feature 15

***Social Media Sharing***

4.15.1 Description and Priority:

Enables users to share event details on social media platforms. Priority: Medium.

4.15.2 Stimulus/Response Sequences:

Stimulus: User clicks the Share button for an event.

Response: System provides options to share on social media platforms.

4.15.3 Functional Requirements:

REQ-1: Provide a Share button for events.

REQ-2: Offer options to share event details on various social media platforms.

REQ-3: Include event details and a link to the event page in shared content.

## System Feature 16

***Promoting Event Sharing***

4.16.1 Description and Priority:

Facilitates event organizers in promoting events through attendee sharing. Priority: Medium.

4.16.2 Stimulus/Response Sequences:

Stimulus: Organizer utilizes the Promote Event feature.

Response: System generates shareable content optimized for social media.

4.16.3 Functional Requirements:

REQ-1: Provide a Promote Event feature for organizers.

REQ-2: Generate shareable content for social media promotion.

REQ-3: Track the number of shares and engagements for promotional purposes.

## System Feature 17

***Dark Mode***

4.17.1 Description and Priority:

Offers users the option to switch to a dark mode for low-light conditions. Priority: Medium.

4.17.2 Stimulus/Response Sequences:

Stimulus: User enables Dark Mode in the app settings.

Response: System changes the background color to a dark theme and adjusts text/UI elements.

4.17.3 Functional Requirements:

REQ-1: Include an option for users to enable Dark Mode in the app settings.

REQ-2: Change the background color and adjust text/UI elements for optimal visibility in Dark Mode.

## System Feature 18

Event Resource Repository

4.19.1 Description and Priority:

Allows organizers to create a repository of past event materials for attendee access. Priority: Medium.

4.19.2 Stimulus/Response Sequences:

Stimulus: Organizer accesses the event organizer dashboard.

Response: System provides a feature for creating a repository of past event materials.

4.19.3 Functional Requirements:

REQ-1: Include a feature in the event organizer dashboard for creating a resource repository.

REQ-2: Ensure the repository includes presentations, videos, and other relevant resources.

## System Feature 19

***Centralized Resource Library***

4.20.1 Description and Priority:

Offers a centralized resource library for past event materials for continuous learning. Priority: Medium.

4.20.2 Stimulus/Response Sequences:

Stimulus: Organizer accesses the event organizer dashboard.

Response: System provides a feature for creating a repository of past event materials.

4.20.3 Functional Requirements:

REQ-1: Include a feature in the event organizer dashboard for creating a resource repository.

REQ-2: Ensure the repository includes presentations, videos, and other relevant resources.

## System Feature 20

# **Other Nonfunctional Requirements**

## **Performance Requirements**

* The portal should have a maximum response time of 3 seconds for any user interaction.
* The system should be able to handle 1000 users concurrently without any compromise on performance.
* The portal should maintain an uptime of at least 95%, excluding scheduled maintenance breaks.

## **Safety Requirements**

* The data of the user should be safe from any unauthorized access, modification or deletion.
* Personal information of users while registration should be handled while keeping in view the related privacy laws.
* Users should only have access to their respective pages to prevent any unauthorized access of options.

## **Security Requirement**s

* All the communication between the portal and the end user should be encrypted by using secure protocols.
* The portal should be immune to any kind of attacks through measures such as firewalls.
* The system should adhere to industry standard security practices and comply with relevant laws.

## **Software Quality Attributes**

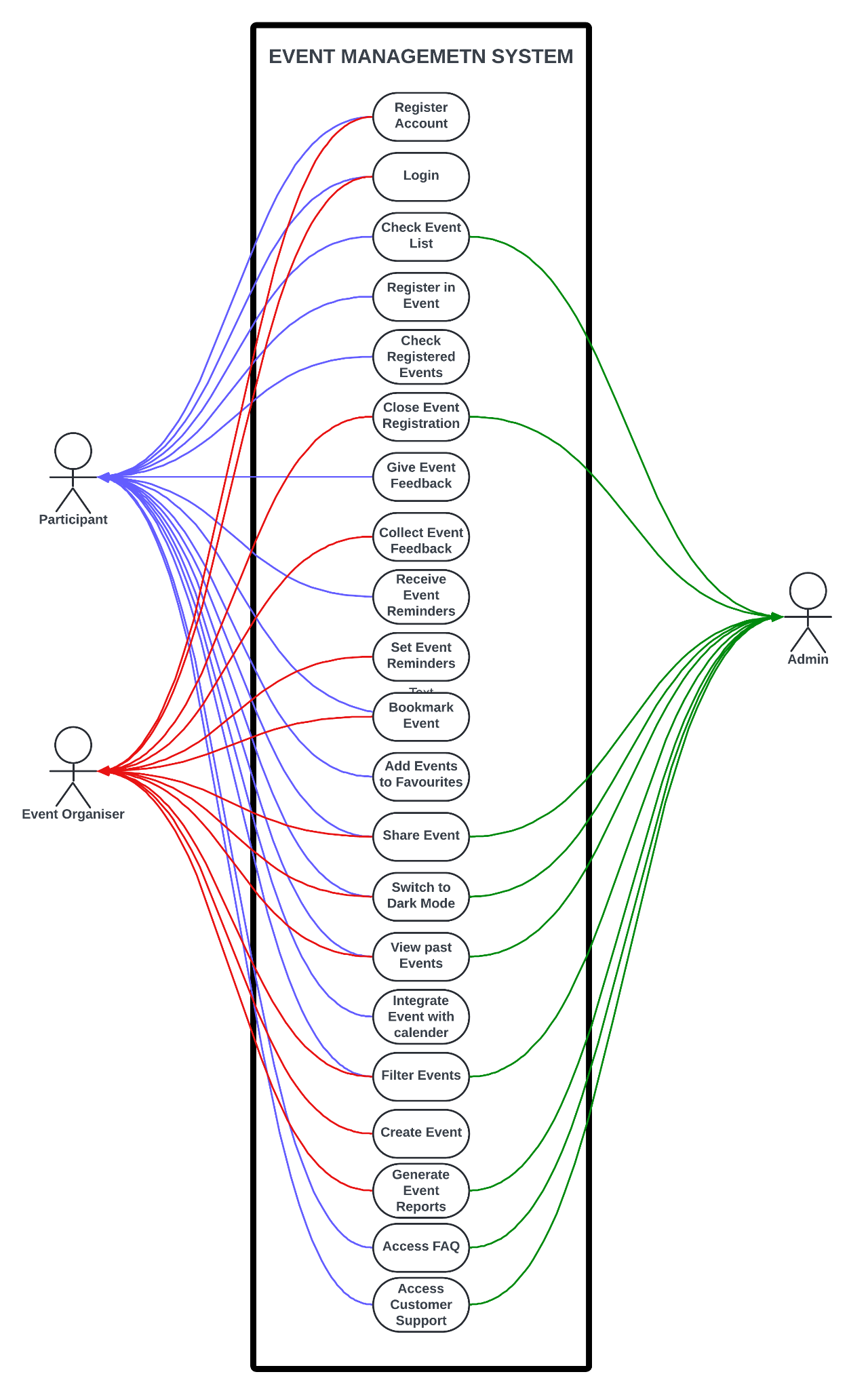
* The portal should have a user-friendly user interface and clear navigation to enhance user satisfaction.
* Codebase should be well documented and modularized to facilitate future updates, maintenance, and enhancements.
* The system should be stable , with low occurrence of errors or crashes.
* The portal should be compatible with various web browsers and devices..

## **Business Rules**

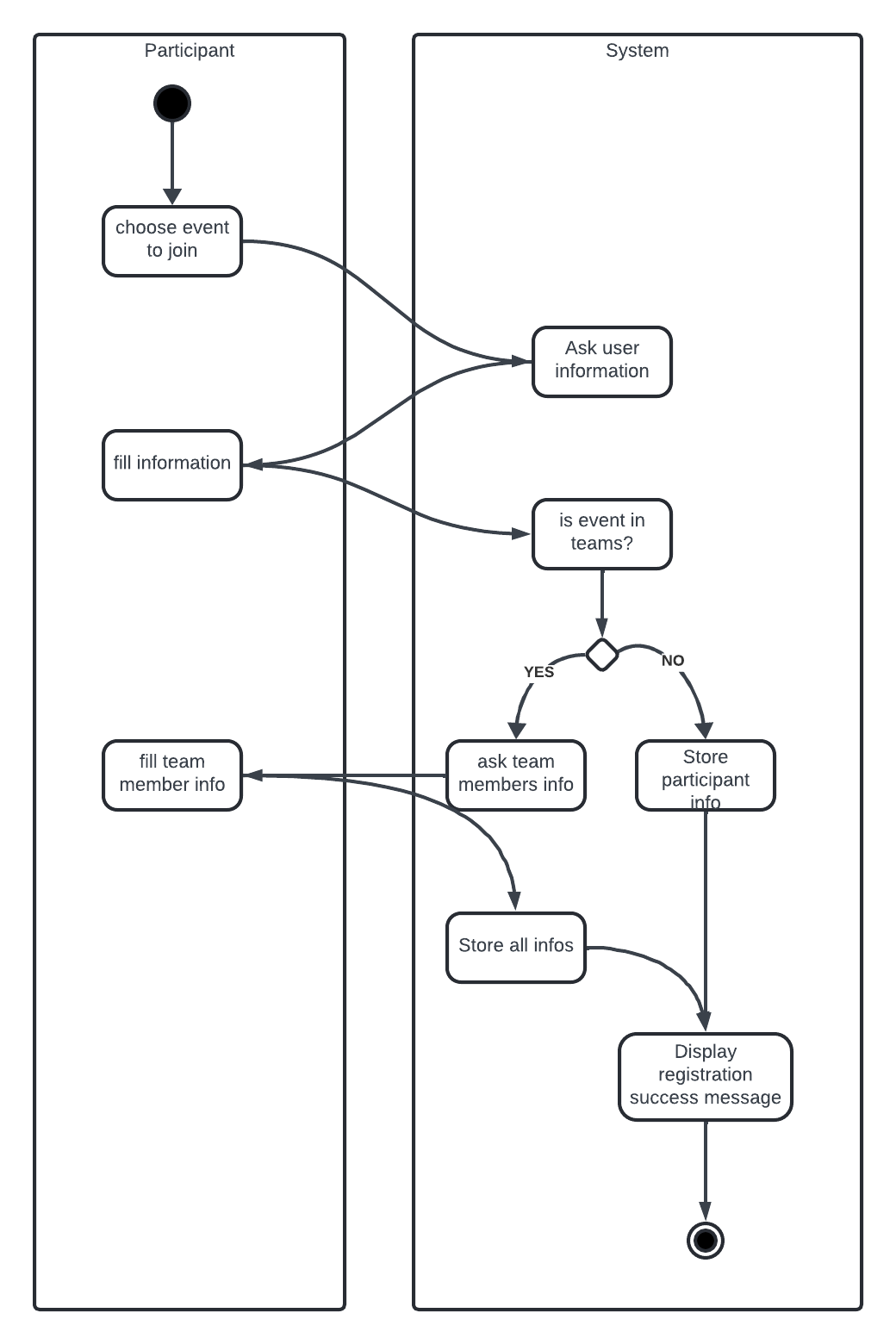
* Events should be authorized by the university before they are posted on the portal.
* A limit should be set for registrations of an event.
* Participants must be given a deadline to submit feedback for timely data collection.

# **Diagrams**

## **Use Case Diagram**

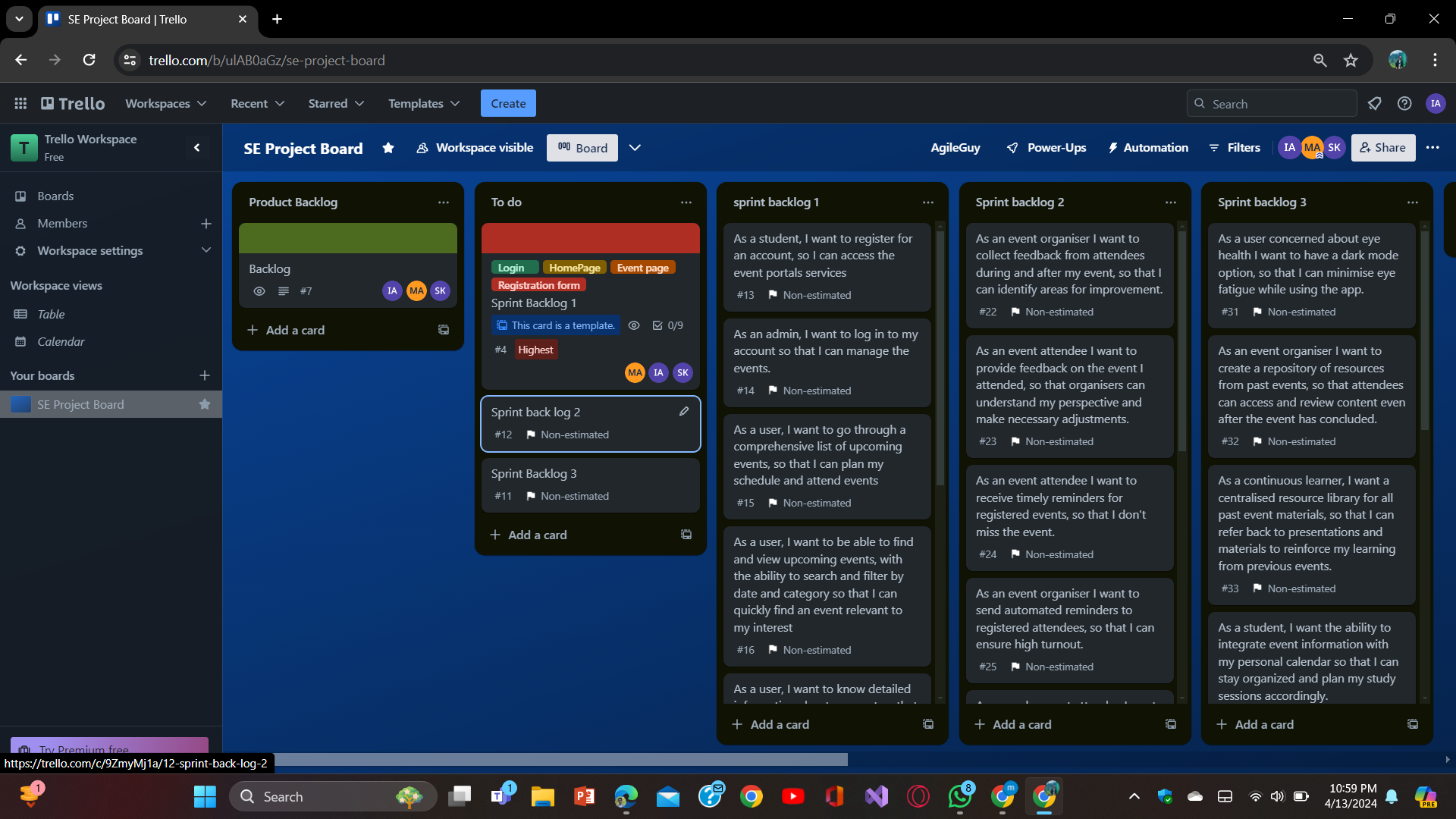


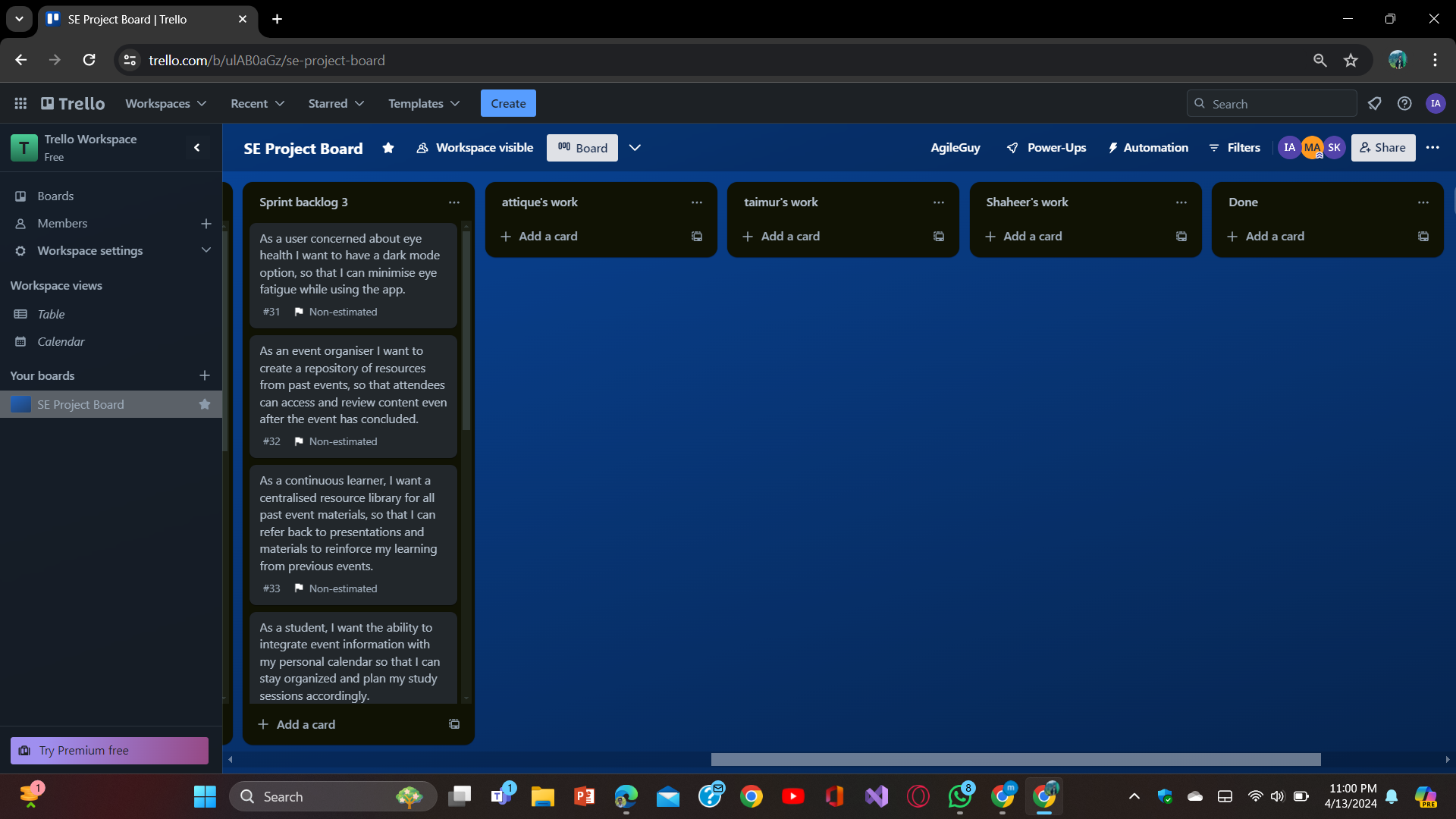
## **Activity Diagram**



## **Sequence Diagram**

# Appendix : Screenshots

***Trello:***

******

***Git:***