King Saud University
College of Computer and Information Sciences
Department of Software Engineering

EVENTHUB
event managment system

Event Management System (EventHub)

Graduation Project, Part-I (SWE 496)
Software Engineering Department
CCIS, KSU

Project Advisor:
Dr. Fayez AL Qahtani

Submitted by
Abdulrahman Almozainy – 441102441
Turki Almutairi – 441103182
Abdulaziz Binbaz – 441102727
Sulaiman Alsunaya - 441102216

Date submitted
13/11/2022

ABSTRACT

This document describes the process of developing a web application that provides the user with the service of creating an event and making it accessible to those interested in this event. All this can be done by allowing the Event Manager to create an event, whether it's for a company or a community and allowing the participant to register for the user's event.

Table of Contents

List of Figures

List of Tables

1. Introduction

Many people are interested and enjoy going to events in Saudi Arabia, it has become a thing that many people do every day. Most people surf different social media platforms looking for some events, however, most social media platforms do not provide complete or classification for events.

In this project we plan to develop an event management system with an interactive social networking platform to ease the work of event management through multiple steps, Online tickets, payments, marketing, scheduling, data gathering, define and organize audience, Sending email notification etc.

Also, it will be a powerful event search tool that can find events around the area and users can browse available events using Date, Category, Rate, location. Then users can apply to an event with only a few clicks by submitting the information to the application.

Moreover, Social networks can have a social purpose, a business purpose, or both through sites such as Twitter, Facebook, and Pinterest.

And in this project, we plan to develop an interactive social networking platform that allows people and corporations to connect with one another so they can develop relationships and share information, ideas, and messages.

Also, users can also connect with different individuals who share the same interests and goals, users can find each other through event category, event lists, and the use of tags.

Currently There is a huge market, and we need to build intelligent recommendation systems. Recommendation systems have proven to play an important role in the field of social networking, e-commerce websites, online shopping, digital marketing, online advertising, etc. by providing personalized recommendations and feedback to users according to their preferences and choices.

Therefore, in this project we plan to include recommendation systems in our project as a subsystem that helps users to find relevant events to them.

2. Domain Analysis

There are multiple software solutions out there in the worldwide market, but we believe there is still room for improvement and added value.

Eye of Riyadh is a marketing company that has an event management system as well. Therefore we are seeing them as one of the domains that could be beneficial for us, As we can learn from their goods and mistakes.

One of the good things they have is they have an easy and forward event registration page.
They have bad user experience for that reason. It is hard to deal with, especially with non-technical people. For example, if we try to register an account it is not an automated process, they do not have a direct payment system.

Eventat is a Kuwaiti website that is specialized in an entertainment event e.g., theatrical performances and courses that are specialized in theatrical performances.

In their website they have a decent user experience and a good payment gate, and you can select that seat you want in the theater.

Faaliat is one of the projects that belongs to the ministry of media that the most well-known event management website with a great user experience with a recommendation system.

But one of the down sides of the system is that it is hard to add an event since it’s required to contact them so they can add a specific event.

Feature	Eventat	Eye of Riyadh	Faaliat	EventHub (Our System)
Social networking system	✕	✕	✕	✓
Event recommendation system using AI	✕	✕	✓	✓
Ability of user to advertise event	✓	✕	✕	✓
Events calendar	✕	✓	✕	✓

Table 2.1 Domain analysis table of other competitors

3. Risk/Constraints

3.1 Risks

No.	Risk	Solution
1	Conflict between project deliverables, exams, quizzes, etc.	We are going to set a time management plan. So, all the members of the have time to work on the project
2	Since the project is web-based. We are going to have a risk for slow connection speed due to a bad server side	We are going to ensure we have a server that is well known and reliable for our website for example. AWS
3	Lack of experience in the specific field we are going to work on (The event management field)	We will have courses that is specialized in event managements

Table 3.1 Risks

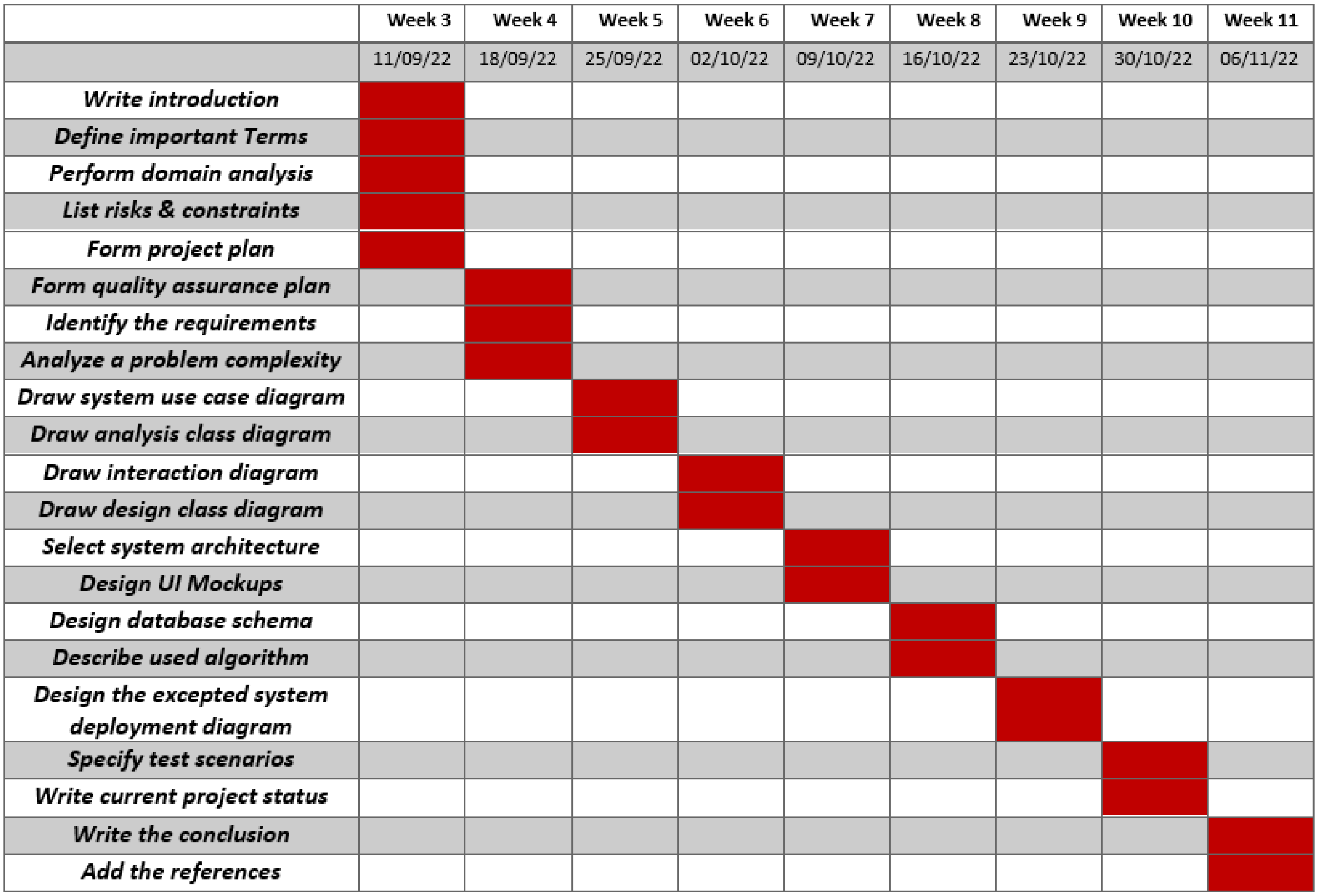
3.2 Constraints

No.	Constrains	How we will manage it
1	the system should be able to work on multiple browsers	after the implementation of the software, we are going to test it on multiple browsers
2	Applying the rules and the regulation of the Saudi government	we will read and keep track that our project does not violate the rule and regulation
3	Since we are on the new system of trimesters instead of the old semester's plan. We have only 10 weeks to work on the first part of the project	We are going to arrange a meeting for all team members. Three times every week so we can keep all team members updated with the latest updates. And we going to split the work evenly between all team members
4	The system should be able to provide multiple methods of payment	we are going to choose a safe payment gateway that provide multiple methods to payment

Table 3.2 Constraints

4. Project Plan

4.1 Gantt chart



	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
	11/09/22	18/09/22	25/09/22	02/10/22	09/10/22	16/10/22	23/10/22	30/10/22	06/11/22
Write introduction	🟥								
Define important Terms	🟥								
Perform domain analysis	🟥								
List risks & constraints	🟥								
Form project plan	🟥								
Form quality assurance plan		🟥							
Identify the requirements		🟥							
Analyze a problem complexity		🟥							
Draw system use case diagram			🟥						
Draw analysis class diagram			🟥						
Draw interaction diagram				🟥					
Draw design class diagram				🟥					
Select system architecture					🟥				
Design UI Mockups					🟥				
Design database schema						🟥			
Describe used algorithm						🟥			
Design the excepted system deployment diagram							🟥		
Specify test scenarios								🟥	
Write current project status								🟥	
Write the conclusion									🟥
Add the references									🟥

Figure 4.1 Gantt chart

5. Quality Assurance Plan

5.1 Review

Every week on Sunday, we will review each other work and give feedback, we will make sure that we have met all constraints and the requirements needed, We will review the work we finished throughout the week, We are going to conduct the walkthrough technique,

5.2 Verification

We will be testing each phase with the previous ones to make sure that our implementation follows our design by using a checklist.

5.3 Validation

We will be checking the whole system with our advisor weather it met all the requirements and the non-functional requirements, by using different test scenarios, and we are going to do a security test for our website.

5.4 Training Team members

Each team member will have a course whether online or an attendance course on web development and will practice the technology before starting to work on the project itself.

5.5 Roles and Responsibilities

Here is a list of typical tasks of the QA specialists:

Role	Name	Responsibilities
QA Lead	Abdulaziz Binbaz	This person watches for the entire team and personally supervises the software testing processes.
QA Analyst	Suliman Alsunaya	This position is kind of a combination of software testing skills with the skills to build long-term web product validation strategies. The QA Analyst position includes work with business logic and matches the target audience.
Manual Test Engineer	Turki Almutairi	gets to know the product manually. This person mostly works with the client part of the software. Web software is tested from the point of view of the end user group, to find the maximum number of bugs that should not get into the release version and to users
UI/UX Test Engineer	Abdulrahman Almozainy	Some companies make a UI/UX department to find the best user experience. The external interface of the application plays an important role, and therefore must be designed and tested with great care

Table 5.1.1 Quality Assurance Plan, Roles, and Responsibilities

6. Requirements

6.1 Functional Requirements

6.1.1 Registration & Login

1. The user shall be able to register.
2. The system shall save user information.
3. The user shall be able login.
4. The user shall be able to logout.
5. The event manager shall be able to register.
6. The system shall save event manager information.
7. The event manager shall be able to login.
8. The event manager shall be able to logout
9. The admin shall be able to login.
10. The admin shall be able to logout.

6.1.2 Account

11. The user shall be able to view account information.
12. The user shall be able to edit account information.
13. The user shall be able to view support accounts.
14. The user shall be able to send a message to the support account.
15. The event manager shall be able to view support accounts.
16. The event manager shall be able to send a message to the support account.
17. The admin shall be able to view user account.
18. The admin shall be able to manage user account.
19. The admin shall be able to view event manager account.
20. The admin shall be able to manage event manager account.

6.1.3 Social Networking

21. The user shall be able to follow another user.
22. The user shall be able to unfollow another user.
23. The user shall be able to follow category.
24. The user shall be able to unfollow category.
25. The user shall be able to follow an Event Manager.
26. The user shall be able to unfollow an Event Manager.
27. The user shall be able to add a post.
28. The user shall be able to delete a post.
29. The user shall be able to share a post.
30. The user shall be able to like a post.
31. The user shall be able to view other user profile.
32. The user shall be able to search for an account.
33. The event manager shall be able to follow a user.
34. The event manager shall be able to unfollow a user.
35. The event manager shall be able to follow category.
36. The event manager shall be able to unfollow category.

37. The event manager shall be able to follow another Event Manager.
38. The event manager shall be able to unfollow another Event Manager.
39. The event manager shall be able to add a post.
40. The event manager shall be able to delete a post.
41. The event manager shall be able to share a post.
42. The event manager shall be able to like a post

6.1.4 Browsing & Filtering

43. The user shall be able to browse events by rating.
44. The user shall be able to browse events by location.
45. The user shall be able to browse events by category.
46. The user shall be able to browse events by date.
47. The user shall be able to search for an event.
48. The user shall be able to see the search result.
49. The system shall recommend events related to user interest.
50. The user shall be able to view the event category.
51. The user shall be able to select an event.

6.1.5 Event

52. The user shall be able to view event’s information.
53. The user shall be able to view event rating.
54. The user shall be able to rate an event.
55. The user shall be able to comment on the event.
56. The user shall be able to see the number of events attended.
57. The user shall be able to see the name of event attended.
58. The user shall be able to see the category of event attended.
59. The event manager shall be able to send a request to add an event.
60. The event manager shall be able to send a request to delete an event.
61. The event manager shall be able to view event’s information.
62. The event manager shall be able to send a request to modify the event's information.
63. The admin shall be able to receive request to add an event.
64. The admin shall be able to approve the event to be added.
65. The admin shall be able to receive request to delete an event.
66. The admin shall be able to approve the event to be deleted.
67. The admin shall be able to receive request to modify event’s information.
68. The admin shall be able to approve event’s information to be modified.

6.1.6 Payment &Ticket

69. The user shall be able to buy a ticket.
70. The system shall save user ticket.
71. The user shall be able to view the ticket.
72. The user shall be able to see payment methods.
73. The user shall be able to select payment method.
74. The user shall be able to make payment.

6.2 Non-Functional Requirements

6.2.1 Usability

75. The user shall be able to apply to an event within 2 minutes.
76. The average time for a new user to navigate through all the functionality in the home page is 4 minutes.

6.2.2 Reliability

77. The system must perform without failures 95% of the time.
78. The system downtime must not be more than 10 minutes per month.

6.2.3 Security

79. The system shall allow only authorized people to access the data.
80. The passwords shall be at least 8 characters.
81. The passwords shall contain at least one letter and one number.
82. The passwords shall contain at least one special character.
83. The system shall be secure against SQL injection.

6.2.4 Performance

84. Average response time shall be no more than 3 seconds.

6.2.5 Supportability

85. The system shall be available in English.

7. Problem Complexity

Diverse groups of stakeholders involved:

Many features will be added to the website to meet the needs of the user and the event manager. As a result, this kind of system requires continuous interaction with a wide range of stakeholders.

Conflicting technical issues:

As our system Recommends events to the user, we want to suggest related events to the user that he is interested in by using an AI (Artificial intelligence) recommendation system.

8. System Use-Cases

8.1 Use Case 1

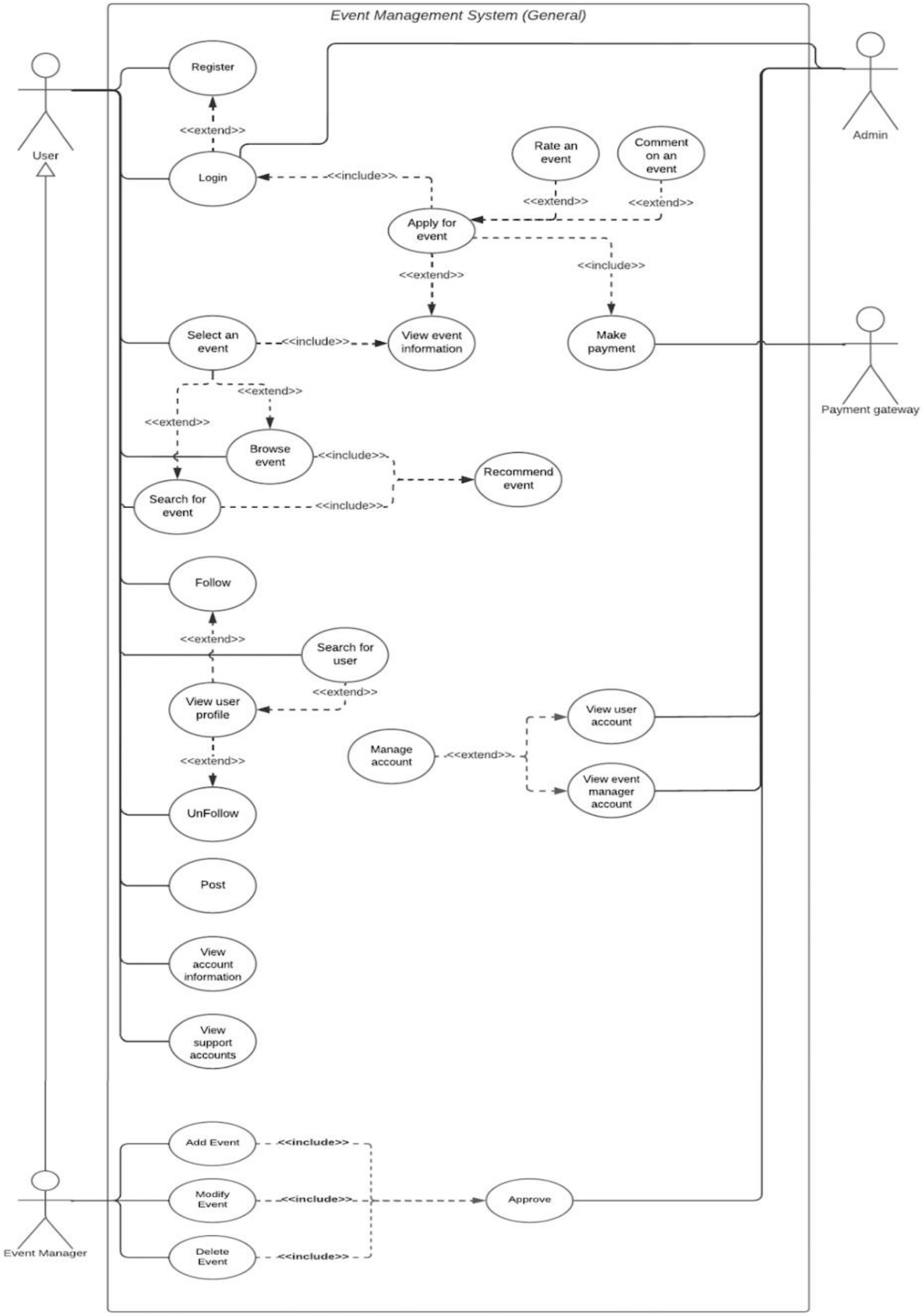
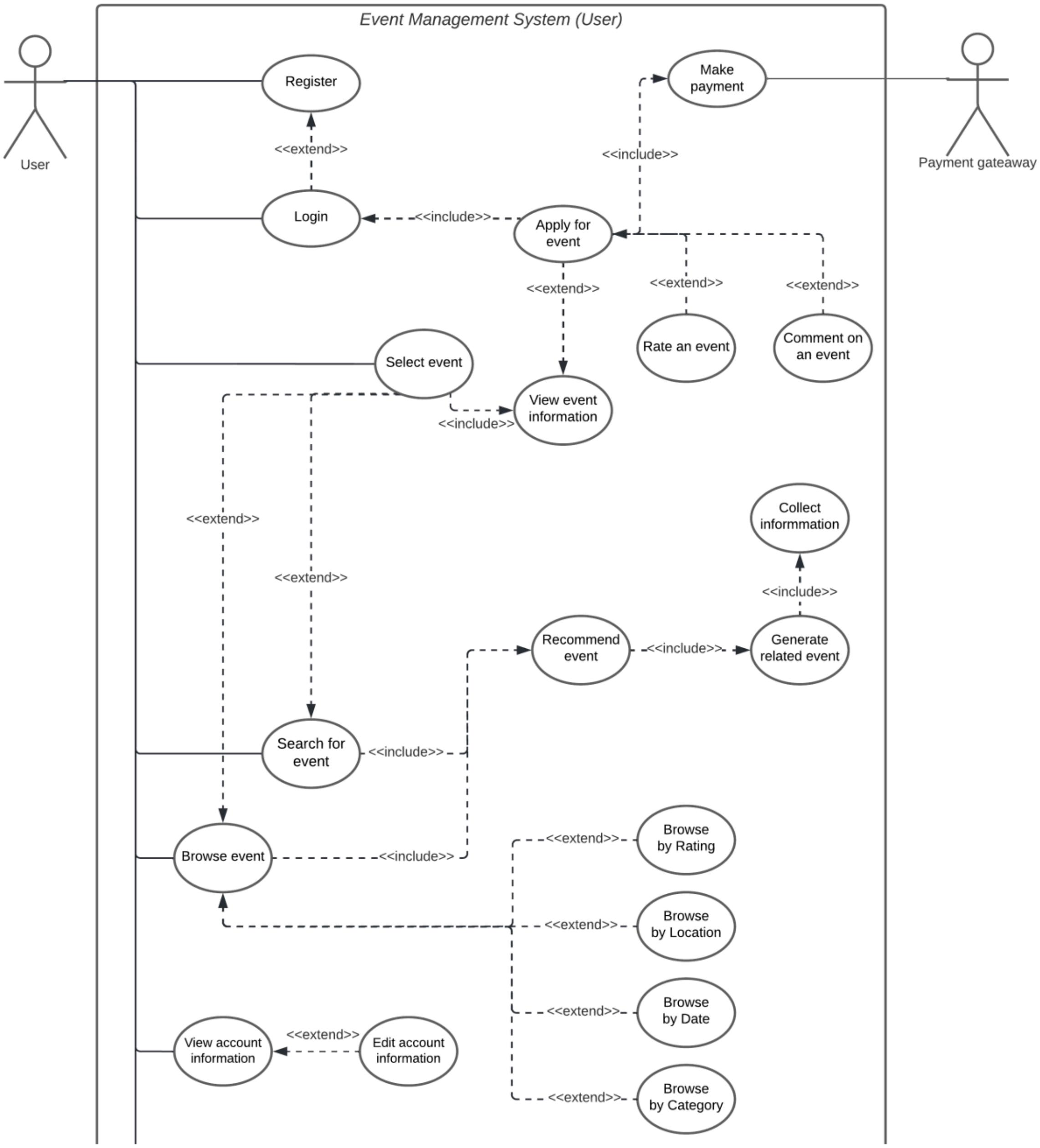


Figure 8.1 EventHub Use Case (General)

8.2 Use Case 2


Event Management System (User)
User
Payment gateaway
Register
<<extend>>
Login
<<include>>
Make payment
<<include>>
Apply for event
<<extend>>
<<extend>>
<<extend>>
Rate an event
Comment on an event
Select event
View event information
<<include>>
<<extend>>
<<extend>>
Collect informmation
<<include>>
Recommend event
<<include>>
Generate related event
Search for event
<<include>>
Browse event
<<include>>
<<extend>>
Browse by Rating
<<extend>>
Browse by Location
<<extend>>
Browse by Date
<<extend>>
Browse by Category
View account information
<<extend>>
Edit account information

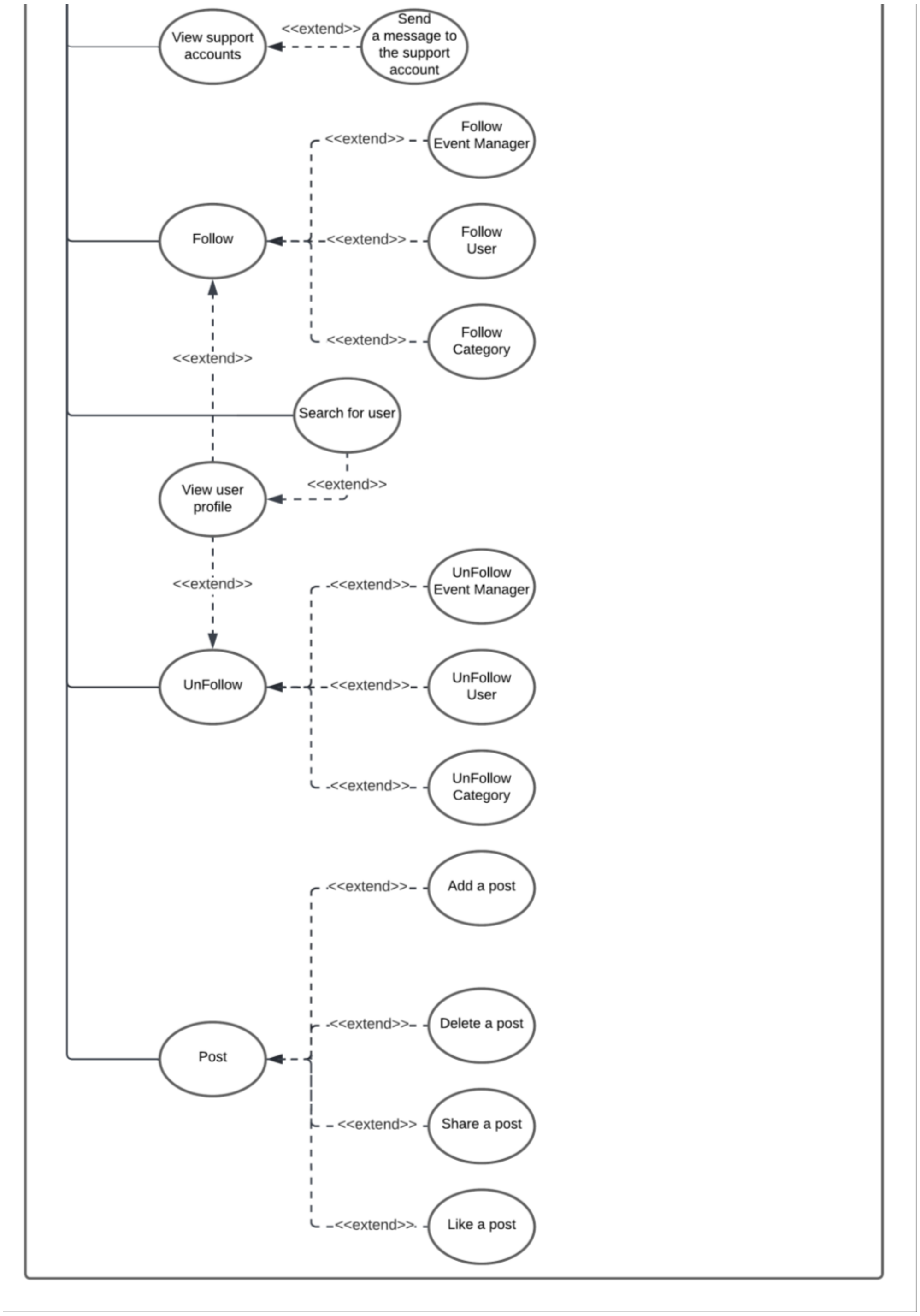


Figure 8.2 EventHub Use Case (Detailed User)

8.3 Use Case3

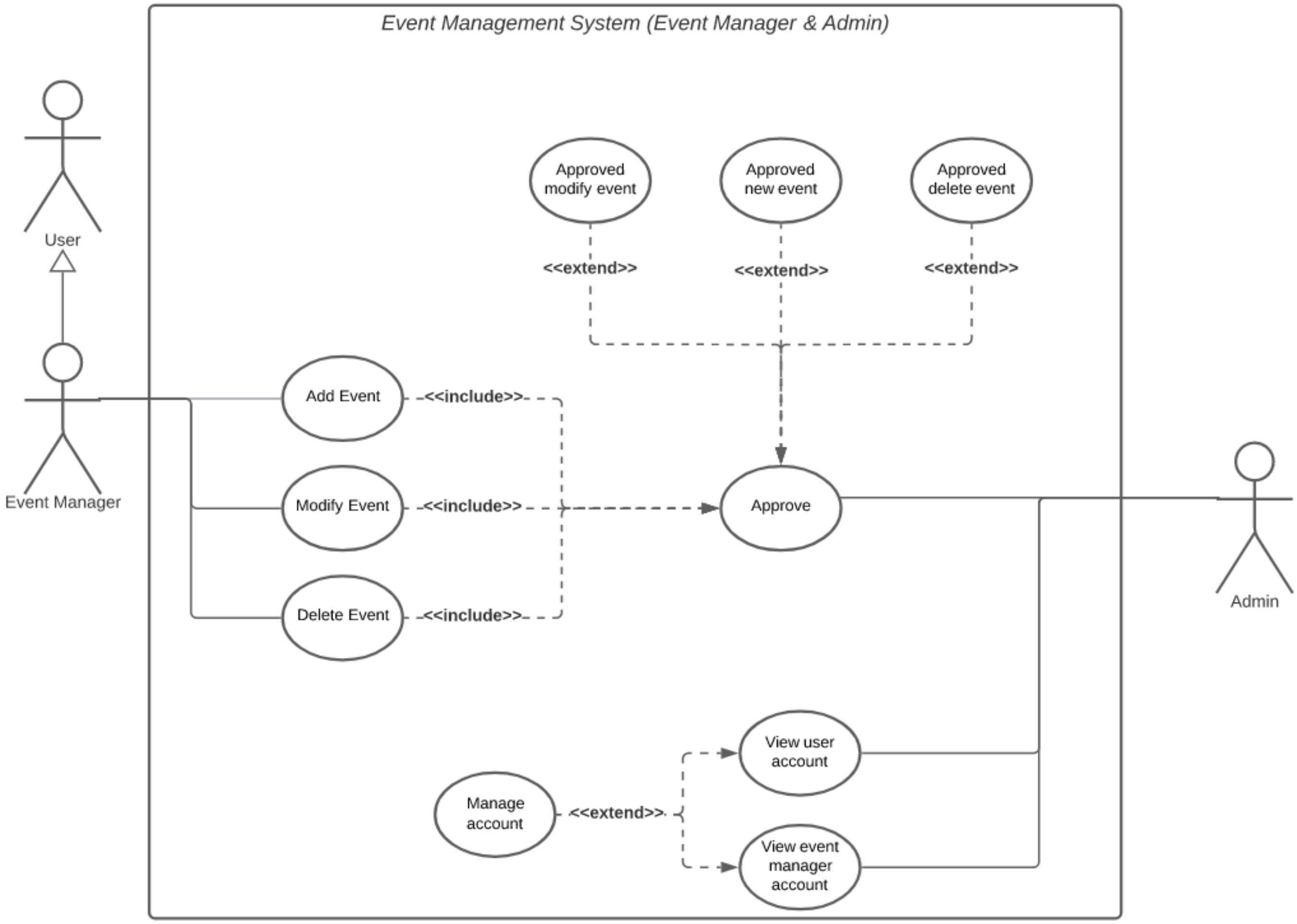


Figure 8.3 EventHub Use Case (Detailed Event Manager & Admin)

9. Use-Cases Description

9.1 Recommend Event

Use case description	
Use case ID:1	
Use case name: Recommend event	
Primary actor: User	**Other actors:** NA
Stakeholders: User	
Description: Give the user a recommended event base on the information collected from the user.	
Relationships: • **Includes:** Generate related event • **Extends:** NA	
Pre-condition: User search for an event or browse an event	
Steps:	
Actor	System
1.The Actor searches or browses an event	2. The system connects to the database 3. The system collects information 4.The system provides the most related event 5.The system display the event
Alternative and exceptional flows: 3a. The Actor have no information. 3a1. The system displays random events.	
Post-condition: The system display the recommend event in the page	

Table 9.1 Recommend event Use Case Description

9.2 View User Profile

Use case description	
Use case ID: 2	
Use case name: View User Profile	
Primary actor: User	**Other actors:** NA
Stakeholders: User	
Description: Allows the Actor to view user profile	
Relationships: • **Includes:** NA • **Extends:** Follow, Unfollow	
Pre-condition: User login in the system	
Steps:	
Actor	System
1. Actor select user account	2. The system connects to the database 3. The system retrieves the selected user profile 4.The system display the selected user profile
Alternative and exceptional flows: NA	
Post-condition: The system displays selected user profile.	

Table 9.2 View User Profile Use Case Description

9.3 View Event Information

Use case description	
Use case ID: 3	
Use case name: View event information	
Primary actor: User, Event manager	**Other actors:** NA
Stakeholders: User, Event manager	
Description: Allows actors to see information about the event	
Relationships: - **Includes:** NA - **Extends:** NA	
Pre-condition: NA	
Steps:	
Actor	System
1. Actor select an event	2.The system displays event information.
Alternative and exceptional flows: NA	
Post-condition: The system displays event information.	

Table 9.3 View Event Information Use Case Description

9.4 Search For Event

Use case description	
Use case ID: 4	
Use case name: Search for event	
Primary actor: User, Event Manager	**Other actors:** NA
Stakeholders: User, Event manager	
Description: Allows Actors to search for event by name of event	
Relationships: - **Includes:** Recommend event - **Extends:** NA	
Pre-condition: NA	
Steps:	
Actor	System
1. Actor enter event name in the search bar	2. System connect to database 3. System retrieves events according to actor input from the database 4. System displays events cards
Alternative and exceptional flows: 3a. The system did not find events according to actor input. 3a1. The system displays message “the event is not found”	
Post-condition: event(s) cards are displayed	

Table 9.4 Search for Event Use Case Description

9.5 Modify Event

Use case description		
Use case ID: 5		
Use case name: Modify event		
Primary actor: Event Manager		**Other actors:** NA
Stakeholders: Event manager		
Description: Allows actor to edit event’s information		
Relationships: • **Includes:** Approve • **Extends:** NA		
Pre-condition: • Event manager is login on the website • Event is already added to the database		
Steps:		
Actor	System	Admin
1.Actor select modify information	2.System connect to database	
	3.System retrieves required data from the database	
5.Actor modify the information	4.system display all the required information	
6.Actor select apply		7.Admin approve modifications
	8.System update all the information to the database	
Alternative and exceptional flows: 5a. The actor entered invalid information. 5a1. The system displays an error message. 7a. Admin disapproves modifications. 7a1. The system does not update the information in the database.		
Post-condition: All the modified information has been updated.		

Table 9.5 Modify Event Use Case Description

10. Analysis Class

10.1 Recommend Event

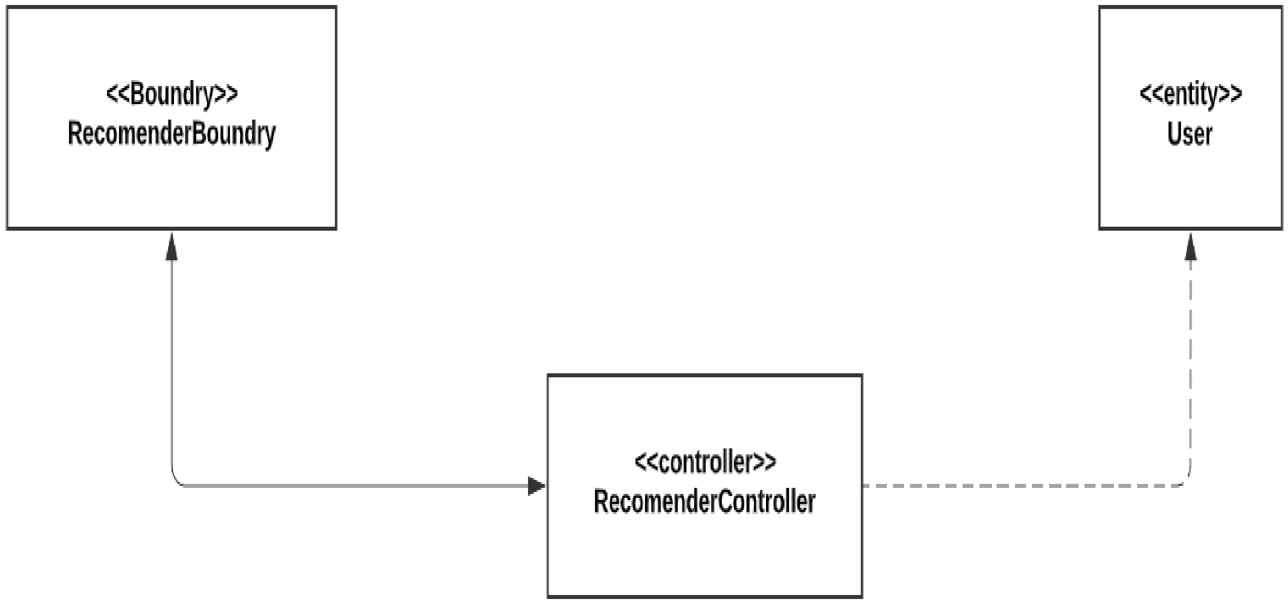


Figure 10.1 Recommend Event Analysis Class

10.2 View User Profile

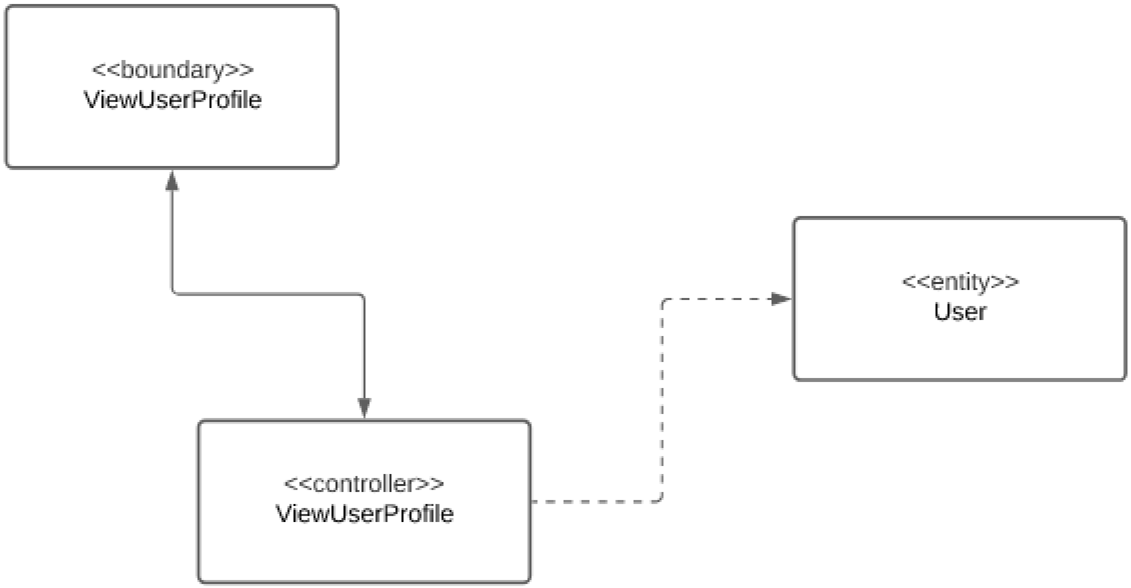


Figure 10.2 View User Profile Analysis Class

10.3 View Event Information

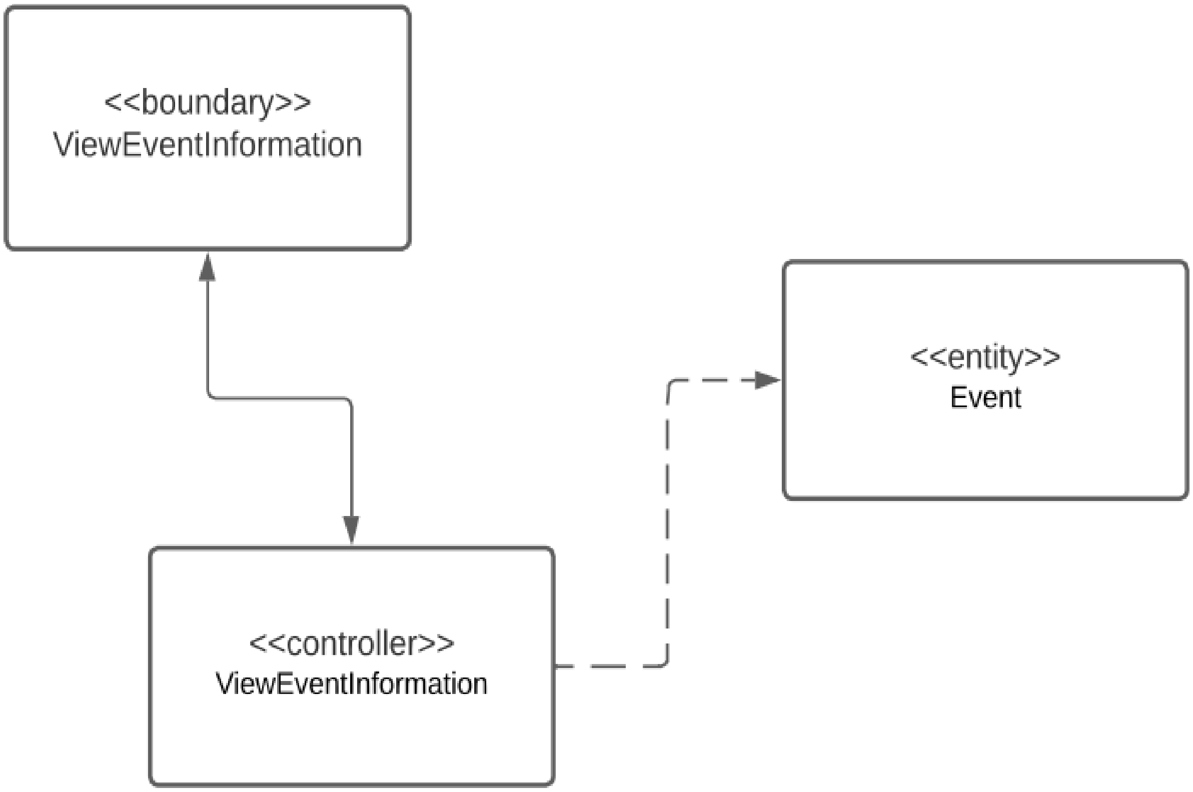


Figure 10.3 View event information Analysis Class

10.4 Search For Event

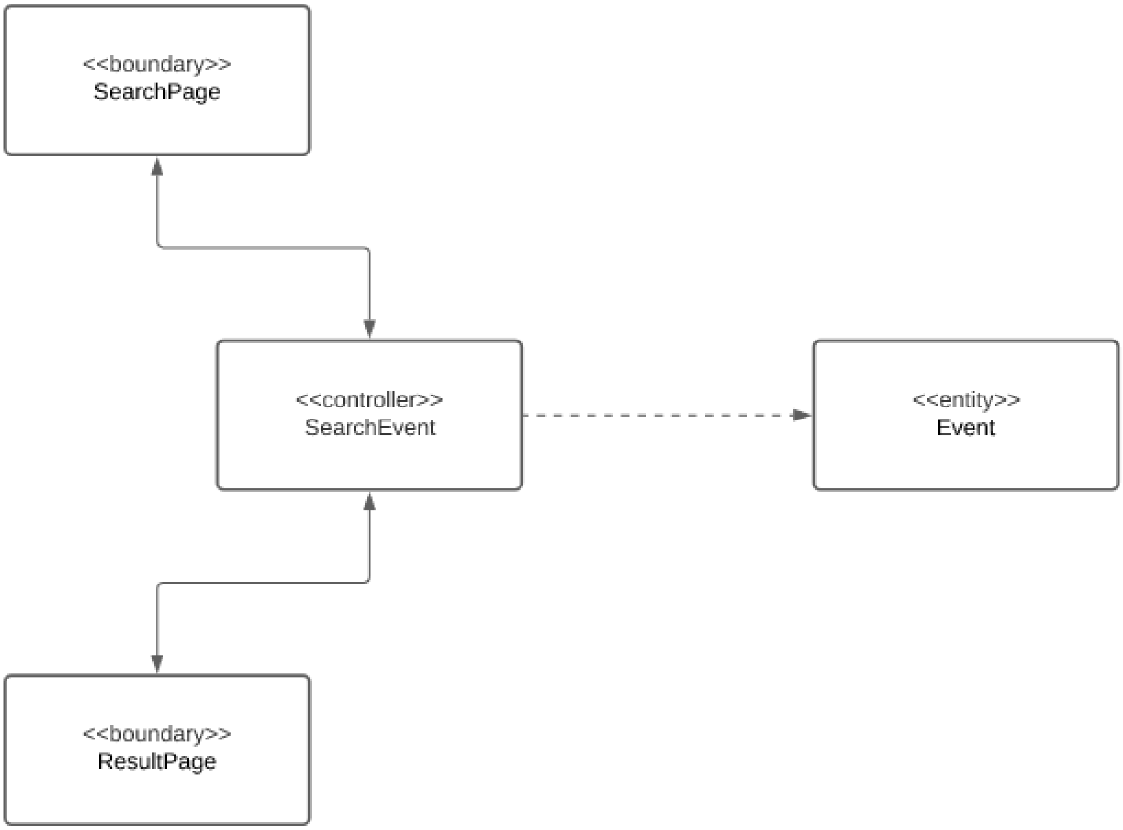


Figure 10.4 Search for Event Analysis Class

10.5 Modify Event

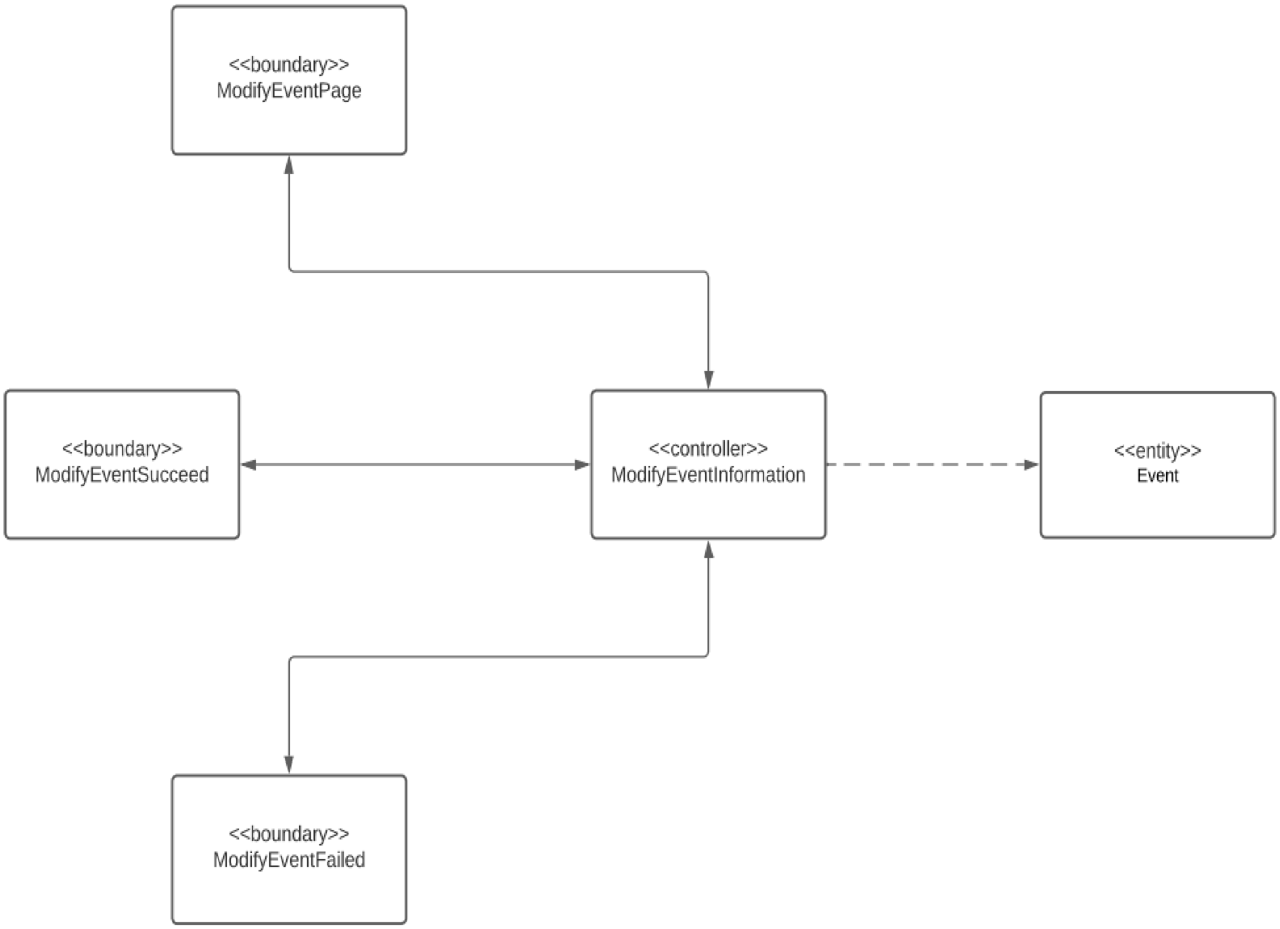


Figure 10.5 Modify Event Analysis Class

11. Interaction Diagram

11.1 Recommend Event

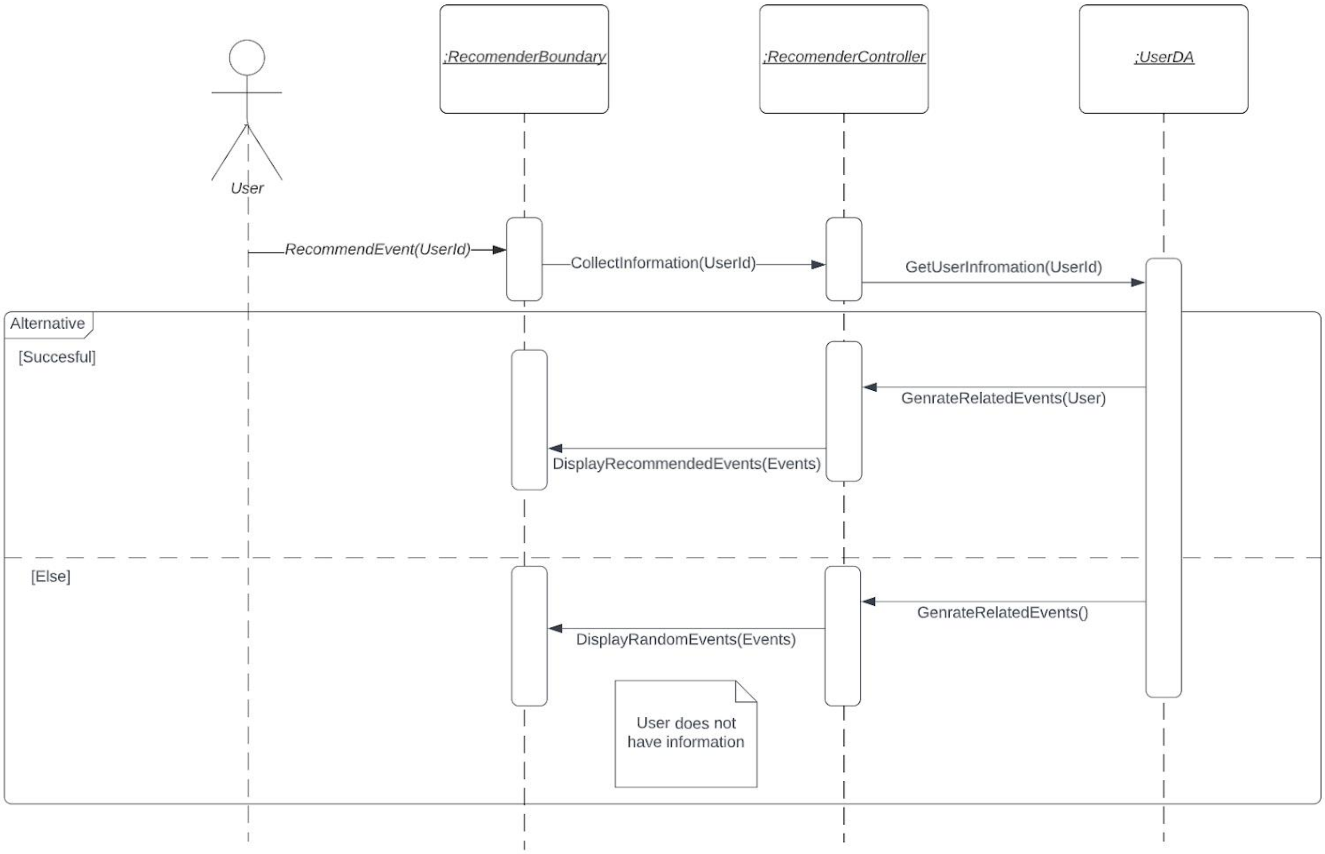

:RecomenderBoundary
:RecomenderController
:UserDA
User
RecommendEvent(UserId)
CollectInformation(UserId)
GetUserInfromation(UserId)
Alternative
[Succesful]
GenrateRelatedEvents(User)
DisplayRecommendedEvents(Events)
[Else]
GenrateRelatedEvents()
DisplayRandomEvents(Events)
User does not have information

Figure 11.1 Recommend Event Sequence Diagram

11.2 View User Profile

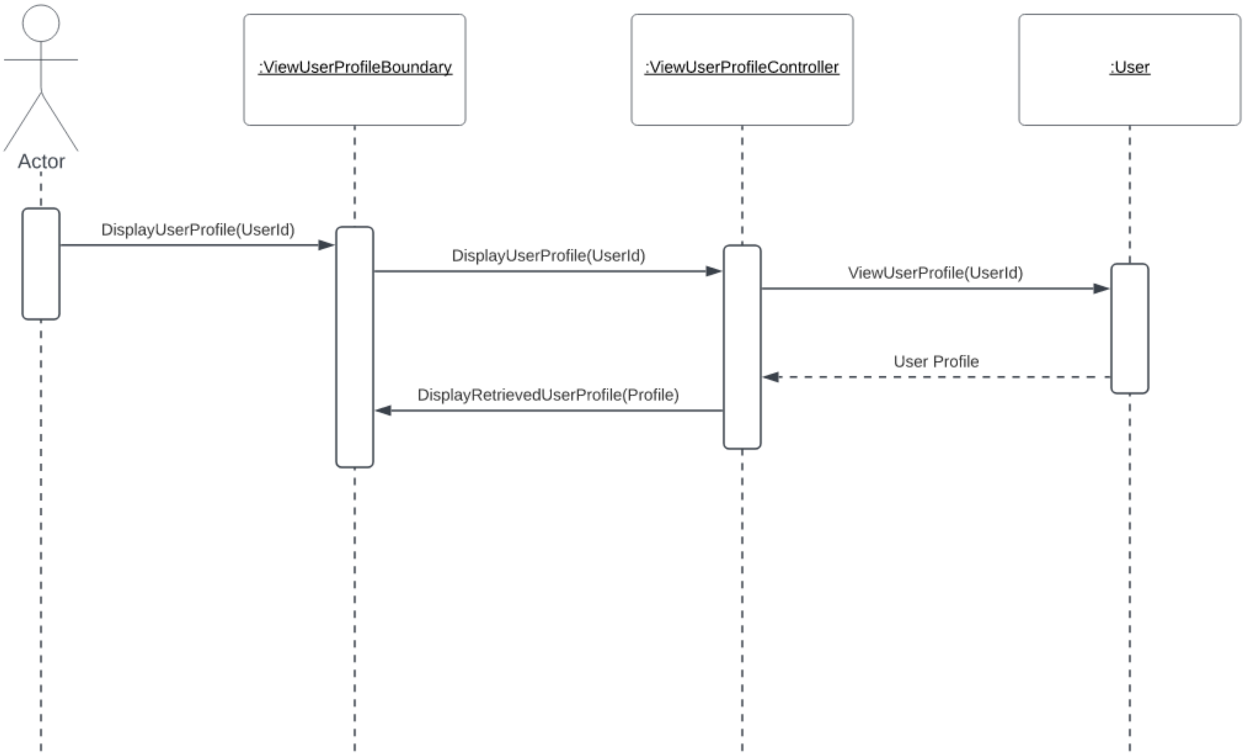
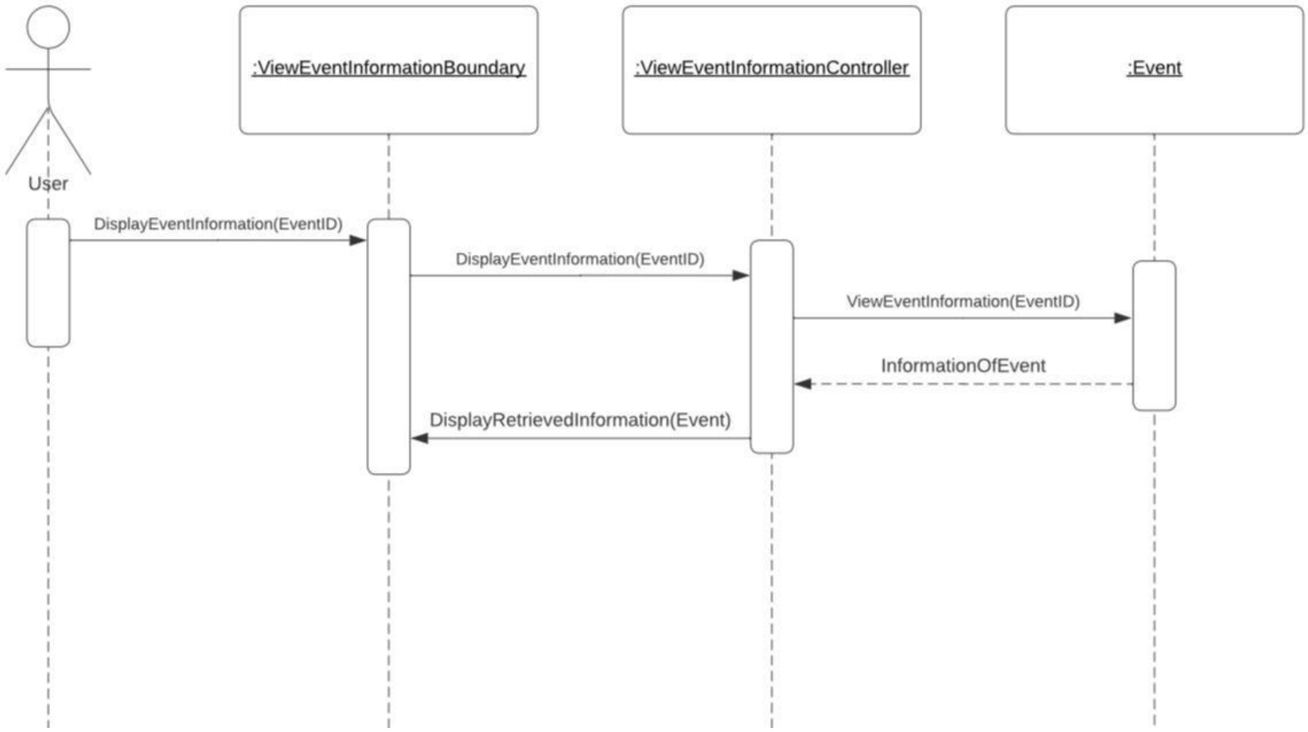

Actor
:ViewUserProfileBoundary
:ViewUserProfileController
:User
DisplayUserProfile(UserId)
DisplayUserProfile(UserId)
ViewUserProfile(UserId)
User Profile
DisplayRetrievedUserProfile(Profile)

Figure 11.2 View User Profile Sequence Diagram

11.3 View Event Information



User
:ViewEventInformationBoundary
:ViewEventInformationController
:Event
DisplayEventInformation(EventID)
DisplayEventInformation(EventID)
ViewEventInformation(EventID)
InformationOfEvent
DisplayRetrievedInformation(Event)

Figure 11.3 View Event Information Sequence Diagram

11.4 Search For Event

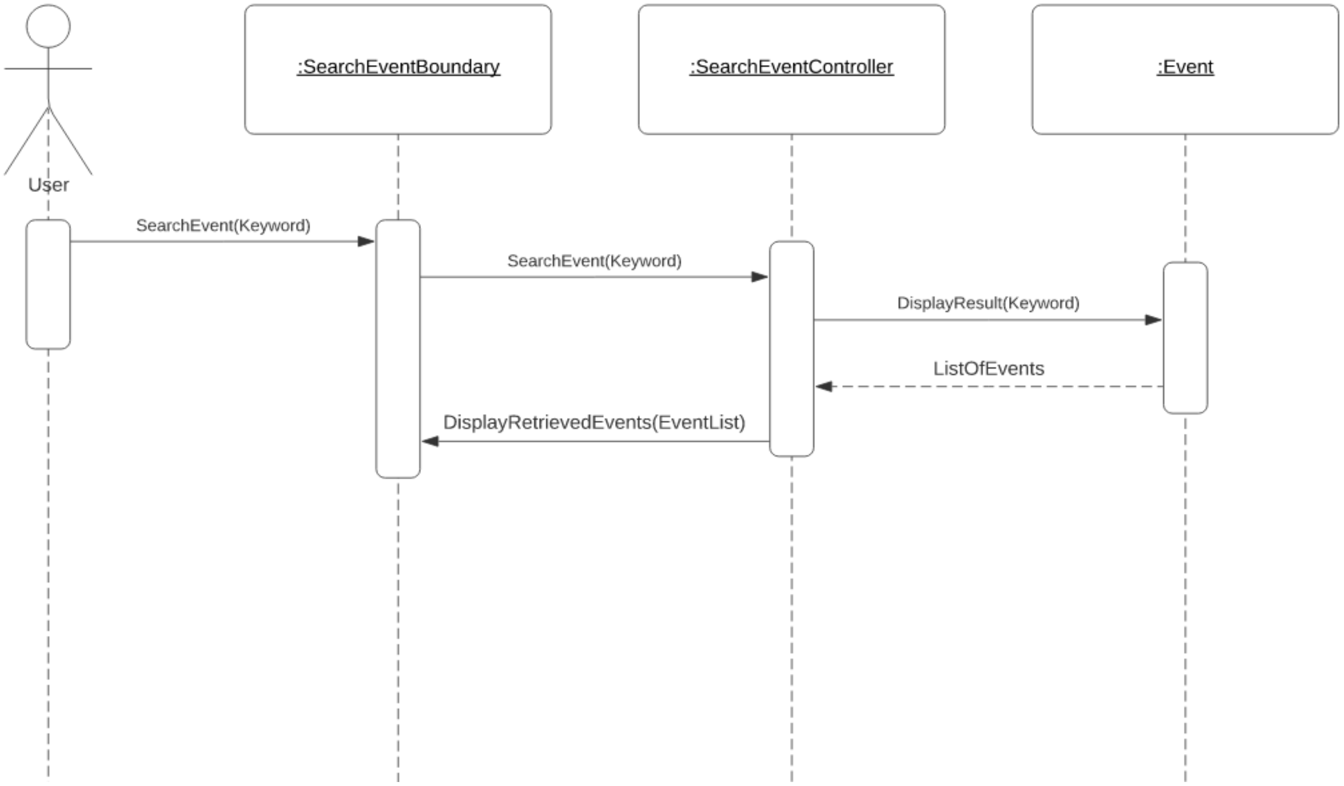


Figure 11.4 Search for Event Sequence Diagram

11.5 Modify Event

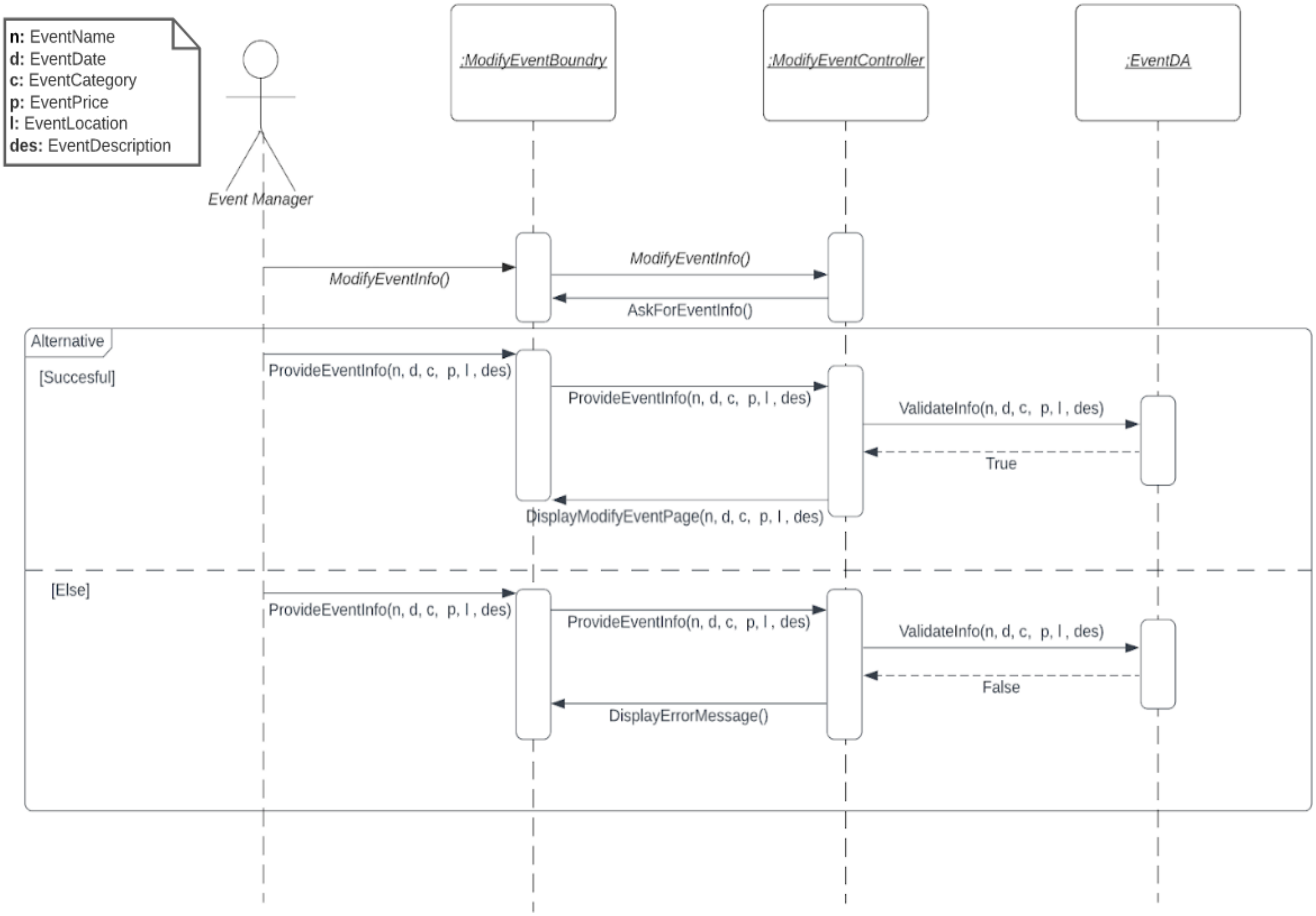
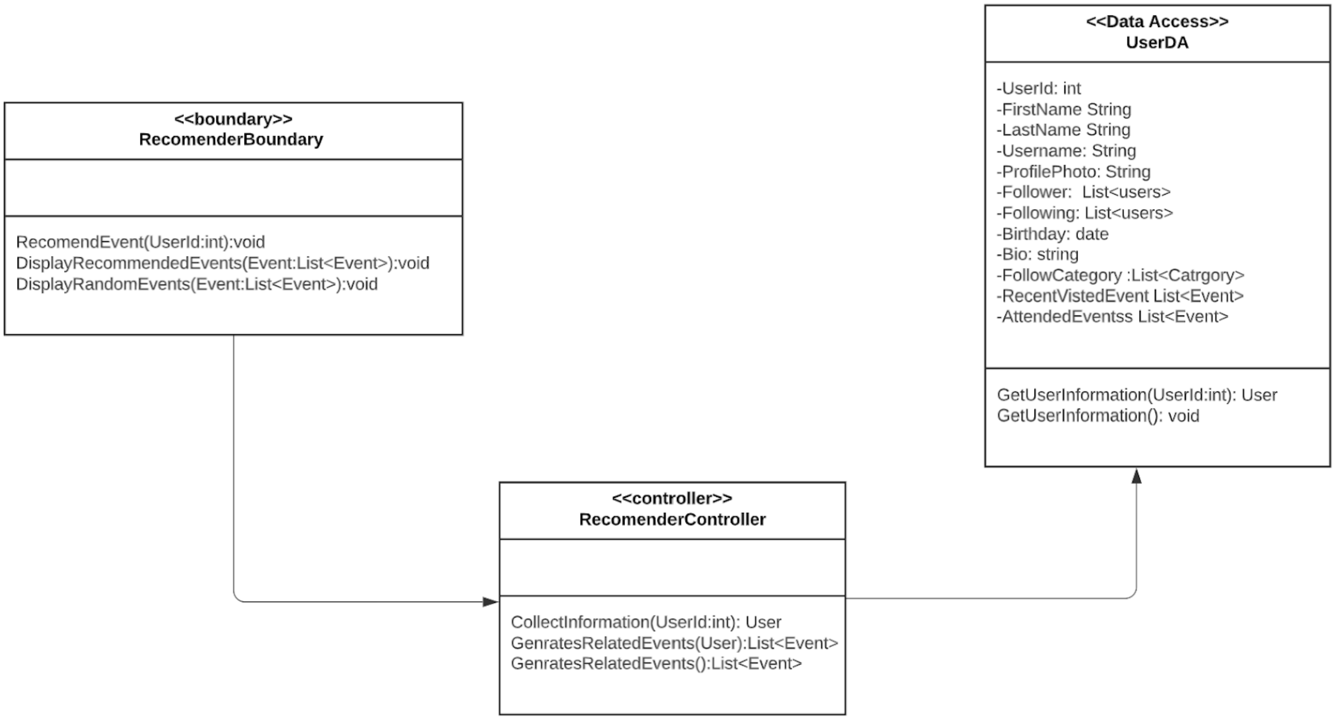


Figure 11.5 Modify Event Sequence Diagram

12. Design Class

12.1 Recommend Event



<<Data Access>>
UserDA
-UserId: int
-FirstName String
-LastName String
-Username: String
-ProfilePhoto: String
-Follower: List<users>
-Following: List<users>
-Birthday: date
-Bio: string
-FollowCategory :List<Catrgory>
-RecentVistedEvent List<Event>
-AttendedEventss List<Event>
GetUserInformation(UserId:int): User
GetUserInformation(): void
<<boundary>>
RecomenderBoundary
RecomendEvent(UserId:int):void
DisplayRecommendedEvents(Event:List<Event>):void
DisplayRandomEvents(Event:List<Event>):void
<<controller>>
RecomenderController
CollectInformation(UserId:int): User
GenratesRelatedEvents(User):List<Event>
GenratesRelatedEvents():List<Event>

Figure 12.1 Recommend Event Design Class

12.2 View User Profile

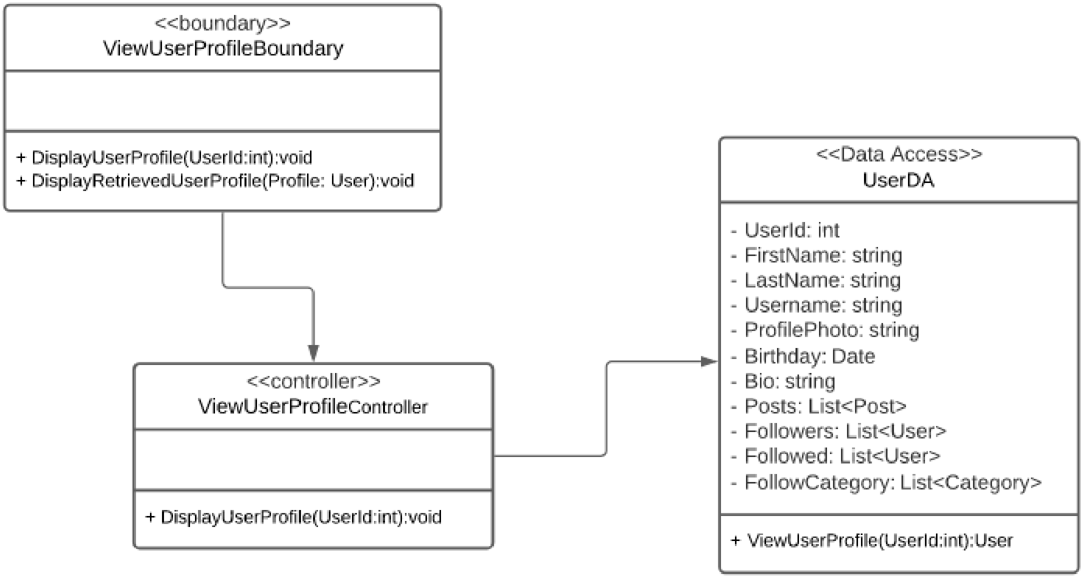
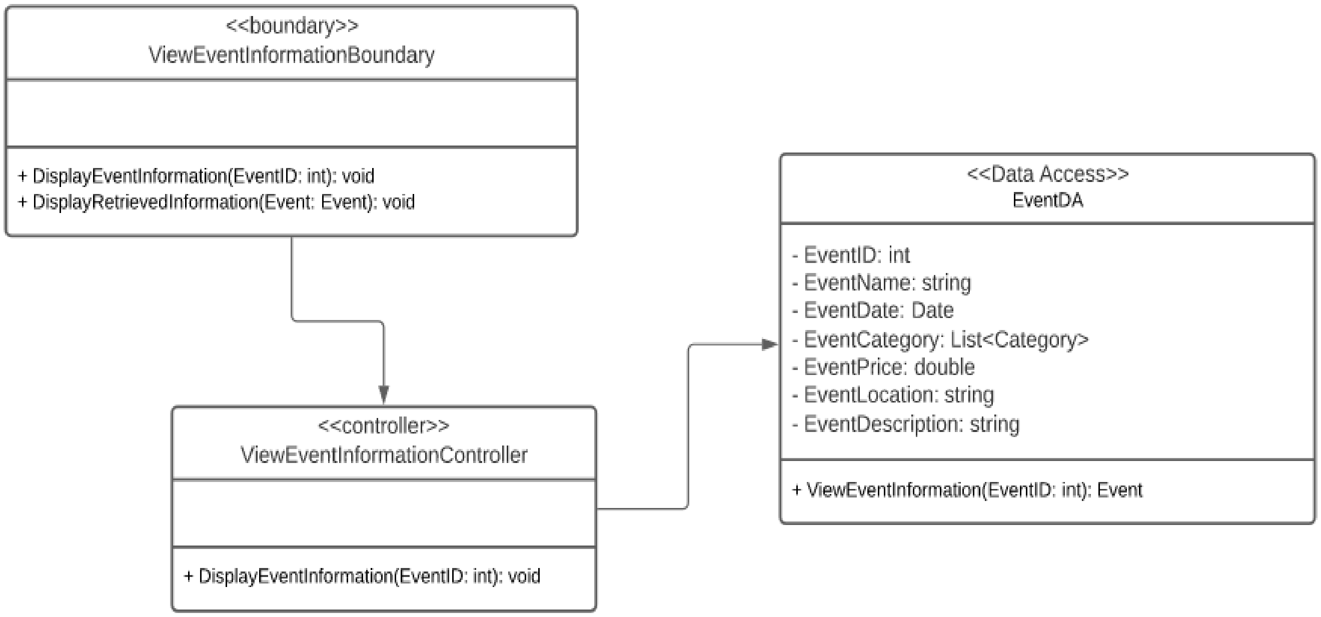


Figure 12.2 View User Profile Design Class

12.3 View Event Information



<<boundary>>
ViewEventInformationBoundary
+ DisplayEventInformation(EventID: int): void
+ DisplayRetrievedInformation(Event: Event): void
<<controller>>
ViewEventInformationController
+ DisplayEventInformation(EventID: int): void
<<Data Access>>
EventDA
- EventID: int
- EventName: string
- EventDate: Date
- EventCategory: List<Category>
- EventPrice: double
- EventLocation: string
- EventDescription: string
+ ViewEventInformation(EventID: int): Event

Figure 12.3 View Event Information Design Class

12.4 Search For Event

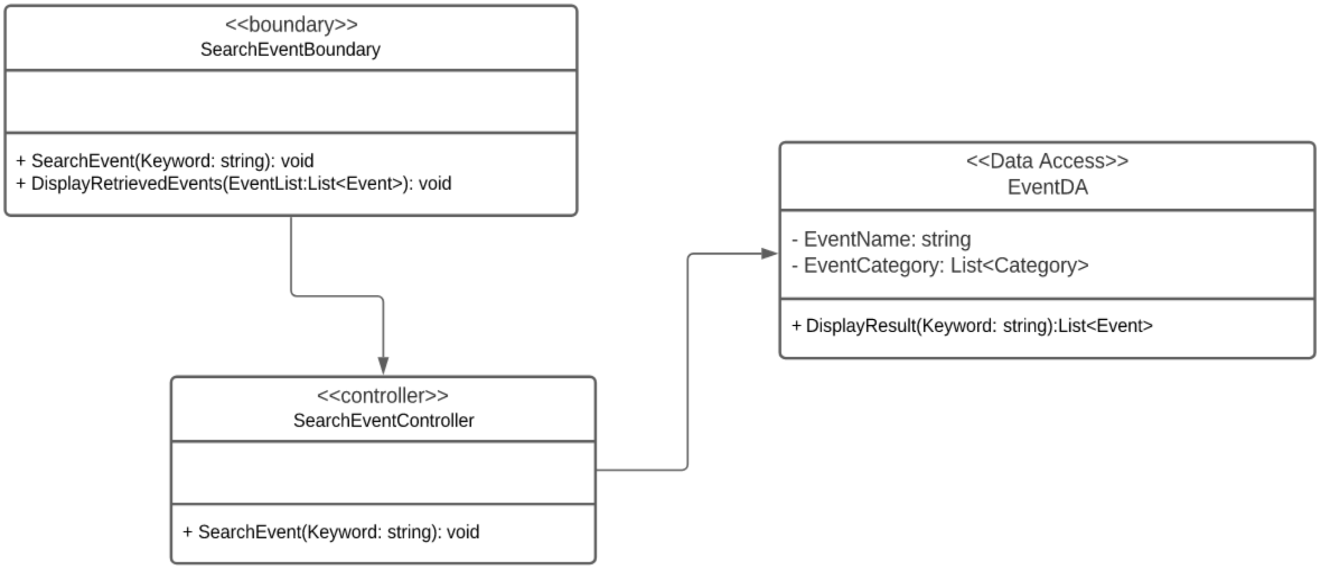

<<boundary>>
SearchEventBoundary
+ SearchEvent(Keyword: string): void
+ DisplayRetrievedEvents(EventList:List<Event>): void
<<Data Access>>
EventDA
- EventName: string
- EventCategory: List<Category>
+ DisplayResult(Keyword: string):List<Event>
<<controller>>
SearchEventController
+ SearchEvent(Keyword: string): void

Figure 12.4 Search for Event Design Class

12.5 Modify Event

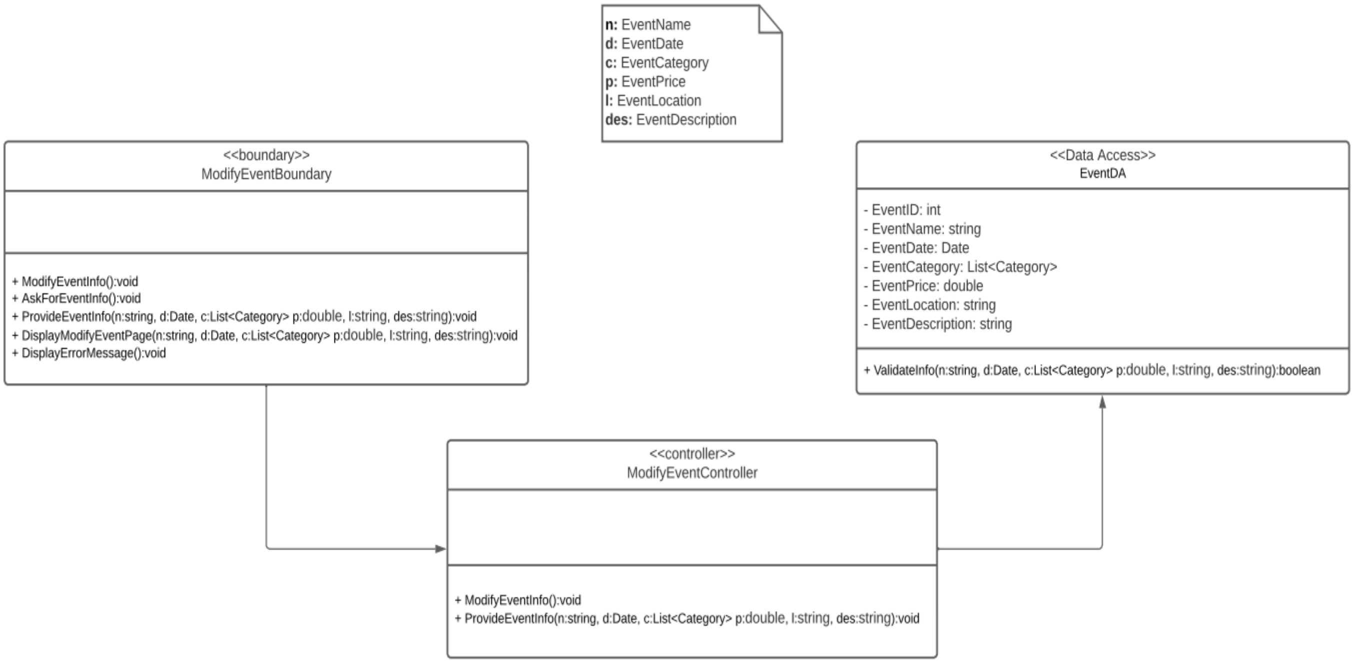

n: EventName
d: EventDate
c: EventCategory
p: EventPrice
l: EventLocation
des: EventDescription
<<boundary>>
ModifyEventBoundary
+ ModifyEventInfo():void
+ AskForEventInfo():void
+ ProvideEventInfo(n:string, d:Date, c:List<Category> p:double, l:string, des:string):void
+ DisplayModifyEventPage(n:string, d:Date, c:List<Category> p:double, l:string, des:string):void
+ DisplayErrorMessage():void
<<Data Access>>
EventDA
- EventID: int
- EventName: string
- EventDate: Date
- EventCategory: List<Category>
- EventPrice: double
- EventLocation: string
- EventDescription: string
+ ValidateInfo(n:string, d:Date, c:List<Category> p:double, l:string, des:string):boolean
<<controller>>
ModifyEventController
+ ModifyEventInfo():void
+ ProvideEventInfo(n:string, d:Date, c:List<Category> p:double, l:string, des:string):void

Figure 12.5 Modify Event Design Class

13. System Architecture

We have considered many system architectures. But after the research we found we need an architecture that supports a fast development process. And support frequent modification.
Therefore, we found the MVC (Model, View, Controller) suitable for our system and needs.

MVC has many frameworks that support it. Also, it is great for the development team. As the three components (Model, View, Controller) are separated, the development team can work concurrently.

Model: The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a customer object will retrieve the customer information from the database, manipulate it and update its data back to the database or use it to render data.

View: The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

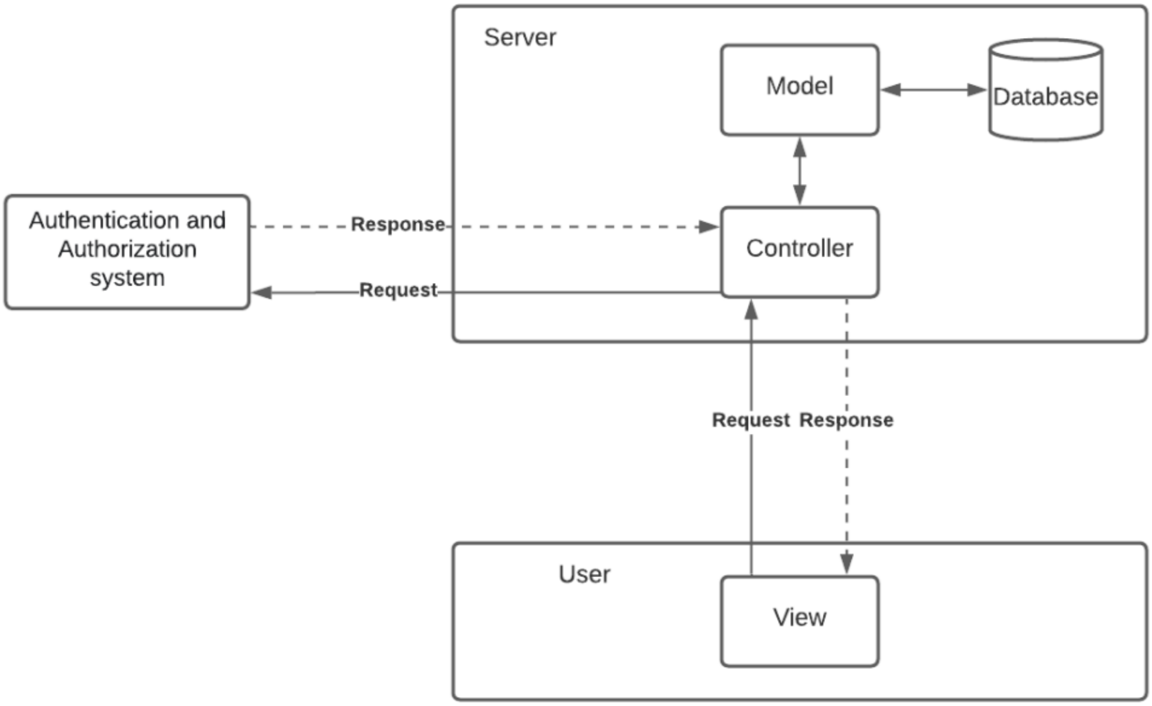


Figure 13.1 System Architecture

14. User Interface Mockup

14.1 Home Page

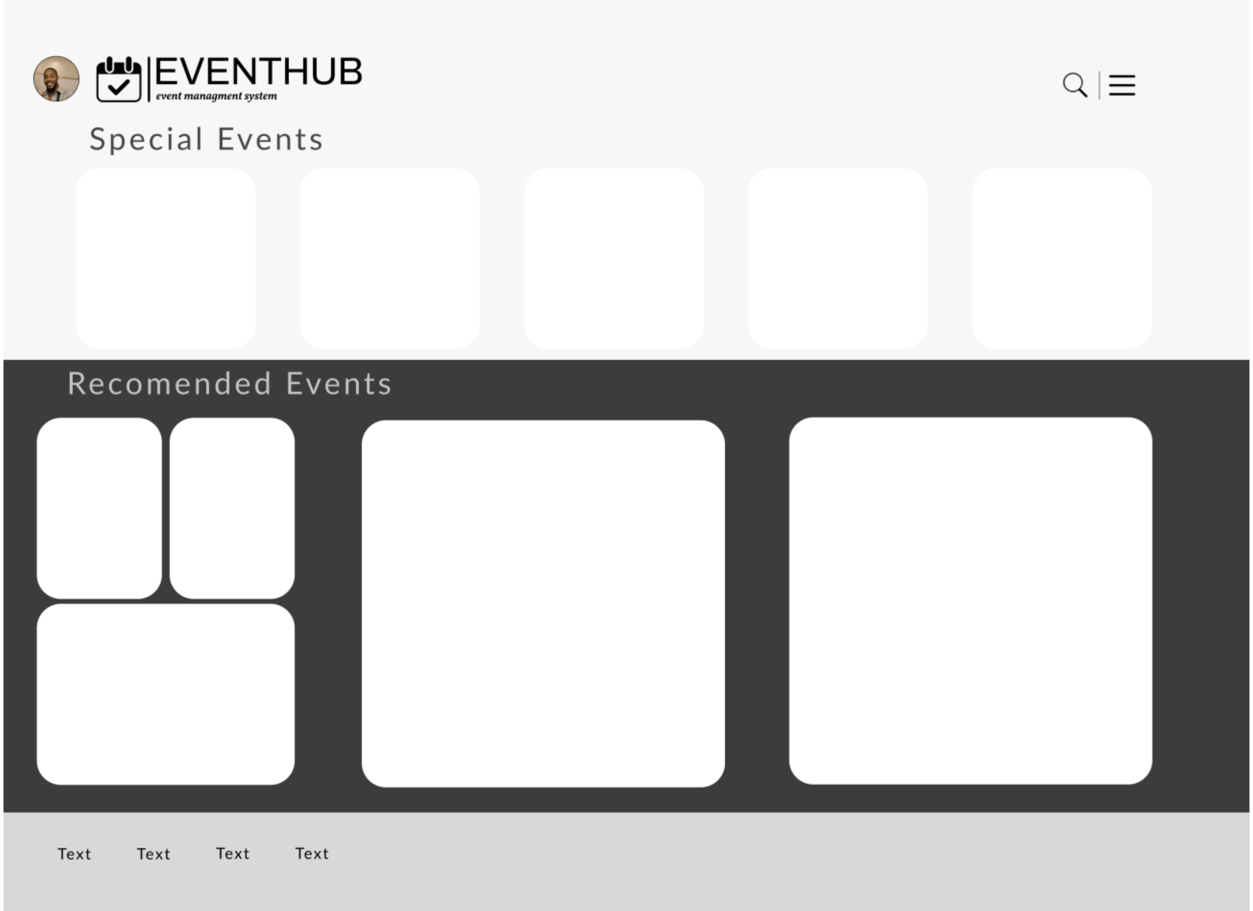


Figure 14.1 Home Page Mockup

14.2 Apply For Event Page

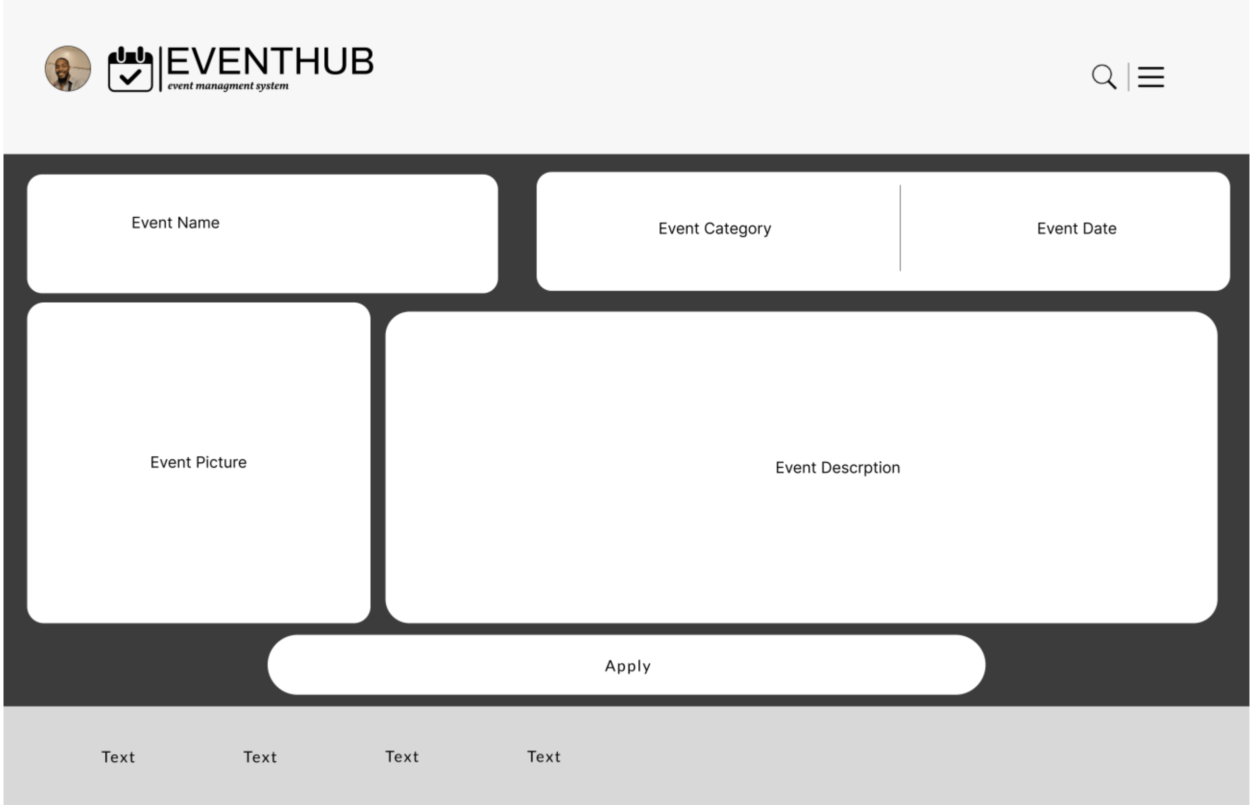


Figure 14.2 Apply for Event Page Mockup

14.3 Attended Event Page

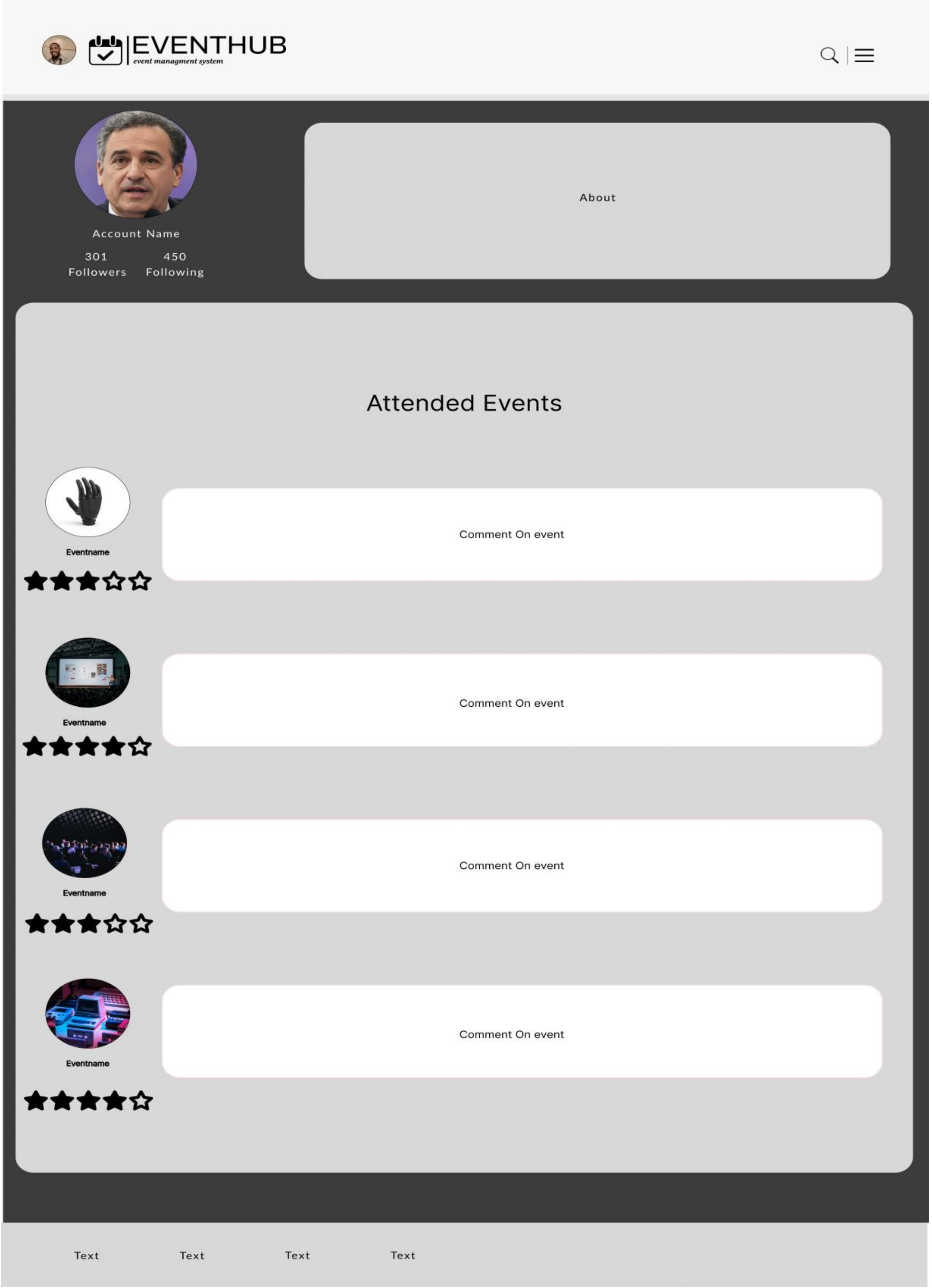


Figure 14.3 Attended Event Page Mockup

14.4 Account Information Page

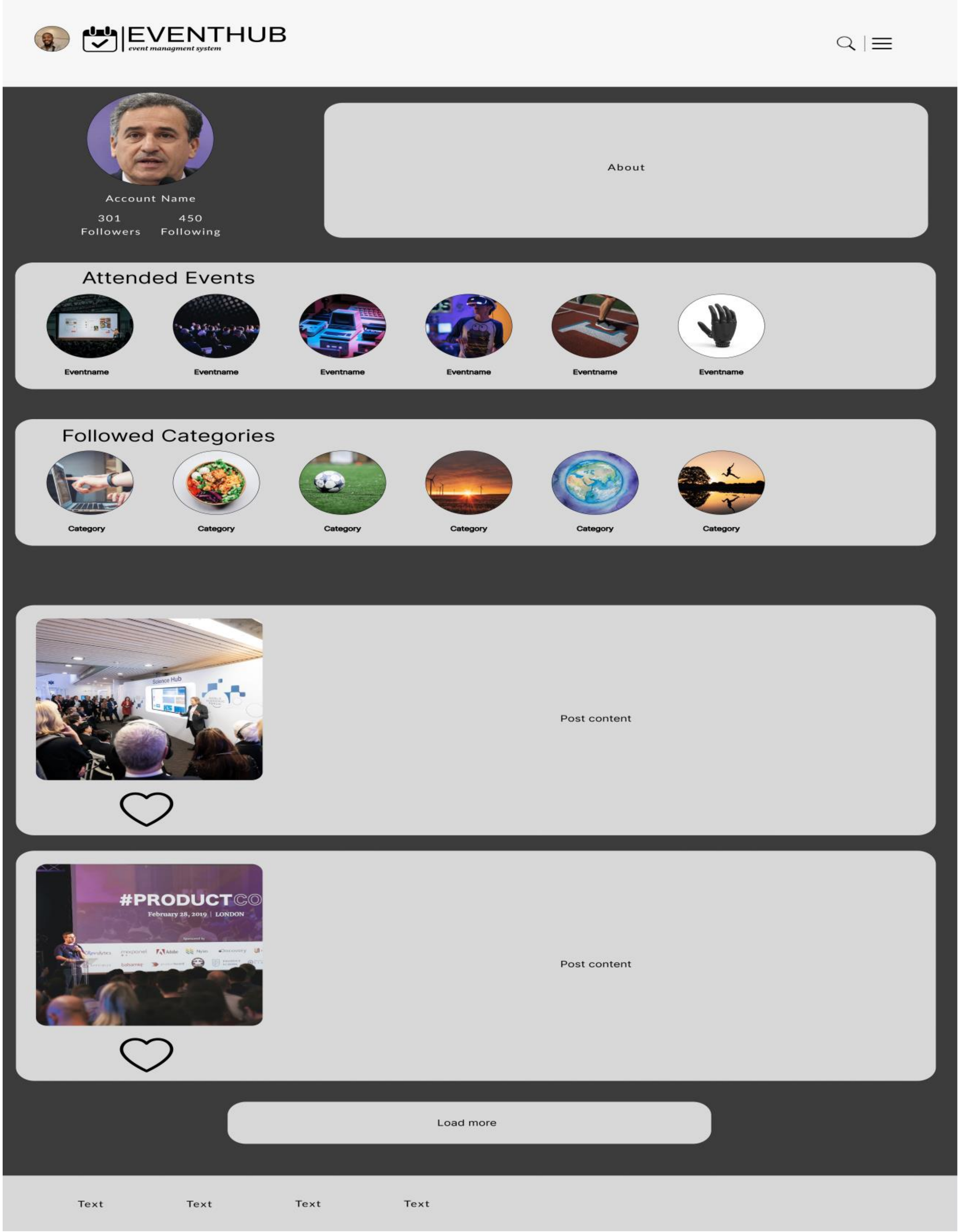


Figure 14.4 Account Information Page Mockup

15. Database Schema

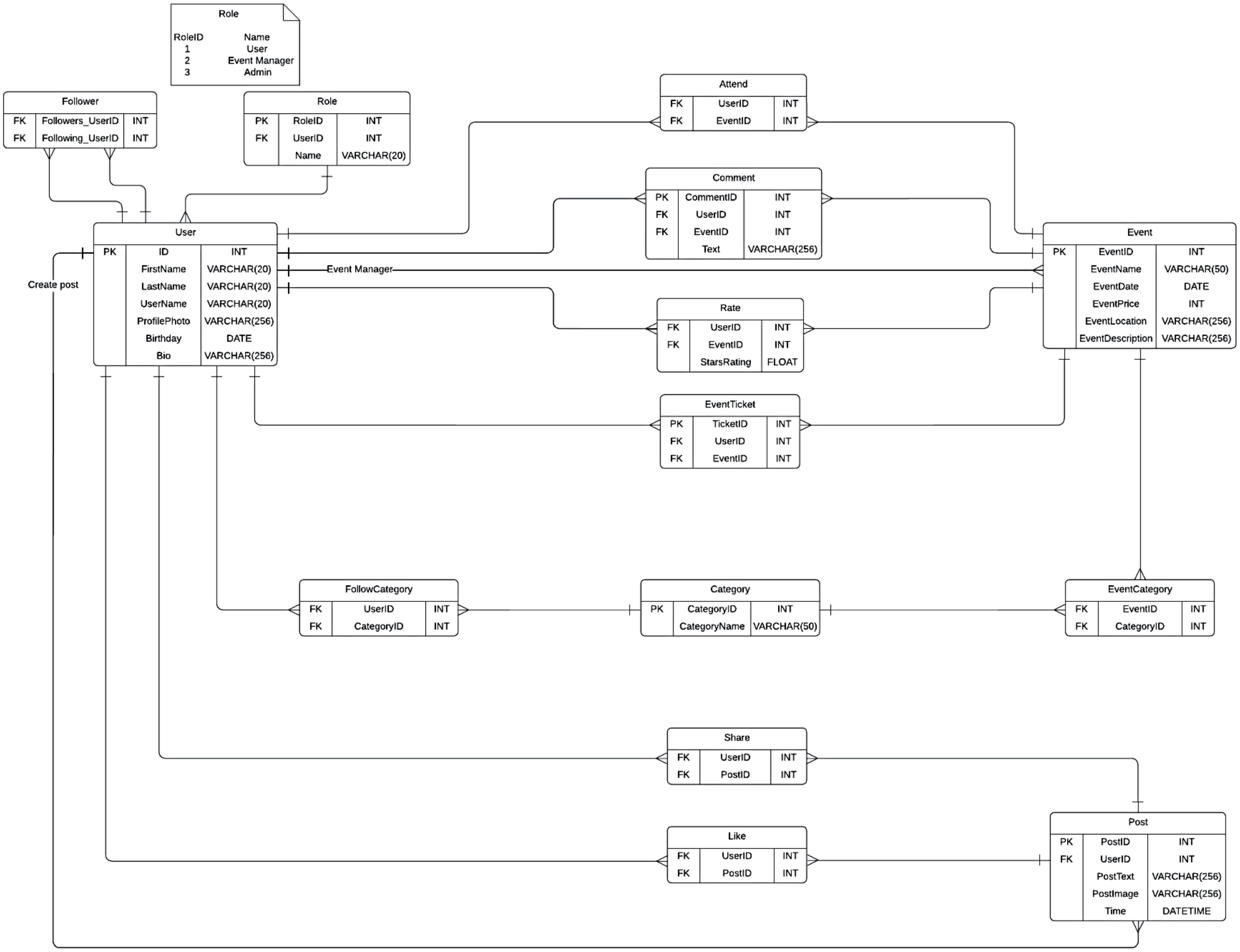
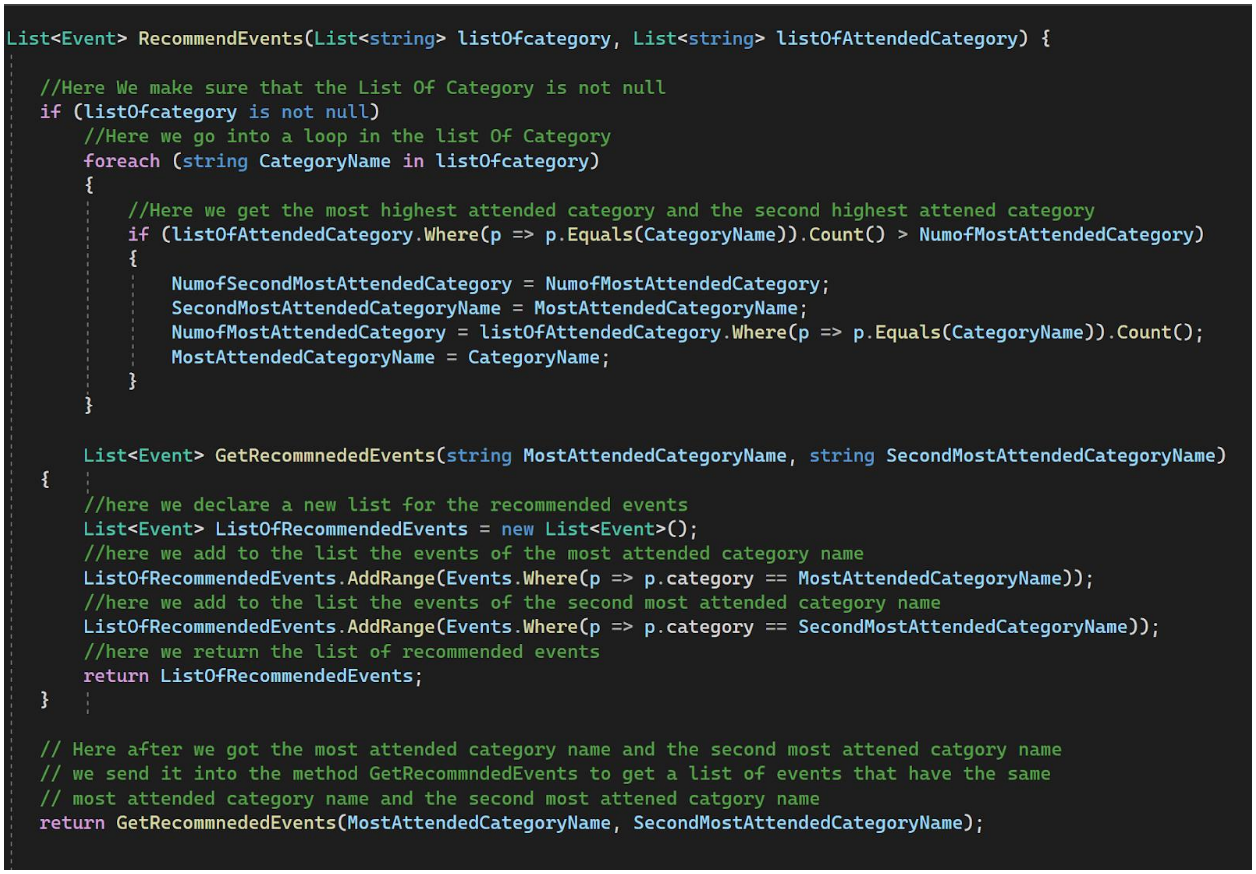


Figure 15.1 Database Schema

16. Algorithms

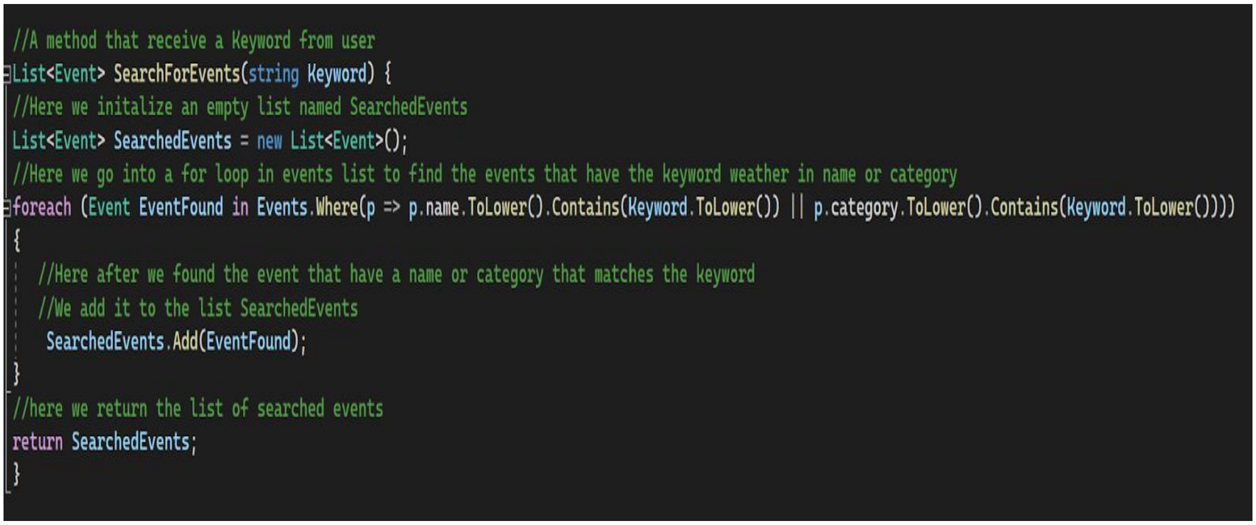
16.1 Recommendation Algorithm



```
List<Event> RecommendEvents(List<string> listOfcategory, List<string> listOfAttendedCategory) {

    //Here We make sure that the List Of Category is not null
    if (listOfcategory is not null)
        //Here we go into a loop in the list Of Category
        foreach (string CategoryName in listOfcategory)
        {
            //Here we get the most highest attended category and the second highest attened category
            if (listOfAttendedCategory.Where(p => p.Equals(CategoryName)).Count() > NumofMostAttendedCategory)
            {
                NumofSecondMostAttendedCategory = NumofMostAttendedCategory;
                SecondMostAttendedCategoryName = MostAttendedCategoryName;
                NumofMostAttendedCategory = listOfAttendedCategory.Where(p => p.Equals(CategoryName)).Count();
                MostAttendedCategoryName = CategoryName;
            }
        }

        List<Event> GetRecommnededEvents(string MostAttendedCategoryName, string SecondMostAttendedCategoryName)
    {
        //here we declare a new list for the recommended events
        List<Event> ListOfRecommendedEvents = new List<Event>();
        //here we add to the list the events of the most attended category name
        ListOfRecommendedEvents.AddRange(Events.Where(p => p.category == MostAttendedCategoryName));
        //here we add to the list the events of the second most attended category name
        ListOfRecommendedEvents.AddRange(Events.Where(p => p.category == SecondMostAttendedCategoryName));
        //here we return the list of recommended events
        return ListOfRecommendedEvents;
    }

    // Here after we got the most attended category name and the second most attened catgory name
    // we send it into the method GetRecommndedEvents to get a list of events that have the same
    // most attended category name and the second most attened catgory name
    return GetRecommnededEvents(MostAttendedCategoryName, SecondMostAttendedCategoryName);
```

Figure 16.1 Recommendation Algorithm

16.2 Search Algorithm



```
//A method that receive a Keyword from user
List<Event> SearchForEvents(string Keyword) {
//Here we initalize an empty list named SearchedEvents
List<Event> SearchedEvents = new List<Event>();
//Here we go into a for loop in events list to find the events that have the keyword weather in name or category
foreach (Event EventFound in Events.Where(p => p.name.ToLower().Contains(Keyword.ToLower()) || p.category.ToLower().Contains(Keyword.ToLower())))
{
   //Here after we found the event that have a name or category that matches the keyword
   //We add it to the list SearchedEvents
    SearchedEvents.Add(EventFound);
}
//here we return the list of searched events
return SearchedEvents;
}
```

Figure 16.2 Search Algorithm

17. Expected Deployment

Global Overview

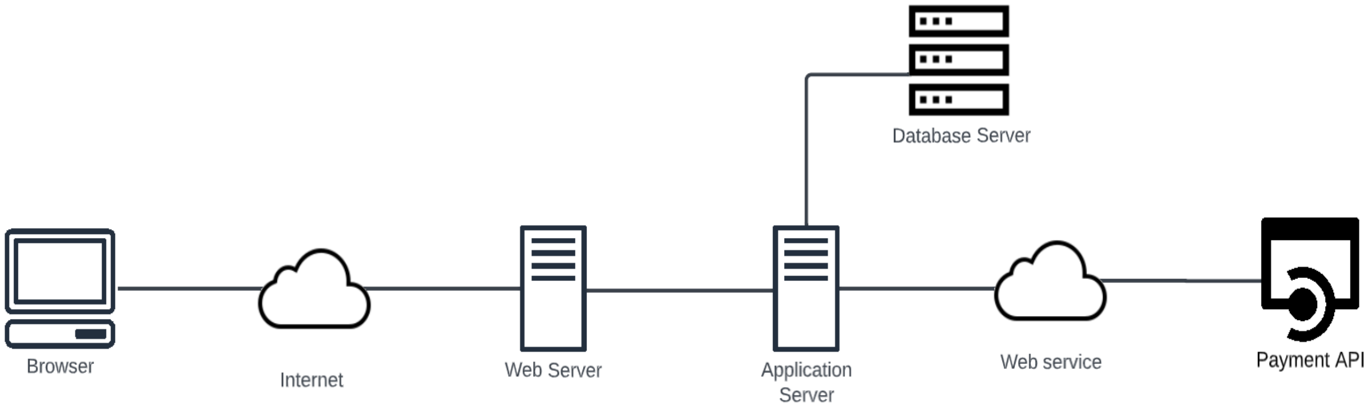


Figure 17.1 System deployment diagram

1. Browser: What the user will use to access the system.

2. Internet: The connector between the user and the server.

3. Web Server: Used to serve pages and static content to the user through https requests.

4. Application Server: Where the system components will be, and it is used to generate the web pages for the user.

5. Database server: A server that will communicate with the database will be used to send/retrieve user related data to/from the system.

6. Payment API: A web service that is used to do all the payment processes with the banks and all the technology needed for that and it validates the payment.

18. Test Scenario

In this section we will specify some of the important test cases for some of the scenarios.

18.1 Event manager Modify event

Here are the test cases for Event manager to modify an event.
Table 18.1.1 specifies the course of the test case when the input is correct.
Table 18.1.2 specifies the course of the test case when the input is not correct.

Test case:

TC1 Goal: To make sure the system behaves correctly when modifying an event with correct input.
Scenario reference: 1
Setup: modify an event with 3rd of April 2023 for the new start date, 11th of September for the new end date, and ‘Valid test’ for the Event description.
Pass criteria: An event is modified.

External event	Reaction
Event manager modifies an event with: new Start date: 3/4/2023 new end date: 11/9/2023 event description: Valid test	The system validates the input, then modifies an Event automatically and assigns the event to the event manager who creates this event.

Table 18.1.1 Modify an event– valid test scenario

Test case:

TC2 Goal: To make sure the system behaves correctly when modifying an event with the end date being before the start date.
Scenario reference: 2
Setup: modify an event 11th of September 2023 for the new start date, 3rd of April 2023 for the new end date, and ‘Failed test’ for the event description.
Pass criteria: An error message is displayed, and the Event is not modified.

External event	Reaction
event manager modifies an event with: new Start date: 11/9/2023 new end date: 3/4/2023 event description: failed test	The system validates the input, then displays an error message informing the event manager that the start date is after the end date.

Table 18.1.2 Modify an event– failed test scenario

18.2 User Search for Event

Here are the test cases for user to search for an event.
Table 18.2.1 specifies the course of the test case when the Searched event is found.
Table 18.2.2 specifies the course of the test case when the Searched event is not found.

Test case:

TC1 Goal: To make sure the system behaves correctly when searching for an event that is found.
Scenario reference: 1
Setup: Searching for an event with 'Car race' which is a found event and 'Valid test' for the Event description.
Pass criteria: An event is created.

External event	Reaction
user search for an event with: input: Car race event description: Valid test	The system validates the input, then displays the searched event to the user.

Table 18.2.1 Search for an event– valid test scenario

Test case:

TC2 Goal: To make sure the system behaves correctly when searching for an event that is not found.
Scenario reference: 2
Setup: Searching for an event with 'calculator' which is an event not found. and 'Failed test' for the event description.
Pass criteria: An error message is displayed, and the Event is not found.

External event	Reaction
user search for an event with: input: calculator event description: failed test	The system validates the input, then displays an error message informing the user that the searched event is not found.

Table 18.2.2 Search for an event– failed test scenario

18.3 Event manager creates an event

Here are the test cases for Event manager to create an event.
Table 18.3.1 specifies the course of the test case when the input is correct.
Table 18.3.2 specifies the course of the test case when the input is not correct.

Test case:

TC1 Goal: To make sure the system behaves correctly when creating a new event with correct input.
Scenario reference: 1
Setup: Create an event with 'Car race' as a title, 3rd of April 2023 for the start date, 11th of September for the end date, and 'Valid test' for the Event description.
Pass criteria: An event is created.

External event	Reaction
event manager creates an event with: Project title: Car race Start date: 3/4/2023 end date: 11/9/2023 event description: Valid test	The system validates the input, then creates an Event automatically and assigns the event to the event manager who create this event.

Table 18.3.1 Create an event– valid test scenario

Test case:
TC2 Goal: To make sure the system behaves correctly when creating a new event with the end date being before the start date.
Scenario reference: 2
Setup: Create an event with 'Car race' as a title, 11th of September 2023 for the start date, 3rd of april 2023 for the end date, and 'Failed test' for the event description.
Pass criteria: An error message is displayed, and the Event is not created.

External event	Reaction
event manager creates an event with: Project title: Car race Start date: 11/9/2023 end date: 3/4/2023 event description: failed test	The system validates the input, then displays an error message informing the event manager that the start date is after the end date.

Table 18.3.2 Create an event– failed test scenario

18.4 System recommends related event to the user

Here are the test cases for a system to recommend related events to the user.
Table 18.4.1 specifies the course of the test case when the system has enough information for the user.
Table 18.4.2 specifies the course of the test case when the system has not enough information to the user due to the user is a new user.

Test case:

TC1 Goal: To make sure the system behaves correctly when recommending an event with enough information to the user.
Scenario reference: 1
Setup: recommend an event when the system has enough information to the user such as user's followed categories, user's attended events, and user's event browsing history, and 'Valid test' for the Event description.
Pass criteria: related event to the user is recommended.

External event	Reaction
system recommend an event when: the system has enough information for the user event description: Valid test	The system collects the information about the user, then generates a model to recommend related events to the user, then the related event is recommended to the user.

Table 18.4.1 System recommends related event to the user– valid test scenario

Test case:

TC2 Goal: To make sure the system behaves correctly when recommending an event with not enough information to the user
Scenario reference: 2
Setup: recommend an event when the system does not have enough information for the user due to the user is a new user. and 'Failed test' for the event description.
Pass criteria: random event is recommended.

External event	Reaction
system recommend an event when: the system does not have enough information for the user event description: failed test	The system generates random events to the user then the random event recommended to the user.

Table 18.4.2 System recommends related event to the user– failed test scenario

19. Project Status

The project has 4 major parts: Analysis, design, implementation, and testing. On the 8th of November we completed the analysis and design phases of the project, On the first 3 weeks we had issues with organizing the work, But we did find a great technique after that the workflow was much better, After that we did face some challenges in understanding the document, But the meeting with our supervisor Dr. Fayez AL Qahtani helped us a lot in understanding the document and give us a very clear image in what we supposed to do.

20. Conclusion

Throughout this report, we have gradually refined the document and outline all difficulties event managers face during the establishment /deployment processes of event management. We have read up on the different aspects of management and how software can aid in organizing the event managers' tasks.

We did not just stop looking at the competitors, we found our own little niche that appears in two exclusive features social networking, recommendation system.

In this report we defined a project plan that shows what will be done each week with the risks we might face. We also defined the functionalities of the system, use cases, and illustrated those use cases to give a better detailed understanding. Furthermore, we have shown different components of our system and how it will be deployed.

To conclude, we hope the second part of this project will be a high-yielding experience for us all and we wish to expand our knowledge in different development aspects and technologies.

21. Reference

- Galin, D. (2004). Software Quality Assurance: From Theory To Implementation. Pearson Education Limited.

- Heim, S. (2008). The Resonant Interface: HCI Foundations for Interactive Design. nltk. (2022, 51).

- Pezze, M., & Young, M. (2007). Software Testing and Analysis: Process, Principles, and Techniques. Wiley.

- Qian, K., Fu, X., Tao, L., & X, C.-w. (2009). Software Architecture and Design Illuminated. Jones & Barlett

- Ramez Elmasri and Shamkant B. Navathe, (2017). Fundemntals of Database Systems Pearson Education Limited.