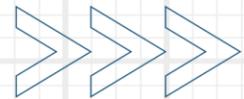




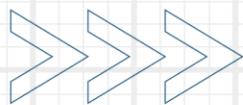
Bulibol

Software Engineering Project



Supervised By: Dr. Hamad Alsawalqah

Ismael Khalil Abu-Zaydeh	0213462
Yasmin Salah Almohtaseb	0216327
Shrouq Hassan Baker	0224734
Layan Mohammed Al-Titi	0225664
Sama Ziad Khamis	0229653



Executive Summary

Bullbol is an innovative web platform designed to streamline event management and enhance engagement within the KASIT community. Serving as a centralized hub, it offers students, faculty, and staff an intuitive interface to discover and stay updated on upcoming events, including key details like date, time, location, and description. Managed by the Vice Dean for Student Affairs, the system supports seamless event organization and promotion, catering to both internal faculty-hosted and cross-faculty activities. With features such as an event archive, real-time updates, and customizable search filters, **Bullbol** ensures easy access to past event data and fosters continuous improvement through user feedback and ratings. By categorizing events based on themes, types, and target audience, the platform enables users to quickly find relevant activities, promoting a vibrant and well-connected faculty environment

Table of Contents

Contents

1.0 Introduction	5
1.1 Project Overview	5
1.2 Problem Definition	5
1.2.1 Problem Statement.....	5
1.2.2 Issues.....	6
1.2.3 Objectives	6
1.2.4 Requirements	6
1.2.5 Constraints	8
1.3 Feasibility Study	8
1.3.1 Technical Feasibility.....	8
1.3.2 Operational Feasibility.....	8
1.3.3 Economic Feasibility	9
1.3.4 Schedule Feasibility.....	11
1.3.5 Legal Feasibility	11
1.4 Recommended Solution and Expected Project Deliverables	14
1.5. Local and Global Impact of the Proposed Solution	15
1.5.1 Local Impact	15
1.5.2 Global Impact	16
1.6. Naming Conventions and Definitions	17
2.0 Project Management Plan	18
2.1 Project Organization.....	18
2.2 Roles and Responsibilities.....	19
2.3 Software Process Model.....	20
2.4 Tools and Techniques:.....	21
2.5 Work breakdown include assigning Team Members to Tasks.....	22
2.5.1 Project's Tasks.....	22
2.5.2 Project's Milestones.....	23
2.6 Project Schedule	25
2.7 Risk Analysis and Plans	26

2.8 Monitoring, Reporting, and Controlling Mechanisms	27
3.0 Software Requirements Specifications (SRS).....	28
3.1 System stakeholders and Requirements Sources.....	28
3.2 User Requirements Definition.....	29
3.3 Use Case Diagram	33
3.4 System Functional Requirement Specification	34
3.5 Textual Description for Each Use Case	36
3.6 Non-Functional Requirements	45
3.6.1 Performance Requirements.....	45
3.6.2 Dependability Requirements	46
3.6.3 Security Requirements.....	46
3.6.4 Usability Requirements	46
3.6.5 Operational and Environmental Requirements.....	46
3.6.6 Maintainability Requirements	47
3.7 Data Requirements	47
4.0 Analysis and Design.....	49
4.1 Activity Diagrams	49
4.2 Sequence Diagrams	53
4.3 Class Diagram	57
4.5 Graphical User Interface Design	58
4.4 Architecture Design.....	69
4.5 Classes and Component Design	70
5.0 Implementation	71
6.0 User Manual	71
7.0 References	71
Appendix	72

1.0 Introduction

1.1 Project Overview

Bullbol is a web-based platform designed to showcase and manage upcoming events within KASIT (King Abdullah II for Information Technology). This system aims to provide students, volunteer teams, faculty, and staff with a centralized, user-friendly interface to view detailed event information, including the event name, description, date, time, and location. By offering features like real-time updates, search and filter functionality, all managed by the authority, which is the Vice Dean for Student Affairs in KASIT, **Bullbol** simplifies the organization and promotion of campus activities, whether set up by individuals within the faculty or from other faculties. Additionally, the platform includes an archive for past events, allowing users to access historical data and insights, which can be useful for planning future activities and maintaining a record of campus initiatives. Furthermore, **Bullbol** provides a classification mechanism for categorizing events based on themes, types and target audience, along with a review feature that enables participants to share feedback and ratings, fostering a culture of continuous improvement and engagement.

1.2 Problem Definition

1.2.1 Problem Statement

The current mechanism and system for announcing KASIT events is ineffective, as it relies on the faculty's or organizer's social media pages, resulting in limited reach and leaving a significant number of people uninformed. Additionally, there is no archive to include past events, and the reliance on papers and scattered social media posts to document events makes it challenging to maintain a comprehensive record. This lack of centralized data complicates efforts to run statistics or generate reports on past events, hindering the ability to analyze engagement trends and improve future planning. These issues contribute to lower attendance and reduced engagement, which the project aims to solve.

1.2.2 Issues

- **Issue 1:** Limited reach of event announcements.
- **Issue 2:** Insufficient details about the event, including its topic, speakers, and target audience.
- **Issue 3:** Absence of a centralized calendar for all events.
- **Issue 4:** Difficulty in gathering real feedback that helps improve upcoming events.
- **Issue 5:** Challenges in coordinating and scheduling events to avoid conflicts and overlaps.

1.2.3 Objectives

These objectives aim to provide solutions to the issues mentioned above and will be kept in mind throughout the entire project process to achieve them.

- **Objective for Issue 1:** Send notifications to registrants via email and target attendees using a database that includes their past registrations and preferences collected by the site.
- **Objective for Issue 2:** Provide a brief event description that includes the topic, speakers (with links to their LinkedIn profiles), the organizer, agenda, date, time, and location.
- **Objective for Issue 3:** Implement a centralized calendar for all events, integrated with Google Calendar for better accessibility and management.
- **Objective for Issue 4:** Enable a rating option on the site, exclusively for registered attendees, to ensure accurate feedback.
- **Objective for Issue 5:** Simplify event management by using the centralized calendar to add new events while preventing scheduling conflicts and overlaps.

1.2.4 Requirements

- **Requirement 1:** User register and login system.
- **Requirement 2:** Ability to change the password.
- **Requirement 3:** Simple registration process for events.
- **Requirement 4:** Event notifications via email for registered users.

- **Requirement 5:** Centralized event calendar that integrates with Google Calendar.
- **Requirement 6:** Ability to view upcoming event details, including topic, speakers, date, time, and location.
- **Requirement 7:** Event feedback system with rating options available to registered attendees.
- **Requirement 8:** User profile displaying personal information (name, email, major, college, and preferences) as well as their activities (event registrations).
- **Requirement 9:** Notification bell to alert users of upcoming events.
- **Requirement 10:** Event search functionality with filters for simplified browsing.
- **Requirement 11:** Access to an archive of past events, including reports with details such as event name, pictures, host, agenda, sponsors, date, and statistics (e.g., number of registrations).
- **Requirement 12:** Event reminder system to send automatic notifications to users before the event starts.
- **Requirement 13:** Automatic event countdown showing the number of days remaining until the event.
- **Requirement 14:** Automatic event expiration system to remove past events from the main page and archive them in archive page.
- **Requirement 15:** An FAQ section to enhance the user experience.
- **Requirement 16:** Comprehensive project documentation to simplify the development process.
- **Requirement 17:** A user-friendly design focusing on both UI and UX.

1.2.5 Constraints

- **Constraint 1:** Dependence on event organizers to provide accurate and timely updates for events, which may cause delays or inconsistencies if not managed effectively.
- **Constraint 2:** Dependence on frequent event updates, which requires real-time data management and may increase the workload for the administrator.
- **Constraint 3:** Reliance on social media advertising for event promotion by hosts or audience, which may overshadow the event site and risk making it secondary, whereas the goal is to establish the site as the official platform for event information and management.
- **Constraint 4:** The reliance on IIS (Internet Information Services) may introduce challenges related to performance, scalability, and feature compatibility. IIS requires specific technical expertise, and misconfigurations could result in slow load times.

1.3 Feasibility Study

1.3.1 Technical Feasibility

After evaluating the programming languages, technical skills, and requirements for the project, it has been determined that the project is technically feasible. For more details, refer to Table 4

1.3.2 Operational Feasibility

Bullbol effectively addresses the needs of the KASIT faculty and simplifies the administrative workload, enhancing both convenience and engagement for students and staff. making it operationally feasible. For additional details, refer to Table 4

1.3.3 Economic Feasibility

- Development Costs:

Table 1 : Development costs

Development activity	Employee position	Cost per hour (JD)	Number of hours	Total cost (JD)
Software development	Backend Developer (C# and .NET core)	25	72	1800
	Frontend Developer (HTML ,CSS, Bootstrap)	20	72	1440
Website design	UI/UX Designer	20	48	960
Testing and Quality Assurance	QA	15	30	450
Requirement and specification analysis	System analyst	15	30	450
Project Management	Project Manager	30	60	1800
Total development costs	6,900 JD			

- Hardware and Software Costs:

Table 2: HW & SW Costs

Item	purpose	Estimated cost
Software Costs		
Development Tools	Web development frameworks (e.g.NET Core)	Free (Open-source)
Database System	relational database (e.g., MySQL)	Free (Open-source)
Google Calendar API Integration	Synchronization for events	Free (basic usage)
Integrated Development Environment (IDE)	Software for coding (e.g., Visual Studio Code)	Free
Hardware Costs		
Development Machines	Computers for coding and development	Assume existing hardware (0 JD)
Internet Costs	Required for development	Assume existing hardware (0 JD)
Testing Devices	Testing the platform on desktop/mobile	Assume existing hardware (0 JD)

The result shown at Table 4

1.3.4 Schedule Feasibility

Table 3 : Schedule Feasibility

Task ID	Task Description	Duration(weeks)
T1	Kickoff meeting and stakeholder identification	1
T2	Requirement and specification analysis	0.5
T3	System architecture design	0.5
T4	Detailed system design	2
T5	Frontend user interface development	2
T6	Database construction	1
T7	Backend development	3
T8	System Integration and testing	1
T9	Deployment and server configuration	1
T10	Project documentation	1
Whole project		13 weeks

1.3.5 Legal Feasibility

The administrator of **Bullbol** ensures that all hosted and managed events align with the Student Code of Conduct established by the University of Jordan. To maintain compliance with university policies, every event will be officially approved by the **Deanship of Student Affairs** at KASIT, ensuring they meet the required standards and guidelines. Result shown at Table 4

Table 4 : Feasibility Study

Feasibility Type	Description	Assessment (Feasible/Not Feasible)
Technical Feasibility	<p>Required Technologies:</p> <ol style="list-style-type: none"> 1. A web development framework (.NET core). 2. A database system to store user and event information. 3. Integration with Google Calendar for event synchronization. 4. Email notification services. 	Feasible
Operational Feasibility	<p>Operational need to meet:</p> <ol style="list-style-type: none"> 1. User Adoption. 2. Resource Availability. 3. Match real needs. 	Feasible
Economic Feasibility	<p>Costs include:</p> <ol style="list-style-type: none"> 1. Wages of development team. 2. Hosting costs. 3. Software and tools. 4. Maintenance. 	Feasible
Schedule Feasibility	<p>Time distribution:</p> <ol style="list-style-type: none"> 1. Kickoff meeting and stakeholder identification 2. Requirement and specification analysis 3. System architecture design 4. Detailed system design 5. Frontend user interface development 	Feasible

	<p>6. Database construction</p> <p>7. Backend development</p> <p>8. System Integration and testing</p> <p>9. Deployment and server configuration</p> <p>10. Project documentation</p>	
Legal Feasibility	<p>It must deal with:</p> <p>1. Student Code of Conduct</p> <p>2. Deanship of Student Affairs</p>	Feasible

Justification/Notes on each feasibility:

Technical Feasibility:

- Web frameworks are free, open-source, and well-supported
- Necessary hardware is already available.

Operational Feasibility:

- The platform is user-friendly and tailored for IT students, who are familiar with such systems.
- The college has programming talent, and the university's IT center supports the development process.
- The database scope is limited to KASIT events, simplifying implementation.

Economic Feasibility:

- The team consists of students or IT center staff, minimizing labor costs.
- Hosting will be managed by the Vice Dean for Student Affairs at KASIT, reducing additional expenses.
- Most tools and frameworks are free and open source.
- Maintenance will be straightforward due to the platform's simple and focused design.

Schedule Feasibility:

- Tasks are evenly distributed, and no major bottlenecks are anticipated.

Legal Feasibility:

- It depends on the administrator who updates and uploads upcoming events.

1.4 Recommended Solution and Expected Project Deliverables

- The system proposes a modern, dynamic website designed to display and manage upcoming faculty's events efficiently. The solution will incorporate the following key features:

- **User-Friendly Interface:**

A responsive and visually appealing interface tailored for students, faculty, and event organizers. It will allow users to browse, search, and filter events effortlessly across devices, including desktops and mobiles.

- **Comprehensive Event Details:**

Each event listing will include essential details such as title, date, time, location, organizer information, and a detailed description. Users will have the option to view these details on a dedicated event page.

- **Search and Filter Functionality:**

Advanced filtering options by date, category, or keywords, enabling users to quickly find events of interest.

- **Real-Time Notifications and Updates:**

Website sends notifications or alerts regarding event changes (e.g., cancellations, time updates) to users who choose to be updated.

- **Calendar Integration:**

Events will be presented in both list and calendar views, offering users flexibility in how they view and interact with upcoming events.

- **Hosting and Scalability:**

A cloud-hosted solution with scalable infrastructure to accommodate increasing traffic and ensure high performance during peak usage times.

- **Optional Feature:**

User registration for personalized event recommendations and reminders.

1.5. Local and Global Impact of the Proposed Solution

1.5.1 Local Impact

- **Enhanced Student Engagement:**

By providing easy access to upcoming events, the website will foster greater participation among students, promoting a vibrant and inclusive faculty culture.

- **Streamlined Event Management:**

The platform will simplify event organization for staff and faculty, reducing manual workload and errors, thus ensuring smoother coordination and execution of college activities.

- **Improved Information Accessibility:**

With real-time updates and a centralized platform, students and faculty will have immediate access to event details, ensuring they stay informed and can plan their schedules effectively.

- **Boosted Collaboration:**

Departments and student organizations will have a streamlined way to promote events, leading to better collaboration and a more dynamic academic environment.

- **Operational Efficiency:**

Automating event updates and notifications will save time for administrators, allowing them to focus on higher-priority tasks.

1.5.2 Global Impact

- **Promoting Digital Transformation in Education:**

The website demonstrates how technology can improve communication and coordination in educational settings, inspiring other institutions to adopt similar solutions.

- **Scalability for Broader Adoption:**

The system can be adapted by other colleges and universities worldwide, showcasing an effective model for event management that can be scaled to suit various institutions.

- **Improved Global Networking Opportunities:**

Highlighting events such as guest lectures, conferences, and international collaborations can enhance the institution's global reputation, attracting participants and partners from around the world.

- **Sustainability In Communication:**

By reducing the reliance on printed materials for event promotion, the system contributes to environmental sustainability, minimizing paper waste and promoting digital communication.

- **Educational Leadership:**

The project positions the college as a leader in adopting innovative solutions, setting an example for others and contributing to the advancement of global education systems.

1.6. Naming Conventions and Definitions

Terms

- **Event:**

An activity or occasion planned to occur at a specific date and time, including workshops, seminars, cultural events, and academic programs.

- **Event Details:**

Information associated with an event, such as the title, description, date, time, location, organizer, and any additional notes or attachments.

- **User:**

Individuals who interact with the system to view or manage events. This includes students, faculty members, event organizers, and administrators.

- **Notification:**

An alert or message sent to users regarding updates, changes, or reminders about events.

Notifications can be sent via email, SMS, or directly on the platform.

2.0 Project Management Plan

2.1 Project Organization

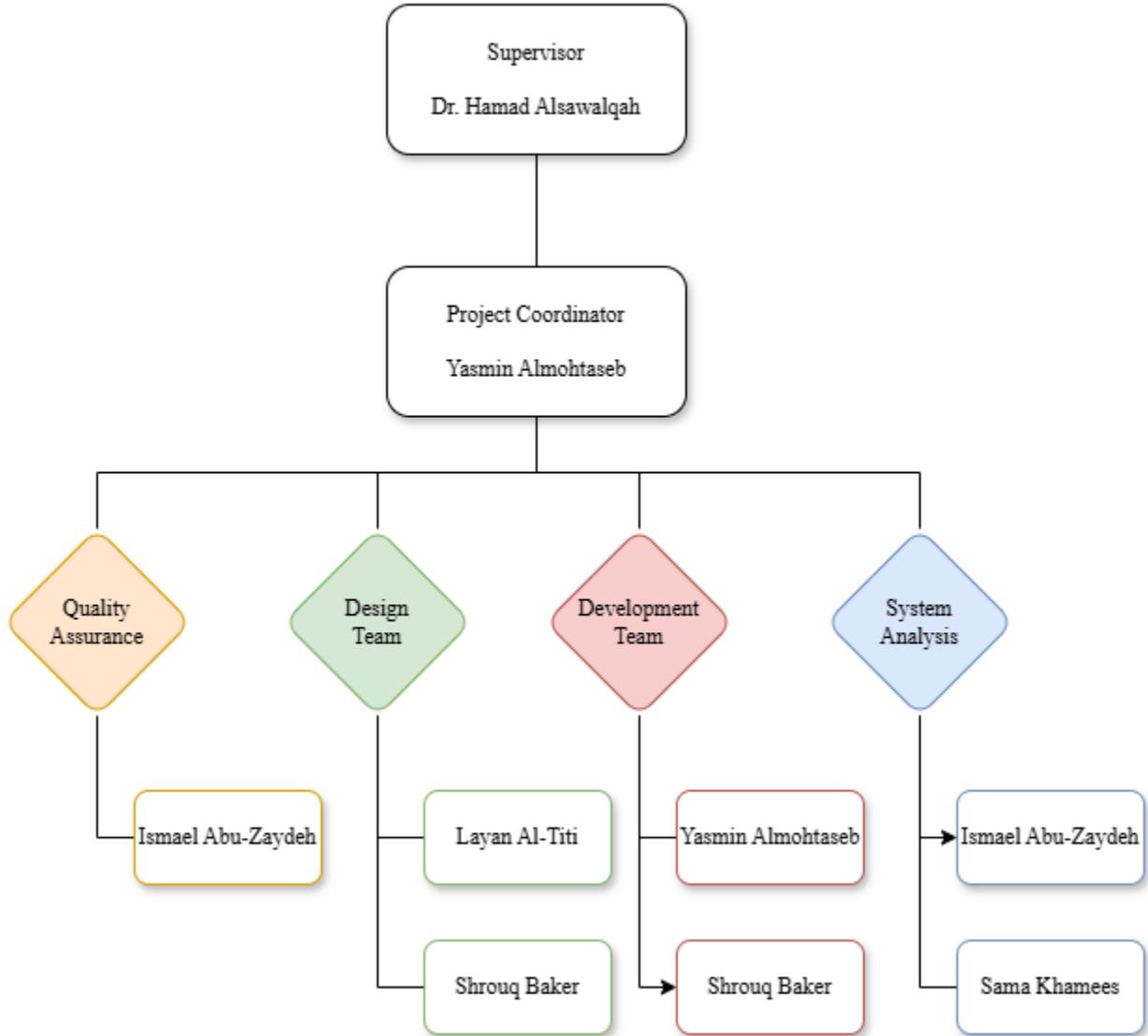


Figure 1: Project Organization

2.2 Roles and Responsibilities

- **Project Coordinator**

- Tracks progress, resolves issues promptly and assures compatibility of outputs from the teams.
- Schedules meetings, oversees communication between teams and prepares progress reports.
- Ensures that goals align with requirements and stakeholders expectations.
- Identifies risks and implements strategies to avoid project delays.

- **Design Team**

- Design prototypes and mockups that align with requirements.
- Focuses on usability and accessibility principles to deliver user-friendly results.
- Collaborate with the development team to ensure accurate implementation of designs.

- **Development Team**

- Implements the visual and interactive elements of the website based on the results of the design team and ensures responsiveness across browsers and devices.
- Deliver high quality code based on clean architecture and appropriate design patterns.
- Troubleshoots technical issues and keeps tracking of logical errors.
- Keep up to date with the latest programming technologies and methodologies.

- **Quality Assurant**

- Develop and execute test plans, test cases and test scripts to validate functionality, performance and usability of the website.
- Track and log software defects and provides feedback.
- Perform security testing to ensure reliability.

- **System Analyst**

- Gather project requirements from stakeholders to define project specifications.
- Perform feasibility studies and risk assessments.
- Document and communicate requirements clearly to the development team.
- Ensure the website aligns with business objectives.

2.3 Software Process Model

The project will follow the Agile methodology due to its flexibility in adapting to changes, its ability to shift project priorities based on requirements, and its collaborative approach, where team members with diverse skill sets work together. These attributes align perfectly with the need to meet specific deadlines. Agile enables iterative development, allowing us to adhere to deadlines while addressing potential challenges early. Additionally, this approach empowers the team to maintain high-quality standards that align with the project requirements and goals, ensuring efficient delivery without delays. Phases of the project are shown below:

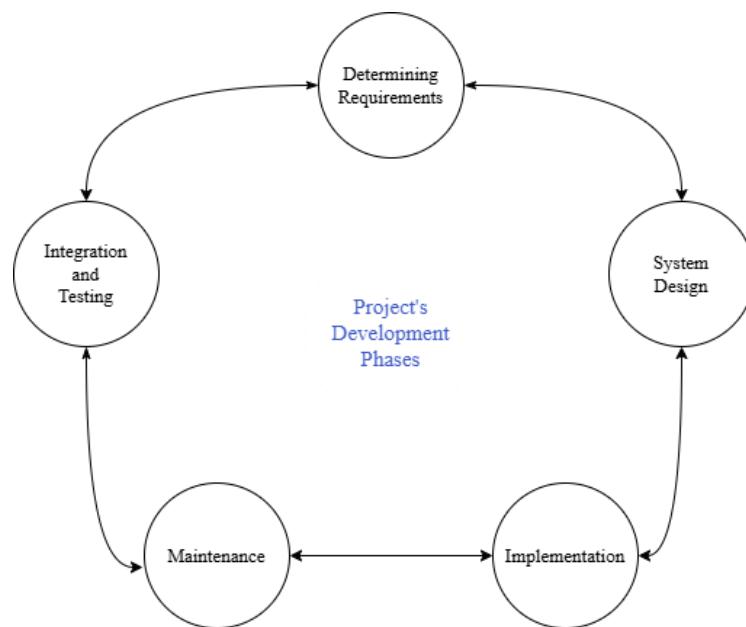


Figure 2 : Project's Development Phase

2.4 Tools and Techniques:

- **Tools:**

Table 5 : Tools Required

Software	Requirement
Development Tools	Visual Studio 2022, Microsoft SQL Server 2019, IIS, VS Code, GitHub
Misc Tools	Google Meet, Draw.io, Lucidchart, Notion
User Interface Design	Figma, Adobe XD, Adobe Illustrator
Backend	C# and .NET core framework
Frontend	HTML, CSS and Bootstrap framework

- **Techniques:**

- Software Design Pattern: The development team aim to deliver a strong and well-structured backend by following MVC (Model-View-Controller) design pattern. This pattern simplifies maintenance, enhances testability, and provides a solid foundation for future expansions and feature additions. In addition to MVC, the team will adopt the Repository pattern, which introduces a layer of abstraction between the application and data access and management.
- Object-Oriented Programming techniques: To ensure a robust codebase, we will apply OOP principles alongside SOLID practices, using well-defined classes and objects. This approach will guide us in creating understandable code while minimizing tight coupling, ensuring long-term maintainability.
- Relational Database Design Techniques: We will translate the requirements into an Entity-Relationship Diagram (ERD) and use a relational database to efficiently structure and manage data. The ERD will define entities, their attributes, and relationships while the relational database will store data in structured tables, supporting CRUD operations through queries.

2.5 Work breakdown include assigning Team Members to Tasks

2.5.1 Project's Tasks

Table 6 : Project Tasks

Phase	Tasks Description	Responsible Staff	Resources Needed (Skills, SW, HW)
Project Initiation	<ul style="list-style-type: none"> • Kickoff meeting • Define project scope and identify stakeholders 	Yasmin Almohtaseb	Project Management Tools
	<ul style="list-style-type: none"> • Analyze existing systems and collect user requirements 	Shrouq Baker, Sama Khamis	Data analysis principles and communication skills
System Design	<ul style="list-style-type: none"> • Determine website's identity and design system architecture 	Layan Al-Titi	Users' experience determination skills
	<ul style="list-style-type: none"> • Detailed system design • Design User Interface 	Layan Al-Titi, Shrouq Baker	Designing softwares
Development	<ul style="list-style-type: none"> • Frontend Development 	Yasmin Almohtaseb	IDE, Bootstrap, GitHub
	<ul style="list-style-type: none"> • Database construction 	Ismael Abu-Zaydeh	Database management tools

	<ul style="list-style-type: none"> • Scaffolding database and backend development 	Yasmin Almohtaseb, Ismael Abu-Zaydeh	IDE, GitHub, framework
Testing	<ul style="list-style-type: none"> • Unit testing • Integration testing 	Ismael Abu-Zaydeh, Layan Al-Titi	Debugging skills, testing framework
Deployment	<ul style="list-style-type: none"> • Web server setup and domain configuration • Project upload to server • Provide Training 	Shrouq Baker, Sama Khamis	Web server software, Cloud host
Documentation	<ul style="list-style-type: none"> • Finalize project's documentation 	Sama Khamis	Technical writing skills, Documentation tools

2.5.2 Project's Milestones

Table 7 : Project Dependencies, Deliverables and Milestones

Task ID	Task Description	Duration	Dependencies	Deliverables	Milestones
T1	Kickoff meeting and stakeholder identification	1	None	Vision document, meeting agenda	Meeting completed, scope defined

T2	Requirement and specification analysis	0.5	T1	Requirements document, specification report	Requirements finalized
T3	System architecture design	0.5	T2	System architecture diagram	Architecture approved (M1)
T4	Detailed system design	2	T3	Detailed design document	Design finalized
T5	Frontend user interface development	2	T2, T4	UI design, responsive design	User interface completed
T6	Database construction	1	T5	ERD design, database setup	Database setup complete (M2)
T7	Backend development	3	T5, T6	Backend code and functionality	Backend development completed
T8	System Integration and testing	1	T5, T7	Test reports	Bugs resolved; testing completed
T9	Deployment and server configuration	1	T8	Live server setup on host	Website deployed live
T10	Project documentation	1	T9	Final report and technical documentation	Documentation complete (M3)

2.6 Project Schedule

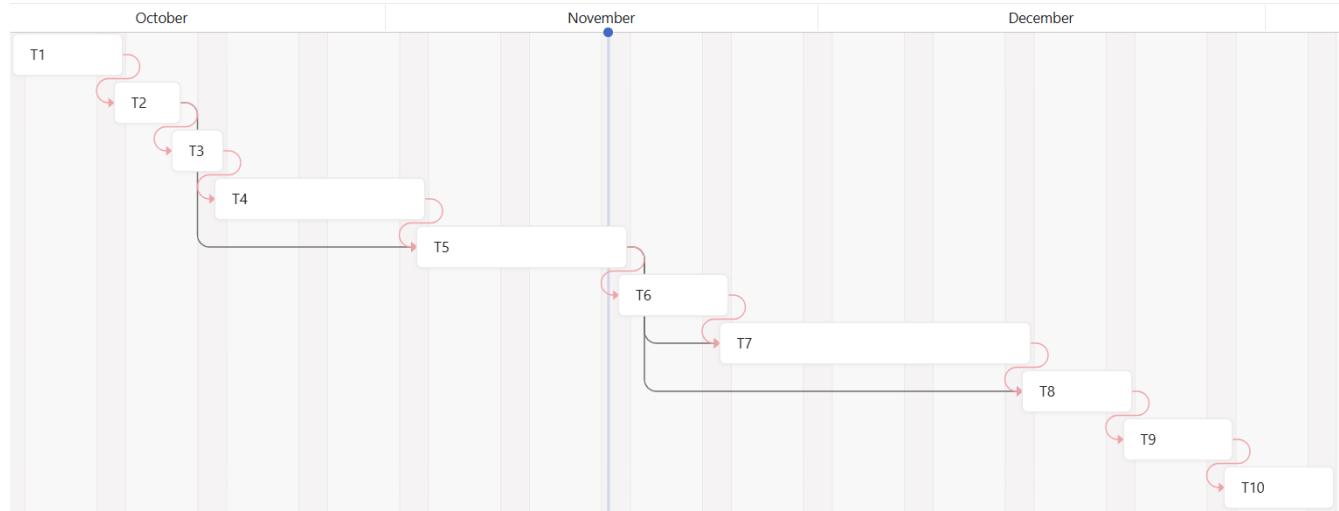


Figure 3 : Gantt Chart for Project Schedule

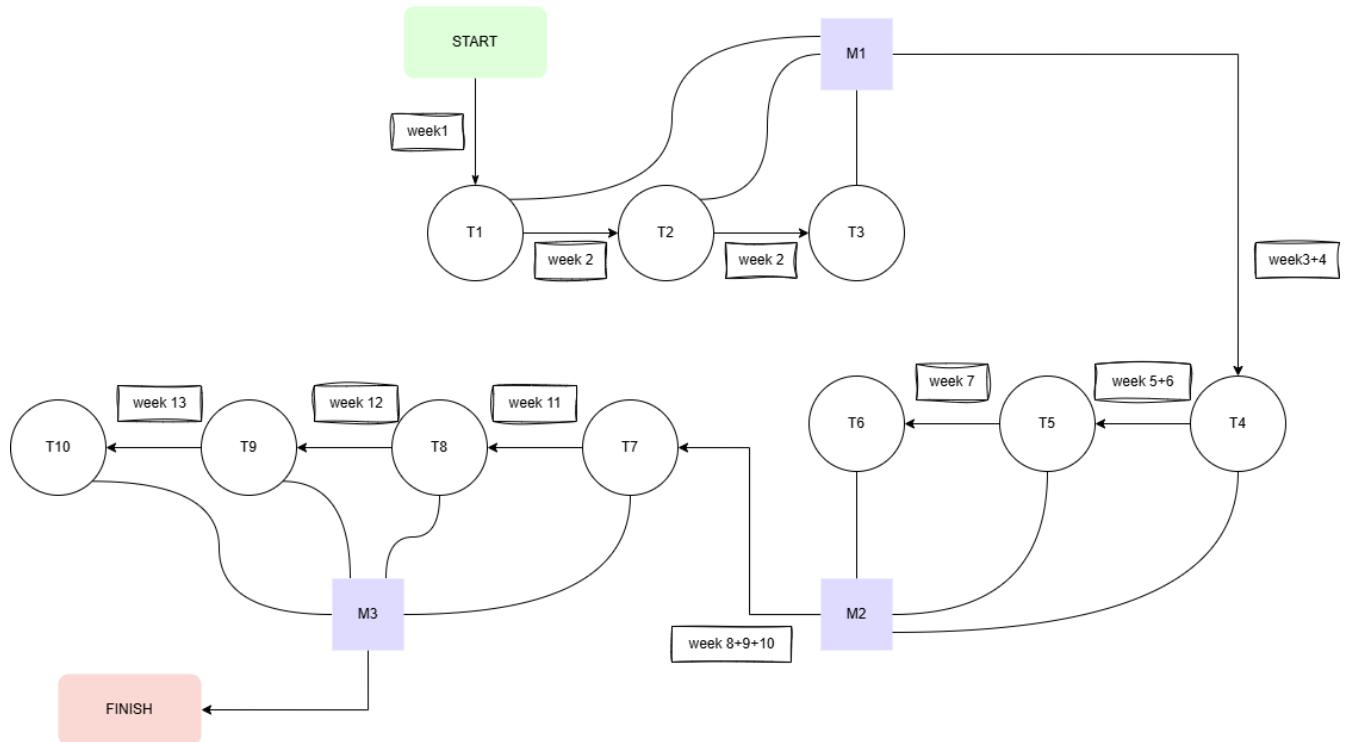


Figure 4 : PERT Figure for Project Schedule

Critical path = $T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 + T10$

Total Time = 13 weeks (91 days).

2.7 Risk Analysis and Plans

Table 8 : Risk Identification and Assessment

Risk	Risk Type	Probability	Effect
Server failure and lag	Technology	Moderate	Catastrophic
Issues due to high traffic	Estimation	Moderate	Tolerable
Data breaches or unauthorized access	Technology	Low	Catastrophic
Data corruption	Technology	Low	Serious
Low user adoption	Estimation	Moderate	Tolerable
Inaccurate event data	Organizational	High	Serious
Uncontrolled requirements or delay of time	Estimation	High	Serious
Limited resources	Organizational	Moderate	Serious

Risk Planning

1. Server failure and lag

- a. Choosing Cloud Hosting.
- b. Distribute software to multiple servers.
- c. Redirect traffic to backup servers.

2. Issues due to high traffic

- a. Test the platform under simulated high-traffic conditions

3. Data breaches or unauthorized access

- a. Use encryption for sensitive data
- b. Conduct regular security checks and penetration testing

4. Data corruption

- a. Setup recovery plan to store data
- b. Perform regular backups and store them in multiple locations

5. Limited Resources

- a. Prioritize features by importance and impact
- b. Use existing university resources to minimize additional costs

2.8 Monitoring, Reporting, and Controlling Mechanisms

Monitoring Mechanisms:

- The project coordinator will oversee the overall progress using project management tools and prioritize tasks based on their importance and dependencies.
- Conduct weekly meetings to ensure alignment among team members and consistency of results.
- Identify problems at each project phase and introduce solutions to solve them.

Reporting Mechanisms:

- The system analyst will keep track of documentation to make sure it meets the project's requirements and specifications.

Controlling Mechanisms:

- If sudden changes come out to the surface, it will be evaluated based on timeline and scope before rejection or acceptance.
- Implement regular quality assurance activities and performance checks.
- Ensure flexibility and accuracy of milestones.

3.0 Software Requirements Specifications (SRS)

3.1 System stakeholders and Requirements Sources

Stakeholders:

1. **System Administrator:** The system administrator can be The Vice Dean of Student Affairs in KASIT or the supervisor who's responsible for giving events permissions and reservations. The administrator creates the event with all its required details to be viewed on the website through the database where only the admin can access it. Also, he/she can make edits to the website as requested and ensure accuracy of event details.
2. **IT Department:** Responsible for providing maintenance, ensuring system security and availability.
3. **Professors and teaching staff:** Can book events through the admin, attend and rate events after registering on the website.
4. **Students:** Can attend and rate events after registering on the website.
5. **Communities in KASIT/University and hosts:** Can book events through the admin.
6. **Sponsors:** Can sponsor booked events and participate in them.
7. **Guests and Speakers:** Can participate in events and their information is mentioned in the event details.

Requirement Sources:

1. **University Policies and Guidelines:** University regulations concerning facility usage, booking protocols, and privacy policies.
2. **Existing University IT Infrastructure:** Existing databases or data that need to be integrated such as calendar systems
3. **User Feedback and Surveys:** Input from students, professors, communities and staff on what they need and expect from **Bullbol**.

- 4. Facility Management Schedules:** Input from faculty managers regarding hall availability, maintenance periods and room capacity, and resources.
- 5. Security Requirements:** Standards and protocols for ensuring system security and data protection.

3.2 User Requirements Definition

In the context of the **Bullbol** platform, user stories illustrate the functionalities required to streamline event management, improve communication, and enhance engagement within the KASIT community. These stories cover the user requirements and define them clearly. By leveraging user stories, the development team can stay agile, adaptable, and focused on delivering a user-friendly solution that solves real-world problems effectively.

User Stories:

User Registration and Login

As a user, I want to create an account with my name, email, major, and college so that I can access personalized event information.

As a user, I want to log in to the system using my email and password so that I can view and manage my registrations.

As a user, I want to be able to change my password so that I can keep my account secure.

Event Discovery

As a user, I want to browse a list of upcoming events with details such as date, time, location, and description so that I can decide which events to attend.

As a user, I want to search for events by keywords, categories, or date filters so that I can easily find events relevant to my interests through tags or keywords.

As a user, I want to see a countdown timer for each event so that I know how many days are left until the event.

Notifications and Reminders

As a registered user, I want to receive email notifications about upcoming events and events that I registered in so that I can stay informed.

As a user, I want to receive reminders before an event starts so that I don't miss it.

As a user, I want a notification bell to alert me to event updates and announcements.

Event Rating

As a registered user, I want to provide ratings on events I attended so that I can help improve future events.

As an admin, I want to collect and review event feedback so that I can analyze engagement and satisfaction.

Event Management

As an admin, I want to create and manage events, including details such as topic, date, time, location, and speakers, so that I can promote events effectively.

As an admin, I want to ensure no scheduling conflicts by using a centralized calendar to plan events.

As an admin, I want to archive past events automatically so that the main event page stays updated with current events.

Calendar Integration

As a user, I want to view events in a centralized calendar that integrates with Google Calendar so that I can manage my schedule easily.

As a user, I want to export event details to my personal Google Calendar so that I can track them alongside my other activities.

User Profiles

As a user, I want to view my profile with details such as my name, major, email, and event registration history so that I can track my activity.

As a user, I want to update my profile preferences so that I receive notifications for relevant events.

Event Archiving

As a user, I want to access an archive of past events with details, images, and statistics so that I can review what events have occurred.

As an admin, I want to be able to write reports based on event statistics so that I can analyze attendance trends.

UI/UX

As a user, I want a user-friendly and visually appealing interface so that I can easily navigate and use the platform.

As a user, I want an FAQ section so that I can find answers to common questions quickly.

Security and Constraints

As an admin, I want the system to follow the Student Code of Conduct and ensure event approval processes so that all hosted events meet university standards.

As an admin, I want secured login and data protection mechanisms to prevent unauthorized access such as password constraints.

3.3 Use Case Diagram

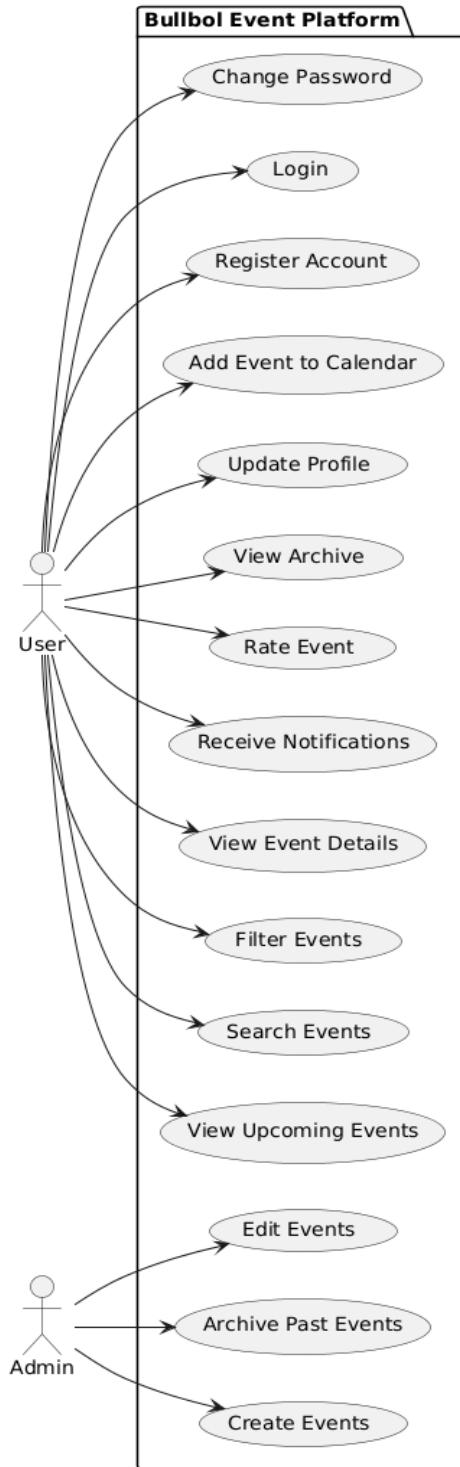


Figure 5 : Use Case Diagram

3.4 System Functional Requirement Specification

Table 9 : Functional Requirements

Function ID	Function Name	Function Description	User Story
1	Register (sign up)	Allows users to create a new account by providing their email, name, role (student: including major and college, or guest), and password. The password must meet the following constraints: it must be at least 8 characters long, include at least one number, one uppercase letter, one lowercase letter, and one special character. Users must also confirm their password to complete the registration process.	User Registration and Login
2	Login	Allows users to log in to the website using email and password.	User Registration and Login
3	View & edit profile	Enables users to view their profile information (name, profile picture, email, preferences, calendar of upcoming events) and update details as needed such as change password.	User Profiles
4	Change password	Allows users to securely change their password via email verification.	User Registration and Login

5	View upcoming events	Displays a list of all upcoming events with key details such as picture, name, date, brief & offer learn more button.	Event Discovery
6	View finished events	Displays a list of all finished events with key details such as pictures, name, host, rating (1-5 stars) & offer learn more button.	Event Archiving
7	View event details	Allows users to view detailed information about events include date, time, location, topic, sponsors, host, agenda, event tags and speakers.	Event Discovery
8	Search for events	Enables users to search for upcoming or finished events & offer filtration option by (date, host, etc..)	Event Discovery
9	Register for events	Enables users to register their attendance for any available event on the website simply by register button.	Event Discovery
10	Email Notifications	Send email reminders to registered users about upcoming events. (optionally by activating the bell button)	Notifications and Reminders
11	Rate my finished event	Enables users to rate and provide feedback on events they attended, simply by ratings (1-5 stars).	Event Rating
12	Log out	Allow users to logout of their accounts.	Security and Constraints

3.5 Textual Description for Each Use Case

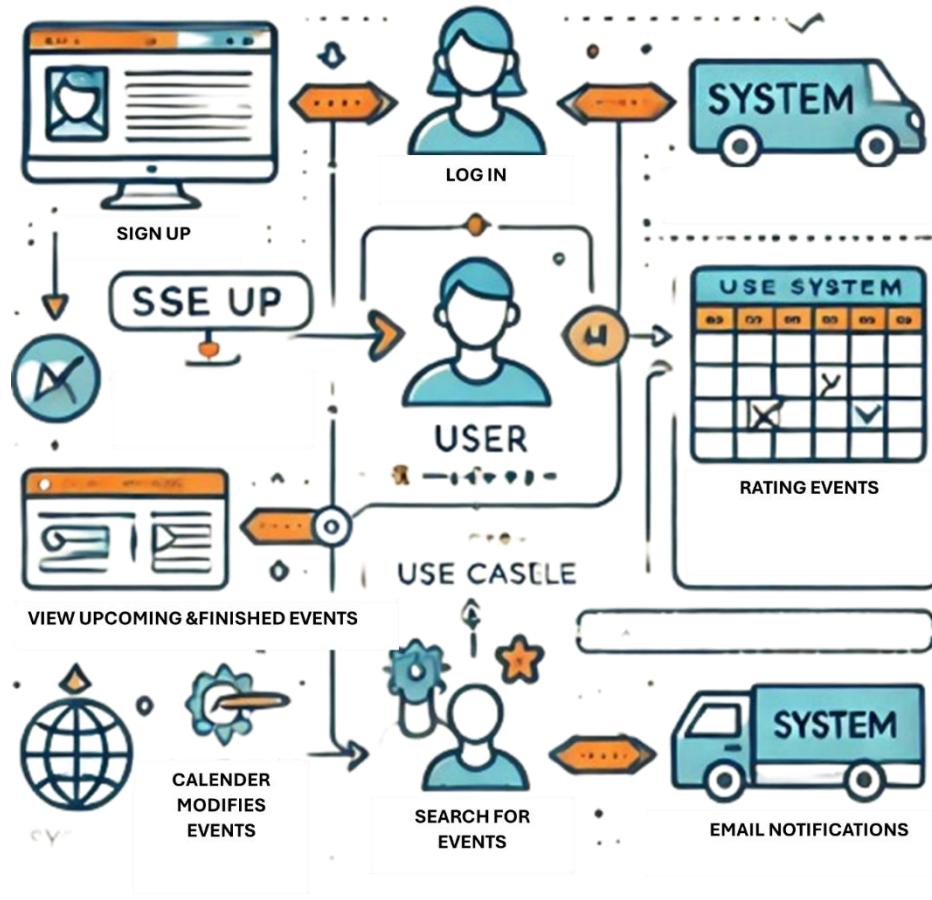


Figure 6 : Use Cases Integrations

Figure 6 : Use Cases Integrations presents how the **user** interacts with various functions of the system include:

- ✓ **Sign-up and Log-in**
- ✓ **Edit Profile**
- ✓ **Finished Events**
- ✓ **Search for Events**
- ✓ **Calendar Integration**
- ✓ **Email Notifications**

Arrows and connections highlight the flow of actions between the **user** and the **system**.

Table 10 : Sign Up Use Case

Use case	Sign up
Actors	user, system
Objective	Enable the user to successfully sign up for Bullbol website.
Preconditions	<ul style="list-style-type: none"> - User must have a valid email account. - User must not already have an account.
Basic Flow	Users open Bullbol website, click sign up button, enter their (name, email ,valid password deal with constrains include (at least 8 characters long, at least one number, one uppercase letter, one lowercase letter, and one special character)) system validates the entered data, if it's right user is redirected to the homepage.
Alternative Flows (What can go wrong)	<ul style="list-style-type: none"> - User enters an invalid password (doesn't meet the constraints). - Password Confirmation Error.
Postconditions	<ul style="list-style-type: none"> ● If the password is invalid: system displays an error message telling the constraints were not met. ● If the confirmation of pass is incorrect: System displays an error message. ● If the info is valid, the system directs user to the homepage.

Table 11 : Login Use Case

Use case	Login
Actors	user, system
Objective	Enable the user to successfully log in to their account.
Preconditions	User must have a previously registered account.
Basic Flow	User open log in page, enter (email, password) system validates the entered data, if it's right user is redirected to the homepage.
Alternative Flows(What can go wrong)	User enter wrong password, system display error message, user redirected to log in page again.
Postconditions	<ul style="list-style-type: none"> ● If the data is correct: User is directed to the homepage. ● If the data is incorrect: System displays an error and redirects the user back to the login page

Table 12 : Change Password Use Case

Use case	Change Password
Actors	user, system
Objective	Enable the user to successfully change password for their account.
Preconditions	<ul style="list-style-type: none"> - User must have a previously registered account. - User must have an active email connected to their account.

Basic Flow	Users open their profile page, enter change password button, system sends a verification code to the user's registered email and show box to enter the sent code.
Alternative Flows (What can go wrong)	<ul style="list-style-type: none"> - Users enter the wrong code. - Code Not Received.
Postconditions	<ul style="list-style-type: none"> ● If the code is correct: user is redirected to enter new pass and confirm it. ● If User enters wrong code: System displays an error message. ● user didn't receive a code: system offers resend button.

Table 13 : Edit Profile Use Case

Use case	Edit Profile
Actors	user, system
Objective	Enable the user to successfully edit their account information.
Preconditions	-User must have a previously registered account.
Basic Flow	Users open their profile page, click edit button, system displays editable fields for the user's account information (e.g., name, email, preferences), user modifies their information finally click save button.

Alternative Flows (What can go wrong)	-New information isn't saved (system error). -invalid modifications.
Postconditions	<ul style="list-style-type: none"> • successful update: system redirect user to their profile page displaying the updated information. • Failed update: system display error message such as "try again later". • Invalid modifications: The system highlights incorrect fields and displays error messages to guide the user in correcting the information.

Table 14 : Notifications Use Case

Use case	Email Notifications
Actors	User, system, third party: Email notification services
Objective	Send a reminder email for the user for upcoming events they have registered for.
Preconditions	<ul style="list-style-type: none"> - The user must have a registered account. - The user's account must be linked to an active email.

	<ul style="list-style-type: none"> - The bell notification must be activated. - The user must have registered for at least one event.
Basic Flow	The system identifies upcoming events that the user has registered for, a reminder email is generated & sent automatically.
Alternative Flows (What can go wrong)	User deactivates notifications.
Postconditions	<ul style="list-style-type: none"> ● If the email is successfully sent, the user receives a reminder for their upcoming event. ● If user deactivates notifications: They are required to reactivate notifications to receive reminder emails.

Table 15 : Archive Use Case

Use case	View Finished Events
Actors	System, user.
Objective	Successfully display finished events to the users.
Preconditions	<ul style="list-style-type: none"> - The user must have an active account. - The system must have archived event data. - Events are automatically moved to the archive after their due date expires.
Basic Flow	The user clicks the "Archive" button, system retrieves and displays finished events, showing up to 8 event cards per page.

Alternative	-No data available.
Flows (What can go wrong)	<p>-System error: the automatic archiving process fails.</p>
Postconditions	<ul style="list-style-type: none"> ● View archive successfully. ● If there is no data available, the system informs the user with a clear message. ● System error: system error handling.

Table 16 : Upcoming Events Use Case

Use case	View Upcoming Events
Actors	System, user.
Objective	Successfully display upcoming events to the users.
Preconditions	<ul style="list-style-type: none"> - The user must have an active account. - The system must have upcoming events data.
Basic Flow	The user clicks the "upcoming events" button, system retrieves and displays upcoming events include (name, date, brief), showing up to 8 event cards per page.
Alternative	-No data available.
Flows (What can go wrong)	
Postconditions	<ul style="list-style-type: none"> ● View upcoming events successfully.

	<ul style="list-style-type: none"> If there is no data available, the system informs the user with a clear message.
--	--

Table 17 : Search Use Case

Use case	Search for events
Actors	System, user.
Objective	System displays accurate results based on the user's search criteria.
Preconditions	<ul style="list-style-type: none"> - The user must have an active account. - The system must have events data.
Basic Flow	Users enter the event's name enhance their search operation using filters include: (date, host, etc..), system display matching events.
Alternative Flows(What can go wrong)	<ul style="list-style-type: none"> -no result match user search. -system errors: include (filter isn't working as expected no data yet).
Postconditions	<ul style="list-style-type: none"> The system displays accurate results. If no result matches: system display massage to user, such as: "No events found matching your search." If there is no data available, the system informs the user with a clear message. System error with filter: system handling filters error.

Table 18 : Rating Use Case

Use case	Rating events
Actors	System, user.
Objective	Allow the user to rate events they attended successfully.
Preconditions	<ul style="list-style-type: none"> - The user must have an active account. - The user must have registered for the event.
Basic Flow	User selects a rating (1 to 5 stars). The user submits the rating, and the system stores the rating.
Alternative Flows (What can go wrong)	-User doesn't attend the event.
Postconditions	<ul style="list-style-type: none"> • The system stores the user's rating for the event. • To avoid rating of users that didn't attend events: The event's rating option is available only for events the user has registered for which appear in their calendar.

Table 19 : Callender Use Case

Use case	Calander modifies
Actors	System, User, Third Party (Google Calendar)
Objective	Automatically modify the user's calendar to reflect upcoming events they've registered for.

Preconditions	<ul style="list-style-type: none"> - The user must have an active account. - The user must have registered for upcoming events. - The system must be integrated with Google Calendar.
Basic Flow	The user registers for an event, system automatically adds the event to the user's Google Calendar.
Alternative Flows(What can go wrong)	-Calendar sync fails.
Postconditions	<ul style="list-style-type: none"> • The event is added to the user's Google Calendar successfully. • If there are any errors such as Calendar sync fails: the system display message "Unable to add event to your calendar. Please try again later."

3.6 Non-Functional Requirements

Non-functional requirements ensure that **Bullbul** system performs effectively under various circumstances, guaranteeing reliability, security, and user satisfaction.

3.6.1 Performance Requirements

- **Response Time:** The platform must respond to user requests (e.g., opening an event page or submitting rating) within 2 seconds under normal load conditions.
- **Scalability:** The system must handle up to 500 concurrent users without significant performance degradation.
- **Throughput:** The platform should process up to 50 event updates per second without exceeding 80% server CPU usage.

3.6.2 Dependability Requirements

- **Availability:** The system must achieve 99.9% uptime, ensuring users can always access the platform except during scheduled maintenance.
- **Data Backup:** Event and user data must be backed up daily, with recovery capabilities ensuring data restoration within 2 hours after any failure.
- **Fault Tolerance:** The platform should maintain functionality during minor server or network failures by implementing redundancy and failover mechanisms.

3.6.3 Security Requirements

- **User Authentication:** All users must log in with secure credentials, with passwords stored using SHA-256 encryption.
- **Data Protection:** Sensitive user and event data must be transmitted using TLS encryption.
- **Access Control:** Only authorized administrators can modify event details or access the full database.
- **Audit Trail:** All administrative changes must be logged for auditing purposes.

3.6.4 Usability Requirements

- **User Interface:** The interface should adhere to accessibility guidelines, ensuring compatibility with screen readers and high-contrast themes for visually impaired users.
- **Ease of Navigation:** Users should find any event-related information within three clicks from the homepage.
- **Multilingual Support:** The platform must support both English and Arabic, with seamless language toggling.

3.6.5 Operational and Environmental Requirements

- **Platform Compatibility:** The system must operate on popular web browsers (e.g., Chrome, Firefox, Edge) and adapt to various screen sizes for mobile responsiveness.
- **Hosting Environment:** The platform should be hosted on a secure, scalable cloud infrastructure.

- **Environmental Considerations:** The system should minimize energy consumption by utilizing energy-efficient servers and optimizing resource allocation.

3.6.6 Maintainability Requirements

- **Code Modularity:** The system must follow modular programming practices to facilitate updates and extensions.
- **Documentation:** Comprehensive technical and user documentation must accompany the platform to assist with future maintenance.
- **Error Logging:** The system should log errors with sufficient detail to enable quick diagnosis and resolution by developers.

3.7 Data Requirements

Bullbol system manages and processes various types of data essential for its operation:

User Data:

- Attributes: Name, email, major, college, event registrations.
- Requirements: Must be securely stored and retrievable for user profile updates and personalized notifications.

Event Data:

- Attributes: Title, description, date, time, organizer, target audience, feedback ratings.
- Requirements: Must support (Create, Read, Update, Delete) operations with archival for past events.

System Logs:

- Attributes: User activity, administrative changes, error messages.
- Requirements: Must support audit trials and troubleshooting efforts.

Backup Data:

- Requirements: Daily snapshots of all critical data stored securely in a geographically separate location for disaster recovery.

4.0 Analysis and Design

4.1 Activity Diagrams

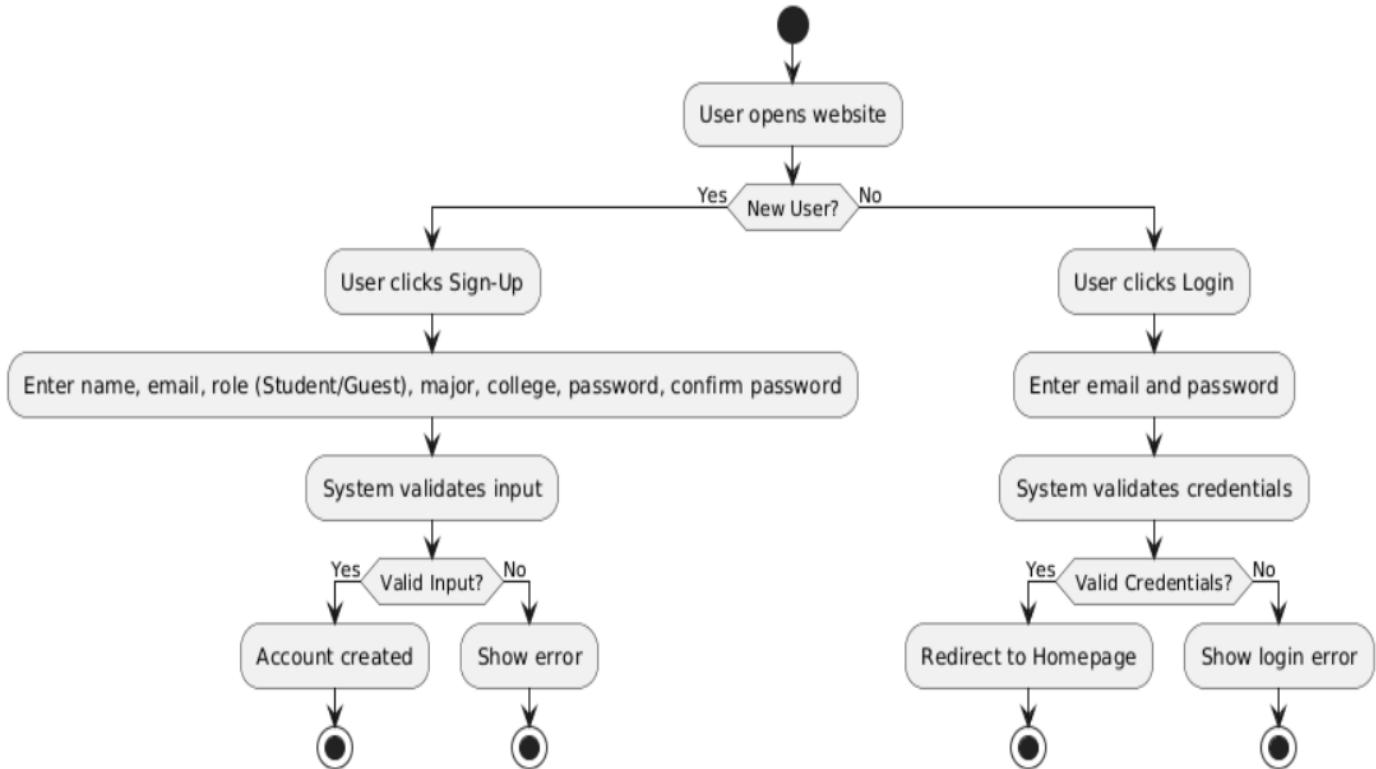


Figure 7 : User Registration and Login Activity Diagram:

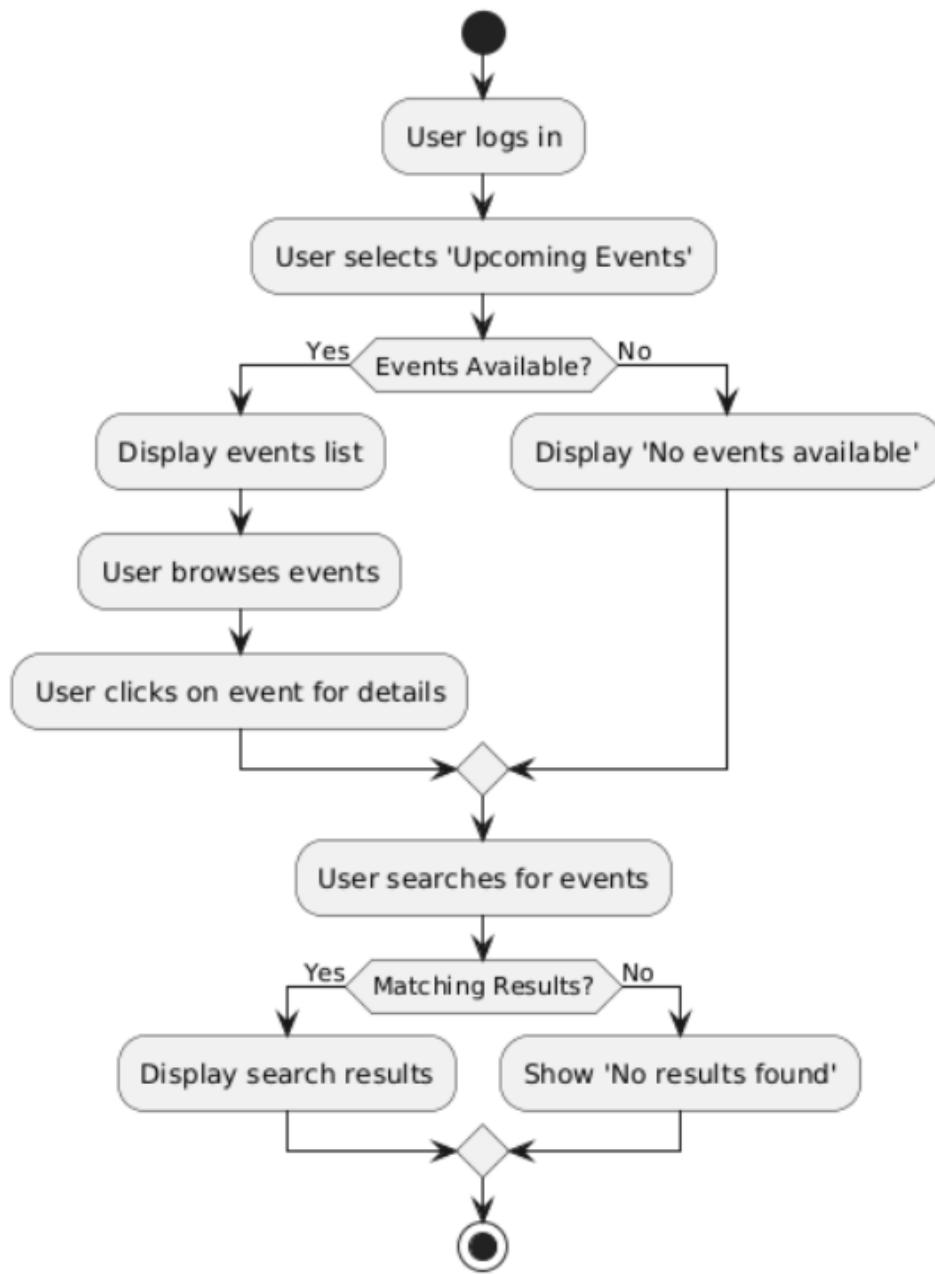


Figure 8 : Event Discovery Activity Diagram:

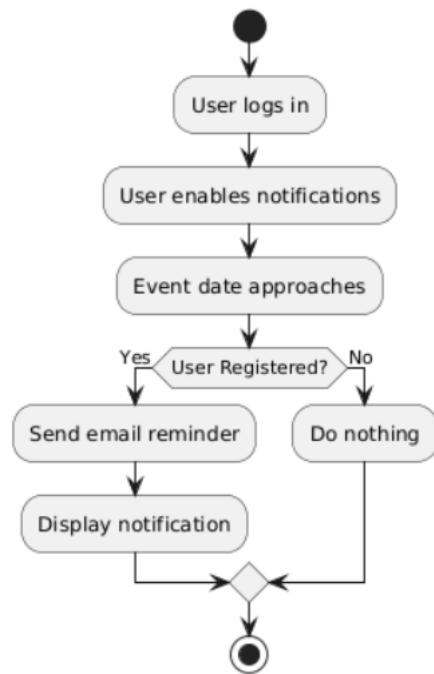


Figure 9 : Notifications and Reminders Activity Diagram

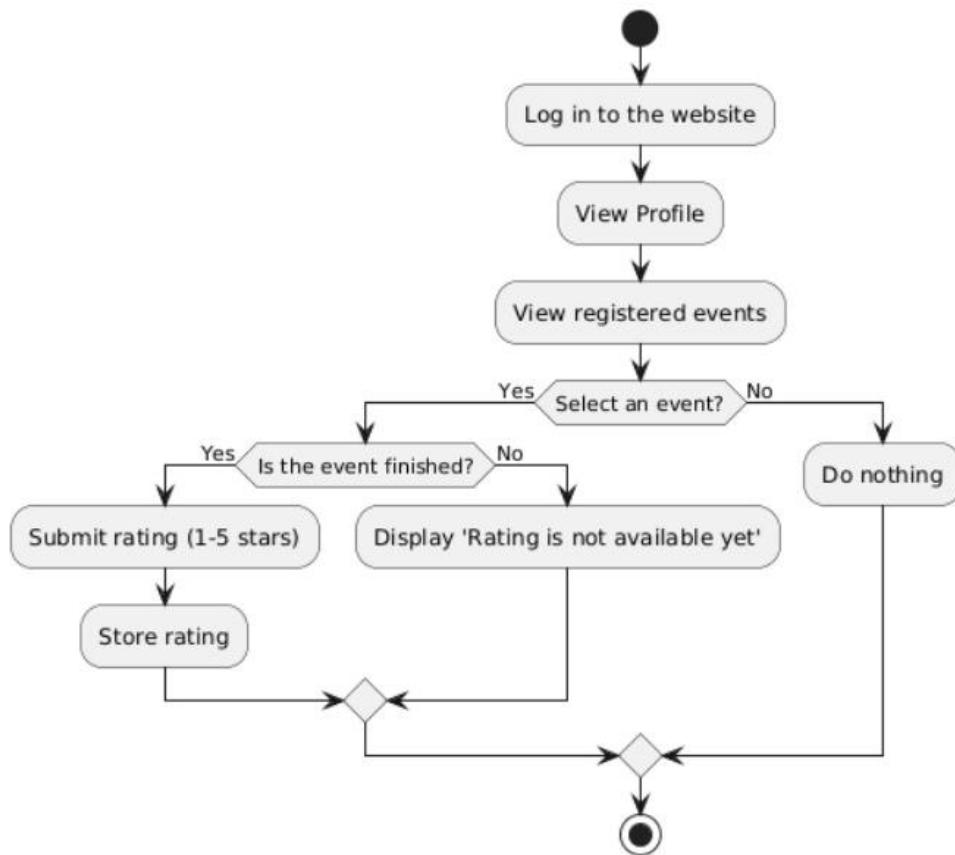


Figure 10 : Rating Events Activity Diagram

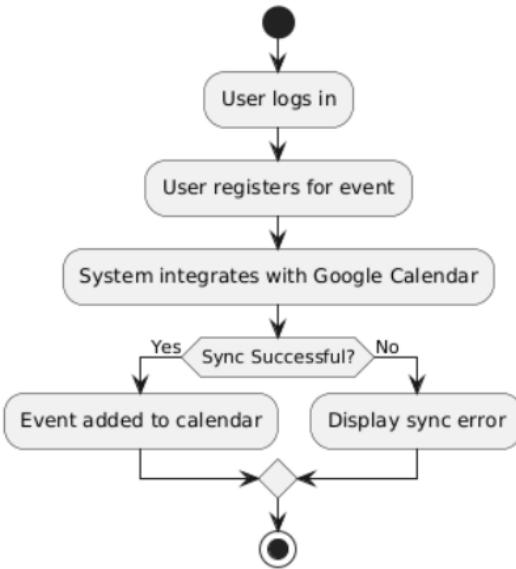


Figure 11 : Calendar Integration Activity Diagram

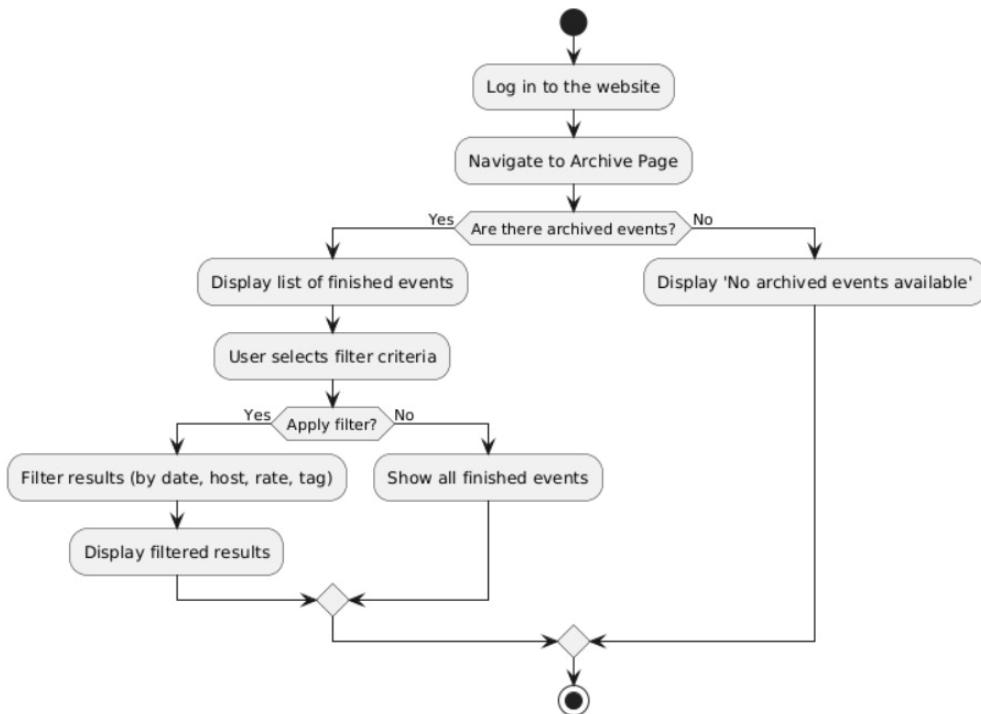


Figure 12 : Archive Activity Diagram

4.2 Sequence Diagrams

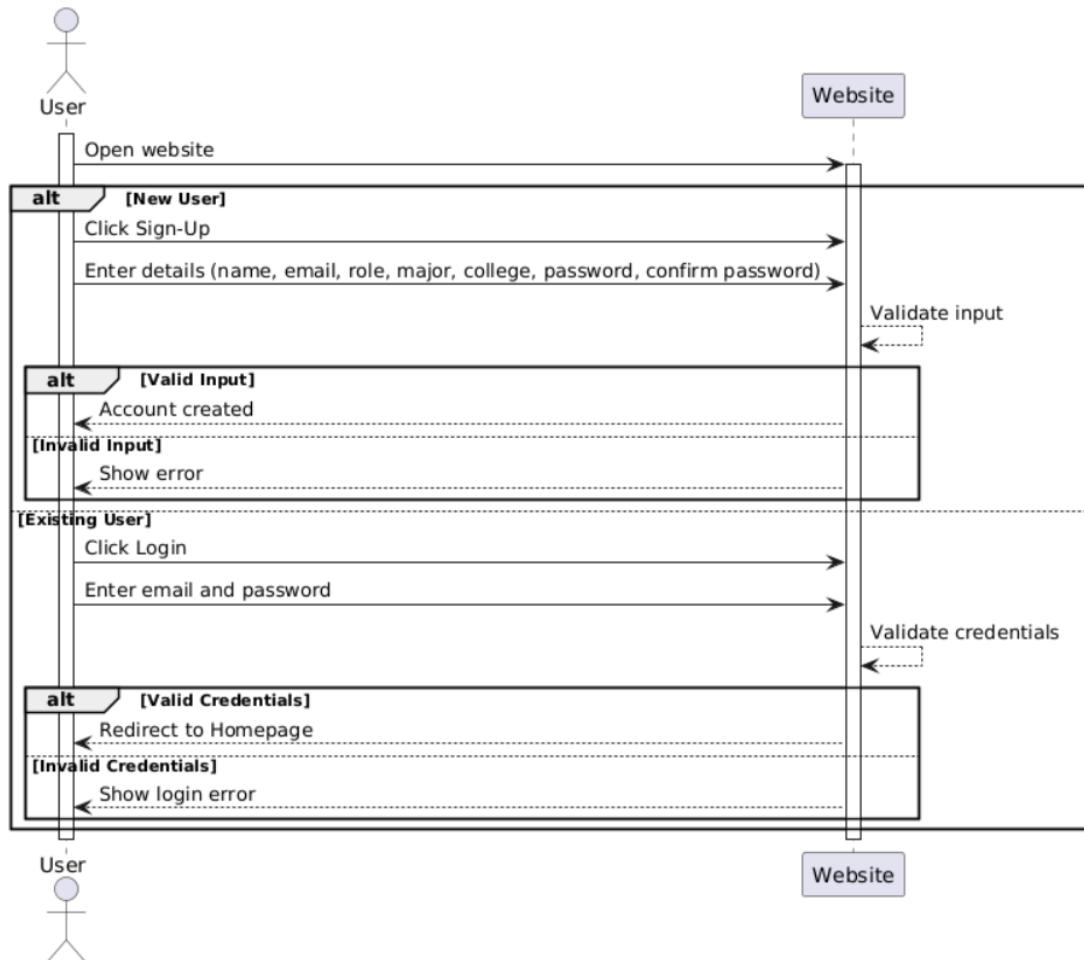


Figure 13 : User Registration and Login Sequence Diagram

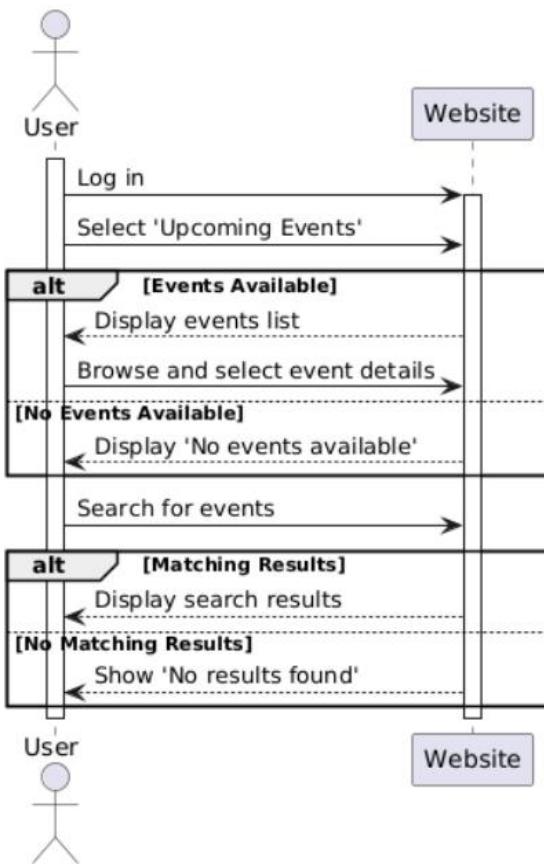


Figure 14 : Event Discovery Sequence Diagram

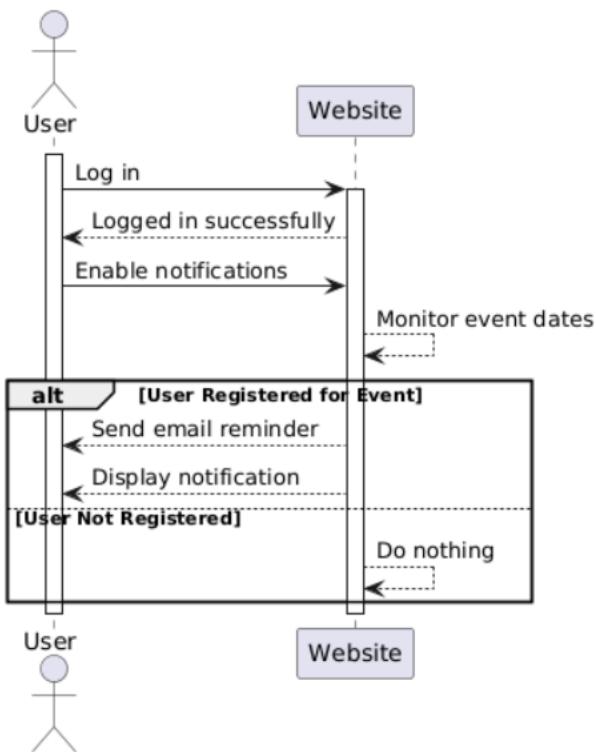


Figure 15 : Notifications and Reminders Sequence Diagram

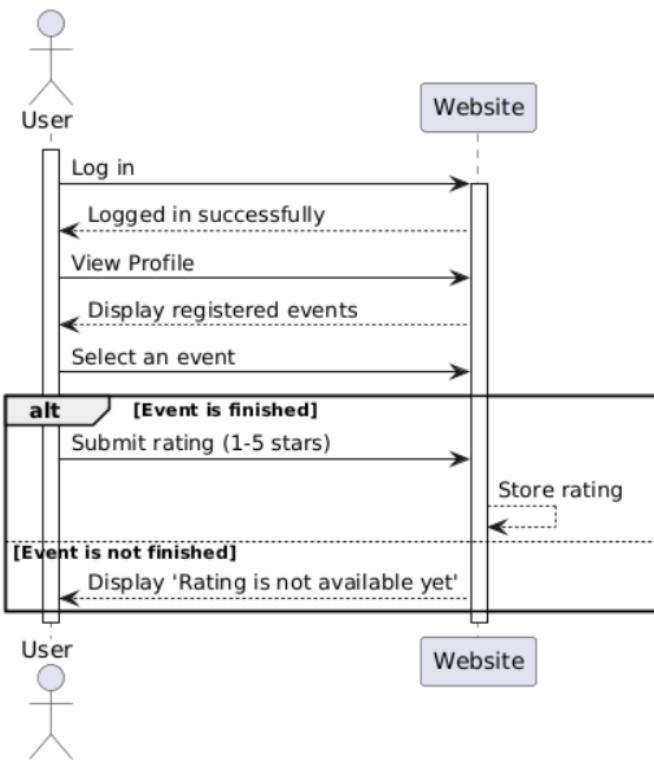


Figure 16 : Rating Events Sequence Diagram

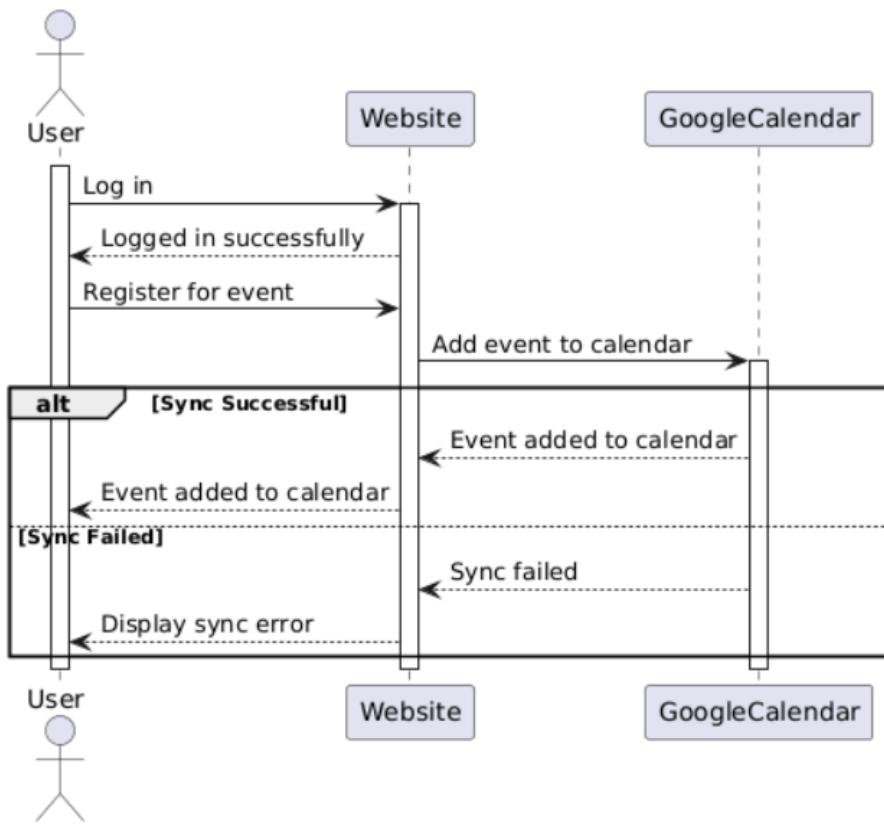


Figure 17 : Calendar Integration Sequence Diagram

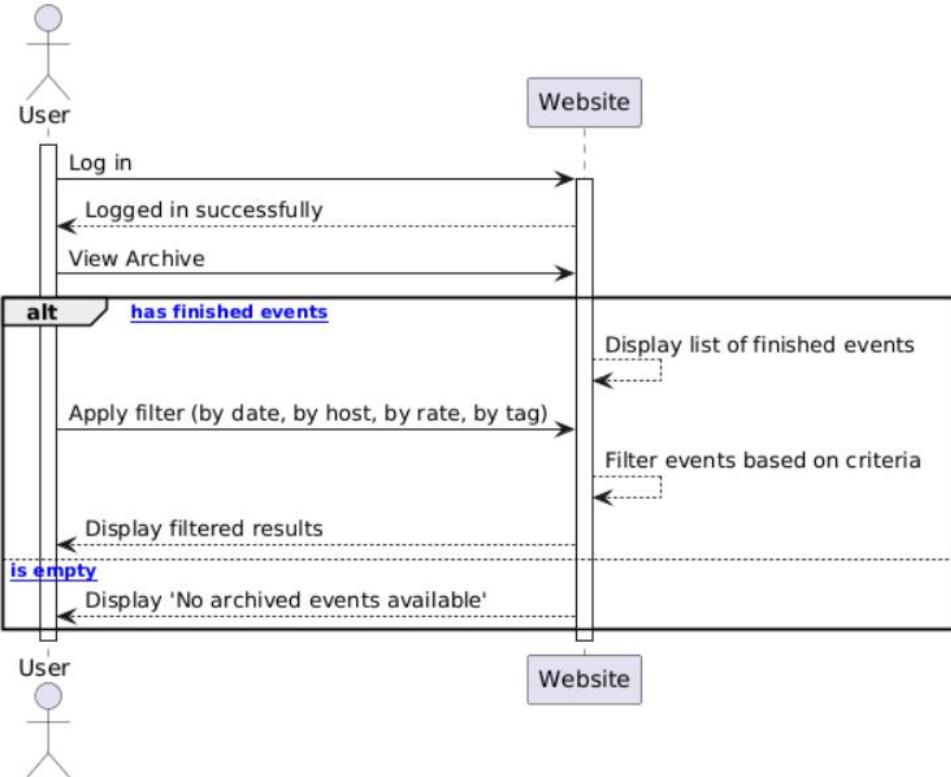


Figure 18 : Archive Sequence Diagram

4.3 Class Diagram

The class diagram provides a high-level design of the system, illustrating how different entities interact. It helps developers and stakeholders understand the architecture, making it easier to implement or modify the system.

Overview of Bullbol Class Diagram:

- Main Classes: user, event, notification, calendar integration.
- Relationships: Inheritance between classes like student and user, aggregation between classes like user and Notification.

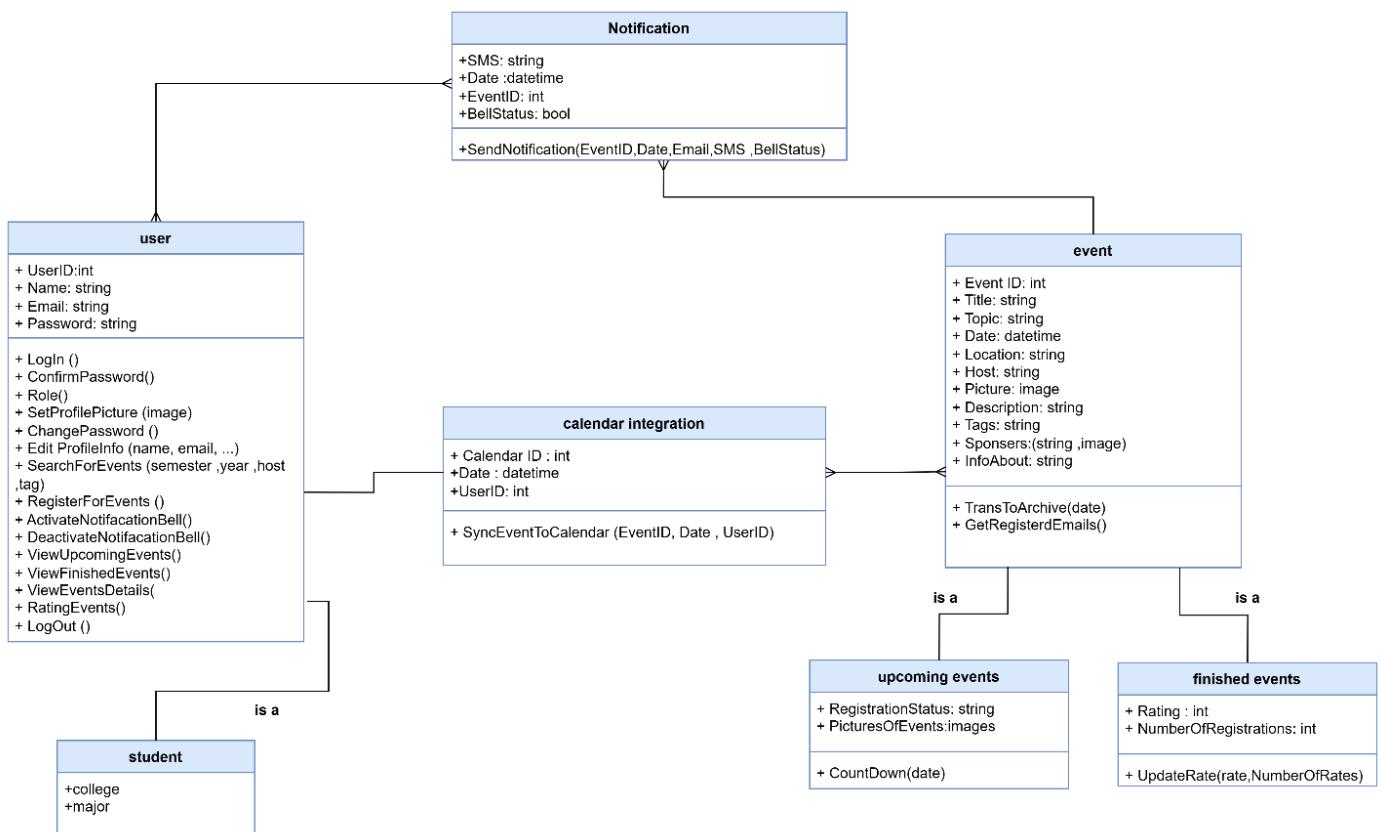


Figure 19 : Class Diagram

4.5 Graphical User Interface Design

- Home page

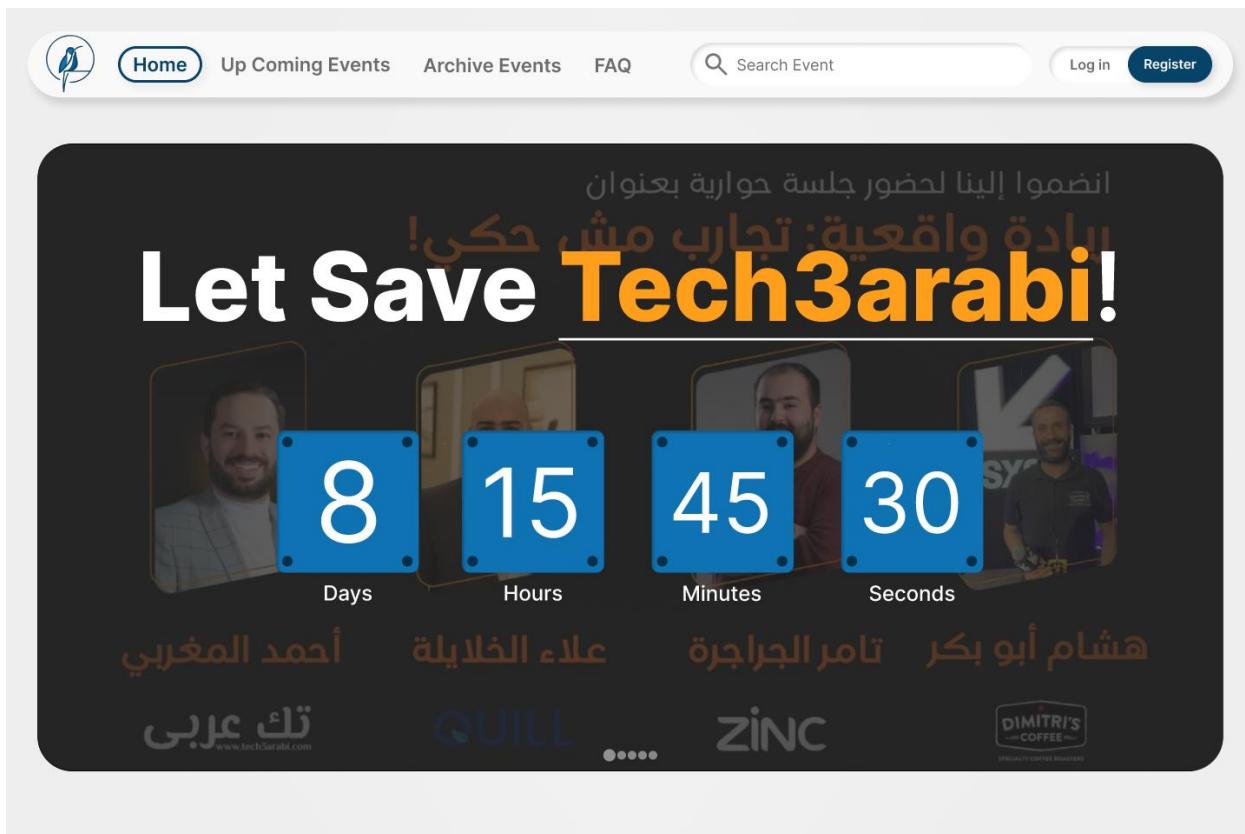


Figure 20 : Home Page _ (header, counter)

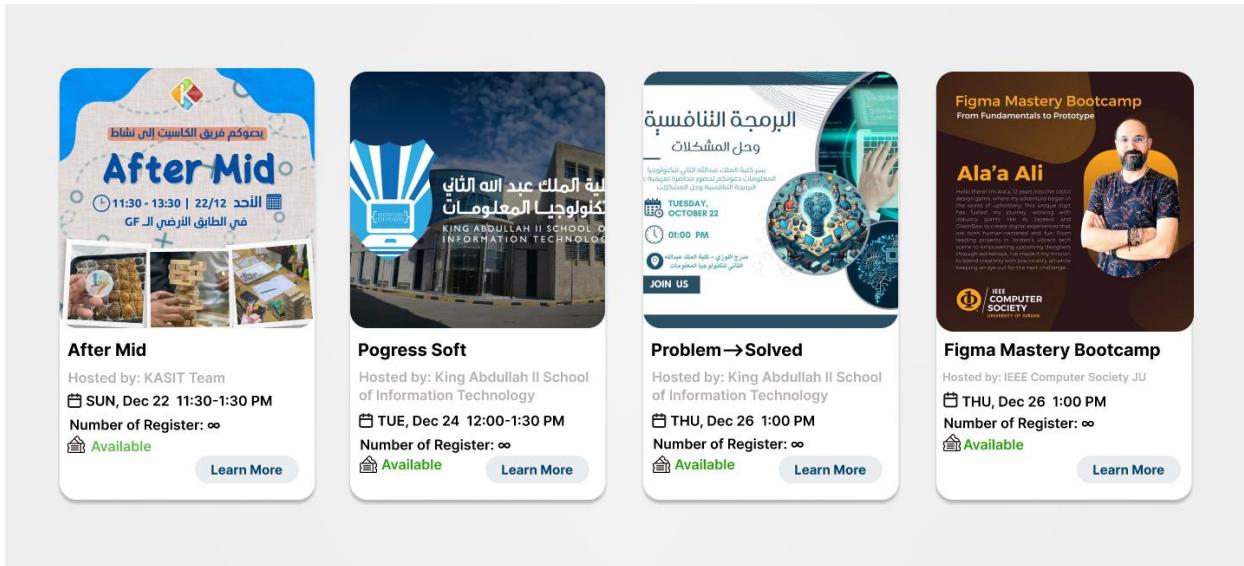


Figure 21 : Home Page Events suggestions

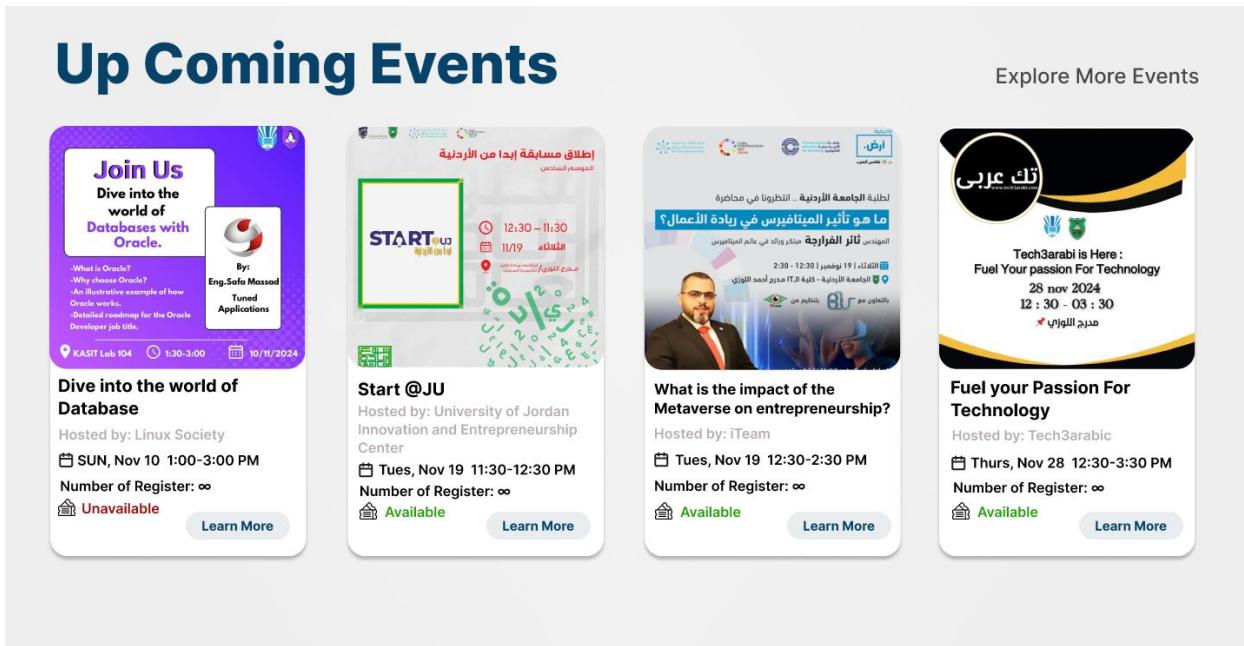


Figure 22 : Home Page _ Upcoming events

Archive Events

[See all events](#)



Problem Solving Training

Hosted by: ACM

Thurs, Oct 3

★★★ 4.8

[More Details](#)



The fresh men welcome party

Hosted by: Kasit College

Tues, Oct 22

★★★ 4.1

[More Details](#)



Hackathon Results

Hosted by: University of Jordan Innovation and Entrepreneurship Center

Tues, Nov 10

★★★ 4.5

[More Details](#)



Play it right

Hosted by: Ahl Alhemmah

Mon, Nov 18

★★★ 4.7

[More Details](#)

Figure 23 : Home Page _Archive events

We've Reached

More than +318 Work Successfully



Events will be Held



+300

Users Visited the Site



Events Successfully Archived

Figure 24 : Home Page _ Statistics



Figure 25 : Footer

- Registration Page

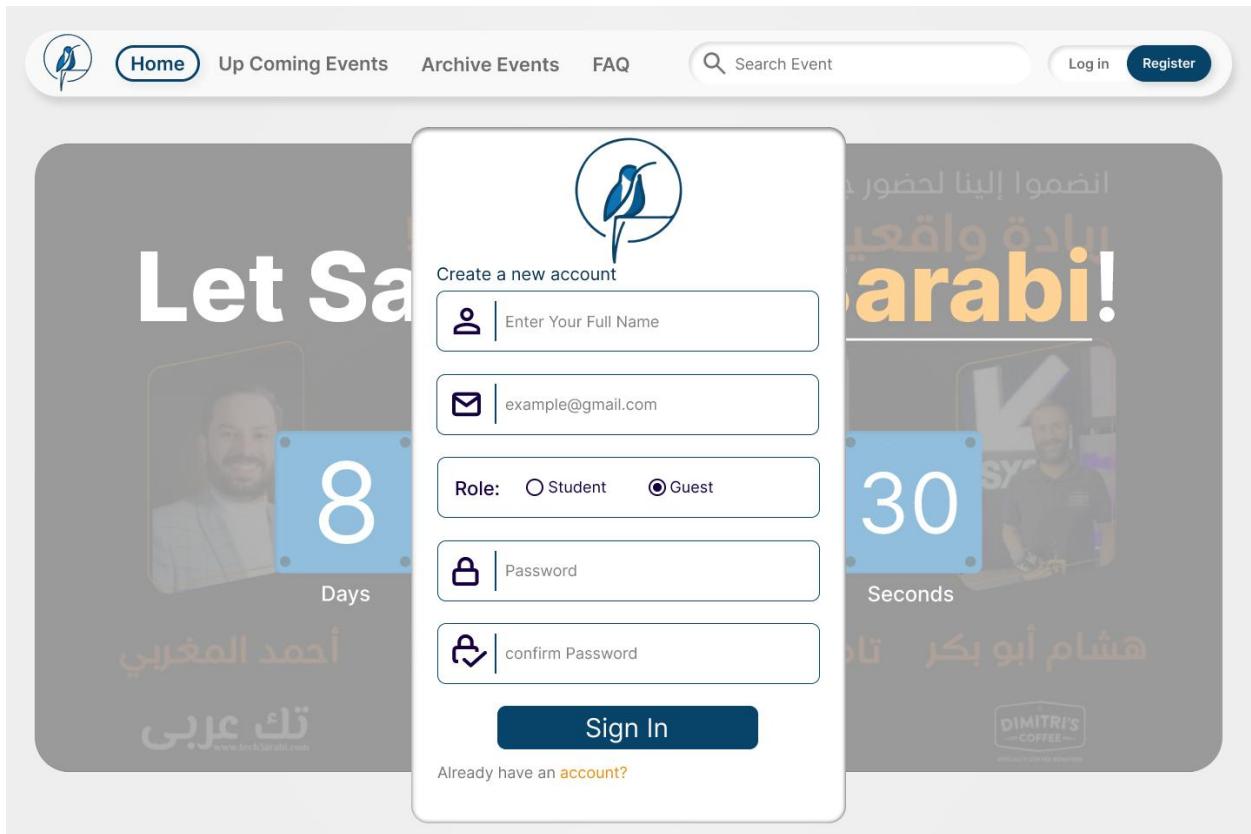


Figure 26 : Registration Page

- Log in Page

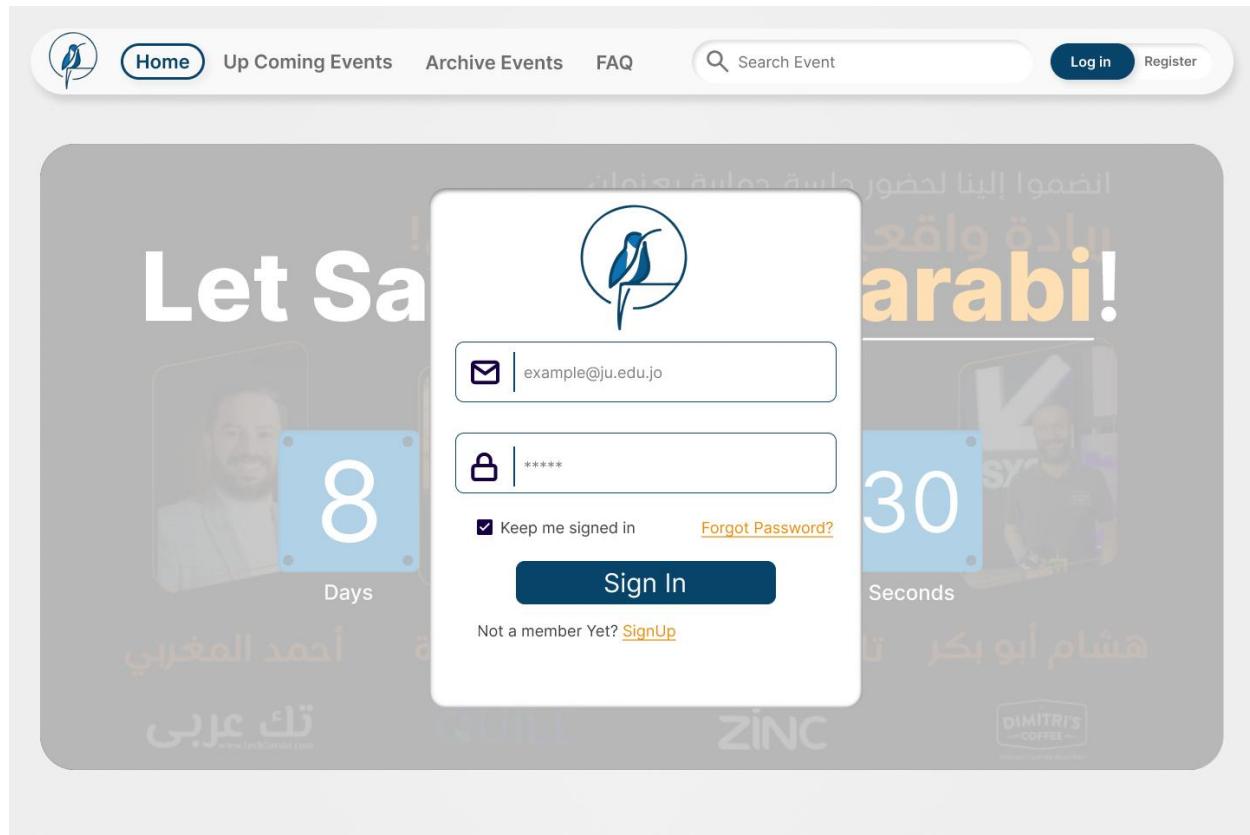


Figure 27 : Log In Page

- Profile Page (include calendar)

The screenshot shows a user profile page. At the top, there is a navigation bar with links for Home, Up Coming Events, Archive Events, and FAQ. A search bar labeled "Search Event" is also present. On the right side of the top bar are icons for notifications and user logout.

My Profile

 Shrouq Hassan Yousef Baker
CIS

Full Name: Shrouq Hassan Yousef Baker	Major: Computer Information System
Email: sro0224734@ju.edu.jo	College: King Abdullah II School of Information Technology
Role: Student	

[Edit profile](#)  [Change password](#) 

Figure 28 : Profile Page _ user info

- Upcoming events page

Up Coming Events

Semesters ▾ Years ▾ Host ▾ Tags ▾

- Join Us**
Dive into the world of Databases with Oracle.
By: Eng-Safa Massad
Tuned Applications
- Start @JU**
Hosted by: University of Jordan Innovation and Entrepreneurship Center
Tues, Nov 19 11:30-12:30 PM
Number of Register: ∞
Available
- What is the impact of the Metaverse on entrepreneurship?**
Hosted by: iTeam
Tues, Nov 19 12:30-2:30 PM
Number of Register: ∞
Available
- Tech3arabi is Here :**
Fuel Your passion For Technology
28 nov 2024
12 : 30 - 03 : 30
مدرج الموزع

Figure 29 : Upcoming events Page

- Archive page

Archive Events

Semesters ▾ Years ▾ Rate ▾ Host ▾ Tags ▾

- Problem Solving Training**
Hosted by: ACM
Thurs, Oct 3
★★★ 4.8
- The fresh men welcome party**
Hosted by: Kasit College
Tues, Oct 22
★★★ 4.1
- HACKTHON RESULTS**
Sunday 2:00PM
10 November 2024
مدرج الموزع كلية الـIT
- Play it right**
Hosted by: Ahl Alhemmeh
Mon, Nov 18
★★★ 4.7

Figure 30 : Archive Events Page

- FAQ Page

The screenshot shows a website's header with a logo, navigation links for Home, Up Coming Events, Archive Events, and a highlighted FAQ link, a search bar, and login/register buttons. Below the header, the word "FAQ" is prominently displayed in large blue letters. Underneath, there is a list of six questions, each preceded by a plus sign, indicating they are expandable:

- What is this website about?
- What are the goals behind it?
- What does it offer?
- Who are the target people?
- Why our website?
- How to register for an event?

Figure 31 : FAQ Page

- Event page (upcoming, past)

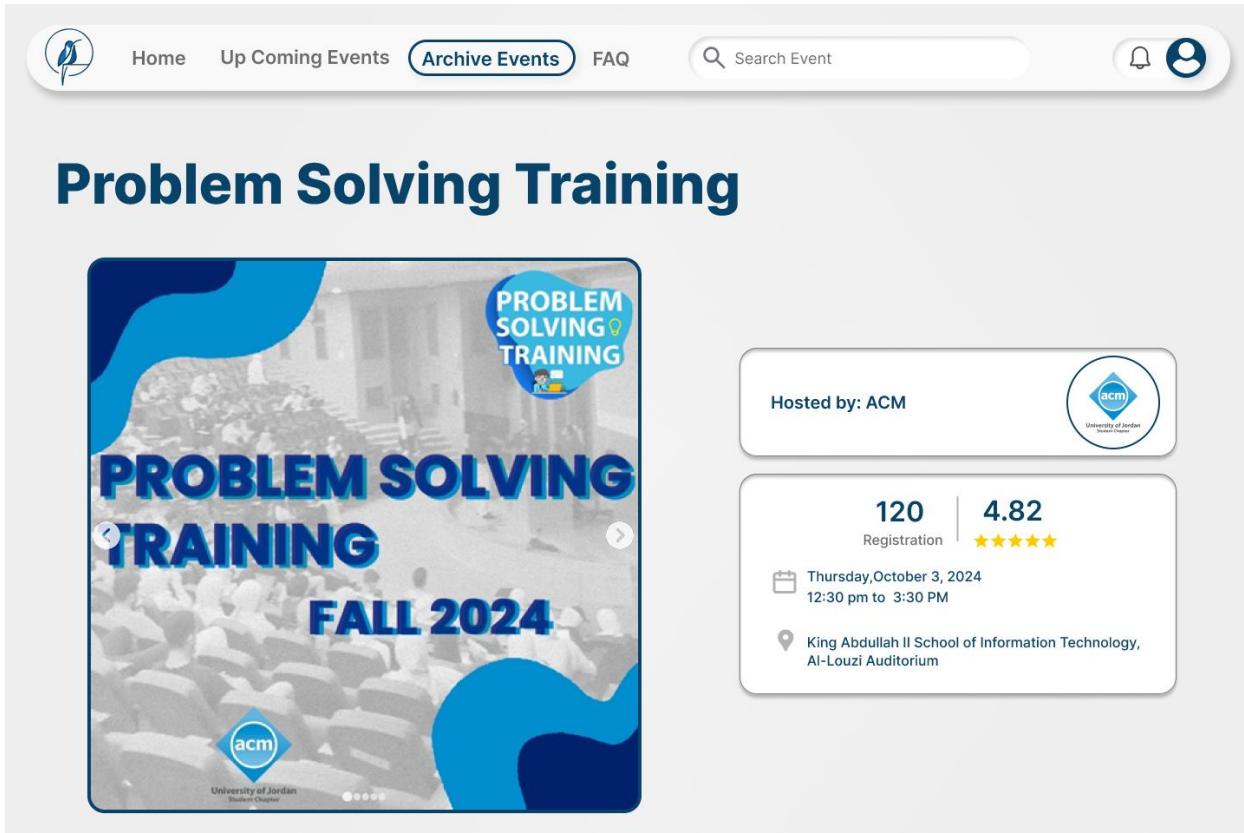


Figure 32 : Past Event Page

About Event

On Thursday, October 3, 2024, the problem-solving skill training sessions organized by the ACM team at King Abdullah II School of Information Technology concluded successfully. The sessions attracted over 200 enthusiastic students from various universities, highlighting the strong interest in developing critical thinking and analytical skills. Spanning three intensive weeks, the program offered a total of 18 hours per level, delivered across nine interactive and engaging lectures. This initiative reflects the ACM team's commitment to fostering a collaborative and innovative learning environment. It provided participants with hands-on experience, equipping them with tools to approach complex challenges effectively. We are incredibly proud of our team for their dedication and hard work in organizing this impactful event. A special thanks to all the trainers, volunteers, and supporters who played a vital role in making this achievement possible. Your efforts have left a lasting impact on the students and the community.

Tags: Problem Solving, Training, Sessions, Critical thinking, Acm

Figure 33 :About Past Event

Event Agenda

Thursday, Oct 3

- ⌚ 12:30 - 12:45 Opening and Welcome Speech
- ⌚ 12:45 - 2:00 Problem-Solving Challenge
- ⌚ 2:00 - 2:15 Break and Coffee Time
- ⌚ 2:15 - 3:15 Q&A Session and Reflections
- ⌚ 3:15 - 3:30 Closing Remarks and Certificate Distribution

Figure 34 : Past Event _ Agenda

Speakers



Thanks to our Sponsors



Figure 35 : Event Spensers & speakers

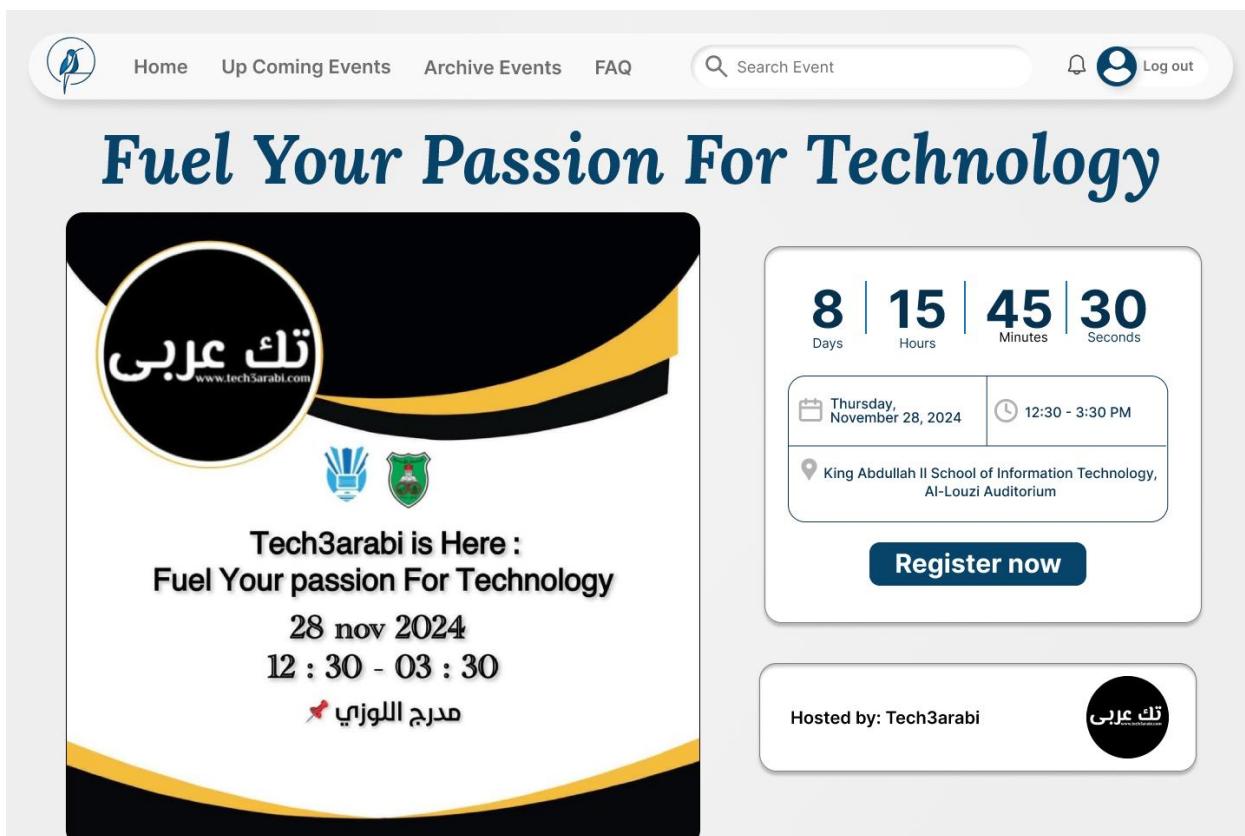


Figure 36 : Upcoming Event Page

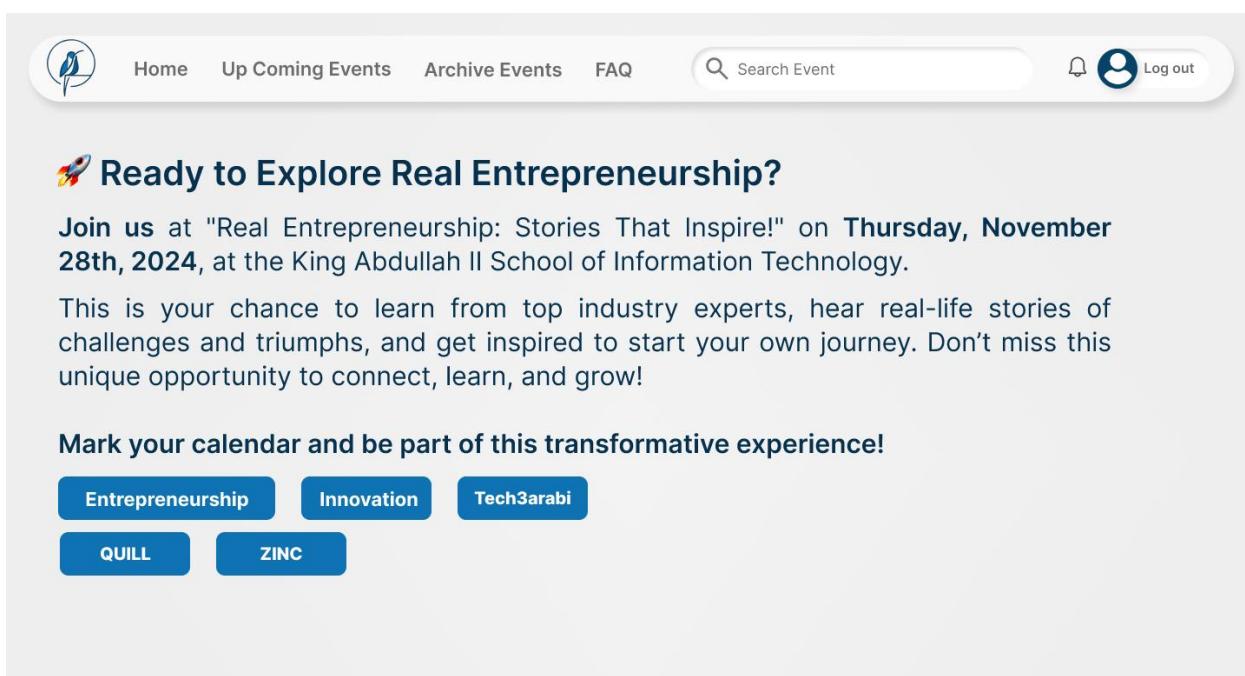


Figure 37 : About Upcoming Event

4.4 Architecture Design

Bullbol Architecture Design

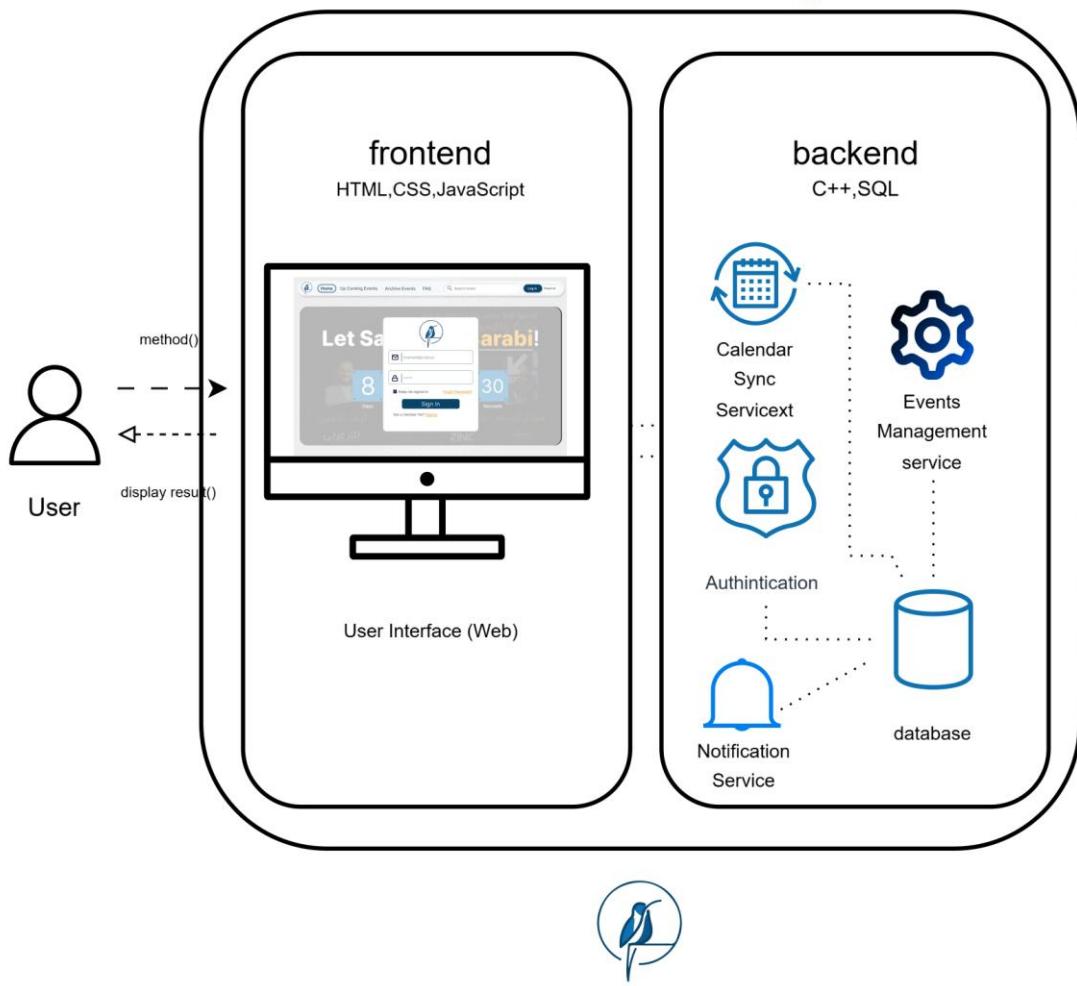


Figure 38 : Architecture Design

4.5 Classes and Component Design

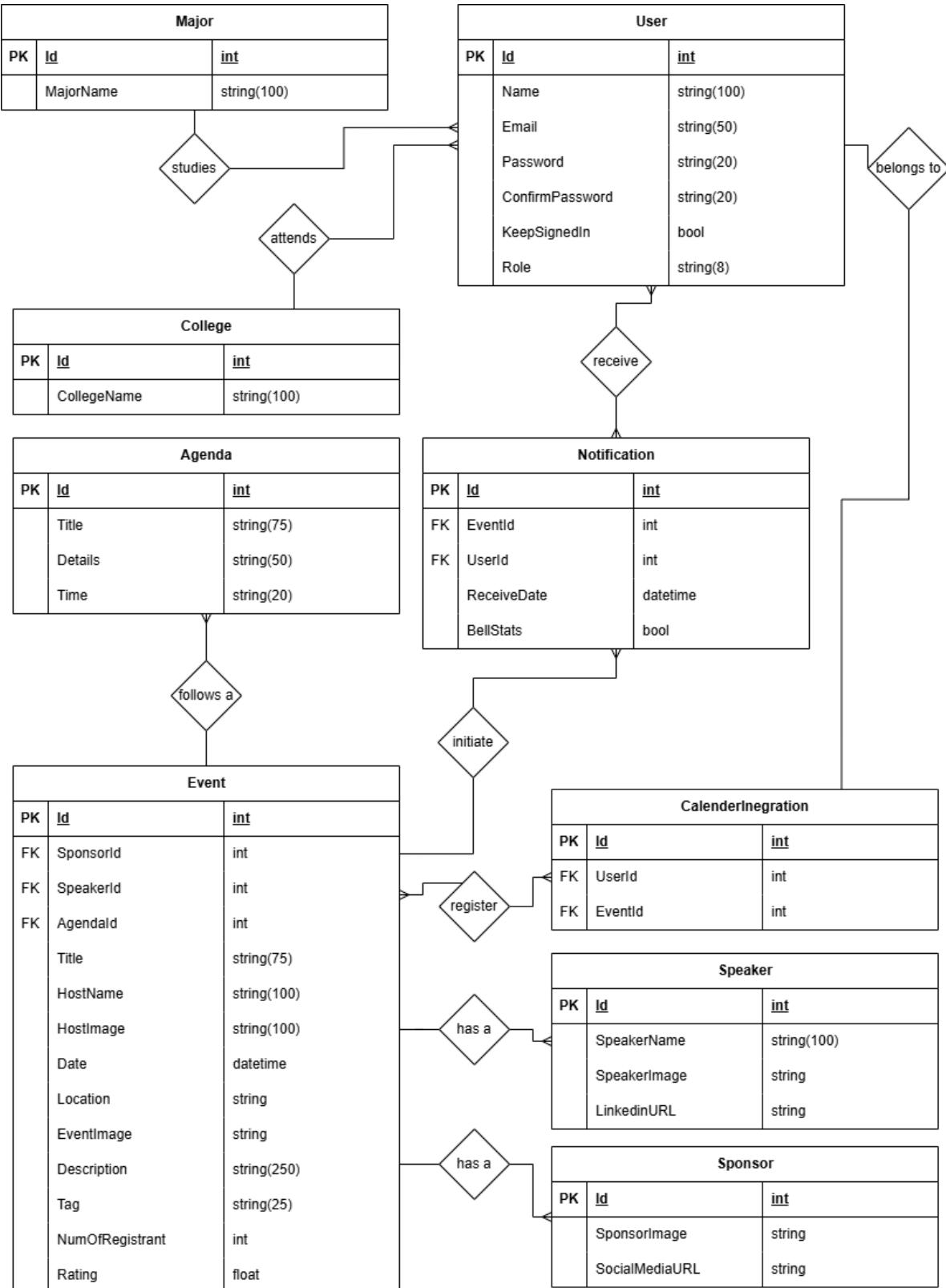


Figure 39: Class and Component Design

5.0 Implementation

To access the Bullbol project code, click the following link:

<https://github.com/Ysmn03/Bullbol>

6.0 User Manual

Click the underlined text to access the user manual and get instructions.

[user manual](#)

7.0 References

Tools:

- Draw,io
- PlantUML
- VS2022
- Notion, Google Drive: Dataset for Content
- Figma, Adobe Illustrator: Create designs and Edit photos.
- Google meet: Online meetings.
- GitHub: Manage, share, and review code.

A variety of tools, frameworks, and libraries, such as Bootstrap 5.2, are also available to assist us with our website development.

Books:

- [Ian Sommerville. Software Engineering, 10th Edition \(Latest Edition\). Addison Wesley, 2017.](#)
- [Ian Sommerville. Engineering Software Products: An Introduction to Modern Software Engineering, Global Edition, 1st Edition, \(Latest Edition\). Pearson, 2020.](#)

Sites:

- <https://jsyp.jordan.ieee.org/>

- <https://www.meetup.com/>
- <https://calendar.jo/>
- <https://www.eventbrite.com/>

Appendix

Acronyms

- CMS: Content Management System
- UI: User Interface
- UX: User Experience

Abbreviations

- TBA: To Be Announced
- EST: Eastern Standard Time
- GMT: Greenwich Mean Time

List of Tables

Table 1 : Development costs	9
Table 2: HW & SW Costs.....	10
Table 3 : Schedule Feasibility.....	11
Table 4 : Feasibility Study	12
Table 5 : Tools Required.....	21
Table 6 : Project Tasks	22
Table 7 : Project Dependencies, Deliverables and Milestones	23
Table 8 : Risk Identification and Assessment	26
Table 9 : Functional Requirements	34
Table 10 : Sign Up Use Case	37
Table 11 : Login Use Case	38
Table 12 : Change Password Use Case.....	38
Table 13 : Edit Profile Use Case.....	39
Table 14 : Notifications Use Case.....	40
Table 15 : Archive Use Case.....	41
Table 16 : Upcoming Events Use Case.....	42
Table 17 : Search Use Case.....	43
Table 18 : Rating Use Case.....	44
Table 19 : Callender Use Case.....	44

Table of Figures

Figure 1: Project Organization	18
Figure 2 : Project's Development Phase.....	20
Figure 3 : Gantt Chart for Project Schedule.....	25
Figure 4 : PERT Figure for Project Schedule	25
Figure 5 : Use Case Diagram	33
Figure 6 : Use Cases Integrations	36
Figure 7 : User Registration and Login Activity Diagram:	49
Figure 8 : Event Discovery Activity Diagram:	50
Figure 9 : Notifications and Reminders Activity Diagram	51
Figure 10 : Rating Events Activity Diagram	51
Figure 11 : Calendar Integration Activity Diagram	52
Figure 12 : Archive Activity Diagram	52
Figure 13 : User Registration and Login Sequence Diagram	53
Figure 14 : Event Discovery Sequence Diagram	54
Figure 15 : Notifications and Reminders Sequence Diagram.....	54
Figure 16 : Rating Events Sequence Diagram	55
Figure 17 : Calendar Integration Sequence Diagram	56
Figure 18 : Archive Sequence Diagram.....	56
Figure 19 : Class Diagram	57
Figure 20 : Home Page _ (header, counter)	58
Figure 21 : Home Page Events suggestions	59

Figure 22 : Home Page _ Upcoming events	59
Figure 23 : Home Page _Archive events	60
Figure 24 : Home Page _ Statistics	60
Figure 25 : Footer.....	61
Figure 26 : Registration Page.....	61
Figure 27 : Log In Page.....	62
Figure 28 : Profile Page _ user info	63
Figure 29 : Upcoming events Page	64
Figure 30 : Archive Events Page.....	64
Figure 31 : FAQ Page	65
Figure 32 : Past Event Page	66
Figure 33 :About Past Event	66
Figure 34 : Past Event _ Agenda.....	67
Figure 35 : Event Spensers & speakers.....	67
Figure 36 : Upcoming Event Page	68
Figure 37 : About Upcoming Event.....	68
Figure 38 : Architecture Design.....	69
Figure 39: Class and Component Design.....	70