**Chapter 3: Design**

**Introduction**

* This is a third phase, which shows how the system looks, runs and perform a certain function that is used by customers. As it includes application, database, user interface, and system interface design.
* Design is one of the step of SDLC. The systematic sequence of steps which helps to describe, several plan/Structure and produce a project is known as design. It helps you to be focus, transparent while developing. Design process are follow to avoid any misses of function. The structured of the project is preplanned and how the project can achieve goals.

Design is important because:

* It clarifies the ways and paths to be taken during the coding and implementation of project Bull’s Club Swimming Pool.
* The client can trust with the decision that we make through the design.
* It makes us easy/efficient when developing.

I used open software **star UML** which is fast, flexible. And also I used **Visual paradigm** to develop ER diagram.And for paper phototypedesigning I have used **Belsamiq Mockup 3.**

**3.1 Structural Modeling**

* Structural model is a view of system that deals with class and object. It is also called static model. It illustrate entities description and connection between entities. It helps to measure stress and strain of the system. Among many structural model, I choose Class diagram and flow chat modeling

**3.1.1 Class Diagram**

* A static view which represent the static view of the system or application is class diagram. It helps to visualize the system and helps to create the executable code of the system. It describes operation of class, its attributes and constraints executed in the system. It is a structural diagram which shows the classes, interface, constraints and associations
* The advantages of creating an class diagram for my project is given below.
* It helps in modelling the static view of an application.
* It helps the programmer in the software coding and implementation.
* It acts as a structural foundation for writing the programming codes.

**Notations used in class diagram:**

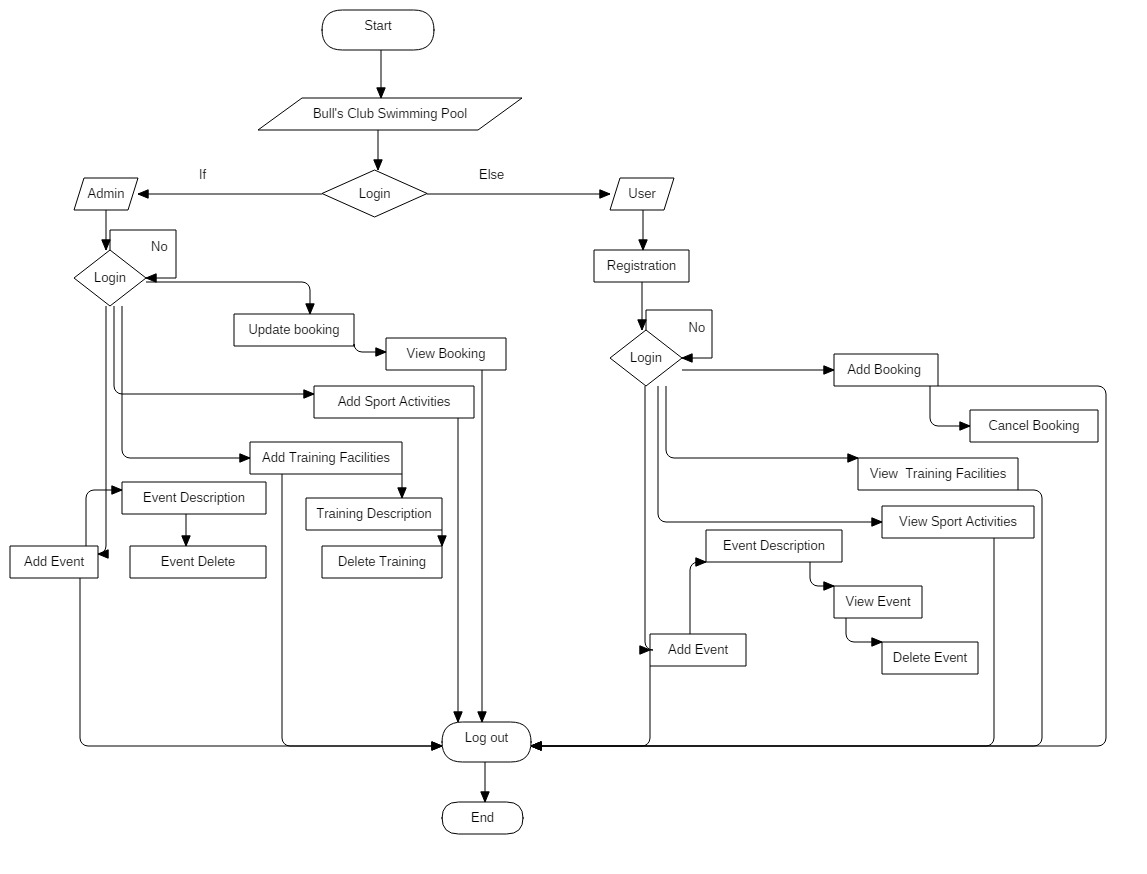
|  |  |  |
| --- | --- | --- |
| **Symbol** | **Notatiom** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**3.1.2 Flowchart modeling**

* A flow chart is a type of diagram representing a process using different symbols containing information about steps or a sequence of events. Each of these symbols is linked with arrows to illustrate the flow direction of the process.
* ***Justification***
* With the help of flowchart, problem can be analysed in more effective way therefore reducing cost and wastage of time.
* It will act as a guide or blueprint during the systems analysis and program development phase.
* It helps in Collecting data about a particular process

***The symbols used in flowchart are shown below in the table:***

|  |  |  |
| --- | --- | --- |
| ***Name*** | ***Symbols*** | ***Function*** |
| Start/End | C:\Users\Techno\Desktop\Screenshot_1.jpg | Used to markup the starting and ending point. |
| Arrows | C:\Users\Techno\Desktop\Screenshot_2.jpg | A arrows is a connector that shows relationship between the representative shapes. |
| Input/Output | C:\Users\Techno\Desktop\Screenshot_3.jpg | Used for the input and output information |
| Process | C:\Users\Techno\Desktop\Screenshot_5.jpg | Used to represent single step. |
| Decision | C:\Users\Techno\Desktop\Screenshot_6.jpg | Used for branching or decision making. |

****

*Fig 3.1.2 Flowchart*

* In above diagram, Admin can easily login to dashboard. Before going into the dashboard, Admin need to correctly type username and password. Once it is correct then admin is directed to the homepage else login is failed. After redirecting to the homepage of Admin, She/He can manipulate event,training, sport acticities and booking , edit and delete any registered user from the database. After logging into the system, user can edit their profile, change password, view sport activities, view training facilities.
  1. **Behaviour Modeling**
* The behavioral diagram is a dynamic diagram and depicts the functionality of the system. When a certain interaction is made, they show what happens or what should happen.It includes a diagram like a use-case, activity, collaboration and sequence diagram.
  + 1. **Activity Diagram**
* An activity diagram visually presents a series of actions or flow of control in a system similar to a [flowchart](https://www.smartdraw.com/flowchart/) or a [data flow diagram](https://www.smartdraw.com/data-flow-diagram/). The dynamic aspect of system is described by activity diagram. Dynamic behavior is also captured by it.

***Justification***

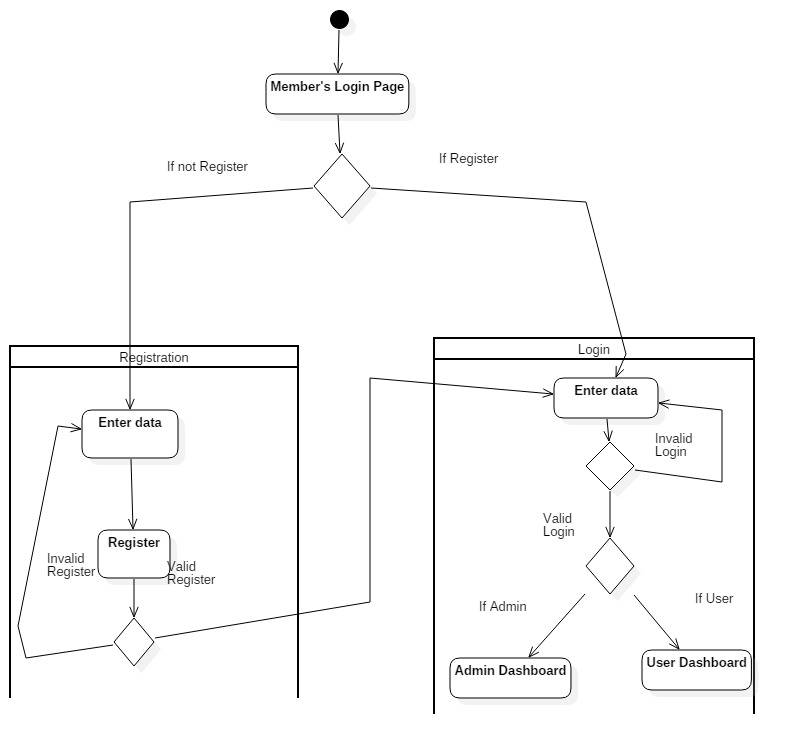
* It records the logic of business in a simple way that makes easy in communicating the complexity of business logic with all the stake holders
* It show the flow of activity that takes place in a system.
* It is simple in understanding for all the end users.

***Notations used in Activity diagram.***

Notation that is used in my project to make an activity diagram are:

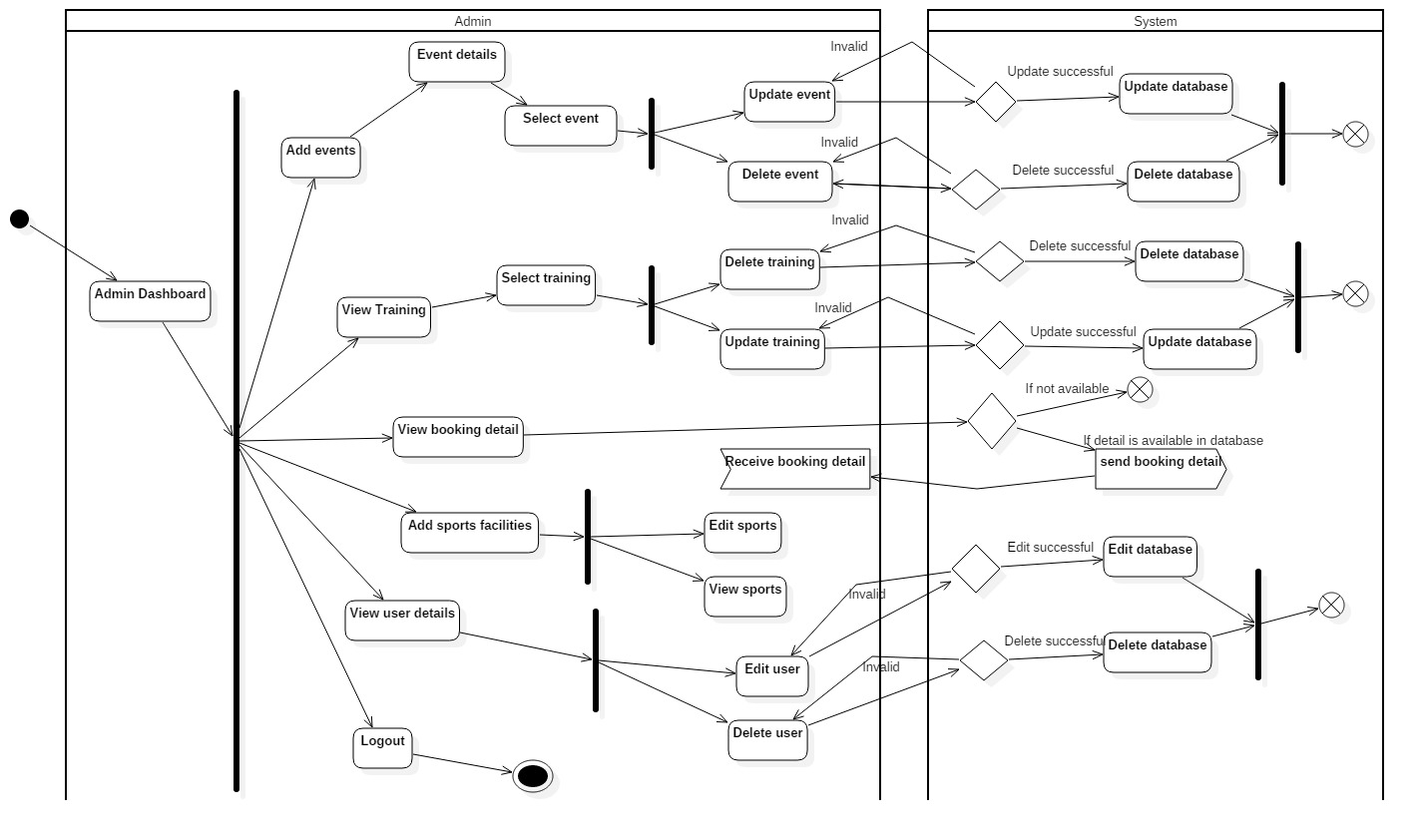
|  |  |  |
| --- | --- | --- |
| Notation | Name | Description |
| C:\Users\Techno\Desktop\Screenshot_7.jpg | Initial | Represent the start of a flow or process of an activity. |
| C:\Users\Techno\Desktop\Screenshot_8.jpg | Control flow | Depict the directional flow and connect two activity. |
|  | Action | Include a short process or activity that occurs within a system. |
| C:\Users\Techno\Desktop\Screenshot_9.jpg | Decision | Represent decision and always merge at least with two conditions. |
|  | Fork | Show that a single activity is split into two or more concurrent activity. |
|  | Join | Represent that two or more concurrent activity are joined to form a single activity. |
| C:\Users\Techno\Desktop\Screenshot_10.jpg | Send | Indicates the action of sending a signal. |
| C:\Users\Techno\Desktop\Screenshot_11.jpg | Receive | Depicts a receipt of a request. |
| C:\Users\Techno\Desktop\Screenshot_12.jpg | Activity Interrupt | It interrupts the flow denoted with a lightning bolt such as a cancellation. |
|  | Accept Time Event | The flow for a time is stopped by time event which is illustrated by an hourglass. |
|  | Final | Indicate the end of a flow of an activity. |
|  | Flow Final | It shows the ending point of a process flow. |

1. **Activity diagram of Login and Registration**

****

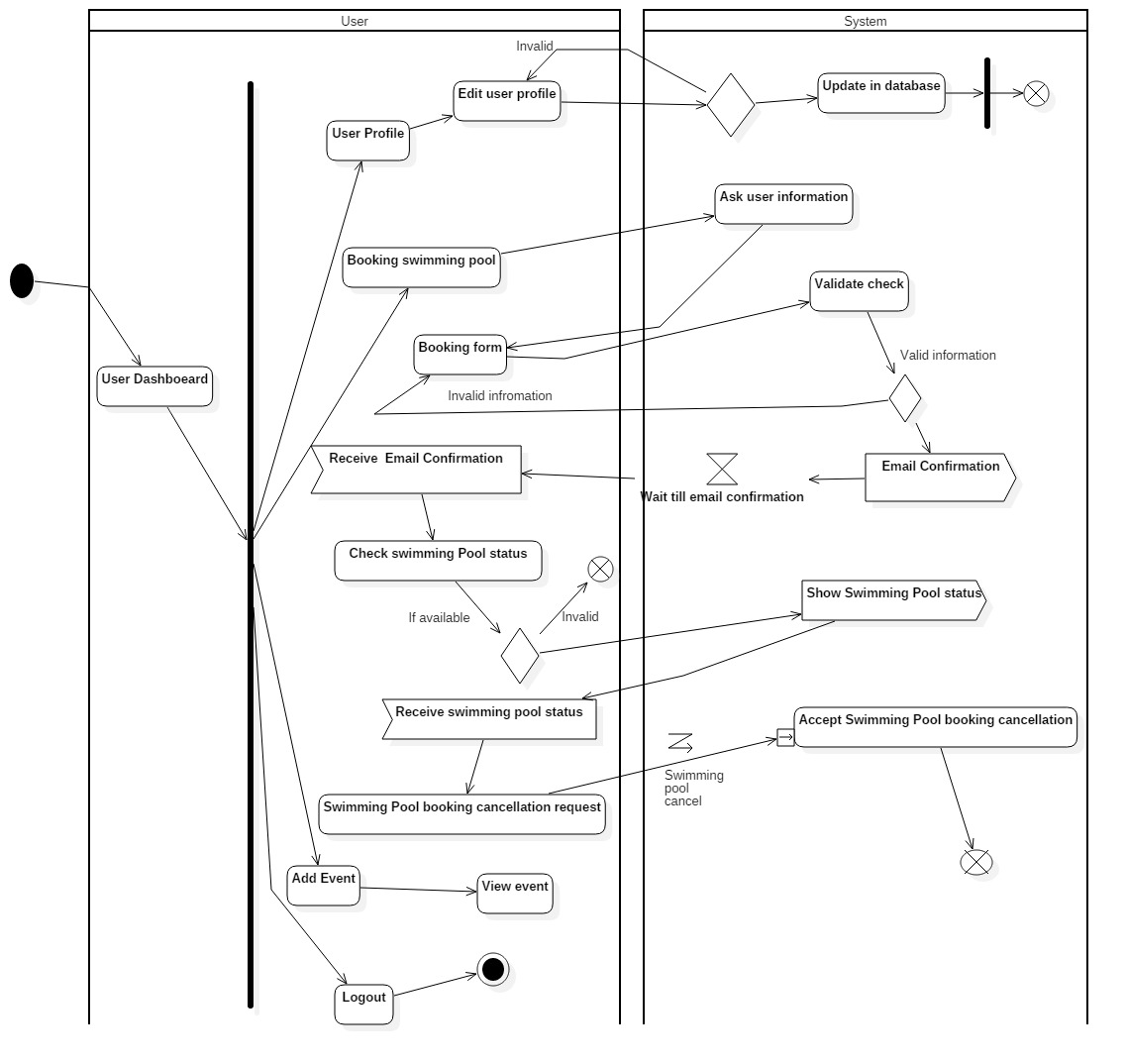
* In the above activity diagram, the activity of the users starts from the login page. After entering the login page a decision notation will separate two different conditions for the registered and unregistered users. If the user is not registered, a registration form will be opened where the details will be provided by users, and if in case there is error in the registration form the user will be taken back to beginning and if the registration is valid the user will be taken to the login page. After the completion of the registration and if the user is registered from the first the user will enter the login page where the successful login will take the user to the admin or user dashboard according to the role they have in the database and the login is unsuccessful the user will be taken back to the beginning of the login page.

1. **Activity diagram of admin dashboard**

****

* When the user is taken into the admin dashboard after the successful login, admin can perform various tasks as illustrated by the fork notation. Briefly, admin can add events and also the events are updated and delete by admin. And also admin can view booking detail, view training, view user details and also add sports facilities .

1. **Activity diagram of user dashboard**

****

* With the user already being logged into the system, the user will be able to edit their profile which will be update in database and aslo they will able to add event and view event. . In the booking term, the user can booking flight but for its confirmation the user is asked for their information, fill reservation form where the system will check whether it is valid or not which will provide email confirmation to the user. They can also check flight status if available and also request the system for the flight cancellation.

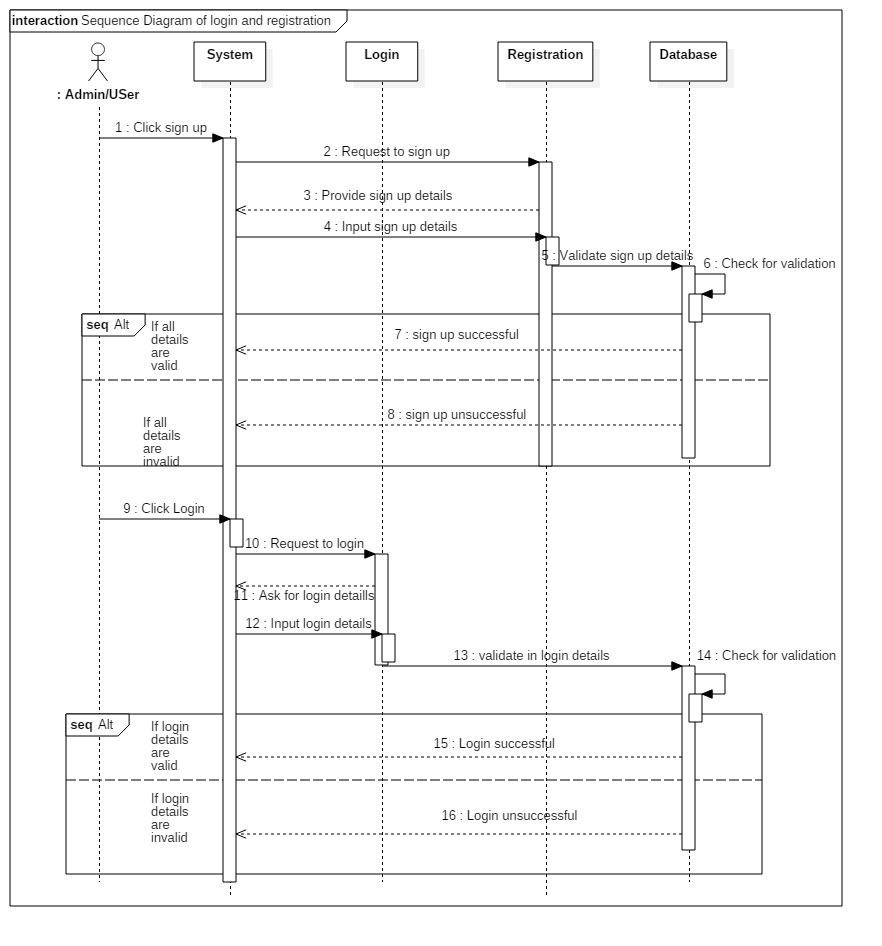
**3.2.2 Sequence diagram**

* An interaction diagram that represent how operations are carried out in a time sequence -- what and when messages are sent. Messages passed among the objects within the use-case in a number of objects are shown.
* The importance of performing sequence diagram in my project are as follows:
* It represent logic interaction between the objects in sequential order.
* It helps in modelling the flow of logic within a system in a visual manner.
* It also represents the messages that are exchanged between the objects needed for carrying out the functionality.

***Notations used in sequence diagram.***

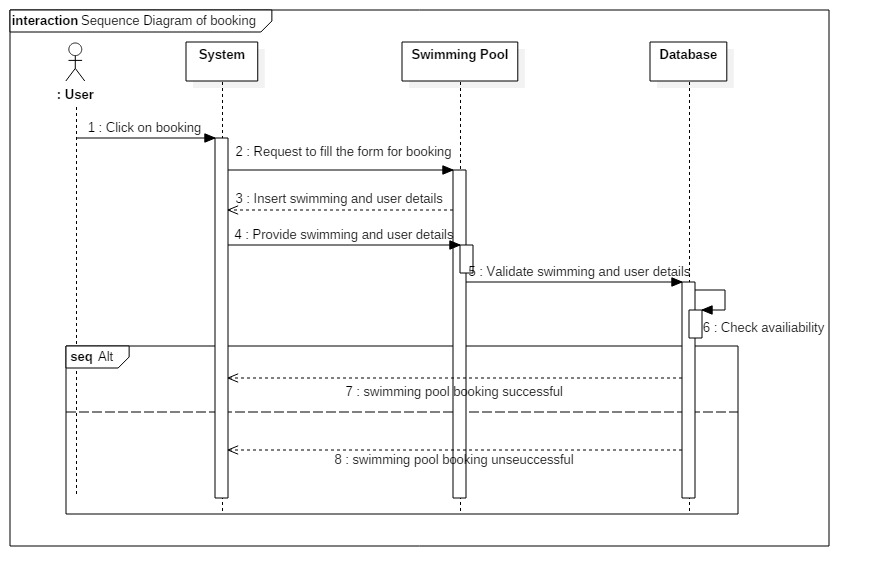
|  |  |  |
| --- | --- | --- |
| Notation | Name | Description |
| C:\Users\Techno\Desktop\Screenshot_1.jpg | Actor | The actor is a user, external entity that interacts with the system. |
| C:\Users\Techno\Desktop\Screenshot_2.jpg | Lifeline | A lifeline represents the participation of an individual in an interaction. |
| C:\Users\Techno\Desktop\Screenshot_3.jpg | Activation | Shows time duration where an operation or task being performed. |
| C:\Users\Techno\Desktop\Screenshot_4.jpg | Send message | Represent the flow of message to objects. |
| C:\Users\Techno\Desktop\7.jpg | Return message | Show return back of message to the callers of the former corresponding message. |
| C:\Users\Techno\Desktop\Screenshot_5.jpg | Recursive message | Represent the call of the message of the same lifeline. |
|  | Loop | Condition in which fragment will occur again and again until the condition is matched. |
|  | alt | A condition where two message sequence occurs. |

1. **Sequence diagram of login and registration**

****

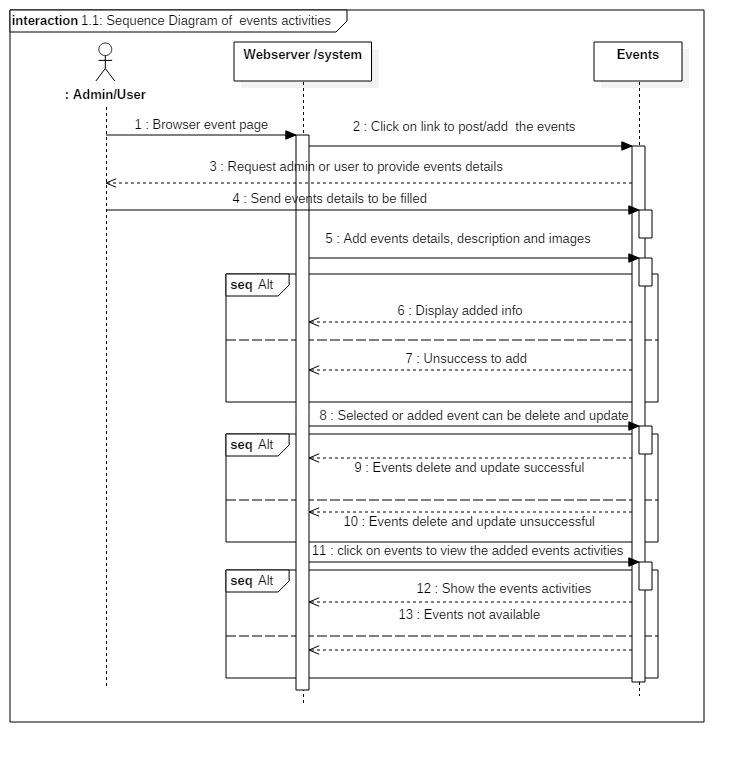
* Here, the user will firstly reach for the signup/login form in which after a click on it the system will provide them a form where the user will be filling their details. After the form is filled and submitted, the system will check for the validation in the database. If the validation is correct, the system will sent a message of successful login or successful signup, otherwise the system will sent a message of unsuccessful login or unsuccessful signup if the validation is incorrect.

1. **Sequence diagram of booking**

****

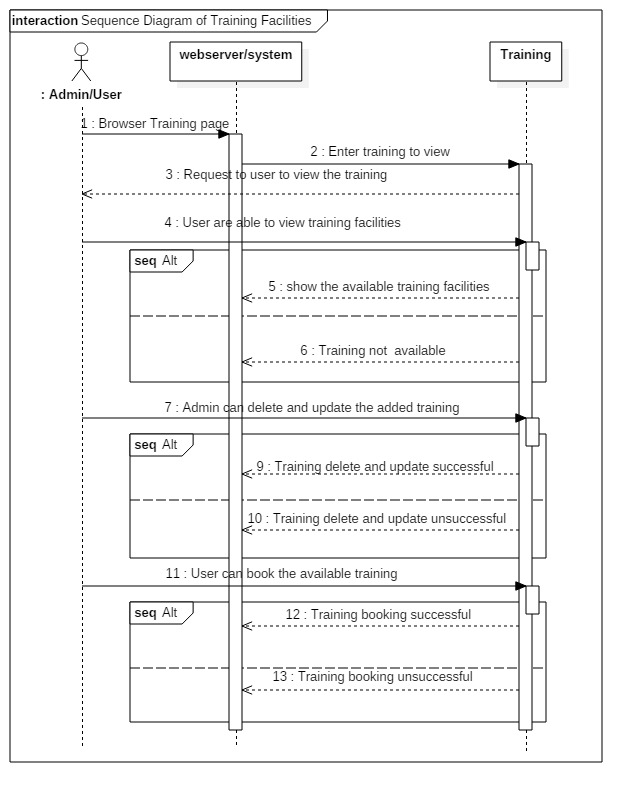
* In the booking term, The user action will be requested by the system to the swimming pool and then to the database. All the swimming pool will be shown if there is any fswimming pool available. When booking the swimming pool system will request to the swimming pool where the user will be asked to input the swimming pool details and the provided swimming pool details will be sent in the database for the validation. If the validation is correct a message will be sent as fswimming pool booking successful else fswimming pool booking unsuccessful.

1. **Sequence diagram of events activities**

****

* Above sequence diagram is about events representing the flow of a sequence of admin/user activities The actions or functions that the admin/user performed will first to add the events by the system to particular page of events After admin/user open the homepage admin and user can view the events activities and aslo the admin and user can do delete and update operation for event.

1. **Sequence diagram of training activities**

****

* Above sequence diagram is about training representing the flow of a sequence of admin/user activities The actions or functions that the admin/user performed will first to view the training by the system to particular page of training. After admin/user open the homepage admin and user can view the training activities and the admin can do delete and update operation for training.and aslo user can book the available tarining from training page.

**3.3 Database modeling**

* A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed. In my project database modeling helps me in designing blueprint of the database that will make me easier to develop a system.

**3.3.1 Data dictionary**

* A data dictionary contains metadata i.e data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. . Data dictionary also used to manipulate the database, to control access and shows the relationship between its different components.
* Purpose of using data dictionary are described below:
* It helps in avoiding data variability throughout a project.
* Makes easy to analyze data.
* The use of data standard is implemented.

***User Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| User\_id | int(10) | Yes | No | No |
| First name | varchar(50) | No | No | No |
| Last name | varchar(50) | No | No | No |
| Email | varchar(50) | No | No | No |
| Contact no | bigint(50) | No | No | No |
| Address | varchar(50) | No | No | No |
| Gender | enum | No | No | No |
| Password | varchar(25) | No | No | No |

***Admin Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Admin\_id | int(10) | Yes | No | No |
| username | varchar(100) | No | No | No |
| password | varchar(50) | No | No | No |
| Email | Varchar(40) | No | No | No |

***Events Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Events\_id | int(10) | Yes | No | No |
| User\_id | int(10) | No | Yes | No |
| Admin\_id | Int(10) | No | Yes | No |
| Events name | varchar(150) | No | No | No |
| Events time | time | No | No | No |
| Events date | date | No | No | No |
| Events description | varchar(200) | No | No | No |
| Events title | varchar(20) | No | No | No |
| Events venue | varchar(20) | No | No | No |

***Booking Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Booking\_id | int(10) | Yes | No | No |
| User\_id | Int(10) | No | Yes | No |
| Training\_id | int(10) | No | Yes | No |
| Booking contact person | varchar(100) | No | No | No |
| Booking time | time | No | No | No |
| Booking date | date | No | No | No |
| Contact no | bigint(50) | No | No | No |
| Contact email | varchar(50) | No | No | No |

***Training Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Training\_id | int(10) | Yes | No | No |
| User\_id | int(10) | No | Yes | No |
| Admin\_id | int(10) | No | Yes | No |
| Training name | varchar(100) | No | No | No |
| Training description | varchar(200) | No | No | No |
| Training time | time | No | No | No |
| Training date | date | No | No | No |

***Sports Table***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Sports\_id | int(10) | Yes | No | No |
| User\_id | Int(10) | No | Yes | No |
| Sport name | varchar(100) | No | No | No |
| Sport description | varchar(250) | No | No | No |

**3.3.2 ER diagram**

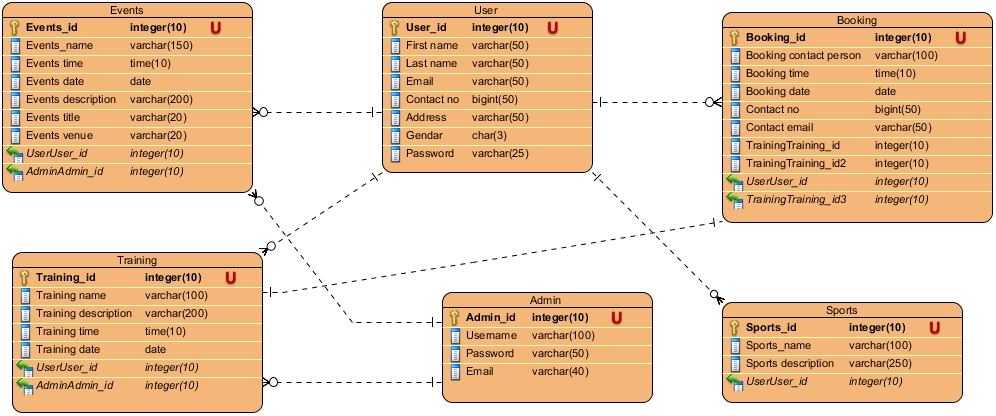
* An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

* Entities
* Relationships
* Attributes

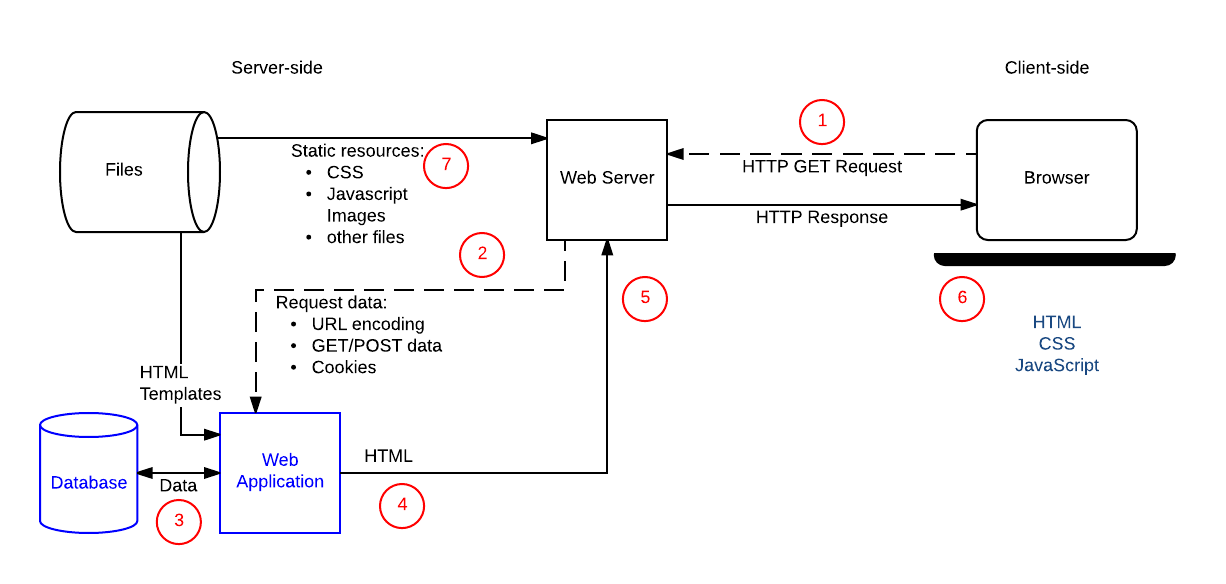
Purpose of using ER diagrams are discussed below:

* Almost anyone can understand it as it is visually presented.
* Changes to database design could be maintained.
* Proper documentation of database design.
* It helps in communicating the logical structure of the database to the users.

****

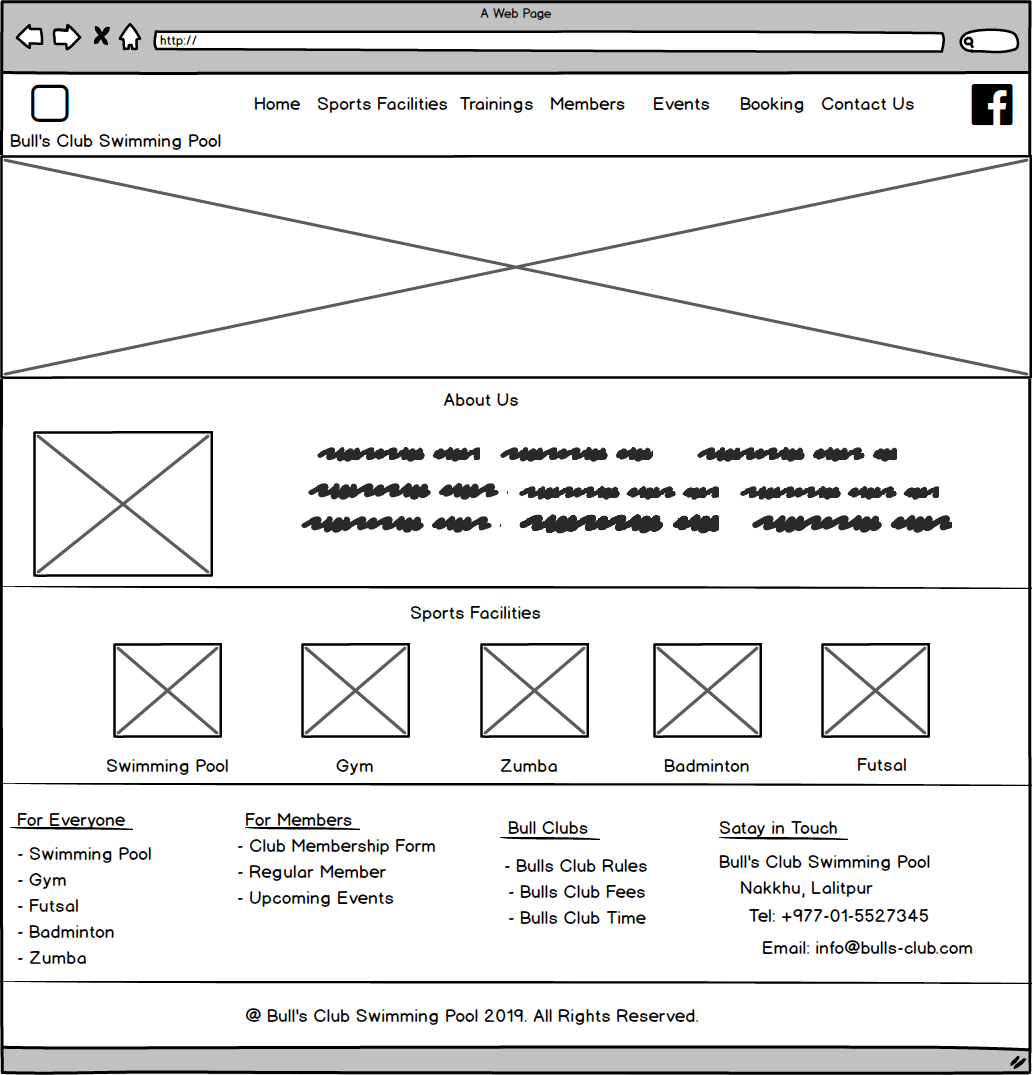
*Fig 3.3.2 ER diagram*

* Above represent ER diagram is based upon the data dictionary prepared on the stage before this design. Here the entities are shown along with the relationship that they have with each other.
  1. **Architectural Design**
* The model used to represent the structure of the overall network used by the system is known as architectural model. For my project I am using the client server architecture. In this process of architecture, server provides and manage the request and services that are requested by client. One or more client can connect to same server. All services are done in network. Client are who runs application .e.g PCs. Client depends on server for files, etc. This architecture allows one or more clients systems to a centrally located server. The server holds the responsibility of controlling the services given to its clients. Client server architecture,architecture of a PC organize in which numerous customers (remote processors) demand and get administration from an incorporated server (have PC). Customer PCs give an interface to enable a PC client to demand administrations of the server and to show the outcomes the server returns. Servers trust that solicitations will touch base from customers and after that react to them. I have prefer to choose client server architecture for my project because of this following reason:
* It helps to maintain good quality of security system.
* It will be easy to repair, replace, relocate and upgrade a client server.
* It shared various resources amongst different platform.
* It improved data sharing process.
* All data are in a centralized location, so recovering and backing up data is easy.

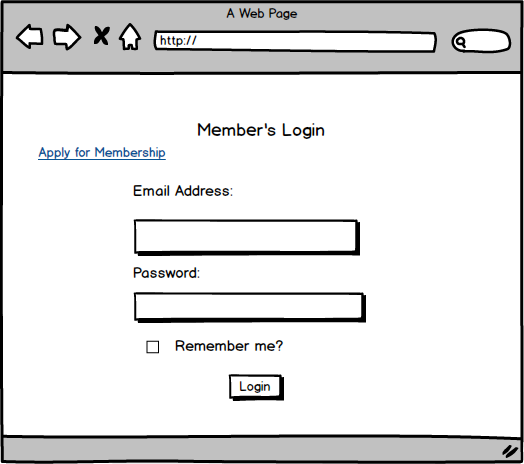


***Fig 3.4.1 Client-Server Architecture***

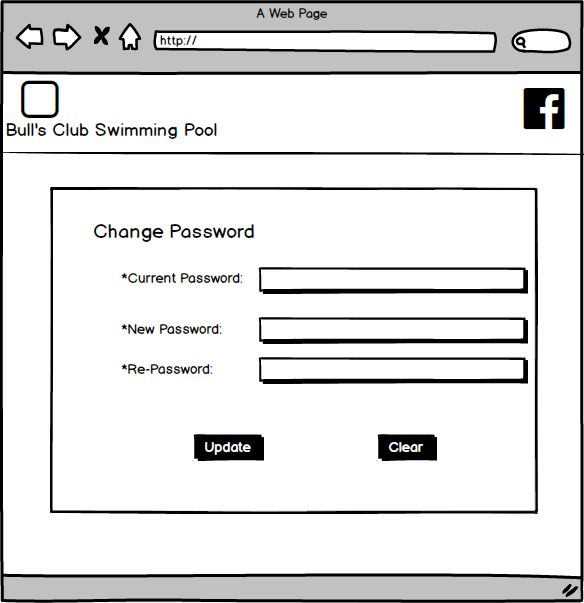
* 1. **Protyping Design**
* Paper prototyping is a technique that allows you to create and test user interfaces quickly and cheaply. It's easier to change a prototype than the final design. It is designed to test and try new design by analyst and users. It gives the overview of system before it is implemented.
* Purpose of using protyping are discussed below:
* Facilitates user for more efficient and effective description.
* It helps you to test and refine the overall functionality of your design.
* Facilitates user for more efficient and effective description.

****

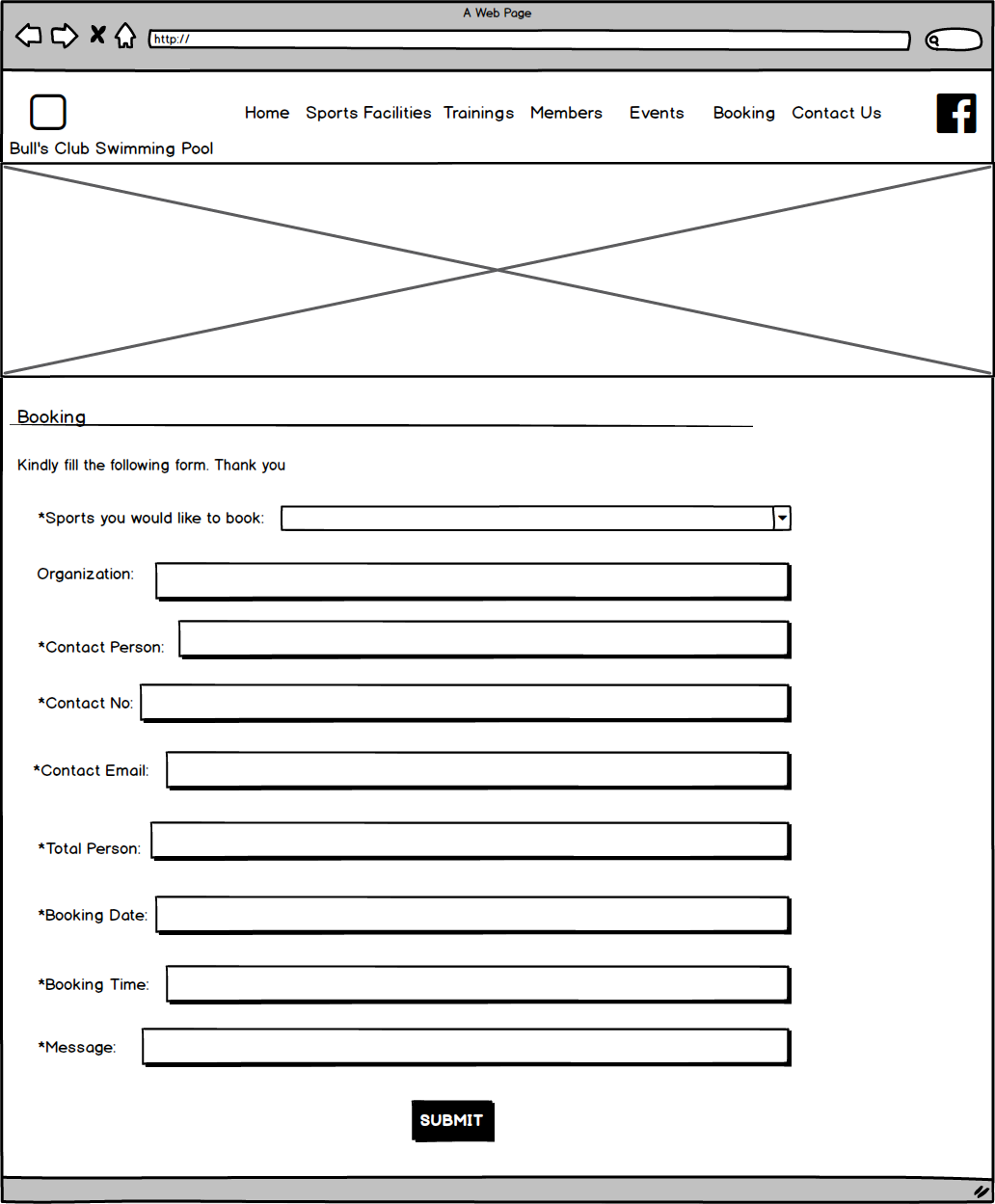
*Fig 1 : Homepage*

****

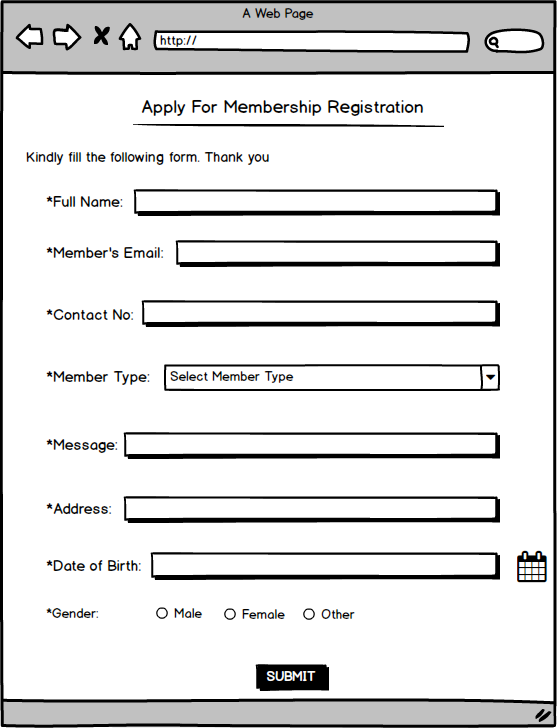
*Fig 2 : Membership Login*

****

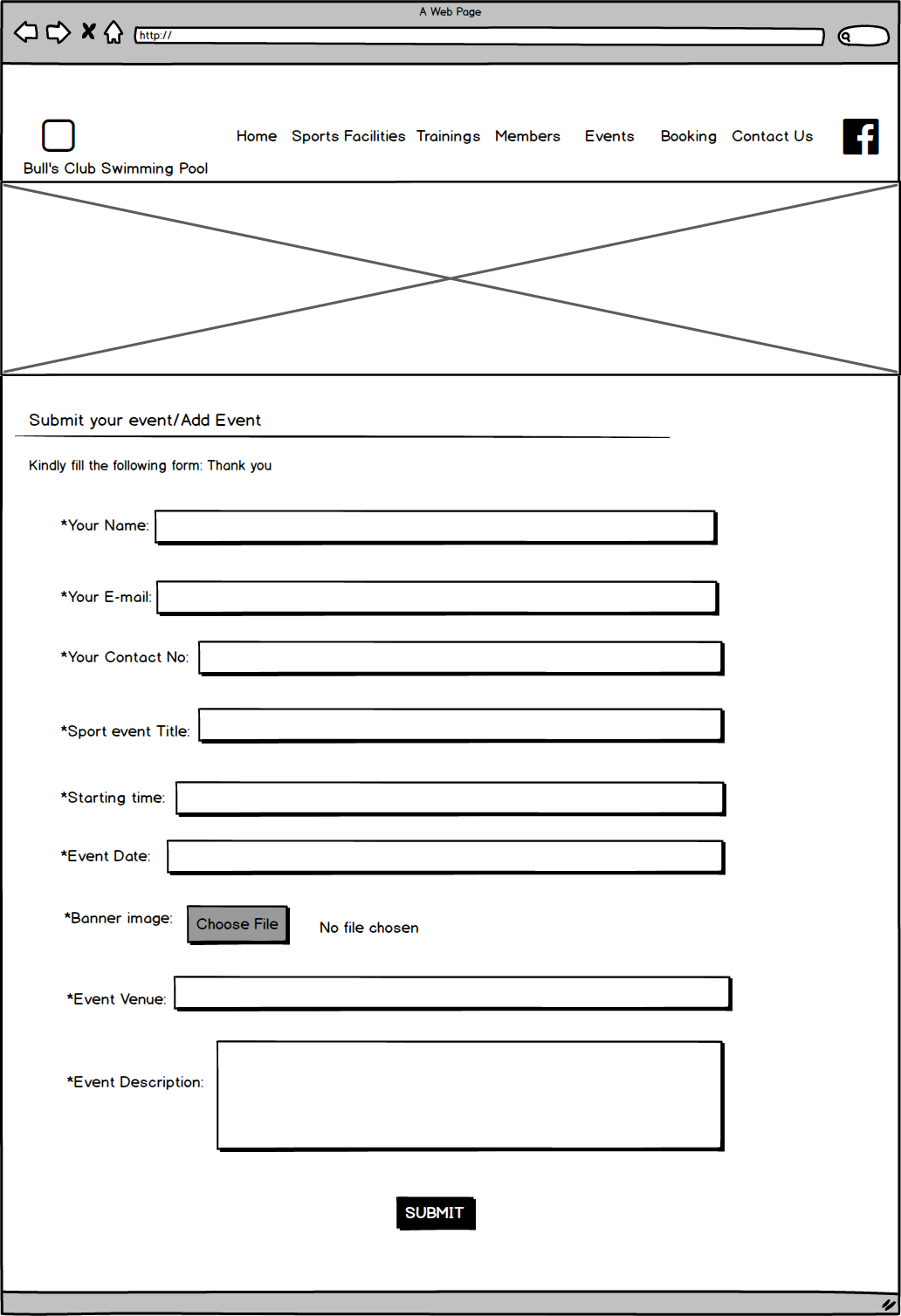
*Fig 3 : Change Password*

****

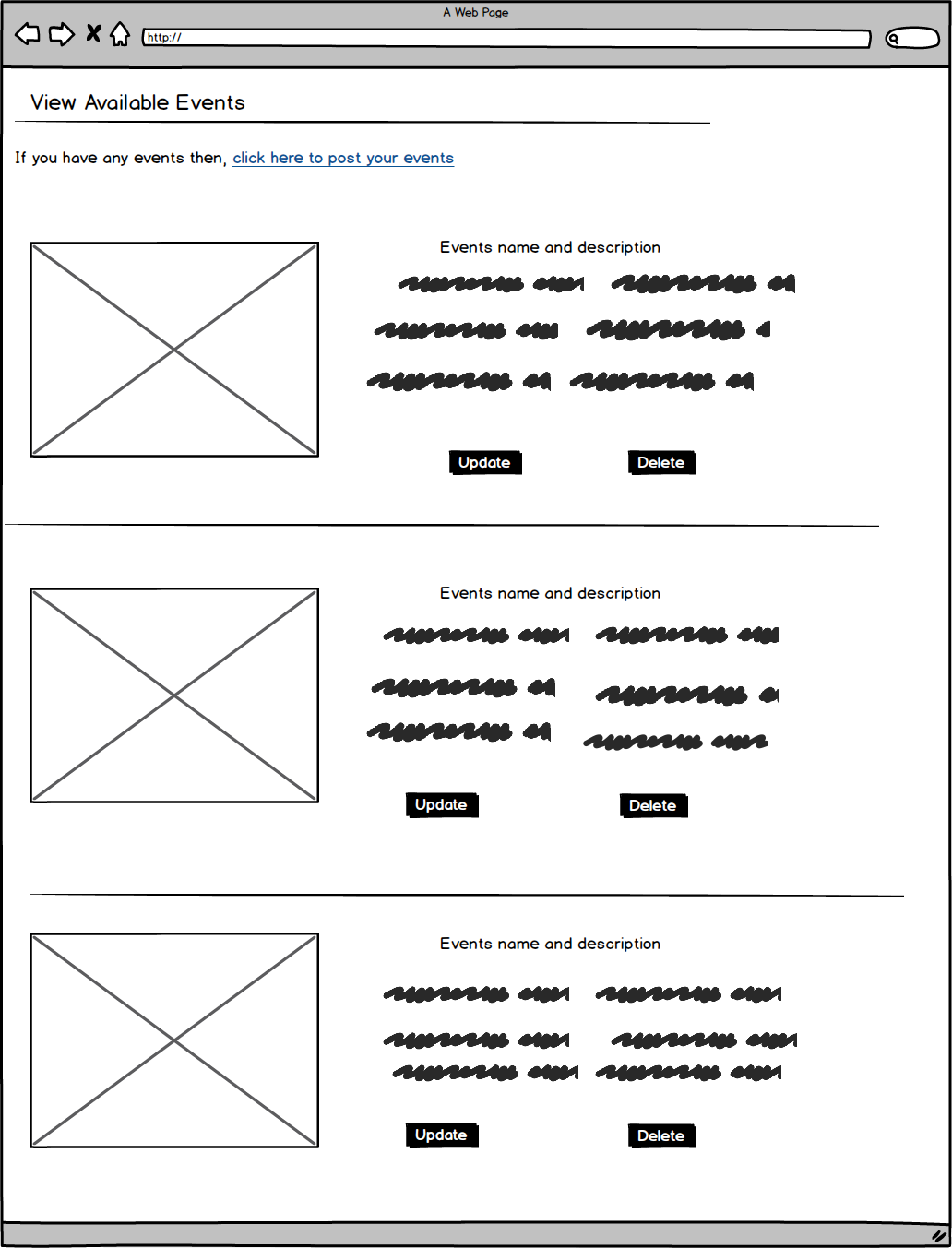
*Fig 4 : Booking form*

****

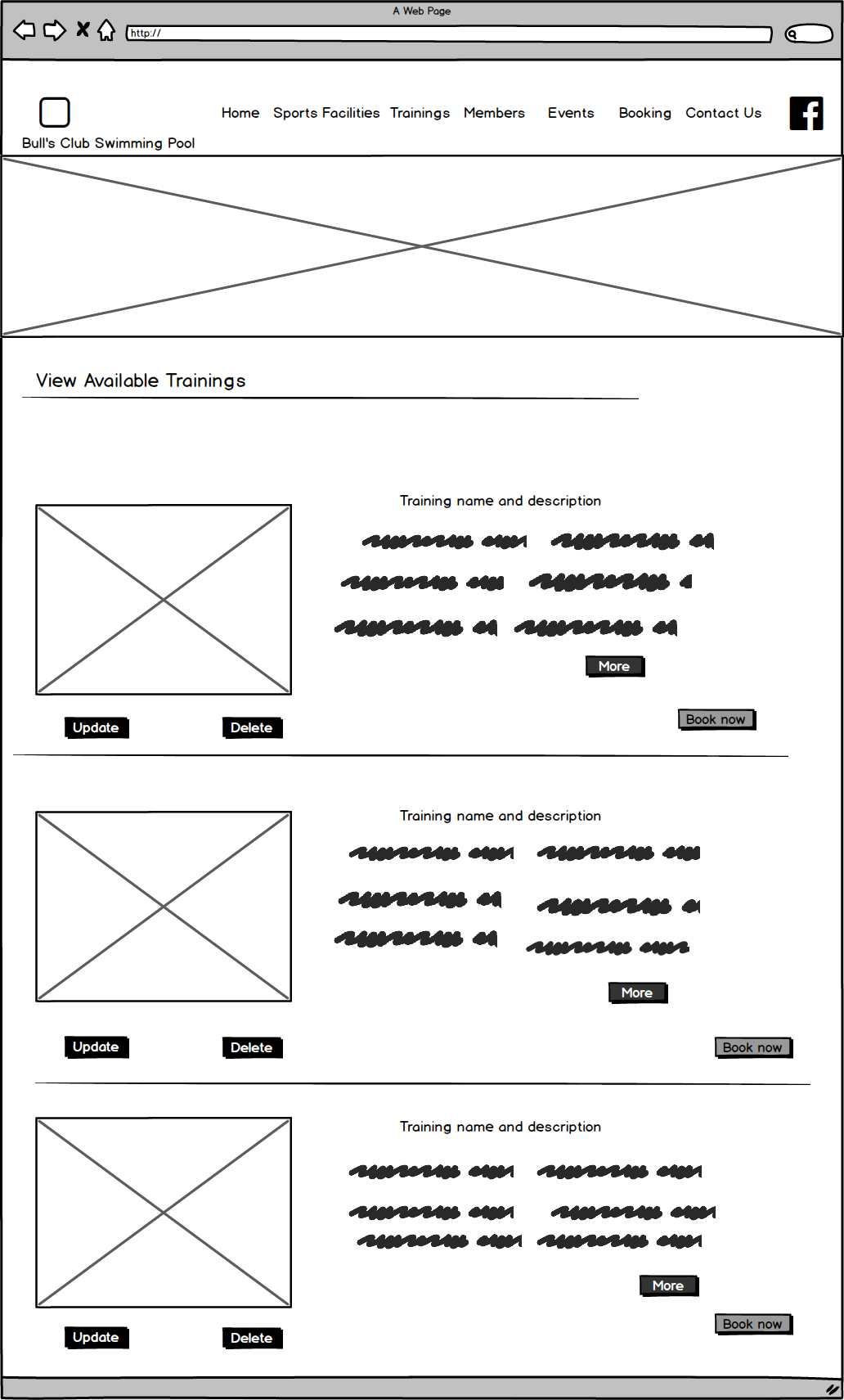
*Fig 5 : Membership Registration*

****

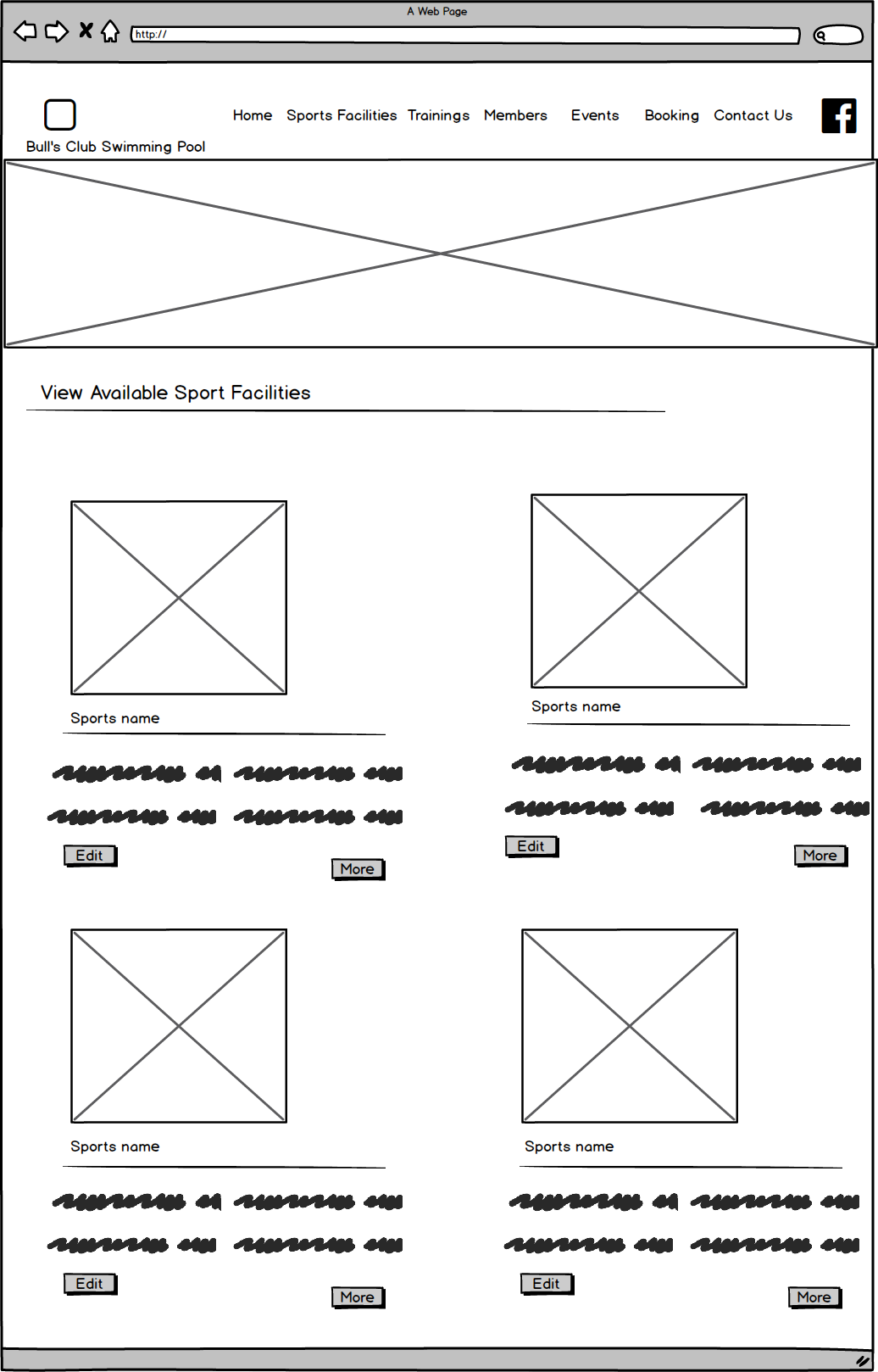
*Fig 6 : Submit or add event form*

****

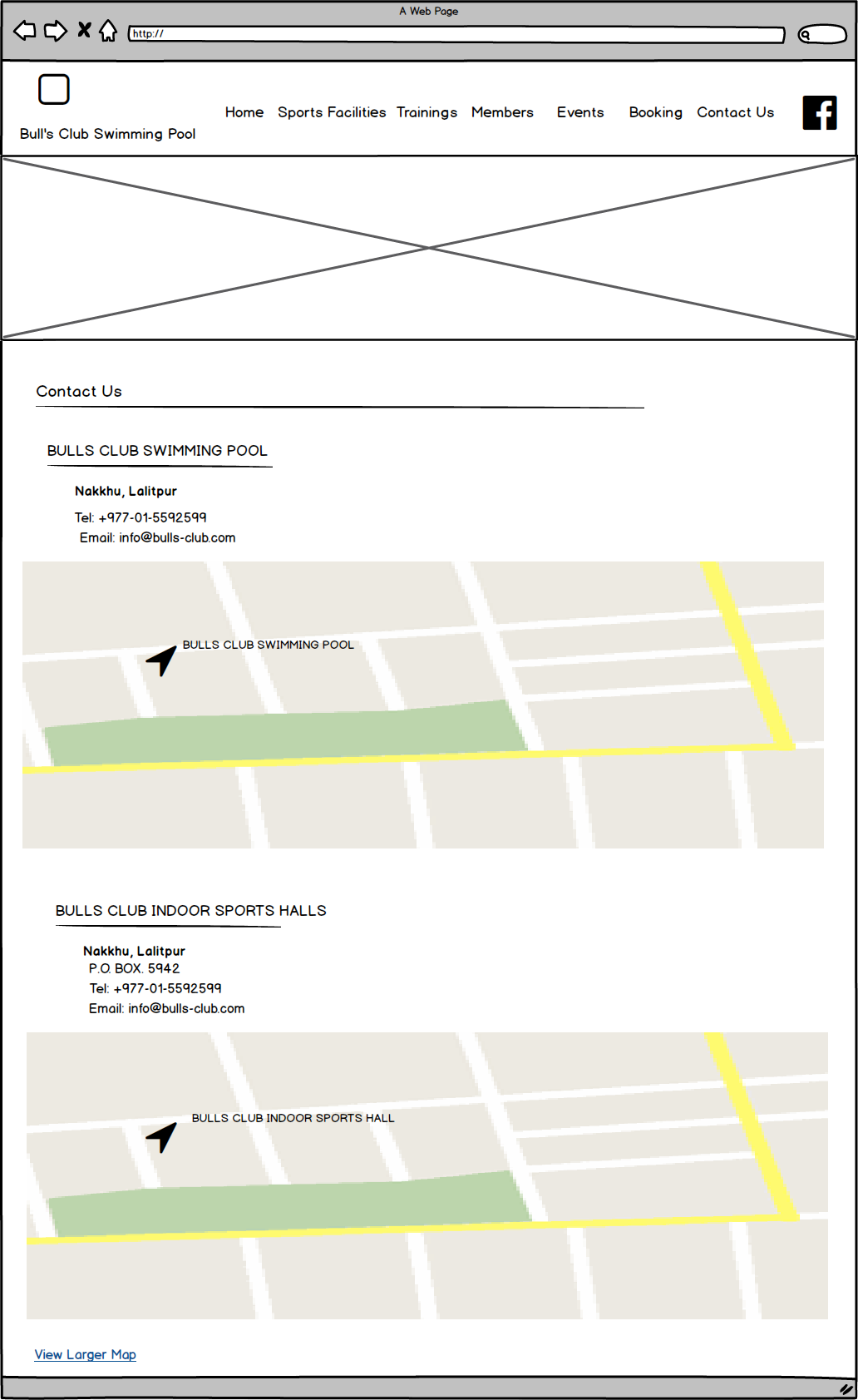
*Fig 7: view, update, delete, description of event*

****

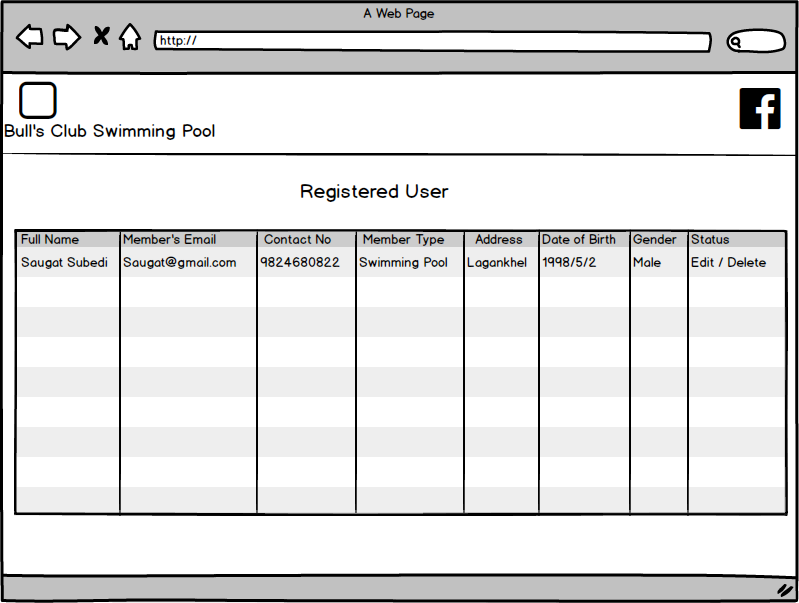
*Fig 8: view, update, delete, description, book of training*

****

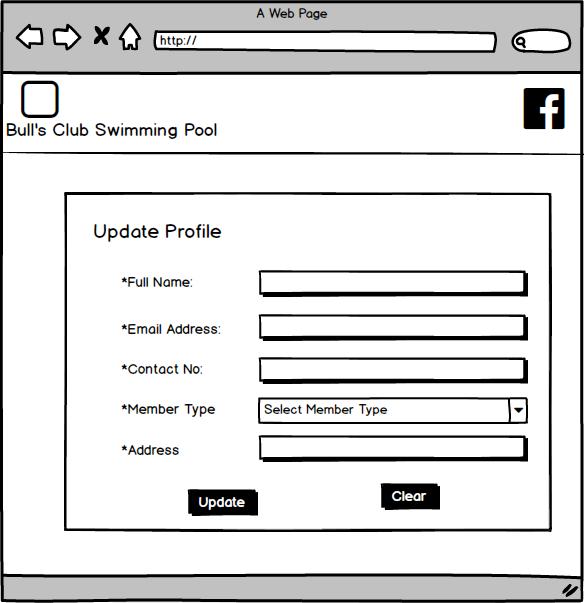
*Fig 9: view, edit and description of sports facilities*

****

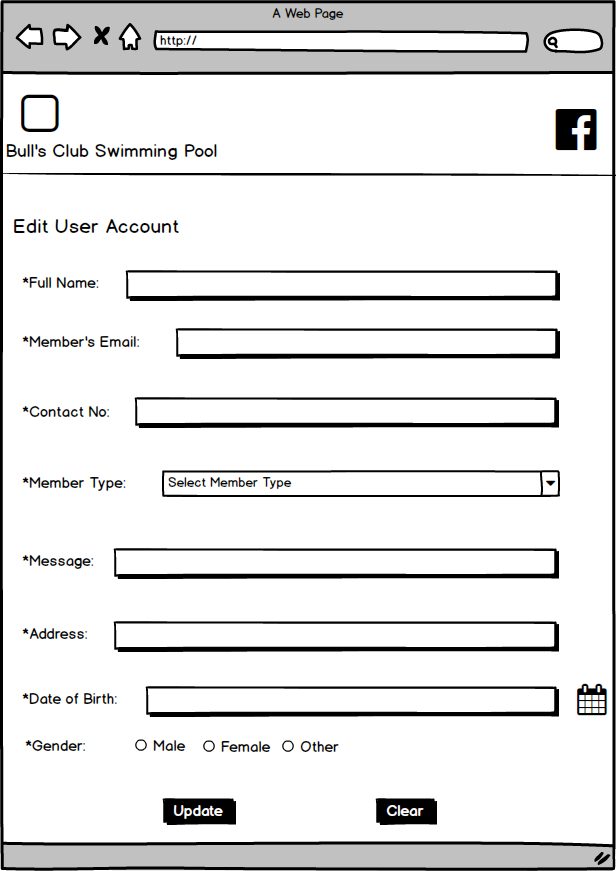
*Fig 10: Contact Us details*

****

*Fig 11: View Registered Users*

****

*Fig 12: Update User’s Details*

**

*Fig 13: Admin Edit User*