

## **ACKNOWLEDGEMENT**

I would like to take this opportunity to thank all the people who helped in the successful development of EventHub. First of all, let me offer my deep regard to Abhishek Koirala sir and Rishav Acharya sir for their valuable guidance, support, and encouragement regarding the project. Their mentorship has gone a long way in shaping my understanding and execution of the platform. I would also like to thank my mentors and instructors for further guidance and impactful feedback. I am indeed grateful to my colleagues for encouragement, sharing knowledge, and expertise which enriched the project. I will extend my special thanks to the opensource community and the developers of Django, a powerful framework which acted as a backbone for this platform. Above all, let me acknowledge hereby, the community inspired me for this project. EventHub is aimed at cultural diversity preservation, improvement in community participation, and my hope is that it does it meaningfully.

## ABSTRACT

EventHub is an online platform that provides a simplified system for easy management and participation of users in community events. From registering, viewing calendar, and viewing of the upcoming events, the users can have their involvement in the events whereas the admin can create, update, and manage the events. This platform provides a provision for two roles: Admin and Users, with specific permissions regarding event management and event viewing using calendar. The Admin is for supervisory purposes in the system, whereas user will do the job of viewing events and registering events to increase the engagement. EventHub uses Django for backend development, providing a secure user-friendly interface for connecting people with their local cultural and community events to enhance cultural diversity and community participation whereas it uses HTML, CSS and JS for the frontend development to ensure the user-friendly experience.

**Keywords:** *EventHub, Django, Cultural Diversity, HTML, CSS, JS.*

## TABLE OF CONTENTS

ACKNOWLEDGEMENT .....	i
ABSTRACT .....	ii
Table of Contents .....	iii
List of Figures .....	v
LIST OF TABLES .....	vi
LIST OF ABBREVIATIONS .....	vii
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 Background.....	1
1.2 Problem Statement .....	1
1.3 Objectives .....	1
1.4 Scope and Limitation.....	2
1.5 Methodology .....	3
Phases of the Iterative Waterfall Model.....	3
CHAPTER 2 .....	5
BACKGROUND STUDY AND LITERATURE REVIEW.....	5
2.1 Background Study .....	5
2.2 Literature Review .....	5
CHAPTER 3 .....	6
SYSTEM ANALYSIS AND DESIGN .....	6
3.1 System Analysis .....	6
3.1.1 Requirement Analysis.....	6

3.1.2 Feasibility Analysis .....	7
i) Technical Feasibility .....	7
ii) Operational Feasibility .....	7
iv) Economic Feasibility .....	7
v) Schedule Feasibility .....	8
.....	8
3.2 System Design .....	9
3.2.1 Class Diagram .....	9
3.2.2 Sequence Diagram .....	11
3.2.3 Activity Diagram .....	12
CHAPTER 4 .....	13
CONCLUSION AND FUTURE RECOMMENDATION .....	13
4.1 Conclusion .....	13
4.2 Future Recommendation .....	13
5) References .....	14

## LIST OF FIGURES

Figure 1.1: Iterative Waterfall Model .....	3
Figure 3.1:Use Case Diagram .....	6
Figure 3.2:Class Diagram .....	9
Figure 3.3:Sequence Diagram.....	11
Figure 3.4:Activity Diagram.....	12

# LIST OF TABLES

Table 1: Gantt Chart .....	8
----------------------------	---

## **LIST OF ABBREVIATIONS**

HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
CRUD	Create, Read, Update, Delete
JS	JavaScript

# CHAPTER 1

## INTRODUCTION

### 1.1 Background

Cultural events like melas, Mahotsav, and community gatherings showcase the rich diversity and heritage of a place, promoting unity and preserving traditions. However, these events often face challenges in reaching a wider audience due to reliance on traditional communication methods and inefficient management of registrations. This project aims to address these gaps by creating an online platform for cultural event management. It will enable admin to showcase events, handle registrations, providing attendees with an easy way to discover and participate in local events.

### 1.2 Problem Statement

In today's digital world, discovering and managing events can be time consuming and complicated process. People often struggle to find the relevant events as the information are distributed across social medias platforms or by the individual people themselves. This lacks a system where people could easily find and get involved in the events. The absence of the integrated calendar also leading to some inconvenience to find the dates of the events happening. Without an efficient way to organize and register for the events, both attendees and event administration faces some challenges to ensure the successful event participations. To address all the issues above, a platform which could leads to the proper management of events is required.

### 1.3 Objectives

The objectives of this project is:

- To make cultural event discovery, registration, and tracking easier through a digital platform.



## 1.4 Scope and Limitation

The scope of the EventHub Event Management System includes:

- **Event Listing:** Providing a single platform where admin can list and manage events, making it easier for users to find upcoming events.
- **Event Registration:** Ensuring that users can register for events through the platform.
- **User Friendly Access:** Allow users to ensure that they can browse events, view details, and register without complication procedures.
- **Calendar Integration:** Displays events in a calendar format so users can see upcoming events and plan accordingly.

The scope of the EventHub Event Management System includes:

- **Limited Event Details:** The platform may not provide deep insights for events.
- **No Payment System:** The system doesn't include any ticketing or payment processing for any events.
- **Single Admin Management:** Only designated admin can create or modify events, limiting the organizer event submission mode.

## 1. 1.5 Methodology

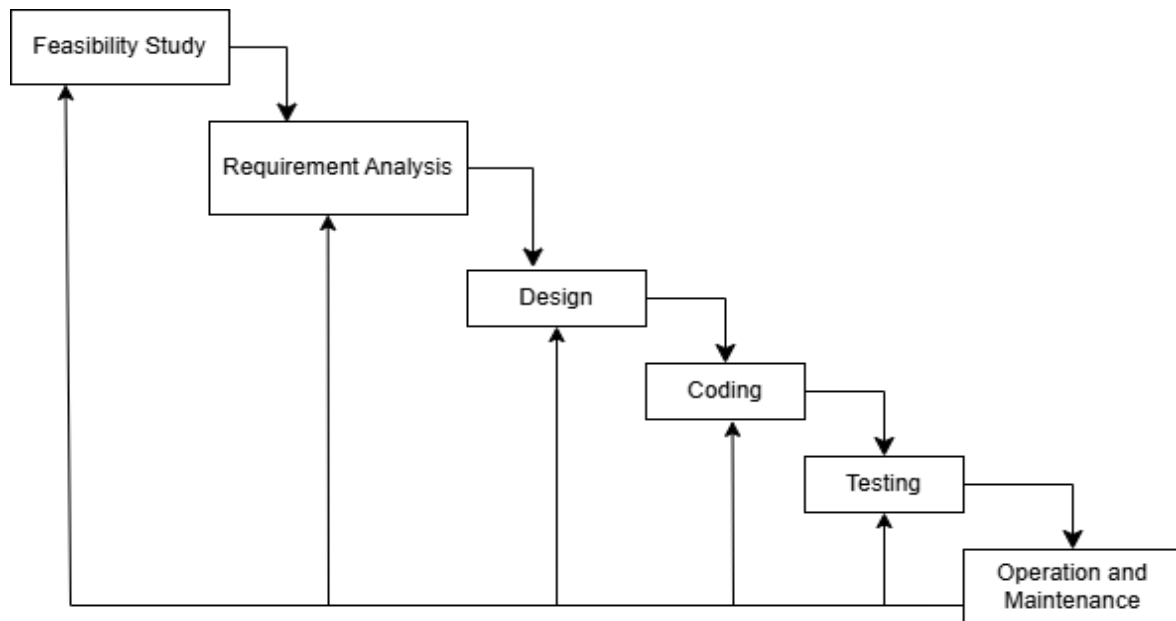


Figure 1.1: Iterative Waterfall Model

The Iterative Waterfall model is a software development method that follows the same steps as traditional waterfall model but allows for changes and improvements at each stage. Unlike the classical waterfall approach, this model includes feedback loops, which means that developers can review and modify previous stages before moving forward.

### Phases of the Iterative Waterfall Model

#### 1) Feasibility Study

This phase checks whether the project is practical and achievable. It evaluates the technical, economic, and operational feasibility of the project that is looking at factors like cost, technology, and potential risks to determine if the project should continue. The result of this phase is a feasibility report, which helps in deciding whether to proceed.

#### 2) Requirement Analysis

Here, the needs of users and system requirements are gathered and documented i.e. functional and non-functional requirement of the system. Developers and stakeholders work together to ensure that all necessary features are identified. The final document from this phase, known as the requirement specification, serves as a guide for designing the system.

### **3) System Design**

In this phase, a plan for how the system will work is created. The system is broken into smaller parts (modules), and their structure and functions are also defined. This process includes both high-level design and detailed design (internal logic, data structures, and flow diagrams). These design documents help programmers during development of the system.

### **4) Coding and Testing of the Modules**

The system design is converted into actual code in this phase. Each module (small part of the system) is built and tested separately through unit testing to ensure that it works correctly. The goal is to create working, error free modules that can later be combined

### **5) Integration and System Testing**

After individual modules are completed, they are connected to form the full system. Integration testing ensures that all modules work well together. Next, system testing checks the entire software against the original requirements to confirm that it functions properly before deployment.

### **6) Operation and Maintenance**

Once testing is complete, the software is released for users. Any bugs or issues found during actual use are fixed in this phase. Over time, improvements may be made to enhance performance or update features in the system. This phase also ensures that the software remains useful and up to date.

## **CHAPTER 2**

### **BACKGROUND STUDY AND LITERATURE REVIEW**

#### **2.1 Background Study**

Cultural events such as festivals, fairs, and local gatherings play an important role in bringing communities together and preserving traditions. However, managing and promoting these events can be difficult due to dependency on word of mouth, posters, or social medias. Many people miss out on events simply because they do not know about them. Similarly, event organizers struggle to reach the right audience and manage registrations efficiently.

Several online platforms exist for event management, such as Eventbrite and Meetup, which help organizers create and promote events while allowing users to discover and register for them. However, these platforms mainly focus on business and professional networking events rather than community-based cultural programs. [1]

In response to this gap, EventHub aims to provide a single platform for organizing and discovering local cultural events. It will allow administrators to list events, manage registrations, and provide users with an easy way to explore events using a calendar-based system. This ensures better participation and improves the overall event experience for both organizers and attendees.

#### **2.2 Literature Review**

Many people struggle to find events because information is scattered across various sources. This lack of a central system makes event promotion and registration difficult for both attendees and organizers.

Platforms like Eventbrite and Meetup provide solutions for event management. Eventbrite enables organizers to create events, sell tickets, and reach a broad audience, while Meetup connects people with similar interests through group activities. However, neither platform is specifically designed for local cultural events. [2]

By learning from these existing platforms and addressing their limitations, EventHub aims to provide a user-friendly system where people can easily find and register specially for cultural events, improving overall community engagement.

## CHAPTER 3

### SYSTEM ANALYSIS AND DESIGN

#### 3.1 System Analysis

System Analysis is the process of studying a system to understand its components, functions, and requirements.

##### 3.1.1 Requirement Analysis

###### i) Functional Requirement

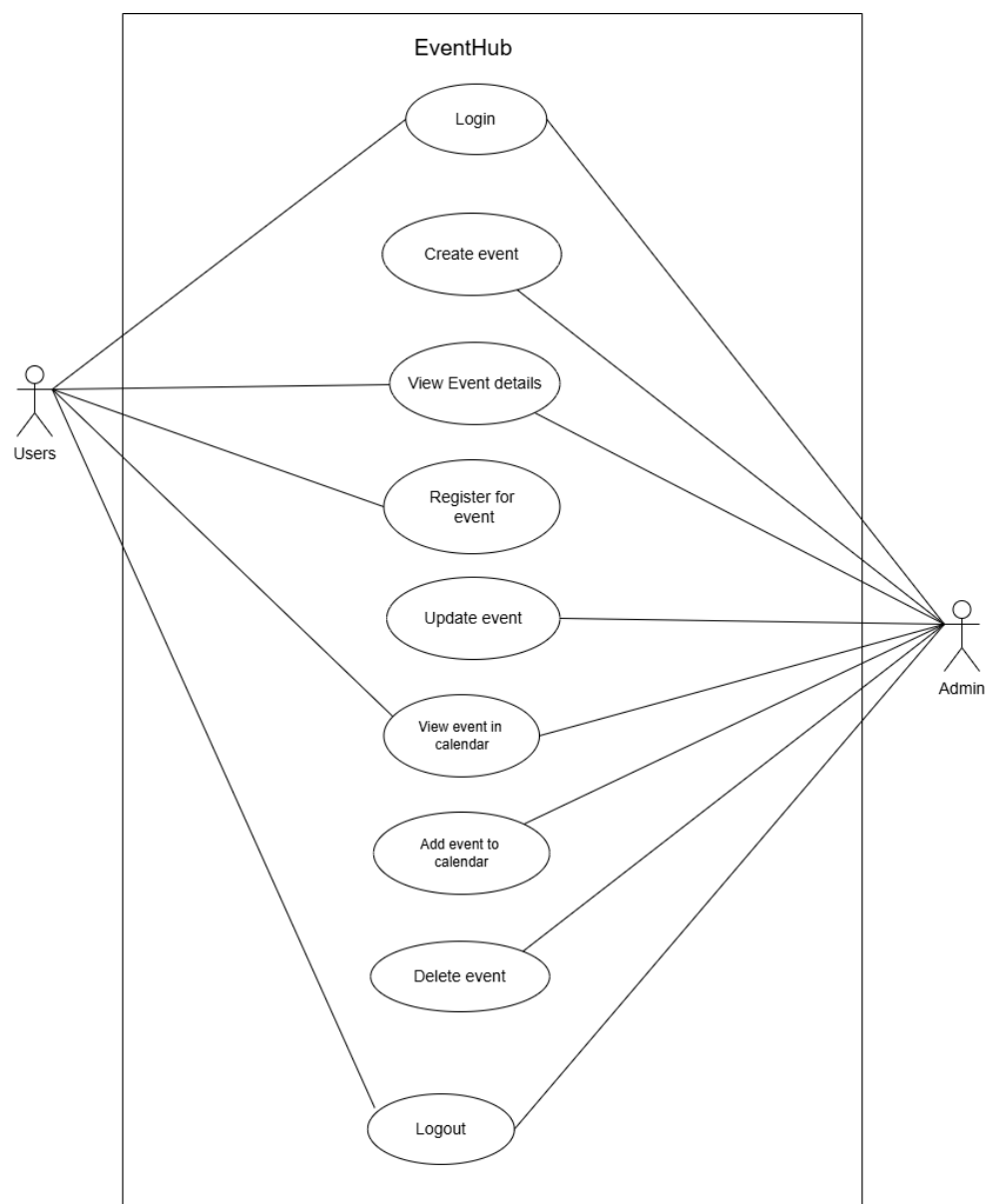


Figure 3.1: Use Case Diagram

## **ii) Non-Functional Requirement**

### **Easy to use**

The user interface and functionality of EventHub is designed ensuring that user can easily navigate and get familiar with the system without difficulty.

### **Security**

The personal information of users and event-related data will be securely stored and accessible only to authorized users, specifically the admin.

### **Responsive**

The system will be designed to respond quickly to user actions, including event browsing, registration, and calendar updates, ensuring a smooth user experience without delays or system lag.

### **Maintainability**

EventHub will be developed using technologies such as HTML, CSS, JavaScript for frontend development, and Django for backend operations. The system will be easy to maintain with clear, organized code;

## **3.1.2 Feasibility Analysis**

### **i) Technical Feasibility**

Technical feasibility determines whether a project is feasible in terms of software, hardware, and expertise. EventHub use Django for the backend, ensuring scalable functionality whereas, HTML/CSS enhances the front-end design for a seamless user experience.

### **ii) Operational Feasibility**

The platform's design will be simple, ensuring ease of use for both end-users (who can register for and view events) and the administrative team (who manage events, users, and registrations). The system is designed to be user-friendly. With real-time updates and a feedback mechanism, it ensures operational feasibility and effective goal achievement

### **iv) Economic Feasibility**

The Economic Feasibility of EventHub shows that the project is financially viable. The use of open-source technologies like Django, HTML, CSS, JS helps minimize both development and maintenance costs.

## v) Schedule Feasibility

Schedule Feasibility for EventHub shows the project can be completed on time. Development will be done in phases, starting with basic features like event listing and registration. The project's flexible approach ensures timely delivery. Below is the Gantt chart that outlines the major milestones and their respective timelines.

Working Days	JAN 2025				FEB 2025				MAR 2025				APRIL 2025			
Week	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
Feasibility Study																
Requirement Analysis																
Design																
Coding																
Testing																
Operation and Maintenance																
Documentation																

Table 1: Gantt Chart

## 3.2 System Design

### 3.2.1 Class Diagram

The main purpose of the class diagram in our project is to represent the system's structure. It serves as a blueprint, showing how different components interact. The UML diagram of EventHub provides an overview of the system's key functionalities and relationships between classes.

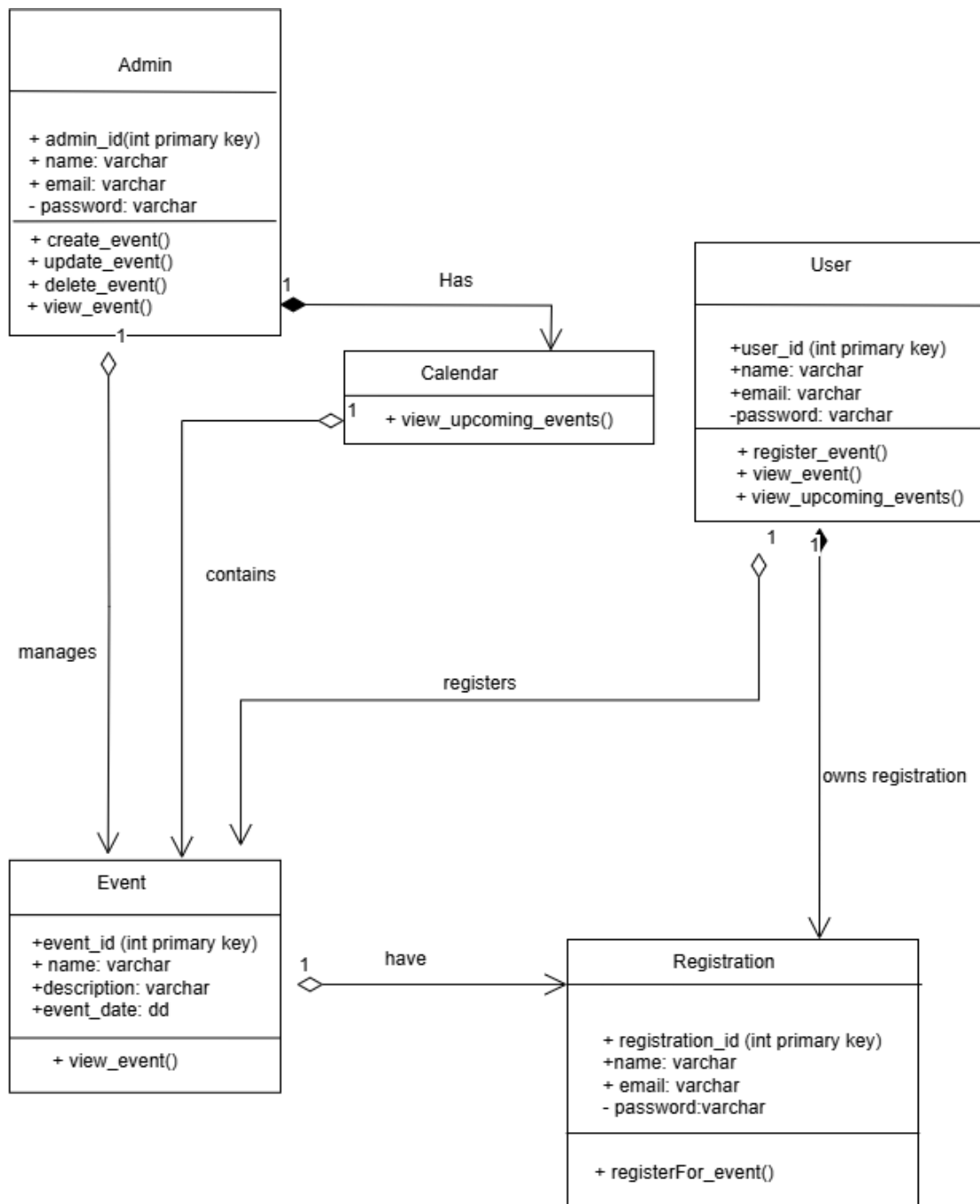


Figure 3.2: Class Diagram



- **Admin & Event (1-to-Many Aggregation)**

An admin can create, update, or delete multiple events, but each event is managed by a single admin.

- **User & Registration (1-to-Many Association)**

A user can register for multiple events, but each registration is linked to a single user.

- **User & Calendar (1-to-1 Association)**

Each user has a calendar to view upcoming events.

- **Event & Registration (1-to-Many Composition)**

An event can have multiple registrations, but each registration is tied to a single event. If an event is deleted, all associated registrations are removed.

- **Admin & Calendar (1-to-1 Association)**

The admin has access to the calendar to manage events.

- **Event & Calendar (Many-to-One Association)**

Multiple events appear in a single calendar view.

### 3.2.2 Sequence Diagram

A sequence diagram is a type of UML diagram that shows how objects interact with each other in a specific sequence of events. It visually represents the flow of messages or actions between objects over time, highlighting the order of operations.

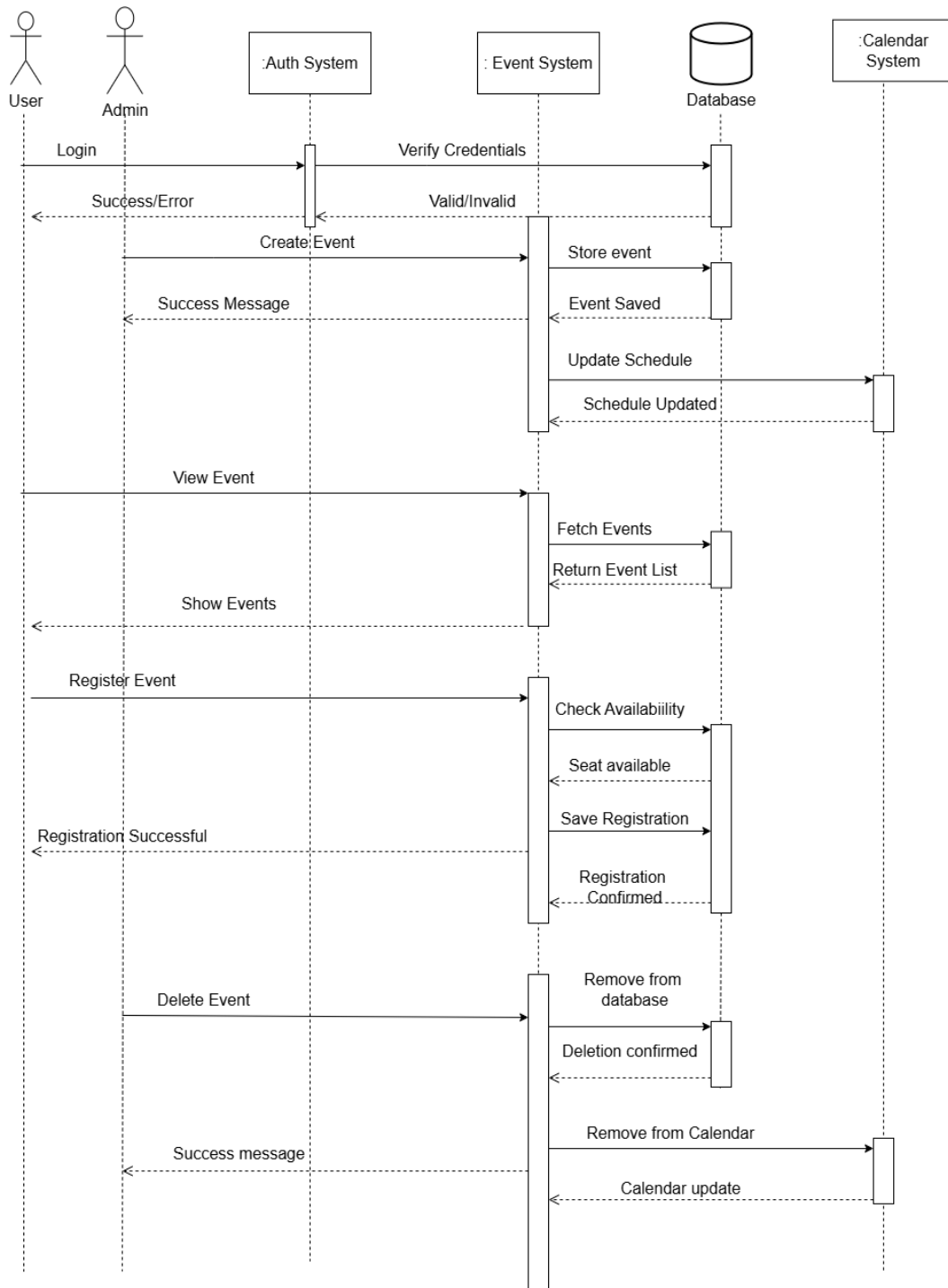


Figure 3.3: Sequence Diagram

### 3.2.3 Activity Diagram

An activity diagram is a type of UML diagram that represents the flow of activities or tasks within a system. It shows the sequence of actions, decision points, and parallel processes, helping to visualize the workflow and logic.

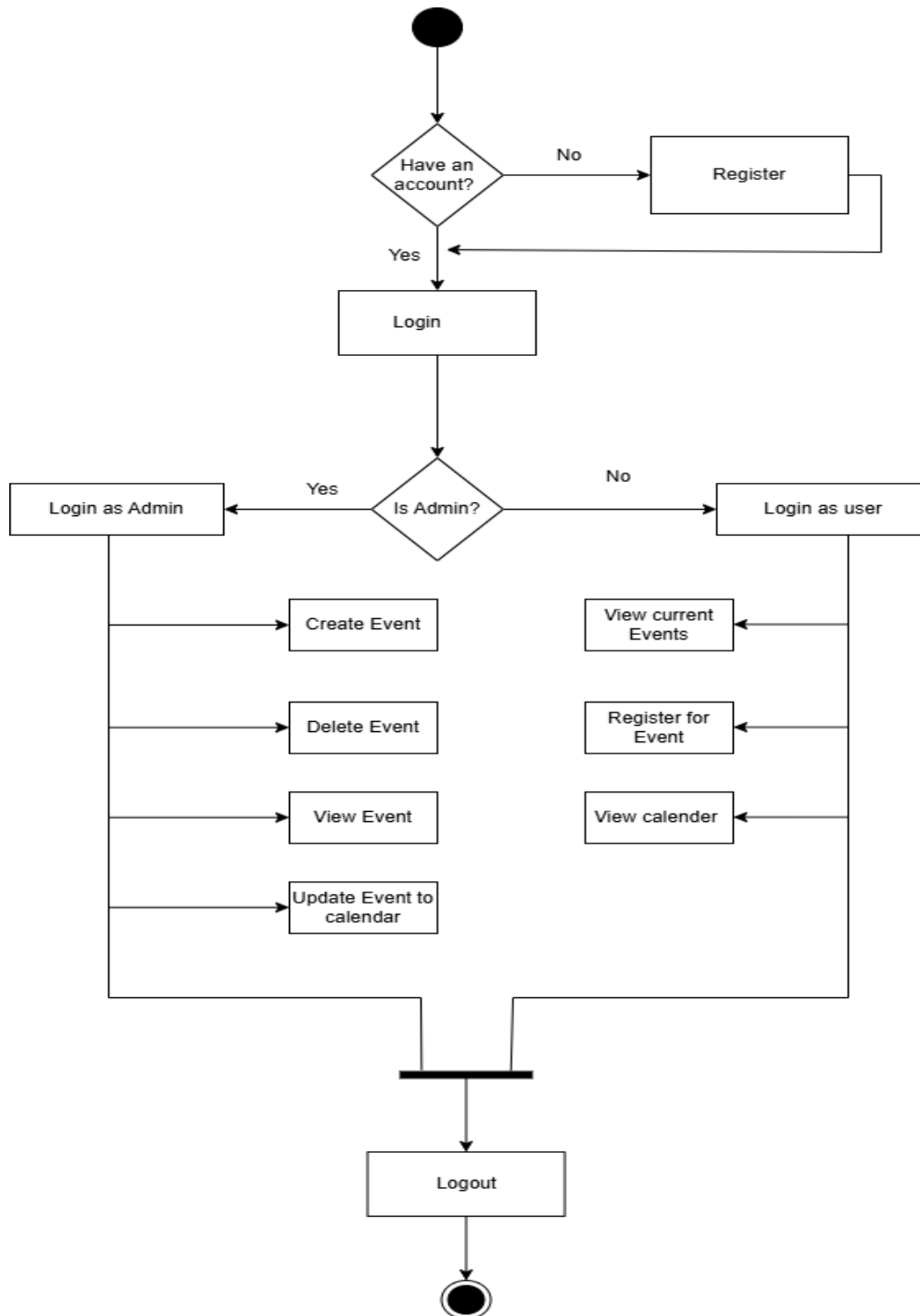


Figure 3.2: Activity Diagram

## CHAPTER 4

### CONCLUSION AND FUTURE RECOMMENDATION

#### 4.1 Conclusion

EventHub aims to provide an easy to use platform for managing events. The system focuses on event listing, user registration, and admin controls, making it simple for users to find and join events. By offering a centralized solution, EventHub helps to reduce the complexity of event management. The project is designed to offer a valuable tool for both event organizers (Admin) and participants. With its user-friendly design, EventHub can help improve event participation and make the process easier for everyone involved.

Ultimately, EventHub can contribute to the smoother organization of events, helping both admins and users save time and effort while promoting greater participation in community activities.

#### 4.2 Future Recommendation

To enhance the project further, consider the following improvements:

##### **Expand Event Categories:**

Add more event types, such as virtual events, workshops, and networking sessions, to attract a wider audience.

##### **Integrate Rating and Reviews:**

Allow users to rate and review events, helping others choose the best ones based on feedback.

##### **Enhance Search & Filtering:**

Implement advanced filtering options, such as category-based search, location-based suggestions, and AI-powered recommendations.

##### **Mobile App Development:**

Develop a mobile application to improve accessibility and allow users to browse and register for events more conveniently.

### **Payment Integration:**

If paid events are introduced, integrate secure payment gateways to allow users to register for premium events easily.

## **5) REFERENCES**

- [1] "Eventbrite," [Online]. Available: <https://www.eventbrite.com..>  
[Accessed 5 Jan 2025].
- [2] Meetup, "Meetup," [Online]. Available: <https://www.meetup.com..>  
[Accessed 7 Jan 2025].
- [3] M. Sharma and P. Singh, "Online Event Management System: A Critical Review of Research and Developments," *International Journal of Innovative Engineering and Science (IJIES)*, vol. 9, no. 5, pp. 1-4, 2024.
- [4] GeeksforGeeks, "Use Case Diagram," GeeksforGeeks, [Online]. Available: <https://www.geeksforgeeks.org/use-case-diagram/>. [Accessed 18 Nov 2024].
- [5] R. Kumar and S. Sharma, "Virtual Events and Use of Technology," *International Journal of Event Management (IJEM)*, vol. 14, no. 1, pp. 4-5, 2021.
- [6] GeeksforGeeks, "Iterative Waterfall Model - Software Engineering," GeeksforGeeks, 22 May 2024. [Online]. Available: <https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/>. [Accessed 07 August 2024].

[7] Asana, "Feasibility Study: A Step-by-Step Guide," [Online]. Available: <https://asana.com/resources/feasibility-study>. [Accessed 12 December 2024].