ALBUKHARY INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTING AND INFORMATICS
SEMESTER 1 2023/2024

CCC2143
SOFTWARE ENGINEERING
GROUP PROJECT ASSIGNMENT (20%)

PROJECT TITLE	COMMUNITY ENGAGEMENT MANAGEMENT SYSTEM	
GROUP LEADER	WINA MUNADA	AIU22102163
GROUP MEMBERS	SITI SOLEHAH YUNITA RAHMAWATI	AIU21102378
	NURSYAZWANI BINTI MOHD RASHID	AIU22102243
	DESY KHALIDA MAHARANI	AIU21102283
	ANIS SOFIANA	AIU22102168

For Examiner's use only

ITEMS	MARKS
REPORT - 40	
PRESENTATION - 20	
TOTAL (60)	

TABLE OF CONTENTS

Introduction

As a result of the dynamic changes in stakeholder interactions and community dynamics, our group sets out on a software engineering journey to create the "Community Engagement Management System," a revolutionary solution. Effective platforms are becoming more and more necessary to enable meaningful connections and collaborations as societies continue to change. An outstanding example is the *Community Engagement Management System,* which was created to close the communication gap between stakeholders and various groups. The goal of this project is to provide a user-friendly, inclusive, and effective software solution that will enable communities to interact, communicate, and work together without any problems.

In considering this, our main goal is to produce a thorough report that not only describes the technical aspects of the system but also emphasizes the need of comprehending all the complex needs of community involvement. Selecting the best model to use at the start of the software development process guarantees that the methodology will work with the complexities of social business software applications. By means of comprehensive requirement analysis, our objective is to breakdown user and system requirements, distinguishing between functional and non-functional elements. The resulting system model includes several types of diagrams to give a comprehensive knowledge of the architecture, behavior, and interactivity of the system: activity diagrams, use case diagrams, sequence diagrams, class diagrams, and state diagrams.

Furthermore, our report delves into the core of the system's architecture design, with an emphasis on scalability, reliability, and maintainability. Adopting a component-based approach ensures modularity and reusability, factors critical to accommodating the diverse and evolving nature of community engagement. As a final crucial dimension, we address software testing strategies, ensuring the robustness of the Community Engagement Management System. By presenting this comprehensive report, we aim to equip stakeholders and developers with insights, strategies, and a clear roadmap for the successful development and deployment of a system poised to redefine community engagement in the digital age.

System Project Objectives

A set of detailed and ambitious goals is driving the creation of the Community Engagement Management System, with the goal of building a comprehensive, dynamic, and user-focused platform that will have a major influence on community interactions. Here are the objectives we have set for our project:

1. **Enhanced Community Connectivity:** Facilitate seamless and meaningful connections among community members and stakeholders.
2. **User-Focused Experience:** Prioritize user experience through an intuitive interface and accessible features.
3. **Comprehensive Requirement Fulfillment:** Address both user and system requirements comprehensively, encompassing functional and non-functional aspects.
4. **Visualized System Understanding:** Develop clear visualizations, including activity diagrams, use case diagrams, sequence diagrams, class diagrams, and state diagrams.
5. **Scalable and Robust Architecture:** Design a system architecture that is scalable, reliable, and easy to maintain.
6. **Modularity and Reusability:** Implement a component-based approach for modularity and reusability of system elements.
7. **Effective Software Testing:** Deploy rigorous software testing strategies to ensure the system's reliability and robustness.

System Requirement Analysis

In this section, for the Community Engagement Management System we present a comprehensive analysis of the System Requirement Analysis user, followed by detailed functional and non-functional system requirements:

1. **User Requirement**
   - **Functional Requirement**
     1. Customers should be able to register on the platform effortlessly, providing necessary information.
     2. Users must create profiles with personal details and preferences.
     3. Communities should be easily accessible, and users must have the ability to join and participate.
     4. Community administrators should have tools for moderation, analytics, and member management.
     5. A real-time messaging system must facilitate instant communication within communities.
     6. Users should be able to share updates, announcements, and multimedia content.

   - **Non-Functional Requirement**
     1. Profiles and community information should be secured with robust privacy measures.
     2. The messaging system should ensure timely and reliable communication.
     3. Content sharing features should provide a seamless and responsive user experience.

2. **System Requirement**
   - **Functional Requirement**
     1. The system should provide a user-friendly registration process. Profiles must store and display user information and preferences.
     2. The platform should support the creation, management, and categorization of communities. Users must be able to join, participate in discussions, and share content within communities.
     3. Community administrators should have access to moderation tools, analytics, and user management features.
     4. Implement a real-time messaging system with customizable notification settings.

5. Develop functionality for users to share updates, announcements, and multimedia content within communities.

- **Non-Functional Requirement**

1. The system must maintain response times under 2 seconds for critical actions. Ensure system availability of at least 99.9%.
2. Support a minimum of 10,000 concurrent users and 100 active communities. The architecture should allow for future scalability.
3. Implement encryption protocols for secure data transmission. Conduct regular security audits to address vulnerabilities.
4. Adhere to industry-standard development practices. Ensure compliance with relevant data protection and privacy standards.

System Model

---

1. **Activity Diagram:** Attached, find an activity diagram for the project showing the flow of activities or processes in the Community Engagement Management System. It works perfectly for complex presentations like those of event management or user registration. Activity diagrams put forth a series of activities that are undertaken in particular tasks and show how one activity is dependent on another.

   The detailed activity diagram corresponding to the processes discussed in this report is available in the appendix (Appendix A).

2. **Use-Case Diagrams:** The use-case diagram for the Community Engagement Management System graphically represents the interactions of the system with the external actors, such as users and administrators plus the particular activities or use cases that link them to this system. It outlines a high-level system functionality, how different actors will interact with it and to specify what potential things the various actors can do within the system as an intended and expected behavior.

   The detailed use case diagram corresponding to the processes discussed in this report is available in the appendix (Appendix B).

3. **Sequence Diagram:** In the Community Engagement Management System project context, a sequence diagram denotes the chronological order that is utilized by the use-case scenarios where objects or components interact and exchange messages. They are very useful for capturing dynamics of the system and showing how actors and components interact in the system over a period of time. Sequences besides class diagram as part of UML are especially useful for understanding communication patterns in complex processes such as event registration or feedback submission. In this report we create 3 sequence diagrams: User Sequence Diagram, Admin Sequence Diagram, and Developer Sequence Diagram.

   The detailed sequence diagram corresponding to the processes discussed in this report is available in the appendix (Appendix C-E).

4. **Class Diagram:** It shows the structure of the Community Engagement Management System in terms of classes, attributes, and relationships. It acts as a blue-print for an organization's data structure when it tries to depict its modeled entities such as users, events, and feedback. Class diagrams are basic in systems design, providing the basis for database schema and object-oriented programming.

   The detailed class diagram corresponding to the processes discussed in this report is available in the appendix (Appendix F).

5. **State Diagram:** State diagrams represent all the possible states and transitions that are necessary in passing from one state to another. In the context of the Community Engagement Management System, it is possible to use the state diagrams in order to model the lifecycle of certain entities like the user accounts or the events. They did provide some insight into the behavior of these entities to ensure that response to different conditions and events in the system is correct as well as increased system reliability and robustness.

   The detailed state diagram corresponding to the processes discussed in this report is available in the appendix (Appendix G).

System Architecture Design

A. System Overview

1. Purpose

The Community Engagement Management System is a feature-rich web-based platform that facilitates community engagement activities in terms of planning, execution, and involvement. Stated differently, the primary goals of the Community Engagement Management System are to enhance project management and facilitate communication and teamwork in order to guarantee a thriving and cohesive community ecosystem.

2. Goals

a. Streamlined Community Engagement:

- Objective: This can promote the smooth design and implementation of community engagement initiatives.
- Implementation: Assist with project management, conversations, and group projects by providing resources and tools.

b. Enhanced Communication:

- Objective: Encourage community members to communicate effectively with each other.
- Implementation: Incorporate interaction features like message boards and direct messaging between users, administrators, and even administrators and administrators. This may promote honest and open communication.

c. User Empowerment:

- Objective: Enable users to take an active role in neighborhood projects.
- Implementation: Implement features that will generate an online certificate for each user who participates in community involvement, as well as user-friendly interfaces, personalized profiles, and features that recognise actions.

d. Adaptability and Customization:

- Objective:Attend to the various needs that the community has.
- Implementation: Provide tools and modules that can be customized to fit the needs of particular tasks and occasions. This also will be able to execute an idea or programme

centered around the Sustainable Development Goals, or even something that would benefit the environment and society

3. **Stakeholders**

- **User:** Users of the Community Engagement Management System can safely log in, browse and sign up for events, communicate privately with other users or administrators, modify their profiles, and quickly look up pertinent community engagement content. These streamlined features guarantee an interface that is easy to use, encouraging efficient and active involvement within the platform.

- **Administrators:** With the Community Engagement Management System, administrators can effectively manage user accounts, supervise community events, have direct conversations with users one-on-one or in groups, create and publish events, generate reports from those events, set up alert settings for events, and curate content for the platform's dashboard. These streamlined features enable administrators to keep the community area responsive and orderly, promoting efficient communication and smooth event preparation and execution.

- **Developers:** The Community Engagement Management System relies heavily on developers to create and update the platform, maintain security, and repair vulnerabilities. Their efforts guarantee the dependability of the system, incorporate fresh functionalities, and expedite resolution of problems, all of which enhance the user experience.

B. Key Components

1. User Profiles
   - **Purpose**: Enable users to establish and manage personalized profiles.
   - **Implementation**: User profiles displaying participation history, preferences, and customizable details.
2. User Engagement Analytics
   - **Purpose**: Collect and analyze user engagement data for improvement.
   - **Implementation**: Implement analytics tools to track user interactions, popular content, and overall user engagement.
3. User Notifications

- **Purpose**: Alert users about important updates, discussions, and project activities.
- **Implementation**: Real-time notification system for new messages, replies, and community announcements.

C. System Design

1. **Architecture**: For the frontend, use AngularJS or React, and for the backend, Java, Spring, and Hibernate.
2. **Technologies**
   - **Frontend:** For responsive and dynamic user interfaces, employ HTML, CSS, and JavaScript in conjunction with a contemporary frontend framework such as AngularJS or React.
   - **Backend:** Use the Spring framework and Java to implement strong backend features.
3. **Database**: For organized and safe data storage, go with a dependable Relational Database Management System (RDBMS), such as MySQL or PostgreSQL.
4. **Real-time Communication**: For real-time communication that guarantees fast updates and notifications, leverage WebSocket technology.
5. **Hosting**: For best performance and availability, deploy the system on a scalable and dependable cloud platform like Microsoft Azure or AWS.
6. **Security Measures**: Use HTTPS to transmit data securely and always update and patch software on a regular basis to fix security flaws.
7. **Scalability**: Build the system with load balancing and effective resource management in mind to handle growing user loads.
8. **Future Improvements**: Arrange the creation of a mobile app to increase accessibility. Moreover, take into account other features like chat support and connection to other community services.

Component-Based Model

The first diagram below represents the component-based model for the Community Engagement Management System. This model has been designed to be modular and reusable which quality development, deployment, and maintenance activities would work out most cost-effectively. Each of the components comes with special characteristics and interfaces representing its roles in this system.

Characteristics:

- Composable: Each component should be to have their composable properties in that they can get combined or rather interleaved with each other to build a working system. For instance, the Login Component may be composed with the Event Management Component to provide the user authentication services whenever a user is registering for an event.
- Deployable: These are components that can be deployed independently as a single entity. Independent deployment of the components brings out a flexible scenario in the deployment of the system and scalability within the need of the organization.
- Documented: Each of the components is rightly documented which provides guidelines about their respective functionality and how they can be utilized through a different interface.
- Independent: They are independent and do not impact on each other for that during the system development and maintenance there is very less chance of any conflict and clash between different components.
- Standardized: It ensures that the components are given standardized interfaces and protocols that enable compatibility and consickness in the entire CEMS.
- Reusable: The style of componentization is such that the components become reusable. For instance, the Communication Component can be reused again with some modification to meet other communication requirements within the system.

Interfaces:

- Provides Interface: For instance, each component provides specific interfaces by which their functionalities are exposed into either other components or external systems. For example, the Event Management Component provides interfaces such as checking event, registering an event, and creating events.

- Interface Required: Some interfaces may be required based on the functionalities of other components a component needs access to. For example, in the case of the Error Display Component, the interfaces will need to be such through which error messages from other components can be captured and displayed.

Component Model:

These components interact in a manner that is defined by the component model within CEMS:

- Interfaces: Components expose well-defined interfaces making clear what functionality a particular service provides and what type of input it requires.
- Use: Components use other components as needed. For example, an Event Management Component may use a Login Component to authenticate the user.
- Deployment: The components can be deployed and managed independently. They can run within their container or server and can interact using interfaces that are standardized.

The component design is composable, deployable, documented, independent, standardized and reusable. The design for each of the components is modular which enables a flexible architecture using the CEMS component model. These components form an efficient synergy with well-defined interfaces and a clear component model retaining a robust Community Engagement Management System.

System Testing Strategy

Creation of a Community Engagement Management System (CEMS), indeed, is a very elaborate process that requires one to be strategic when setting up, testing, and rolling out such a system. This is intended to enhance and facilitate an organization's interaction with its communities as such a CEMS ensures that the system performs flawlessly as required. The strategy covers three crucial phases of testing which include the development testing, release testing and user testing all with variant purposes in the CEMS development process.

1. **Development Testing**

   Development Testing includes unit testing, component testing and system testing is the basis of our policy that examines each of the components of CEMS for anticipated functionality. Unit testing checks independent units like Login, Edit Profile, and Verify Password in isolation to establish correctness. Component testing is performed to validate the interaction between components such as Event Management and Communication for proper integration. System testing verifies CEMS in its completeness and confirms whether all interconnected components work properly. It also identifies whether problems of integration, any issue pertaining to data flow as well as any lack in functionality subsist.

2. **Release Testing**

   Release testing includes requirements-based testing, scenario testing, performance testing checks and ensures that the CEMS satisfies the requirements for which it is specified to have been designed and works fine under normal scenarios as well as performs good. On the other hand, Requirements-Based Testing clearly ensures flows of actions according to preset criteria, and Scenario Testing models user functions within a system across components involving even things such as Event Management, User Management or Communication. Performance Testing is concerned with speed and scaling of a system under different conditions. It is ensured that the CEMS should fulfill user's expectation, project requirements and exactly meets with optimum performance.

3. **User Testing**

   User testing includes alpha testing, beta testing, acceptance testing. User testing includes usability check and functional aspects of the CEMS. Alpha Testing, which takes place in-house

to flush the flaws and usage problems of components such as Feedback, Error Display, and Content Upload. Beta Testing, widens the test bed by allowing additional recruits into the test group by including the users from outside of the organization for deriving better feedback on the components such as Event Notification and Search Content. Acceptance Testing, performed by end users, ensures that the CEMS is as per organizational expectations and requirements. It includes the elements of Event Reports and User Management.

Roles and Responsibility of Project Members

In the successful execution of the Community Engagement Management System project, each project member will play a crucial role in contributing to the overall development report. The following statements outline the responsibilities for project members:

Name	Role and Responsibility
Wina Munada	Software Testing Strategies
Siti Solehah Yunita Rahmawati	System Project Objectives, Requirement analysis
Nursyazwani Binti Mohd Rashid	System Architecture Design
Desy Khalida Maharani	Component-based System
Anis Sofiana	System Model: Activity Diagram, Use Case Diagram, Sequence Diagram, Class Diagram, and State Diagram

Conclusion

In conclusion, the detailed analysis and planning of the Community Engagement Management System (CEMS) project have provided a sound foundation for development and successful implementation. Our dedication to the ideals of this project, the in-depth analysis of system requirements, effective design of a system model and architecture, as well as the development and implementation of a component-based model serve to lay ground for a CEMS that would be nothing short of being the greatest thing since sliced bread.

The system testing strategy that is outlined herein is meant to ensure the CEMS gets a thorough evaluation at all stages of its development, from the unit testing stage through the systems integration, and climaxes in the final acceptance by end users, and other stakeholders. This strategy will ensure that CEMS is able to run smoothly thereby meeting the users' expectations and corresponds with the organizational mission of facilitating interaction within the community that is meaningful. Having a well-defined roadmap and commitment towards quality assurance, the project of Community Engagement Management System is poised with the desired objectives, enhancing the aspects of community engagement efforts and thus ensuring a positive impact over the organization, and respective stakeholders.

References

Arsat, N. (2023). Component-based Development [Lecture slides]. Software Engineering Course, Albukhary International University. https://elearning.aiu.edu.my/pluginfile.php/31998/mod_resource/content/2/CCC2143%20WK8%20Component-Based%20Development.pdf

Arsat, N. (2023). Software Testing [Lecture slides]. Software Engineering Course, Albukhary International University. https://elearning.aiu.edu.my/pluginfile.php/32271/mod_resource/content/3/CCC2143%20WK9_10%20Software%20Testing.pdf

Halawati Abd Jalil Safuan. (2023). Requirement Engineering [Lecture slides]. Software Engineering Course, Albukhary International University. https://elearning.aiu.edu.my/pluginfile.php/29671/mod_resource/content/1/CCC2143%20Week%205.pdf

Halawati Abd Jalil Safuan. (2023). System Design [Lecture slides]. Software Engineering Course, Albukhary International University. https://elearning.aiu.edu.my/pluginfile.php/31433/mod_resource/content/1/CCC2143%20Week%207%20%282023%29.pdf

Coders, A. (2021, April 2). UML state machine diagram [Video]. In *YouTube*. https://www.youtube.com/watch?v=obLemkvbWr0

Master2Teach. (2020, April 30). Activity diagram - Step by step guide with example [Video]. In *YouTube*. https://www.youtube.com/watch?v=knM8BGY9yVI

Software, L. (2018, August 27). How to make a UML sequence diagram [Video]. In *YouTube*. https://www.youtube.com/watch?v=pCK6prSq8aw&list=PLUoebdZqEHTxpGCwKrb82cIvHNoNaBb4R&index=7

Software, L. (2023, August 11). UML class diagrams [Video]. In *YouTube*. https://www.youtube.com/watch?v=6XrL5jXmTwM&list=PLUoebdZqEHTxpGCwKrb82cI

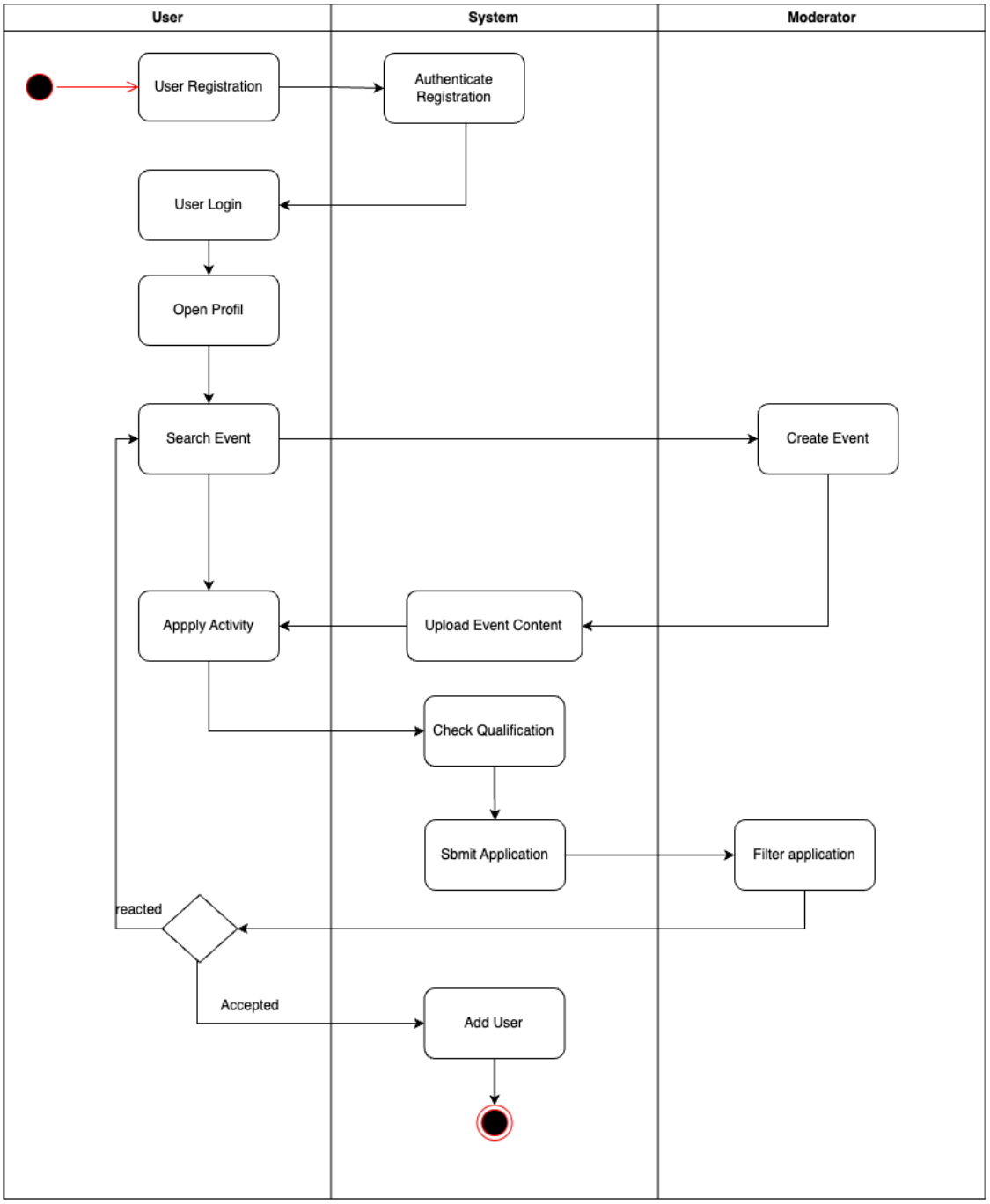
vHNoNaBb4R&index=5

Software, L. (2023b, September 21). UML use case diagrams [Video]. In *YouTube*. https://www.youtube.com/watch?v=4emxjxonNRI&list=PLUoebdZqEHTxpGCwKrb82cIvHNoNaBb4R&index=7

Appendix

Appendix A

Activity Diagram

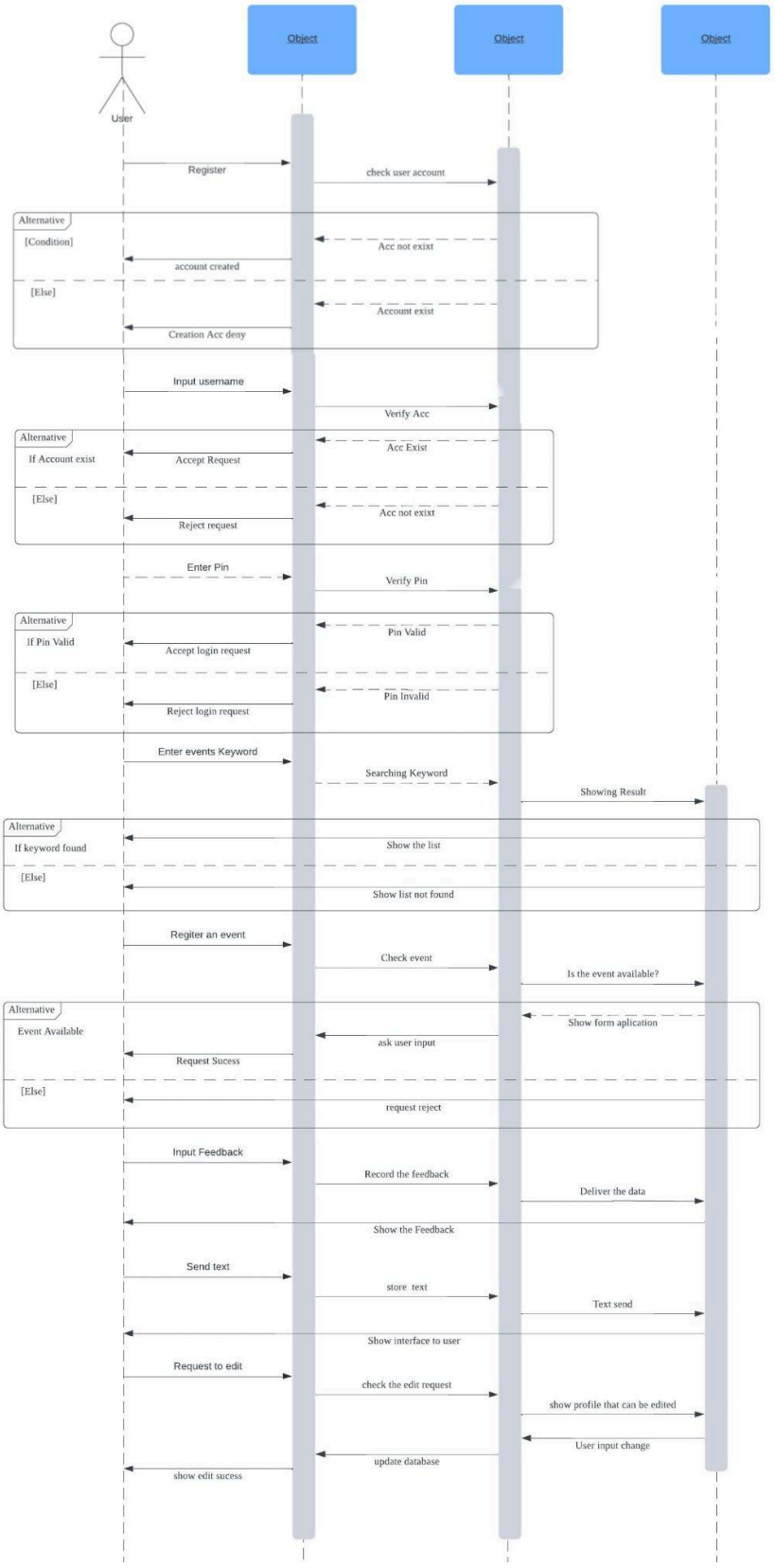


Appendix B

Use Case Diagram

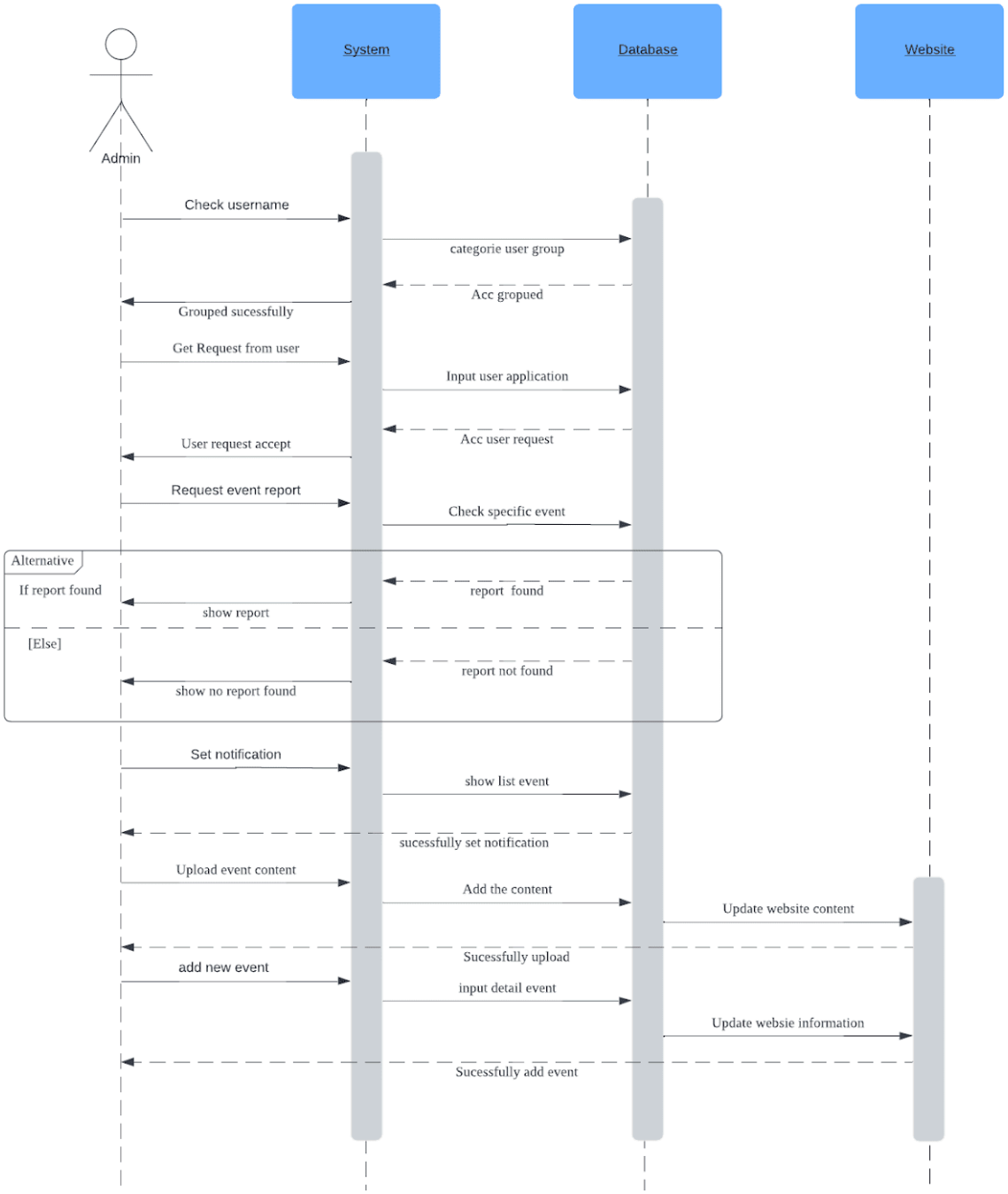
Appendix C

User Sequence Diagram



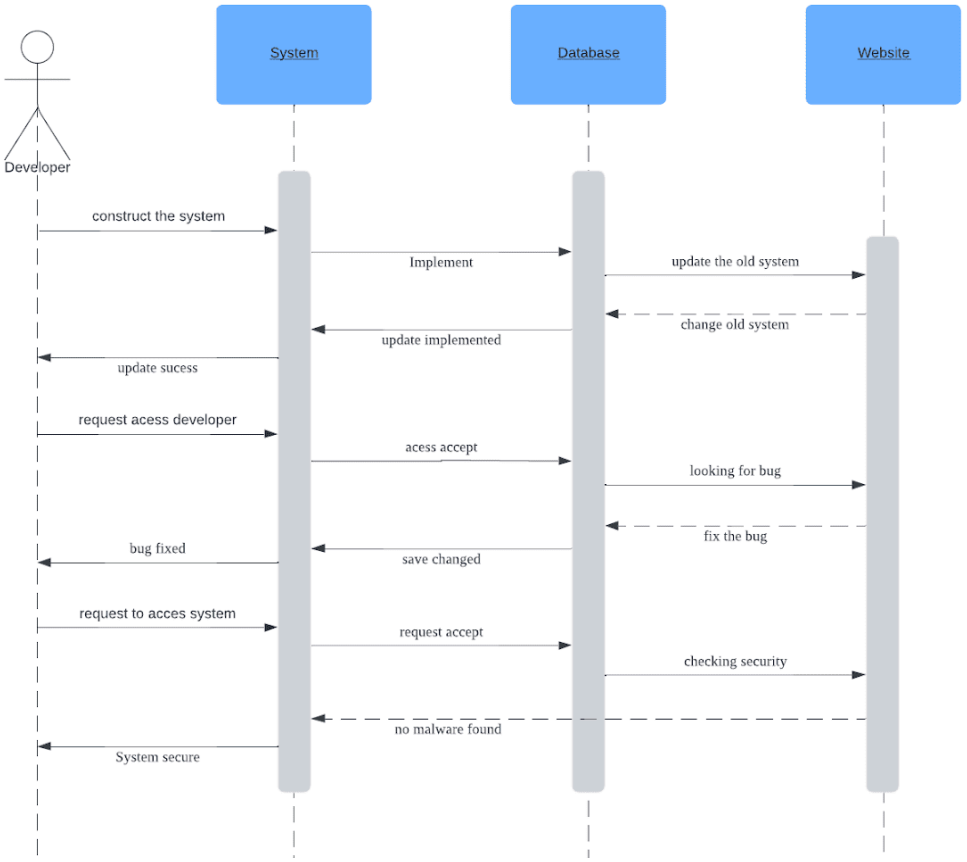
Appendix D

Admin Sequence Diagram



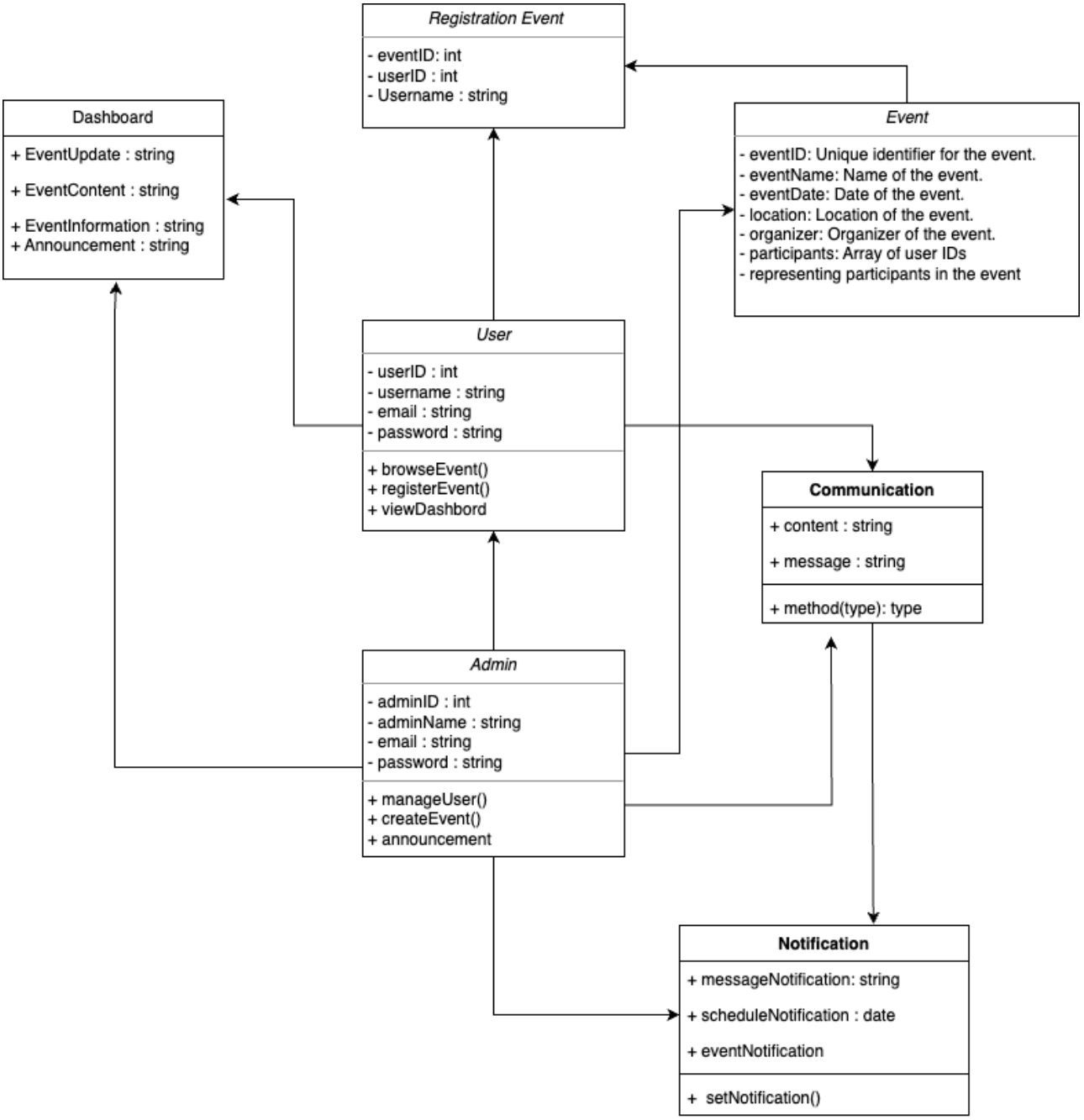
Appendix E

Developer Sequence Diagram



Appendix F

Class Diagram



Appendix G

State Diagram