Software Requirements Specification

for

Fast Events Portal

**Team Members:**Abdullah Tahir 21i-2960  
Farhan Ahmed 21i-0600  
Ruhail Rizwan 21i-2952  
 **Course:** Software Engineering **Degree Program:** BS (CS) **Section:** E **Date of Submission:** 13th April 2024 **Submitted To:** Ma’am Shafaq Riaz

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2.** **Overall Description 1**

2.1 Product Perspective 2

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 6

2.5 Design and Implementation Constraints 7

2.6 User Documentation 8

2.7 Assumptions and Dependencies 9

**3.** **External Interface Requirements 10**

3.1 User Interfaces 10

3.2 Hardware Interfaces 11

3.3 Software Interfaces 12

3.4 Communications Interfaces 14

**4.** **System Features 16**

4.1 System Feature 1 16

4.2 System Feature 2 (and so on) 17

**5.** **Other Nonfunctional Requirements 18**

5.1 Performance Requirements **18**

5.2 Safety Requirements **18**

5.3 Security Requirements 18

5.4 Software Quality Attributes 18

5.5 Business Rules 18

**6.** **Diagrams 19**

6.1 Use Case Diagram 19

6.2 Activity Diagram 20

6.3 Sequence Diagram 23

**Revision History**

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

# **Introductio**n

**1.1 Purpose**

The software requirements specified in this document pertain to the development of the Fast Events Portal, targeted at addressing the limited visibility and accessibility to university events and activities at FAST University. This document outlines the functional and non-functional requirements necessary for the successful implementation of the portal.

**1.2 Document Conventions**

This SRS follows standard conventions for documenting software requirements. Priorities for higher-level requirements are inherited by detailed requirements. Each requirement statement is assigned its own priority to ensure clarity and facilitate traceability.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for developers, project managers, marketing staff, users, testers, and documentation writers involved in the development and deployment of the Fast Events Portal. It is organized to provide an overview of the project, followed by detailed functional and non-functional requirements. Readers are suggested to begin with the overview sections and proceed through sections pertinent to their roles or interests.

**1.4 Product Scope**

The Fast Events Portal is a user-friendly online platform tailored for FAST University. It aims to enhance student engagement and participation in campus activities by providing easy access to event information and registration. The portal will integrate seamlessly with FAST's existing student authentication system to ensure security and personalized access. Key features include event browsing, registration, personalized recommendations, event creation, and management tools for organizers.

# **Overall Description**

## **Product Perspective**

*The Fast Event’s Portal is envisioned as a new, self-contained product designed specifically for FAST University. It is not a follow-on to any existing product family but is intended to replace and enhance any informal or disjointed methods currently used by the university community to manage and participate in events. This portal aims to centralize and streamline the visibility and accessibility of university events, thereby improving student engagement and participation.*

*The product will seamlessly integrate into the larger ecosystem of FAST University’s existing digital tools. It is designed to interact with the university's student authentication system to ensure secure and personalized access for users. This integration allows the portal to leverage existing databases for user authentication, ensuring that students and faculty can use their university credentials to log in, enhancing both security and user experience.*

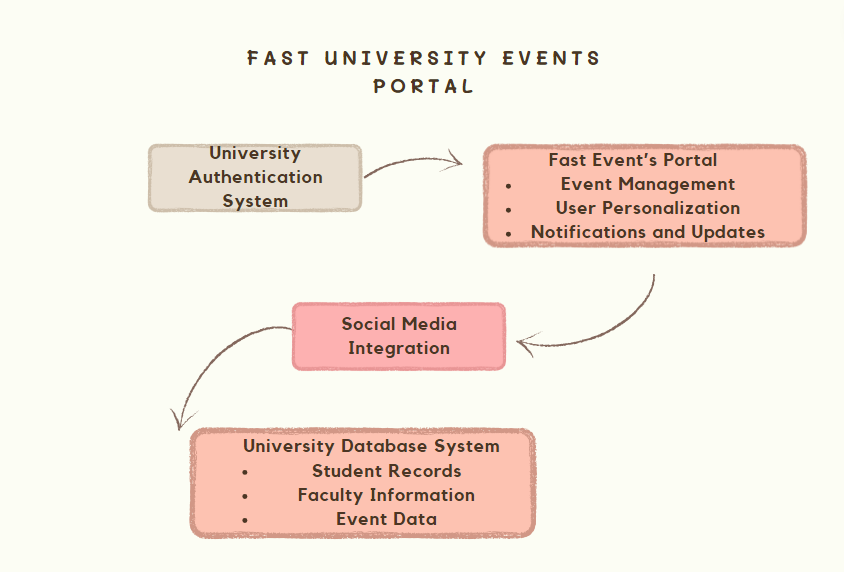
*The Fast Event’s Portal acts as a central hub for all event-related activities within the university. It will have interfaces with:*

*University Authentication System: To authenticate users and ensure secure access.*

*Social Media Platforms: To allow for the sharing of event information and increase engagement.*

*University Database Systems: For retrieving and updating information related to students, faculty, events, and other related data.*

*Here's a simple diagram conceptually illustrating how the Fast Event’s Portal fits within the larger university system:*

**

## **Product Functions**

*The Fast Event’s Portal is designed to offer a comprehensive suite of functions to facilitate effective event management and participation within FAST University. Here is a high-level summary of the major functions that the product will perform:*

***Event Browsing and Search:***

*Allows users to easily search and browse through a list of university events using filters such as date, category, and keywords.*

***Event Creation and Management:***

* *Enables event organizers to create new events, providing necessary details such as event name, date, time, location, and description.*
* *Provides tools for managing event details, updating information, or cancelling events as needed.*

***User Registration and Authentication:***

* *Integrates with FAST University's existing authentication systems to provide secure and personalized user access.*
* *Supports registration processes for new users, allowing them to sign up and participate in events.*

***Event Registration and Participation:***

*Allows users to register for events, manage their registrations, and receive updates and notifications related to the events they are interested in.*

***Notifications and Updates:***

*Sends automated notifications to users about event registrations, reminders for upcoming events, and updates on events they are registered for or interested in.*

***Personalized Recommendations:***

*Provides personalized event recommendations based on users' interests and past participation, enhancing user engagement and satisfaction.*

***Feedback and Interaction:***

*Enables participants to provide feedback on events and interact with event organizers and other participants to share their experiences and suggestions.*

***Social Media Integration:***

*Allows sharing of events on social media platforms to increase visibility and engagement, and integrates social media features directly within the portal.*

***Analytics and Reporting:***

*Offers dashboard functionalities for administrators and event organizers to view analytics related to event attendance, engagement levels, and participant feedback.*

***Calendar Integration:***

*Features a calendar view where users can see all registered and upcoming events, allowing them to plan their participation effectively.*

*These functions are organized to ensure that the portal is user-friendly and effective in engaging both students and faculty in campus life, fostering a vibrant community through enhanced participation in university events. Further details about these functionalities will be discussed in Section 3 of the SRS, focusing on specific requirements and operational details.*

## **User Classes and Characteristics**

*The Fast Event’s Portal is designed to cater to various user classes within the FAST University community, each with distinct characteristics and needs. Below is a detailed identification of these user classes along with their pertinent characteristics:*

***1. Students***

* *Frequency of Use: High, as students will regularly check for and participate in events.*
* *Functions Used: Browsing and searching events, registering for events, providing feedback, and receiving personalized recommendations.*
* *Technical Expertise: Moderate to high; familiar with digital platforms but varying levels of proficiency.*
* *Educational Level: Undergraduate and postgraduate students.*
* *Experience: Users are generally young adults who are tech-savvy and actively engage with social media and digital platforms.*
* *Security/Privilege Levels: General user access with permissions to view and register for events, manage their profiles, and provide feedback.*

***2. Faculty and Staff***

* *Frequency of Use: Moderate; use the portal for creating, managing, and overseeing events.*
* *Functions Used: Event management, analytics viewing, and feedback collection.*
* *Technical Expertise: Moderate; familiar with administrative and academic systems.*
* *Educational Level: Higher educational qualifications, typically with advanced degrees.*
* *Experience: Experienced users who require more robust functionalities for event oversight and analytics.*
* *Security/Privilege Levels: Elevated access to include event creation, editing, and analytics functionalities.*

***3. Event Organizers***

* *Frequency of Use: High; actively involved in listing and managing events.*
* *Functions Used: Full suite of event management tools, including creating, updating, and canceling events, as well as accessing reports and analytics.*
* *Technical Expertise: High; often designated personnel skilled in managing digital platforms.*
* *Educational Level: Varied, depending on their role within the university.*
* *Experience: Users with specific roles relating to event planning and management, requiring detailed interaction with the platform.*
* *Security/Privilege Levels: Similar to faculty but focused specifically on event-related functionalities.*

***4. Administrators***

* *Frequency of Use: Moderate to high; monitor and manage the overall use of the portal.*
* *Functions Used: Oversight of all portal activities, user management, security settings, approval and moderation of content.*
* *Technical Expertise: High; deep understanding of both the technical and administrative aspects of the platform.*
* *Educational Level: Typically higher educational qualifications with a background in administration or IT.*
* *Experience: Experienced professionals adept at managing information systems and ensuring compliance with university policies.*
* *Security/Privilege Levels: Highest, with access to all system settings, user accounts, and confidential reports.*

***Importance Hierarchy***

* *Primary Users: Students, as the main beneficiary of the portal, represent the most critical user class, with the system primarily designed to enhance their campus experience.*
* *Secondary Users: Faculty and event organizers, who play a crucial role in providing content and managing the events.*
* *Tertiary Users: Administrators, essential for the oversight and smooth operation of the portal but less frequently engaged with its day-to-day functionalities.*

*This classification ensures that the portal meets the diverse needs of its user base, providing tailored functionalities that enhance usability and engagement across all user classes.*

## **Operating Environment**

*The Fast Event’s Portal is designed to operate within a robust and flexible technological environment to ensure accessibility and reliability for all users. Below is a detailed description of the operating environment for the portal:*

***Hardware Platform:***

* *Client-side: The portal is accessible on various devices including desktop computers, laptops, tablets, and smartphones. This ensures users can access the portal both on-campus and remotely.*
* *Server-side: The backend of the portal runs on scalable server infrastructure capable of supporting a large number of concurrent users typical of a university setting. This might include dedicated servers or cloud-based services to provide flexibility and scalability.*

***Operating Systems:***

* *Client-side: The portal is platform-agnostic on the client side, supporting major operating systems such as Windows, macOS, iOS, and Android. This ensures a broad accessibility spectrum for users across different devices.*
* *Server-side: The server-side components are designed to run on Linux-based servers, which are known for their stability and scalability. This choice supports a variety of enterprise-level backend solutions.*

***Software Components:***

* *Web Servers: Uses Apache or Nginx as the web server, known for their robust performance and compatibility with various applications.*
* *Database: Utilizes MySQL or PostgreSQL to manage data storage, ensuring efficient data retrieval and security for user and event data.*
* *Backend Frameworks: Employs frameworks such as Node.js or Django to manage the server-side logic, providing a stable and efficient environment for application operations.*

***Coexisting Software:***

* *University Authentication System: The portal integrates with FAST University’s existing authentication systems, which may include LDAP or Active Directory, to manage user logins and security.*
* *Email Systems: Integrates with email servers for sending notifications and confirmations to users, requiring compatibility with SMTP protocols.*
* *Social Media Platforms: Works alongside social media APIs (like Facebook, Twitter) for event sharing and engagement features.*
* *Academic and Administrative Software: Needs to coexist with other university systems used for academic and administrative purposes, such as learning management systems (LMS) and student information systems (SIS).*

***Network Requirements:***

* *Requires a reliable internet connection with adequate bandwidth to support high user traffic and data transfers, especially during peak event registration periods.*

***Security Considerations:***

* *Must comply with university IT security policies and best practices, including secure connections (HTTPS), data encryption, and regular security audits to protect sensitive user information.*

## **Design and Implementation Constraints**

*The development of the Fast Event’s Portal must adhere to a set of constraints that limit certain design and implementation choices. These constraints are derived from a combination of university policies, technical requirements, and industry standards, which include:*

***Corporate and Regulatory Policies:***

* *Data Privacy and Protection: Must comply with university privacy policies and potentially applicable regulations such as GDPR or FERPA, which govern the handling of student and faculty information.*
* *Accessibility Standards: The portal must be accessible according to the Web Content Accessibility Guidelines (WCAG) to ensure it is usable by people with disabilities.*

***Hardware Limitations:***

* *Scalability: The system should be scalable to accommodate the growing number of users and data over time without degradation of performance.*
* *Memory and Processing Requirements: The server hardware must be capable of handling high processing loads, especially during peak usage times such as event sign-ups or deadlines.*

***Interfaces to Other Applications:***

* *University Systems Integration: The portal must integrate seamlessly with existing university systems like the authentication servers, student information systems (SIS), and learning management systems (LMS).*
* *Third-party Services: Integration with third-party services such as email servers for notifications and social media platforms for sharing events must be managed without disrupting existing functionalities.*

***Specific Technologies, Tools, and Databases:***

* *Programming Languages: The development of the portal will predominantly use C# as the main programming language, leveraging its robustness and integration with the .NET framework for both front-end and backend services.*
* *Databases: MySQL or PostgreSQL is preferred for their robustness and widespread use within the academic community. However, considering the use of the .NET framework, Microsoft SQL Server could also be a viable option due to its seamless integration with .NET technologies.*
* *Development Frameworks: The application will be built on the .NET framework, which supports rapid development and has strong community support. The .NET framework provides a comprehensive development environment that offers tools and libraries for building applications efficiently, including web services, database connectivity, and user interface components.*

***Parallel Operations:***

* *The system must support concurrent access by multiple users without loss of data integrity or performance issues. This requires careful management of database transactions and user session management.*

***Language Requirements:***

* *The portal must support multilingual capabilities to accommodate international students or faculty, starting with English and potentially expanding to other languages based on university demographics.*

***Communications Protocols:***

* *Must use secure communication protocols such as HTTPS for all data exchanges over the network to protect data integrity and confidentiality.*

***Security Considerations:***

* *Implementation of robust security measures including, but not limited to, data encryption, secure authentication mechanisms, and regular security audits.*
* *The portal must enforce strong password policies and use two-factor authentication (2FA) where feasible.*

***Design Conventions or Programming Standards:***

* *Adhere to industry best practices and programming standards to ensure high-quality, maintainable code.*
* *The university’s IT department may be responsible for ongoing maintenance, so the codebase must be well-documented and easy to understand.*

## **User Documentation**

*To ensure that all users can effectively utilize the Fast Event’s Portal, comprehensive user documentation will be provided. This documentation will help users navigate the portal, understand its features, and troubleshoot common issues. The following are the main components of the user documentation that will be delivered along with the software:*

***User Manuals:***

* *General User Manual: A detailed guide covering all features available to students and general users, including how to search for events, register for events, and manage personal profiles.*
* *Organizer Manual: Specific documentation for event organizers and faculty detailing how to create, manage, and update events, along with instructions on using advanced features like analytics and reporting tools.*
* *Administrator Manual: A comprehensive manual for administrators on managing user roles, overseeing portal activity, and performing system configurations and updates.*

***Online Help:***

* *An integrated help system within the portal that provides immediate, context-sensitive assistance to users. This feature will offer quick answers to common questions and step-by-step guides for completing various tasks within the application.*
* *FAQ Sections: Regularly updated FAQs to address the most common challenges and queries that users might encounter.*

***Tutorials:***

* *Interactive Tutorials: Step-by-step interactive guides that users can follow to familiarize themselves with different functionalities of the portal. These tutorials will be particularly useful for new users to get acquainted with the system.*
* *Video Tutorials: Short, engaging instructional videos available within the portal and possibly on external platforms like YouTube, demonstrating key processes such as registering for an event, setting up an event, or configuring user settings.*

***Quick Start Guides:***

* *Concise documents aimed at getting users up and running with the basic functionalities of the portal quickly. These guides will be beneficial during initial rollouts or orientations.*

***Documentation Delivery Formats and Standards:***

* ***PDF Documents:*** *Most of the manuals and quick start guides will be available in PDF format for easy download and offline access.*
* ***HTML Format:*** *Online help and tutorials will be available in HTML format directly accessible within the portal.*
* ***Accessible Formats:*** *All documentation will also be available in accessible formats like ePub or equivalent, which are compliant with accessibility standards to ensure all users, including those with disabilities, can utilize them.*
* ***Multimedia Formats:*** *Video tutorials will be delivered in widely supported video formats (e.g., MP4) to ensure compatibility across different devices and platforms.*

## **Assumptions and Dependencies**

*In the development of the Fast Event’s Portal, several assumptions and dependencies play crucial roles in shaping the requirements and expectations of the software system. Understanding these factors is essential for accurate planning and effective implementation.*

***Assumptions***

* *User Adoption Rates: It is assumed that there will be a high rate of adoption among the students and faculty at FAST University, which influences the design to handle large volumes of traffic and interactions.*
* *Stable Internet Connectivity: The portal assumes reliable and continuous internet connectivity for users to access all its features without disruptions, particularly when hosting or participating in events.*
* *Integration Compatibility: It is assumed that the existing university systems, such as the student authentication system and database servers, are compatible with the technologies used in the portal (e.g., .NET framework, C#) and can be integrated without significant modifications.*
* *User Technical Proficiency: The design assumes a moderate level of technical proficiency among users, which affects the complexity of the user interface and the extent of the documentation provided.*
* *Continued Support and Maintenance: The project assumes that the university’s IT department will provide ongoing support and maintenance after the deployment of the portal.*

***Dependencies***

* *Third-Party Software Components: The project is dependent on specific third-party software components, such as the .NET framework and potentially third-party libraries for additional functionalities like charting or social media integration.*
* *Database Systems: The portal’s operation is heavily reliant on the chosen database systems (e.g., Microsoft SQL Server) for all data storage and retrieval operations. This includes dependencies on their robustness, scalability, and security features.*
* *Email Systems: For sending notifications and confirmations, the portal depends on integration with an email system capable of handling bulk email distribution and that is reliable for communications.*
* *Software Reuse: Components from other university systems, particularly those involved in user authentication and data management, are expected to be reused in this project. This includes libraries and code modules that handle secure login and session management.*
* *Regulatory Compliance Software: Compliance with data protection regulations (such as GDPR or FERPA) may require the use of specific compliance software or modules, particularly for data encryption and user privacy management.*

***Potential Impacts if Assumptions and Dependencies Change***

* *Changes in user adoption rates could impact system performance and scalability requirements.*
* *Any disruptions in internet connectivity or lower than expected connectivity standards could severely affect the user experience.*
* *Incompatibility with existing university systems would require additional development work, potentially leading to delays and increased costs.*
* *Underestimation of user technical proficiency might result in the need for enhanced user training and support services.*
* *Reduction in support and maintenance capabilities from the university's IT department could affect the long-term viability and security of the portal.*

# **External Interface Requirements**

## **User Interfaces**

### User Interfaces:

* **Web Interface for Students, Faculty, and Staff:**
  + **Logical Characteristics:** The web interface will provide an intuitive and user-friendly experience accessible via desktop and mobile devices.
  + **Sample Screen Images:** Screenshots/mockups of the main dashboard, event browsing/searching interface, event creation/editing interface, user profile management, and feedback submission.
  + **GUI Standards**: Follows modern web design principles with clear navigation menus, consistent layout, and visually appealing elements.
  + **Screen Layout Constraints:** Responsive design to adapt to various screen sizes, ensuring usability on both desktop and mobile devices.
  + **Standard Buttons and Functions:** Common buttons for actions such as 'Register for Event', 'Create Event', 'Update Profile', 'Submit Feedback', and 'Log Out'.
  + **Keyboard Shortcuts:** Minimal keyboard shortcuts to enhance accessibility and efficiency, such as keyboard navigation for menus and form fields.
  + **Error Message Display Standards:** Clear and concise error messages displayed inline or in modal dialogs, indicating the nature of the error and possible solutions.
* **Admin Dashboard:**
  + **Logical Characteristics:** A separate interface for administrators to manage system settings, user accounts, and oversee portal activities.
  + **Sample Screen Images**: Screenshots/mockups of the admin dashboard displaying analytics, user management tools, security settings, and content moderation features.
  + **GUI Standards:** Follows a consistent design language with the user interface for general users, but with additional administrative functionalities.
  + **Standard Buttons and Functions:** Admin-specific actions such as 'Manage Users', 'View Analytics', 'Configure Security Settings', and 'Approve/Reject Events'.
  + **Error Message Display Standards:** Similar to the user interface, with clear error messages for any admin-specific actions or configurations.
* **Organizer Interface:**
  + **Logical Characteristics:** An interface tailored for event organizers to manage event listings, registrations, and analytics.
  + **Sample Screen Images:** Screenshots/mockups of the organizer dashboard showing event creation/editing tools, attendee management features, and reporting functionalities.
  + **GUI Standards:** Shares common design elements with the general user interface but with specialized tools for event management.
  + **Standard Buttons and Functions:** Organizer-specific actions such as 'Create Event', 'Manage Registrations', 'Generate Reports', and 'Communicate with Attendees'.
  + **Error Message Display Standards:** Consistent with other interfaces, providing clear guidance for any errors encountered during event management tasks.
* **Help and Support Interface:**
  + **Logical Characteristics:** Provides users with access to help resources, FAQs, and support channels within the portal.
  + **Sample Screen Images:** Screenshots/mockups of the help center interface featuring FAQs, user manuals, tutorials, and contact information for support.
  + **GUI Standards**: Easily accessible from any page within the portal, with clear navigation and search functionalities.
  + **Standard Buttons and Functions:** Buttons or links to access help resources, submit support tickets, or contact support staff directly.
  + **Error Message Display Standards:** If applicable, error messages related to support inquiries or ticket submissions should be informative and guide users on next steps.

## **Hardware Interfaces**

### *Hardware Interfaces:*

* ***Client-Side Devices:***
  + *Logical Characteristics: The software product interfaces with various client-side devices such as desktop computers, laptops, tablets, and smartphones.*
  + *Physical Characteristics: Supports different hardware configurations, screen sizes, and input methods (e.g., touchscreens, keyboards, mice).*
  + *Data and Control Interactions: The software interacts with client-side hardware to display the user interface, receive user inputs (e.g., clicks, taps, keyboard input), and render multimedia content.*
  + *Communication Protocols: Utilizes standard communication protocols over different hardware interfaces, including USB, Wi-Fi, Bluetooth, and cellular data connections.*
* ***Server-Side Hardware:***
  + *Logical Characteristics: The software product interacts with server-side hardware responsible for hosting and running the backend components of the portal.*
  + *Physical Characteristics: Server hardware includes scalable infrastructure capable of handling a large number of concurrent users and data processing tasks.*
  + *Data and Control Interactions: The software communicates with server-side hardware to request and process data, execute business logic, and perform database operations.*
  + *Communication Protocols: Utilizes network protocols such as TCP/IP for communication between client-side devices and server-side hardware, ensuring reliable data transmission over the internet.*
* ***Networking Equipment:***
  + *Logical Characteristics: The software product relies on networking equipment such as routers, switches, and firewalls to facilitate communication between client-side devices and server-side hardware.*
  + *Physical Characteristics: Networking equipment may vary in terms of capacity, speed, and security features depending on the network infrastructure.*
  + *Data and Control Interactions: The software interacts with networking equipment to establish connections, transmit data packets, and ensure network security and integrity.*
  + *Communication Protocols: Utilizes standard networking protocols such as TCP/IP, HTTP/HTTPS, and DNS for communication between different network components, ensuring interoperability and security.*
* ***Cloud-Based Services:***
  + *Logical Characteristics: The software product may utilize cloud-based services for hosting server-side components, storing data, and performing computational tasks.*
  + *Physical Characteristics: Cloud-based services are hosted on remote data centers operated by cloud service providers, offering scalability, reliability, and accessibility.*
  + *Data and Control Interactions: The software interacts with cloud-based services through APIs and web services to access resources, execute functions, and store/retrieve data.*
  + *Communication Protocols: Utilizes standard internet protocols such as HTTP/HTTPS, RESTful APIs, and OAuth for communication with cloud-based services, ensuring secure and efficient data exchange.*

## **Software Interfaces**

### *Software Interfaces:*

* ***Database Management System (DBMS):***
  + *Name and Version: MySQL or PostgreSQL (specific version to be determined during implementation).*
  + *Data Items:*
    - *Incoming Data: Event details (name, date, time, location, description), user profiles, registration information, feedback, analytics data.*
    - *Outgoing Data: Query results, updated event information, user authentication status.*
  + *Purpose: The portal interacts with the DBMS to store and retrieve various data related to events, user profiles, registrations, and analytics. It uses SQL queries to insert, update, delete, and retrieve data from the database.*
* ***Operating Systems:***
  + *Supported Platforms: Windows, macOS, Linux.*
  + *Data Items: N/A.*
  + *Purpose: The software product must be compatible with multiple operating systems to ensure accessibility for users on different devices. It utilizes standard APIs and system calls provided by the respective operating systems for tasks such as file I/O, networking, and process management.*
* ***Web Servers:***
  + *Name and Version: Apache or Nginx (specific version to be determined during implementation).*
  + *Data Items: HTTP requests, responses, and session data.*
  + *Purpose: The portal runs on a web server to serve web pages, handle HTTP requests from client-side devices, and manage user sessions. It utilizes server-side scripting (e.g., PHP, Python, Node.js) to generate dynamic content and interact with backend components.*
* ***Backend Frameworks:***
  + *Name and Version: Node.js or Django (specific version to be determined during implementation).*
  + *Data Items: Application logic, business rules, and data processing tasks.*
  + *Purpose: The software product uses backend frameworks to manage server-side logic, handle user requests, and interact with the database. It defines routes, controllers, and middleware to process incoming requests, execute business logic, and generate appropriate responses.*
* ***University Authentication System:***
  + *Name and Version: LDAP or Active Directory (specific version to be determined by the university's IT department).*
  + *Data Items: User credentials, authentication tokens.*
  + *Purpose: The portal integrates with the university's authentication system to verify user identities and ensure secure access to the portal. It utilizes authentication protocols (e.g., LDAP, Kerberos) to validate user credentials and generate authentication tokens for session management.*
* ***Email Systems:***
  + *Name and Version: SMTP server (specific version to be determined by the university's IT department).*
  + *Data Items: Email messages, recipient addresses, message metadata.*
  + *Purpose: The portal integrates with an SMTP server to send automated notifications, reminders, and confirmations to users. It uses SMTP protocols for message transmission and may utilize email templates for consistent messaging.*
* ***Social Media Platforms:***
  + *Name and Version: Facebook Graph API, Twitter API (specific versions to be determined by the respective platforms).*
  + *Data Items: Event details, user interactions (e.g., shares, likes, comments).*
  + *Purpose: The portal integrates with social media platforms to allow users to share event information, increase visibility, and engage with social media features. It utilizes APIs provided by social media platforms for authentication, posting content, and retrieving user interactions.*
* ***Integrated Development Environment (IDE) and Tools:***
  + *Name and Version: Visual Studio, JetBrains Rider, Sublime Text, etc.*
  + *Data Items: Source code files, project configurations, debugging information.*
  + *Purpose: Developers use IDEs and tools to write, debug, and maintain the software product's source code. They utilize features such as code editors, version control integration, and debugging tools to streamline the development process.*

## **Communications Interfaces**

* ***HTTP/HTTPS Protocol:***
  + ***Functionality****: Enables communication between client-side devices (web browsers) and the web server hosting the Fast Event Portal.*
  + ***Message Formatting:*** *Utilizes HTTP methods (GET, POST, PUT, DELETE) for sending requests and receiving responses. Data may be transmitted in JSON or XML format.*
  + ***Communication Standards:*** *Adheres to HTTP/1.1 or HTTP/2 standards for secure (HTTPS) and efficient data transmission.*
  + ***Security and Encryption:*** *Utilizes TLS encryption for HTTPS connections to ensure data confidentiality and integrity.*
  + ***Data Transfer Rates:*** *Data transfer rates depend on factors such as network bandwidth, server capacity, and client-side device performance.*
  + ***Synchronization Mechanisms****: Relies on asynchronous communication between client-side devices and the server, with client requests triggering server-side actions and responses delivered asynchronously.*
* ***SMTP Protocol:***
  + ***Functionality****: Facilitates communication between the Fast Event Portal and email servers for sending automated notifications and confirmations to users.*
  + ***Message Formatting****: Utilizes standard email message formats (e.g., MIME) for composing and transmitting email messages.*
  + ***Communication Standards:*** *Adheres to the SMTP protocol for email transmission, utilizing SMTP commands for message delivery and SMTP extensions for enhanced functionality.*
  + ***Security and Encryption:*** *Supports SMTP over TLS (SMTPS) for secure email transmission, ensuring data confidentiality during message delivery.*
  + ***Data Transfer Rates:*** *Data transfer rates depend on factors such as email server capacity, network latency, and recipient email server responsiveness.*
  + ***Synchronization Mechanisms:*** *Relies on synchronous communication between the Fast Event Portal and email servers, with email messages queued for delivery and delivery status updates received in real-time.*
* ***API Communication:***
  + ***Functionality****: Enables communication between the Fast Event Portal and external APIs (e.g., social media platforms, university authentication systems) for data exchange and integration.*
  + ***Message Formatting:*** *Utilizes JSON or XML formats for sending and receiving data payloads in API requests and responses.*
  + ***Communication Standards:*** *Adheres to the standards specified by each API provider, following API documentation and guidelines for authentication, endpoint usage, and data formatting.*
  + ***Security and Encryption:*** *Implements API authentication mechanisms (e.g., OAuth, API keys) for secure access to external services and may utilize HTTPS for encrypted data transmission.*
  + ***Data Transfer Rates:*** *Data transfer rates depend on factors such as API response times, network latency, and API rate limits imposed by external service providers.*
  + ***Synchronization Mechanisms:*** *Relies on synchronous or asynchronous communication patterns depending on the nature of API interactions, with requests triggering immediate responses or queued for processing asynchronously.*
* ***Database Communication:***
  + ***Functionality****: Facilitates communication between the Fast Event Portal and the database management system (DBMS) for data storage and retrieval.*
  + ***Message Formatting:*** *Utilizes SQL queries for sending database commands and retrieving query results.*
  + ***Communication Standards:*** *Adheres to SQL standards for database interaction, including SQL syntax for query construction and execution.*
  + ***Security and Encryption:*** *Implements database authentication mechanisms (e.g., username/password authentication) and may utilize database encryption features for data security.*
  + ***Data Transfer Rates:*** *Data transfer rates depend on factors such as database server performance, network latency, and query complexity.*
  + ***Synchronization Mechanisms:*** *Relies on synchronous communication between the Fast Event Portal and the DBMS, with SQL queries executed in real-time and query results returned promptly.*

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature: Event Browsing and Searching

4.1.1 Description and Priority

*This feature allows users to browse and search for university events using various criteria like keywords, dates, and categories. Priority: High*

* *Benefit: 9 (Critical for user engagement)*
* *Penalty: 9 (Lack of this feature could significantly reduce usability)*
* *Cost: 5 (Moderate complexity)*
* *Risk: 3 (Low risk of technical issues)*

4.1.2 Stimulus/Response Sequences

* *Stimulus: User accesses the portal and inputs search criteria.*
* *Response: The system displays a list of events matching the criteria.*

4.1.3 Functional Requirements

* *REQ-1: The system must allow filtering of events by date, category, and keywords.*
* *REQ-2: The system must provide instant feedback when no events match the search criteria, suggesting alternative search terms.*
* *REQ-3: Events must be sortable by date, relevance, and popularity.*

## System Feature: User Registration and Authentication

4.2.1 Description and Priority

Enables users to register for an account using their university email and authenticate to access personalized features. Priority: High

* Benefit: 8 (Essential for personalized experience)
* Penalty: 7 (Without it, personalization and security are compromised)
* Cost: 4 (Integration with existing systems may be challenging)
* Risk: 5 (Medium risk related to data privacy)

4.2.2 Stimulus/Response Sequences

* Stimulus: User chooses to register.
* Response: The system prompts the user to enter their details and sends a verification email.

4.2.3 Functional Requirements

* REQ-4: The system must verify the email address to ensure it is unique and linked to the university domain.
* REQ-5: The system must securely store user credentials and personal data.
* REQ-6: The system should allow users to recover lost passwords through a secure process.

**4.3 System Feature: Event Management**

4.3.1 Description and Priority

Allows event organizers to create, update, and delete event listings. Priority: High

* Benefit: 9 (Direct impact on the portal's functionality)
* Penalty: 8 (Inability to manage events would render the portal ineffective)
* Cost: 6 (Requires robust backend features)
* Risk: 4 (Moderate risk of misuse or errors in event management)

4.3.2 Stimulus/Response Sequences

* Stimulus: Organizer logs in and opts to manage events.
* Response: The system provides options to create a new event, edit existing ones, or delete them.

4.3.3 Functional Requirements

* REQ-7: The system must allow organizers to input detailed event information including title, date, time, location, and description.
* REQ-8: The system must ensure that changes to event details are updated in real time and notify registered attendees of significant changes.
* REQ-9: The system must allow deletion of events with automatic notifications to registered attendees.

**4.4 System Feature: Notifications and Updates**

4.4.1 Description and Priority

The system sends notifications and updates about event registrations and changes to users. Priority: Medium

* Benefit: 7 (Keeps users informed and engaged)
* Penalty: 6 (Users may miss important updates)
* Cost: 3 (Relatively simple to implement with existing technology)
* Risk: 2 (Low risk, mostly related to user notification preferences)

4.4.2 Stimulus/Response Sequences

* Stimulus: An event the user is registered for is updated.
* Response: The system sends an email and/or app notification with the updated details.

4.4.3 Functional Requirements

* REQ-10: The system must provide options for users to customize their notification preferences.
* REQ-11: Notifications must be timely and reflect the latest updates or changes.
* REQ-12: The system must handle failed notification deliveries gracefully, with retry mechanisms in place.

This structured approach helps clarify the development needs, prioritize functionalities based on user needs, and ensure a logical flow of user interaction with the system. Each feature is designed to cater to the core functionalities of the portal, ensuring it meets the diverse needs of the university community effectively.

# **Other Nonfunctional Requirements**

**5.1 Performance Requirements**

- The portal should load event listings within 2 seconds under normal server load conditions.

- Registration and login processes should have a response time of less than 1 second.

- The system should handle concurrent user registrations without degradation in performance.

**5.2 Safety Requirements**

- The portal should ensure the confidentiality and integrity of user data, complying with relevant privacy laws and regulations.

- Event organizers must not have access to sensitive user information unless explicitly permitted.

**5.3 Security Requirements**

- User identity authentication should be implemented using FAST Student List.

- The portal must enforce access controls to prevent unauthorized users from modifying event listings or user data.

- All user inputs must be validated to prevent injection attacks and other security vulnerabilities.

**5.4 Software Quality Attributes**

- The portal should prioritize usability and ease of navigation to enhance user experience.

- System availability should be maintained at 99.9% uptime to ensure uninterrupted access to event information.

- Code maintainability and extensibility should be emphasized to facilitate future enhancements and updates.

**5.5 Business Rules**

- Only registered users should be allowed to create, update, or delete event listings.

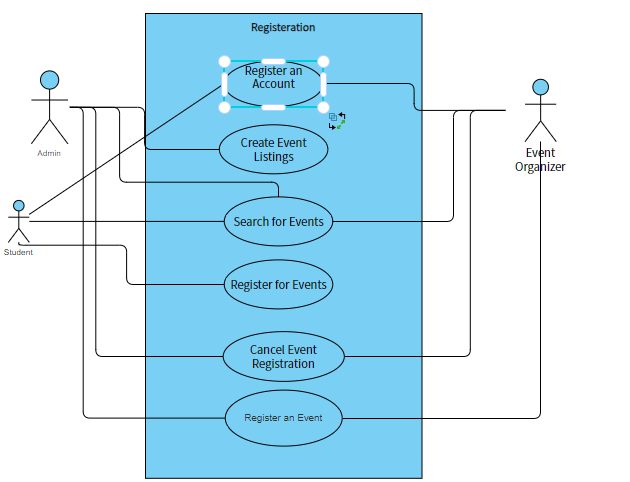
- Event organizers must adhere to community guidelines and university policies when creating event content.

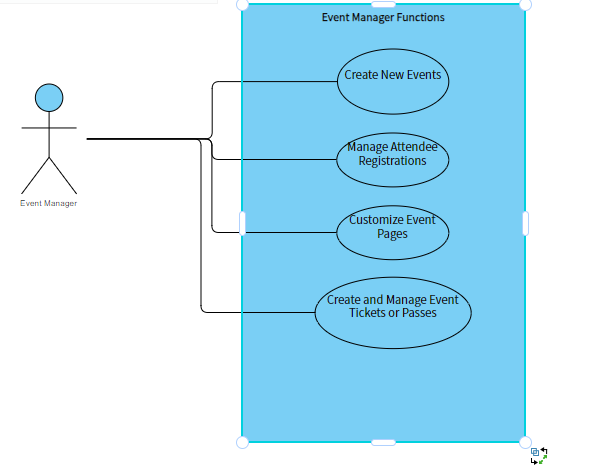
- Events should be categorized based on academic departments, student organizations, or other relevant criteria to facilitate targeted recommendations.

# **Diagrams**

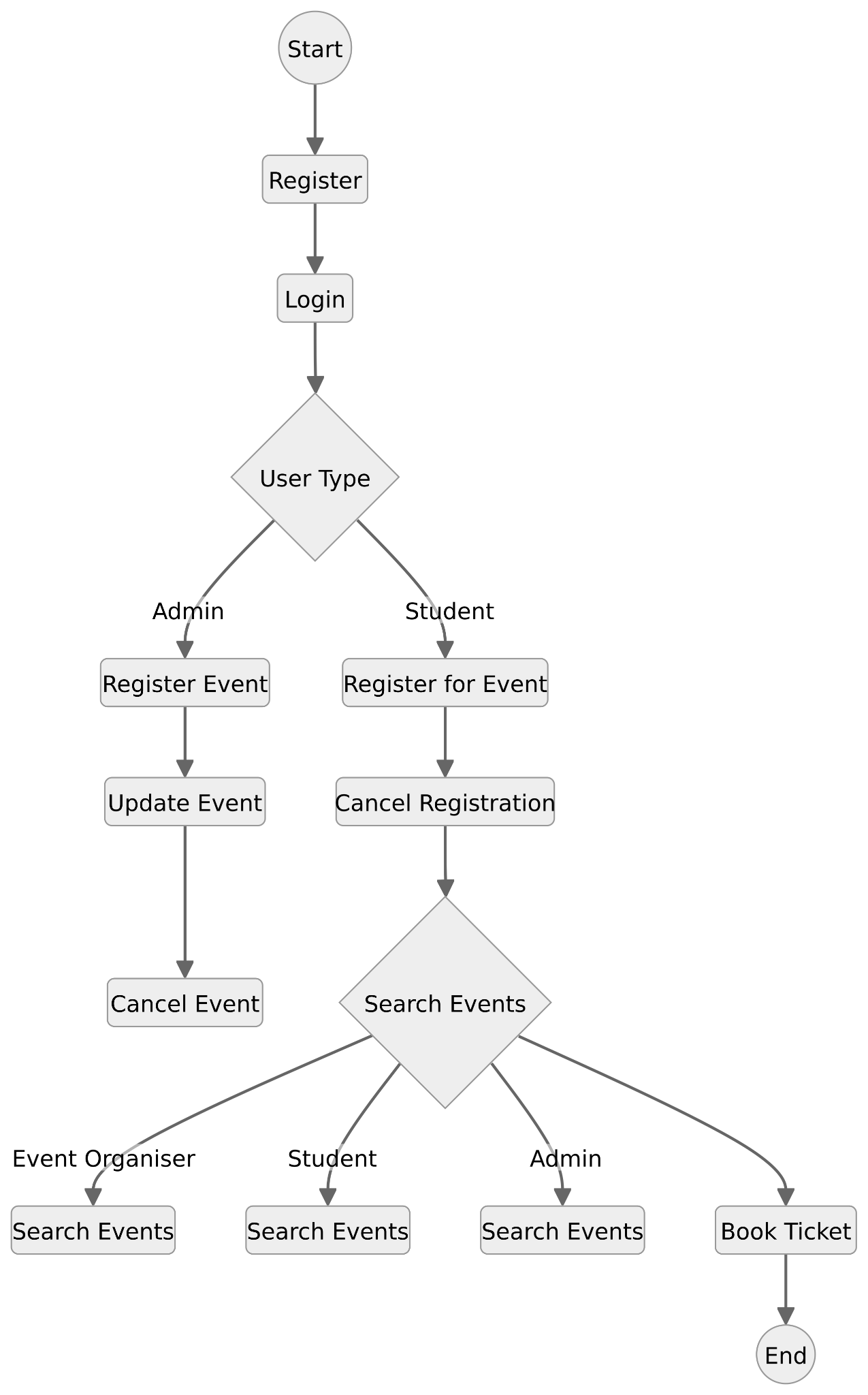
*<Include visual representations that help convey the structure, behavior, and interactions within the software system.*

## **Use Case Diagram**





## **Activity Diagram**



## **Sequence Diagram**

