King Fahd University of Petroleum & Minerals

Department of Information and Computer Science
SWE363: Web Engineering & Development

Term 192

Project

Volunteering Teams Management Website

Team Members:

1. Mojtaba Alamer   [201648240]
2. Haitham alsaeed   [201628180]

Contents

Description of the requirements:

Purpose of the document:

The purpose of this document is to give a description of the whole structure of a web-based system that will allow Volunteering teams to manage their events, assigning members to events teams and managing the attendances.

Project Goals and requirements:

1. The system shall allow the guest user to register as a new volunteer.
2. The system shall allow the user to browse the Incoming events.
3. The system shall allow the user to search for a particular event.
4. The system shall allow guest users to sign in.
5. The system shall allow the signed-in users to sign out.
6. The system shall allow the signed-in users to view profiles.
7. The system shall allow a signed-in user to apply to participate in an event.
8. The system shall allow signed-in users to check the status of their applications.
9. The system shall allow the signed-in user to cancel his application.
10. The system shall allow the signed-in user to view the previous events that he did participate in it.
11. The user shall be able to see an overview of the website.
12. The user shall be able to sign up.
13. The user shall be able to sign in.
14. The user shall be able to Browse the incoming events.
15. The user shall be able to search for a particular event.
16. The Team member shall be able to check his profile.
17. The Team member shall be able to Browse the incoming events.
18. The Team member shall be able to apply for events.
19. The Team member shall be able to check his application status.
20. The Team member shall be able to cancel his application.
21. The Team member shall be able to check his attendance.
22. The System admin shall be able to add members.

23. The System admin shall be able to delete members.

24. The System admin shall be able to block members.

25. The System admin shall be able to create an event.

26. The System admin shall be able to delete an event.

27. The System admin shall be able to edit an event.

28. The System admin shall be able to assign a member to a specific team.

29. The System admin shall be able to generate a report about Event.

30. The System admin shall be able to generate a report about Team Events.

31. The System admin shall be able to generate a report about Team Members.

32. The System admin shall be able to assign an event to a specific team.

33. The System admin shall be able to schedule an event.

34. The team leader shall be able to create an event.

35. The team leader shall be able to delete an event.

36. The team leader shall be able to edit an event.

37. The team leader shall be able to browse assigned events by the admin

38. The team leader shall be able to schedule an event.

39. The team leader shall be able to take attendance.

Content of the Website and Organization:

The website will have 12 different webpages:

- Login page.

- Register page.

- Home page, which contains information about the Volunteering Teams Management system and the upcoming events with the Search option.

- Search page.

- member Dashboard page: which contains all the information about the member and his attendance history.

- Event details page: which contains all the information about the event time, date, gender and description

- Events page: which contains all the Events information like time, date, gender and description

- Admin Dashboard page: which contains buttons for the options for maintaining the members, events and reports

- Leader dashboard page: which contains buttons for the options for maintaining the application and events.

- Manage Members page: to maintains members

- Manage Events page: to maintains events

- Edit Profile page: to Edit personal information

Models

Use Case Diagram
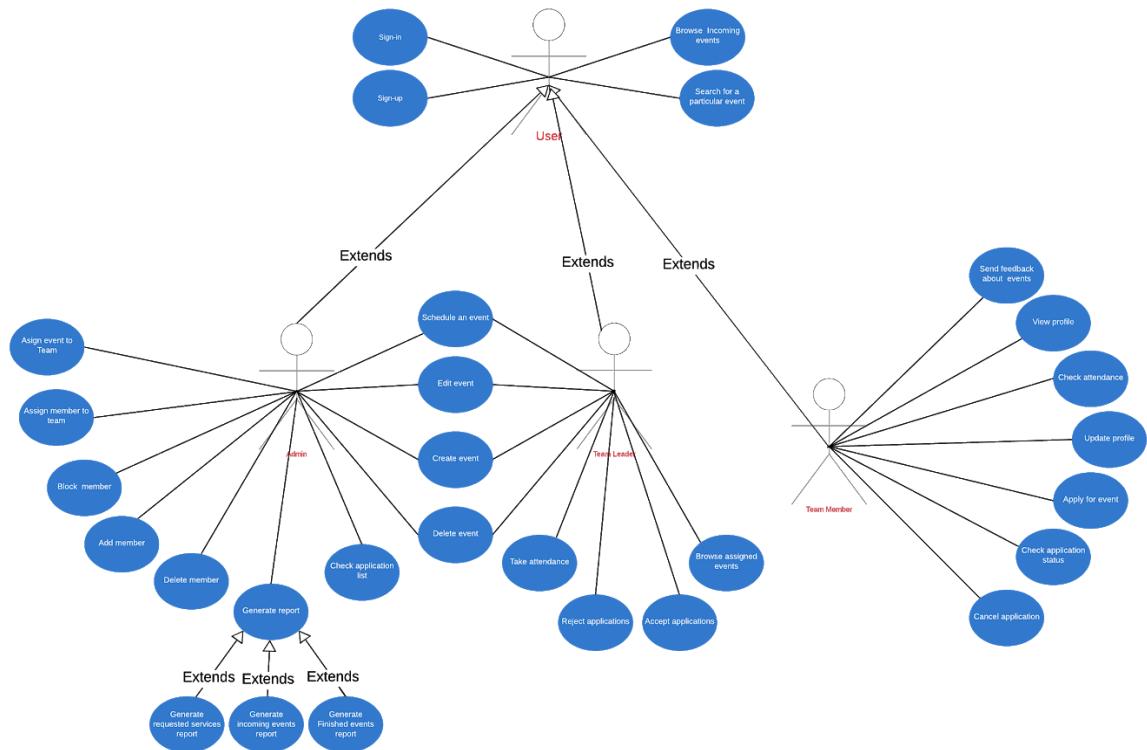
The relation between required functionality and the use cases

Functionality	Use Cases
Display services provided by your web site, Browse the available services	Browse Upcoming events
Sign in, sign up, Register	Sign-up,Sign-in
Search, Search for a particular service	Search for a particular event
Update profile	Update profile
Request a service, Track the status of the sent request, Cancel a request	Apply for event, Check application status, Cancel application
Add a new staff, Delete a staff, Block a registered user	Add member, Delete member, Block member
Maintain the services information	Create event, Edit event, Delete event
Assign request to staff	Assign a member to team
Generate a report about services and staff Reports are: • Statistics about services requested per period • Pending requests • Closed requests per time interval (ex. Closed requested during last month)	generate a report about Event. generate a report about Team Events. generate a report about Team Members.
Browse assigned services by admin	Browse assigned events
Schedule a service	Schedule an event

Activity Diagrams

Apply for event:

Assign member to team:

State Machine

State of the Member Application :

State of the Event:

Class Diagram

<<enum>> Gender

Male
Female

User

- govID: int
- firstName: String
- lastName: String
- gender: Gender
- email: String

+ getUserName(): String
+ getFirstName(): String
+ getLastName(): String
+ getGovID(): int
+ getEmail(): String
+ setUserName(String s): void
+ setFirstName(String s): void
+ setLastName(String s): void
+ setEmail(String s): bool
+ GetGender(): Gender

ApplicationsList

- applicationsList: Map<Event,Member>
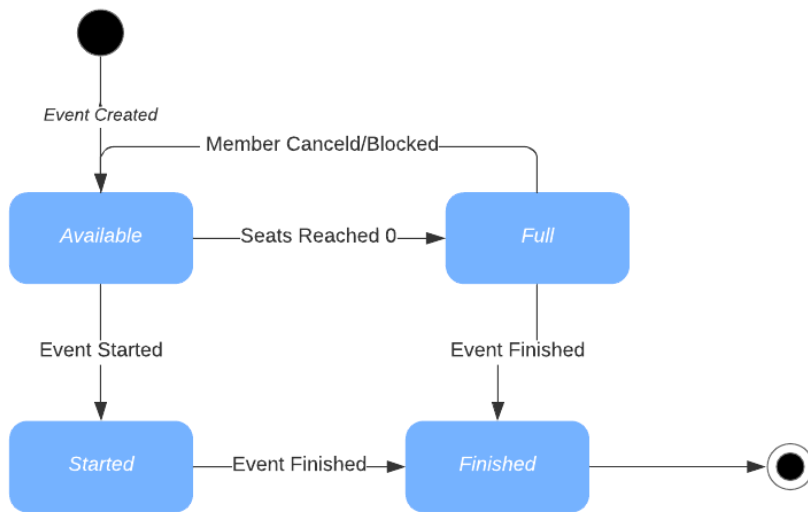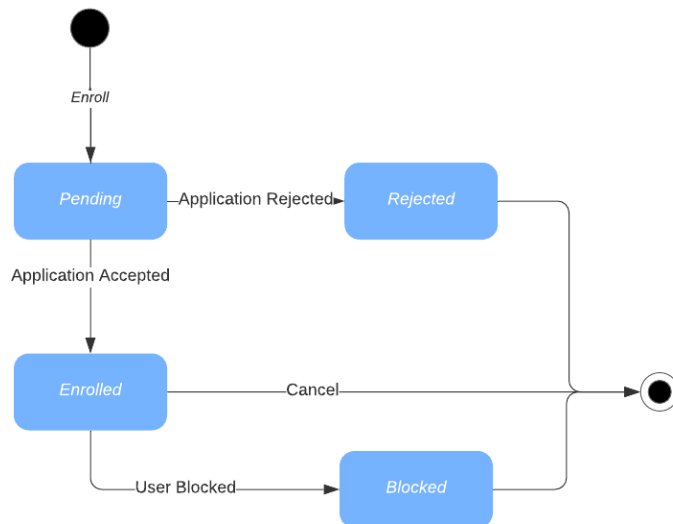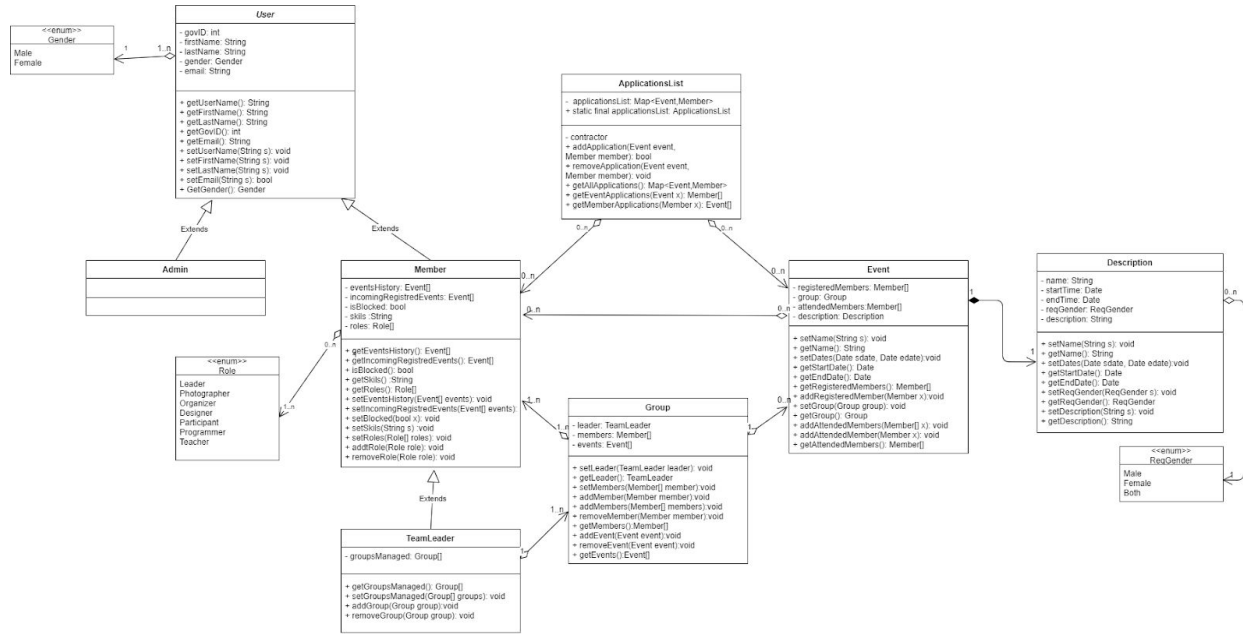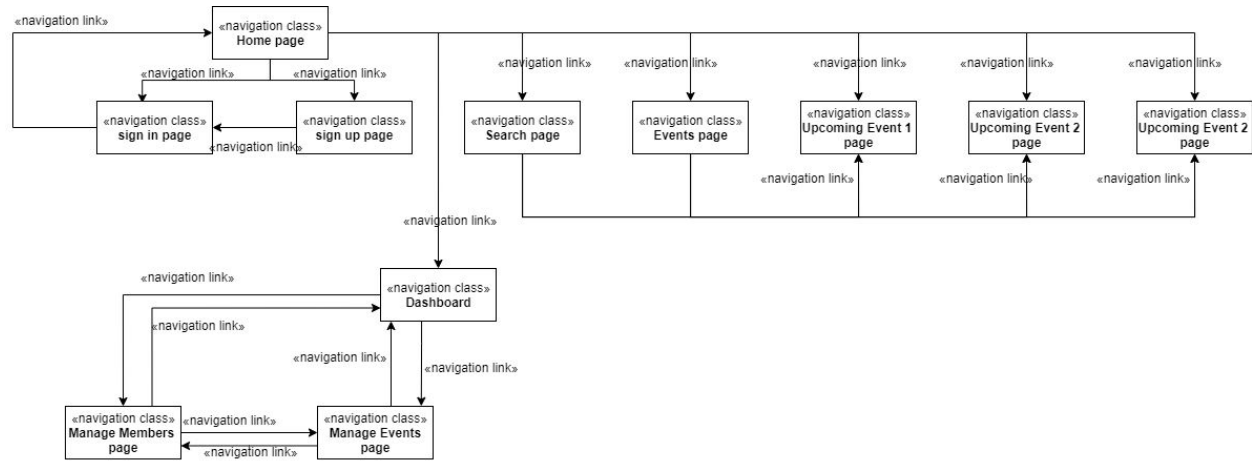+ static final applicationsList: ApplicationsList

- contractor
+ addApplication(Event event, Member member): bool
+ removeApplication(Event event, Member member): void
+ getAllApplications(): Map<Event,Member>
+ getEventApplications(Event x): Member[]
+ getMemberApplications(Member x): Event[]

Admin

Extends

Member

- eventsHistory: Event[]
- incomingRegistredEvents: Event[]
- isBlocked: bool
- skills :String
- roles: Role[]

+ getEventsHistory(): Event[]
+ getIncomingRegistredEvents(): Event[]
+ isBlocked(): bool
+ getSkils(): String
+ getRoles(): Role[]
+ setEventsHistory(Event[] events): void
+ setIncomingRegistredEvents(Event[] events):
+ setBlocked(bool x): void
+ setSkils(String s) :void
+ setRoles(Role[] roles): void
+ addRole(Role role): void
+ removeRole(Role role): void

Extends

<<enum>> Role

Leader
Photographer
Organizer
Designer
Participant
Programmer
Teacher

Event

- registeredMembers: Member[]
- group: Group
- attendedMembers: Member[]
- description: Description

+ setName(String s): void
+ getName(): String
+ setDates(Date sdate, Date edate):void
+ getStartDate(): Date
+ getEndDate(): Date
+ getRegisteredMembers(): Member[]
+ addRegisteredMember(Member x):void
+ setGroup(Group group): void
+ getGroup(): Group
+ addAttendedMembers(Member[] x): void
+ addAttendedMember(Member x): void
+ getAttendedMembers(): Member[]

Group

- leader: TeamLeader
- members: Member[]
- events: Event[]

+ setLeader(TeamLeader leader): void
+ getLeader(): TeamLeader
+ setMembers(Member[] member):void
+ addMember(Member member):void
+ addMembers(Member[] members):void
+ removeMember(Member member):void
+ getMembers():Member[]
+ addEvent(Event event):void
+ removeEvent(Event event):void
+ getEvents():Event[]

TeamLeader

- groupsManaged: Group[]

+ getGroupsManaged(): Group[]
+ setGroupsManaged(Group[] groups): void
+ addGroup(Group group):void
+ removeGroup(Group group): void

Extends

Description

- name: String
- startTime: Date
- endTime: Date
- reqGender: ReqGender
- description: String

+ setName(String s): void
+ getName(): String
+ setDates(Date sdate, Date edate):void
+ getStartDate(): Date
+ getEndDate(): Date
+ setReqGender(ReqGender s): void
+ getReqGender(): ReqGender
+ setDescription(String s): void
+ getDescription(): String

<<enum>> ReqGender

Male
Female
Both

Navigational Diagram

«navigation link»

«navigation class»
Home page

«navigation link»   «navigation link»

«navigation class»
sign in page

«navigation link»

«navigation class»
sign up page

«navigation link»

«navigation link»   «navigation link»   «navigation link»   «navigation link»   «navigation link»

«navigation class»
Search page

«navigation class»
Events page

«navigation class»
Upcoming Event 1 page

«navigation class»
Upcoming Event 2 page

«navigation class»
Upcoming Event 2 page

«navigation link»   «navigation link»   «navigation link»

«navigation link»

«navigation link»

«navigation class»
Dashboard

«navigation link»

«navigation link»

«navigation link»   «navigation link»

«navigation class»
Manage Members page

«navigation link»

«navigation class»
Manage Events page

«navigation link»

Presentation Models

Login page

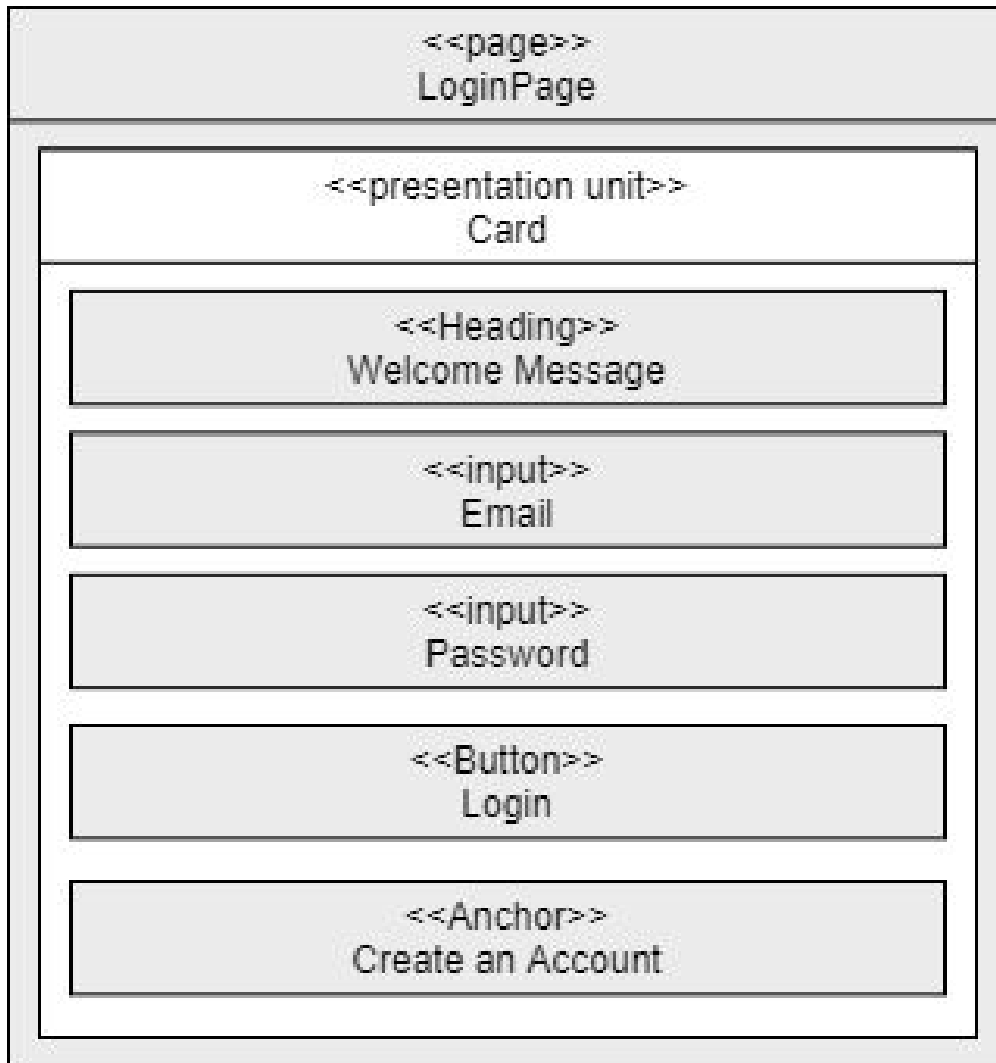# Register page

<<page>>
RegisterPage

<<presentation unit>>
Card

<<Heading>>
Create an Account

<<input>>
First Name

<<input>>
LastName

<<input>>
National ID

<<input>>
Email
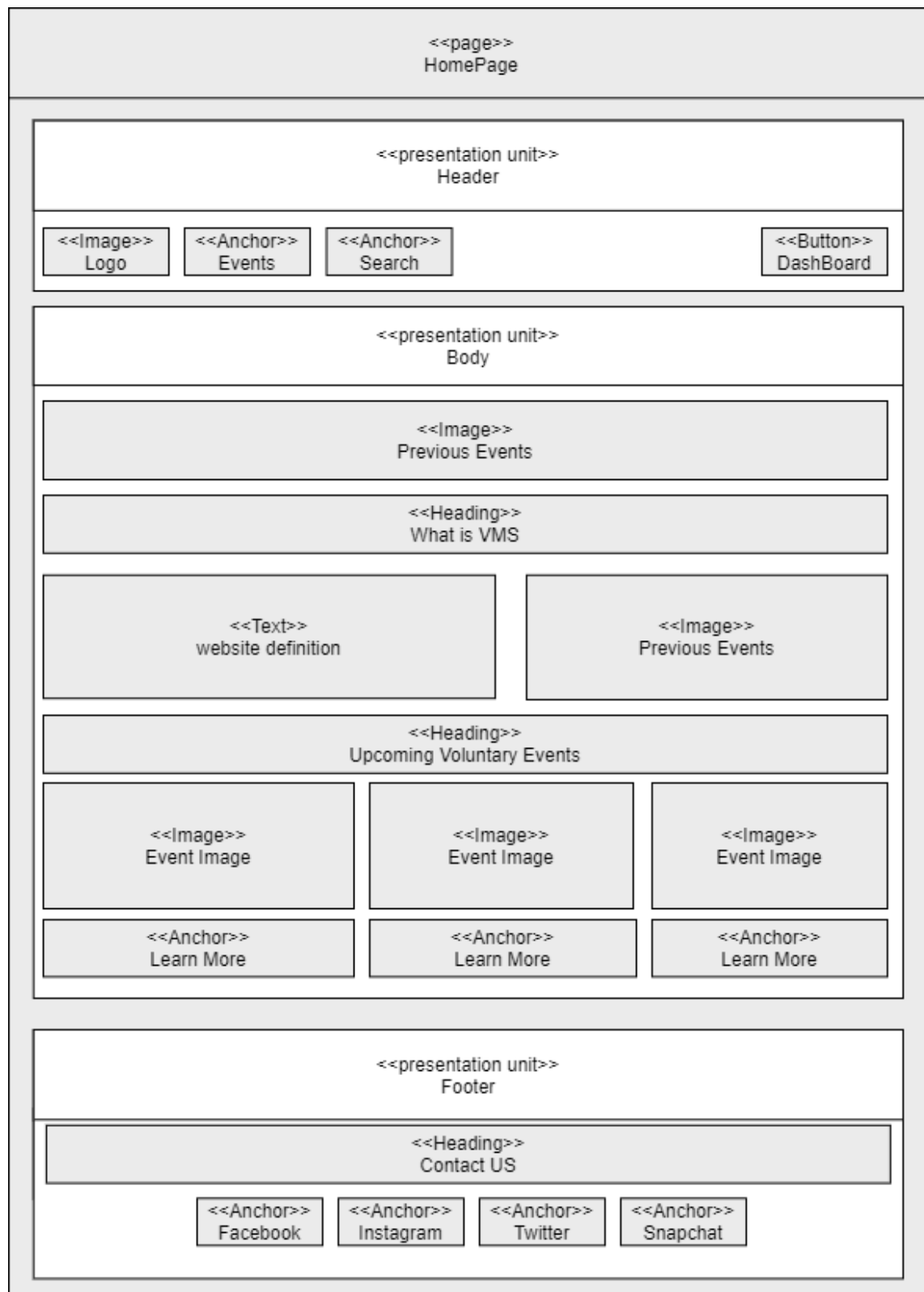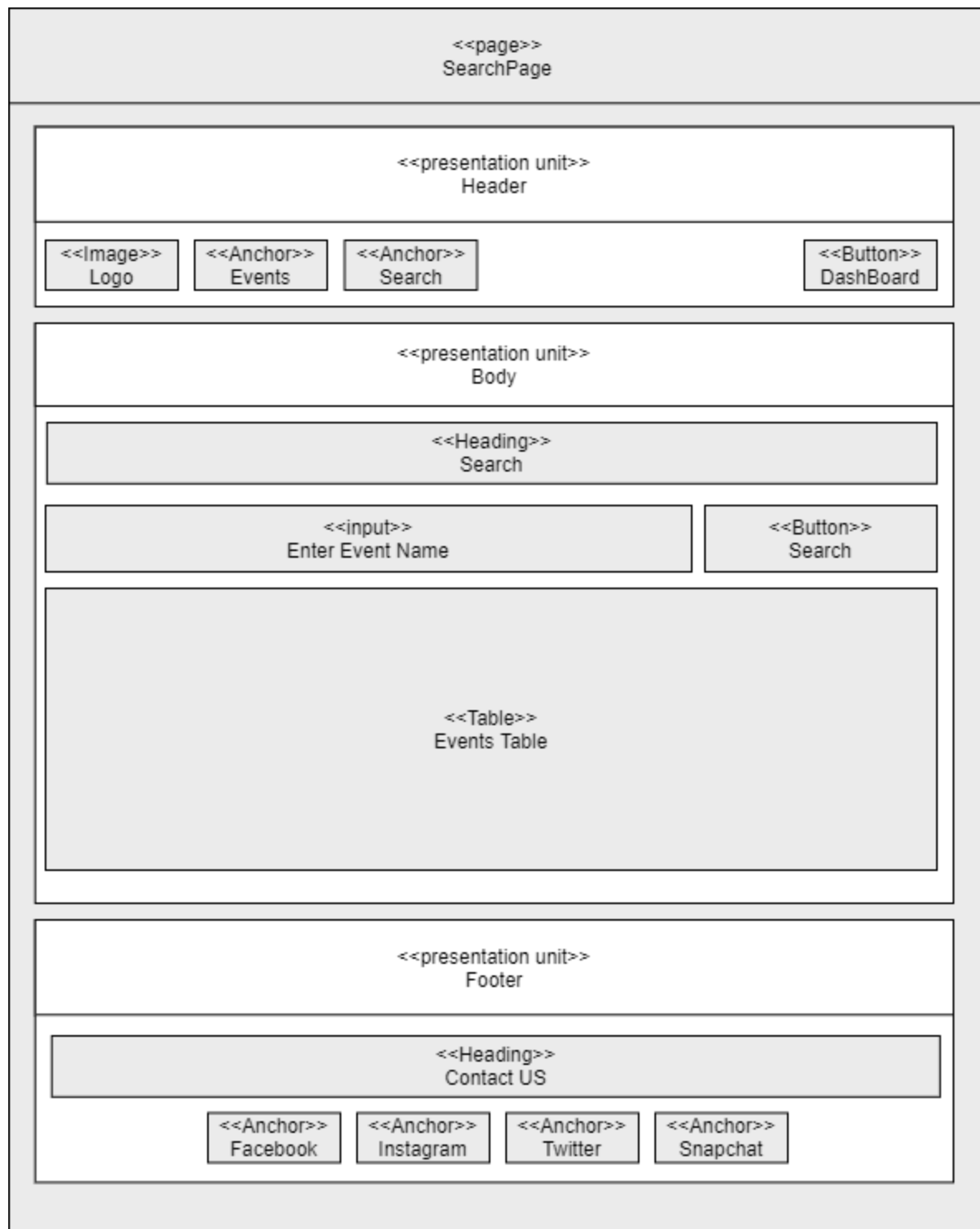
<<input>>
Password

<<input>>
Repeat Password

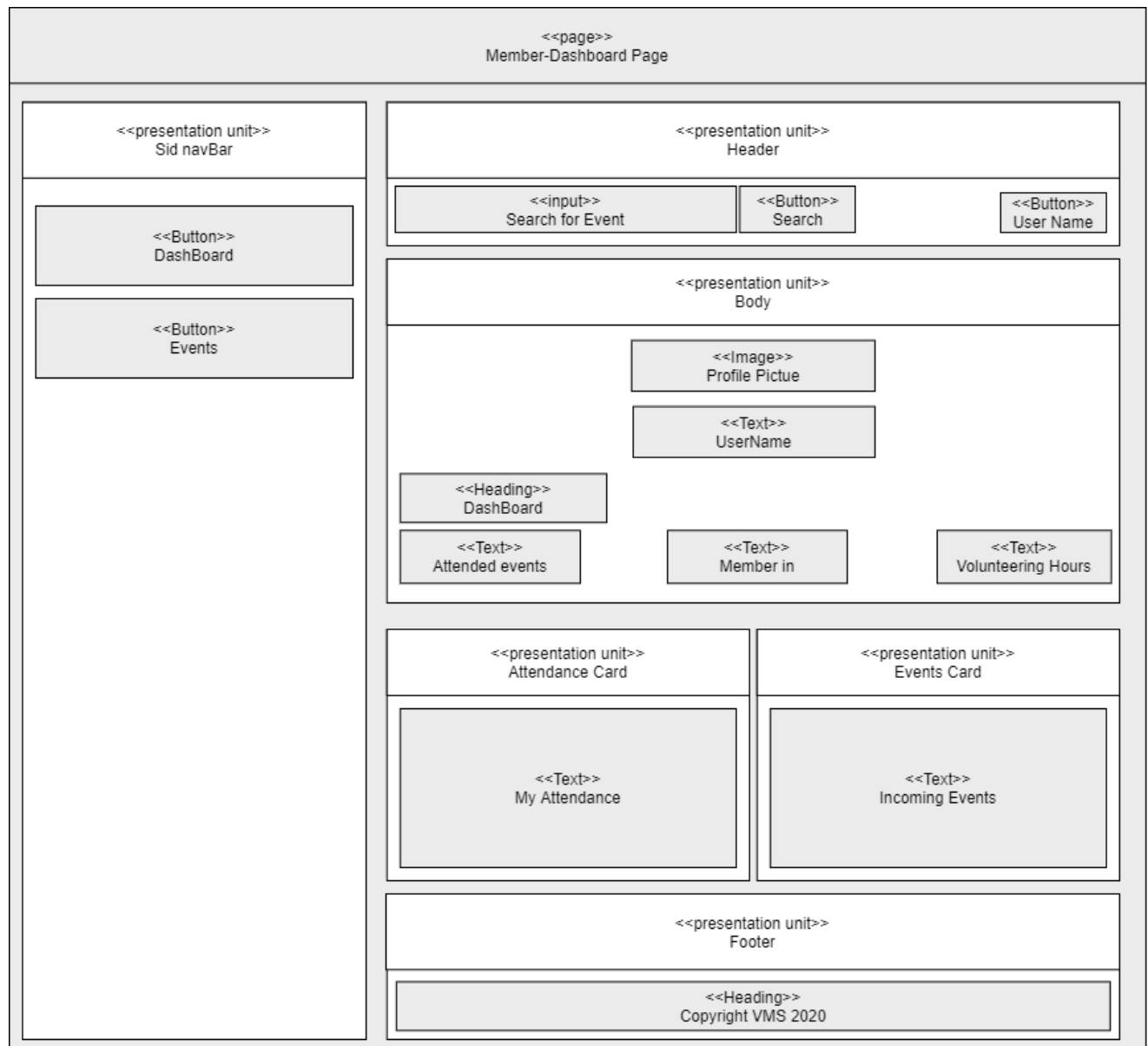<<input>>
Gender

<<input>>
BirthDate

<<Button>>
Register Account

<<Anchor>>
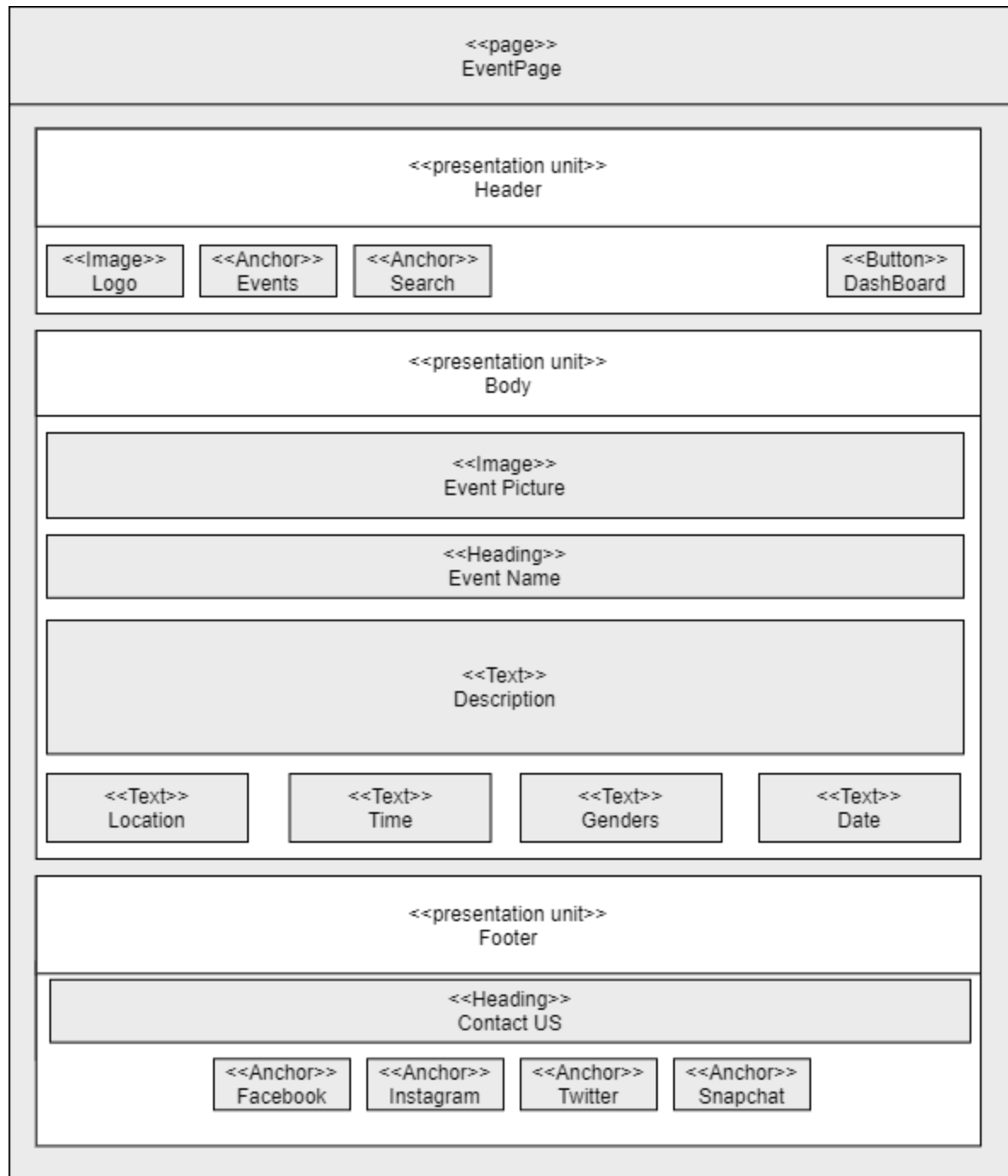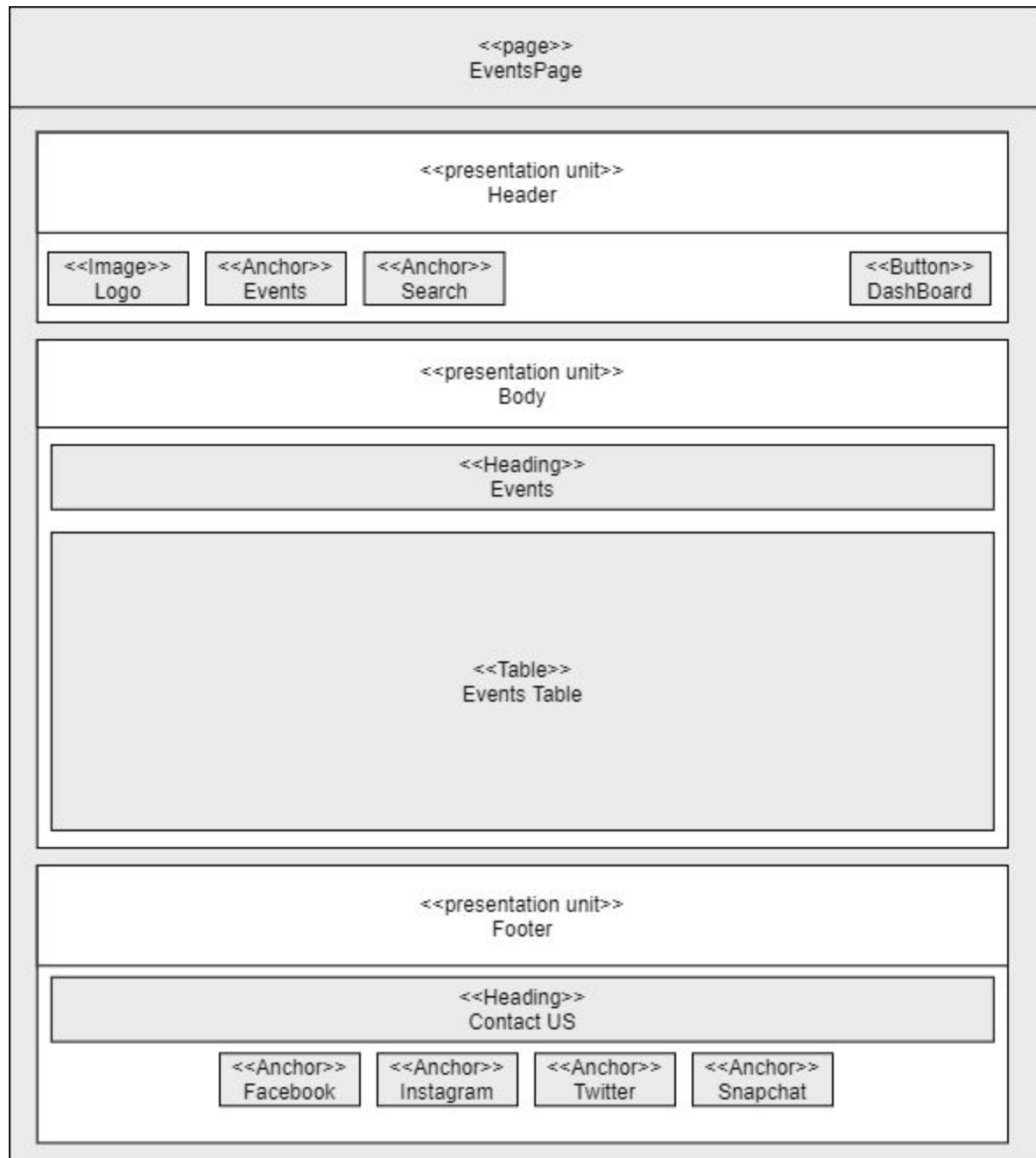Already have an account? Login

Home page:

<<page>>
HomePage

<<presentation unit>>
Header

| <<Image>> Logo | <<Anchor>> Events | <<Anchor>> Search | <<Button>> DashBoard |

<<presentation unit>>
Body

<<Image>>
Previous Events

<<Heading>>
What is VMS

| <<Text>> website definition | <<Image>> Previous Events |

<<Heading>>
Upcoming Voluntary Events

| <<Image>> Event Image | <<Image>> Event Image | <<Image>> Event Image |
| <<Anchor>> Learn More | <<Anchor>> Learn More | <<Anchor>> Learn More |

<<presentation unit>>
Footer

<<Heading>>
Contact US

| <<Anchor>> Facebook | <<Anchor>> Instagram | <<Anchor>> Twitter | <<Anchor>> Snapchat |

Search page

<<page>>
SearchPage

<<presentation unit>>
Header

<<Image>>
Logo

<<Anchor>>
Events

<<Anchor>>
Search

<<Button>>
DashBoard

<<presentation unit>>
Body

<<Heading>>
Search

<<input>>
Enter Event Name

<<Button>>
Search
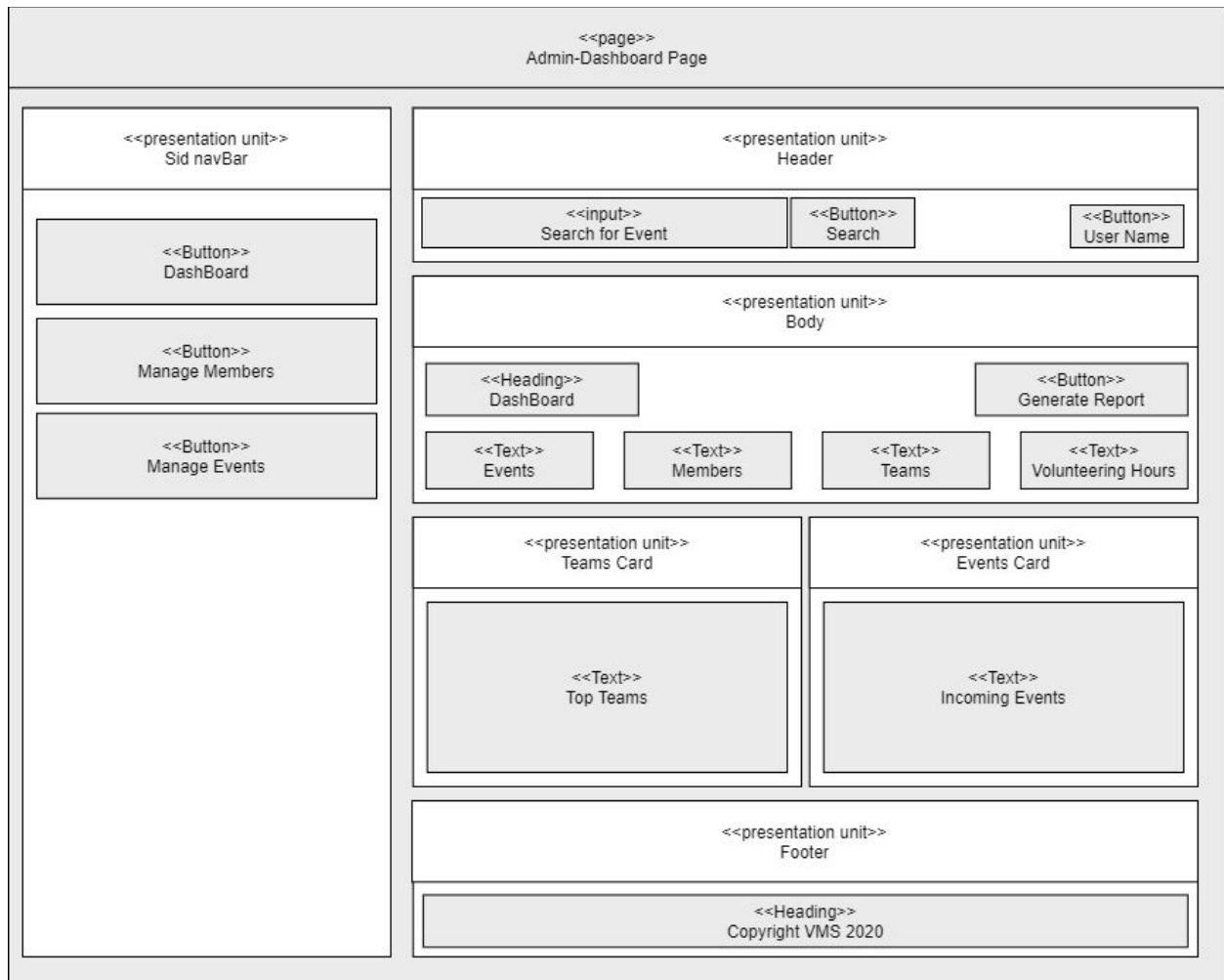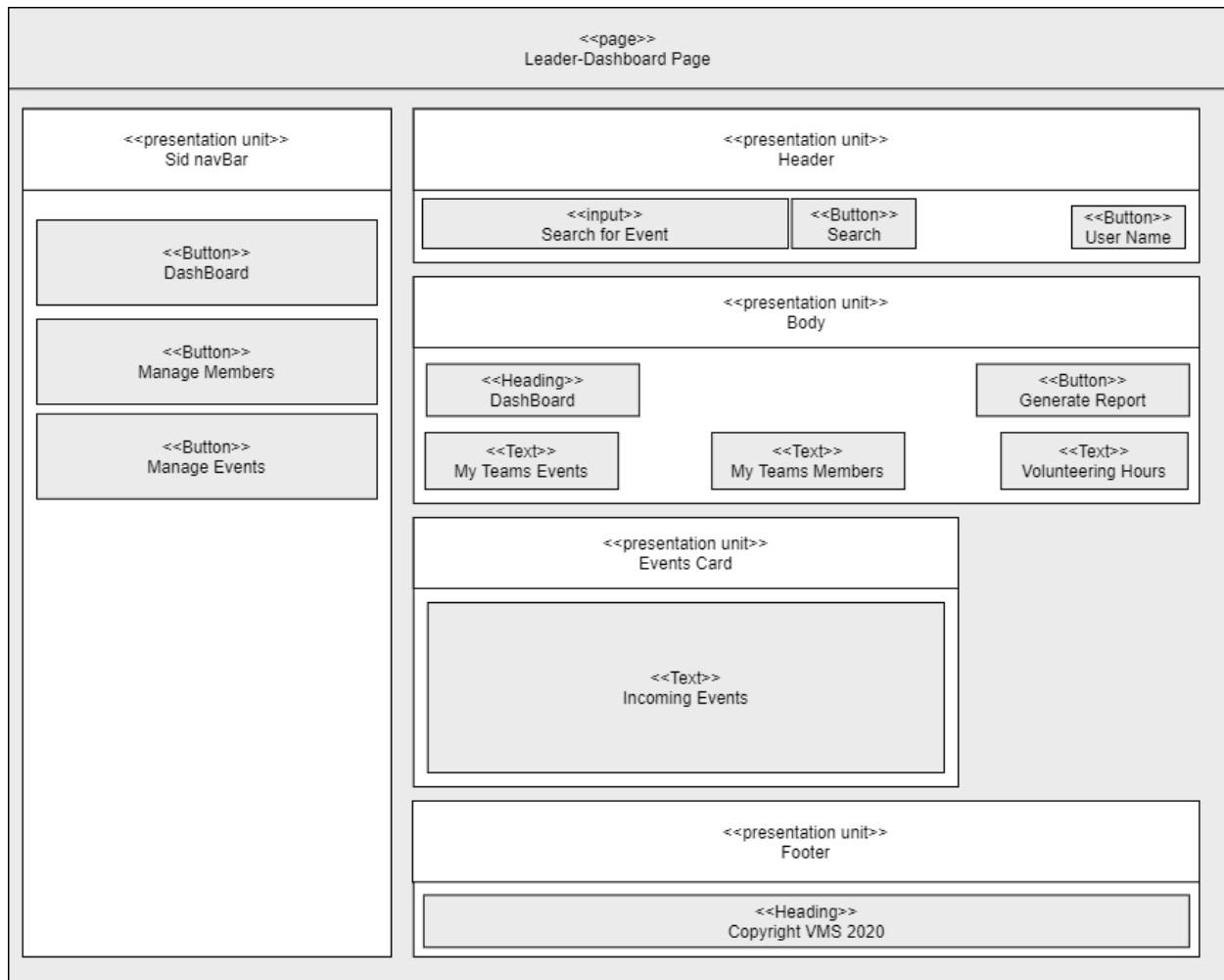
<<Table>>
Events Table

<<presentation unit>>
Footer

<<Heading>>
Contact US

<<Anchor>>
Facebook

<<Anchor>>
Instagram

<<Anchor>>
Twitter

<<Anchor>>
Snapchat

Member Dashboard page

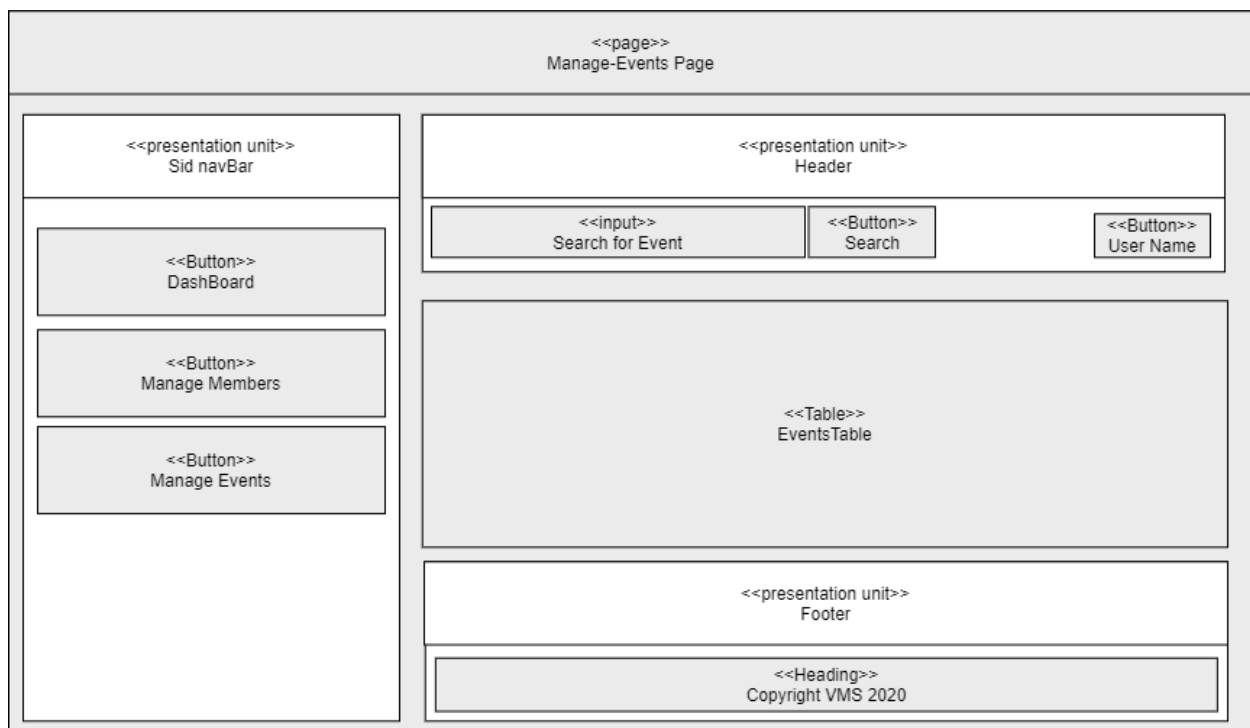# Event details page

Events page

Admin Dashboard page

Leader dashboard page

<<page>>
Leader-Dashboard Page

<<presentation unit>>
Sid navBar

<<Button>>
DashBoard

<<Button>>
Manage Members

<<Button>>
Manage Events

<<presentation unit>>
Header

<<input>>
Search for Event

<<Button>>
Search

<<Button>>
User Name

<<presentation unit>>
Body

<<Heading>>
DashBoard

<<Button>>
Generate Report

<<Text>>
My Teams Events

<<Text>>
My Teams Members

<<Text>>
Volunteering Hours

<<presentation unit>>
Events Card

<<Text>>
Incoming Events

<<presentation unit>>
Footer

<<Heading>>
Copyright VMS 2020

Manage Members page


```
                        <<page>>
                    Manage-Members Page

  <<presentation unit>>        <<presentation unit>>
     Sid navBar                     Header

   <<Button>>          <<input>>      <<Button>>    <<Button>>
    DashBoard       Search for Event    Search      User Name

   <<Button>>
  Manage Members              <<Table>>
                            Members Table
   <<Button>>
  Manage Events

                        <<presentation unit>>
                             Footer

                          <<Heading>>
                       Copyright VMS 2020
```
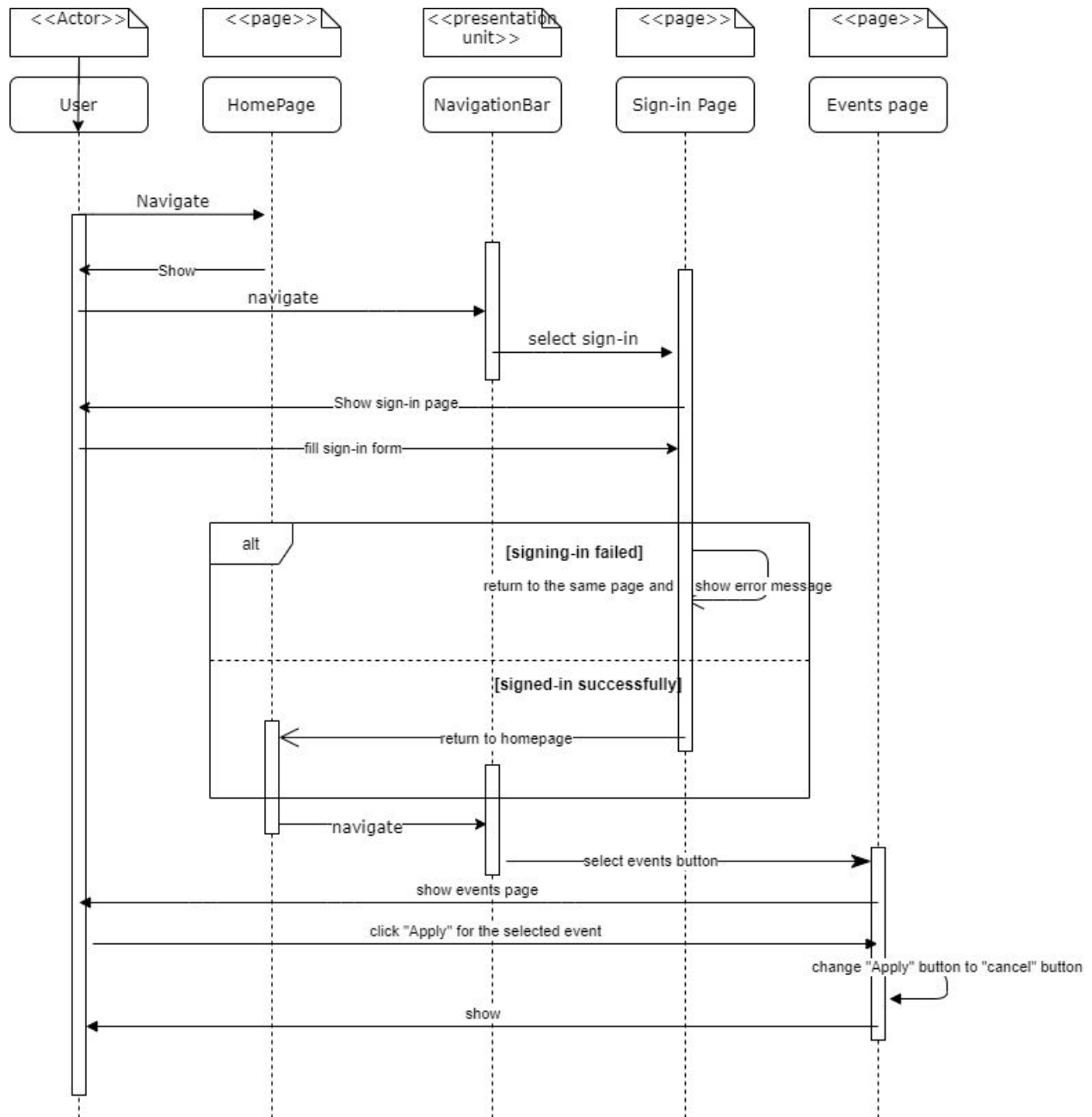
Manage Events page


```
                        <<page>>
                    Manage-Events Page

  <<presentation unit>>        <<presentation unit>>
     Sid navBar                     Header

   <<Button>>          <<input>>      <<Button>>    <<Button>>
    DashBoard       Search for Event    Search      User Name

   <<Button>>
  Manage Members              <<Table>>
                             EventsTable
   <<Button>>
  Manage Events

                        <<presentation unit>>
                             Footer

                          <<Heading>>
                       Copyright VMS 2020
```

Edit Profile page

Sequence Model

Conclusion

WebSite Link

http://ialamer.co.technology

Distribution of work

Mojtaba Alamer:

First Phase:

- Derived The Requirements.
- Derived The Use Cases from the Requirements
- Draw The Use Case Diagram
- Derived the relation between required functionality and the use cases
- Draw Activity Diagram
- Draw state machine Diagram
- Draw the relations type between Classes in Class Diagram
- Designed login page
- Designed Register page
- Designed Events page
- Designed Admin Dashboard page
- Designed Admin Manage Members page
- Designed Admin Manage Events page
- Designed Leader Dashboard page
- Designed Leader Manage Members page
- Designed Leader Manage Events page
- Linking between pages for different actors
- Provided screenshot for each page in the Presentation Models

Second Phase:

- Implemented GenerateReport page
- Implemented Restriction of unauthorised access to some pages
- Implemented DashBoard page for different type of users
- Implemented edit profile page
- Implemented manage Events page and its subsections(delete event,add,etc.)
- Implemented manage Members page and its subsections(delete member,add,etc.)
- Designed Presentational mode
- Designed the DataBase

- Implemented Take Attendance page

Haitham Alsaeed:

First Phase:

- Lead the team
- Derived The Requirements.
- Derived The Use Cases from the Requirements
- Draw Class Diagram
- Draw navigational Diagram
- Designed Home page
- Designed Search page
- Search result page
- Designed Member Dashboard page
- Designed Event details page
- Select the developing environment
- Choose a color style
- Search for similar websites to benefit from the design
- Add alt attribute to imgs
- Choose images used in the design webpages

Second Phase:

- Implemented the DataBase
- Implemented Registration page and the validation of informations
- Implemented Edit Profile page and the validation of informations
- Implemented Login page
- Designed Sequence Model
- Implemented Sessions
- Implemented Events Page
- Implemented Event Page