# **CH-3 Design**

Design is the most crucial phase in the development of a system. Logical design is formed in analysis phase and converted into physical design in this phase. In the design phase the SDLC process continues to move from the **what** questions of the analysis phase to the **how***.* A detailed description of what is needed to solve original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided.

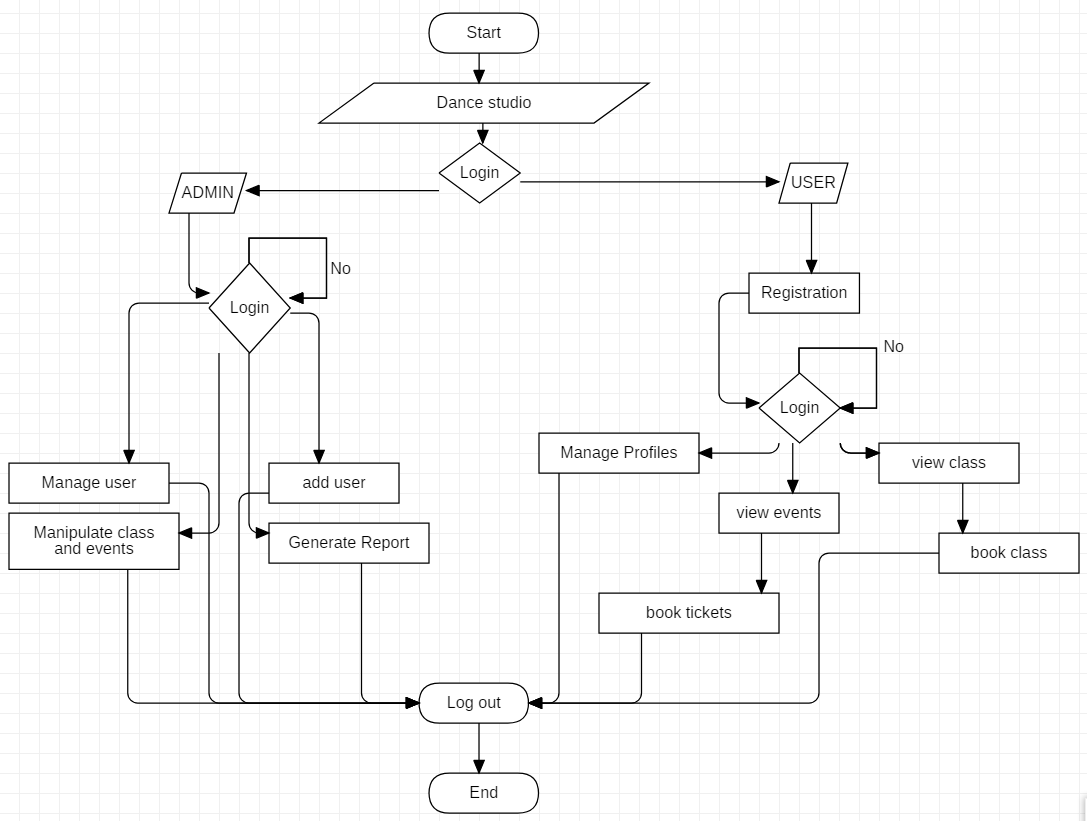
There are several tools and techniques used for describing the system design of the system. These tools and techniques are: Class diagram, Flow chart, Data flow diagram (DFD), Data dictionary, Decision table and Decision tree.

# **3.1 Structural modelling**

## **3.1.1 Class diagram**

## **3.1.2 Flow chart**

Flow chat is a pictorial representation which shows the flow of process which helps in understanding the flow in which system works.

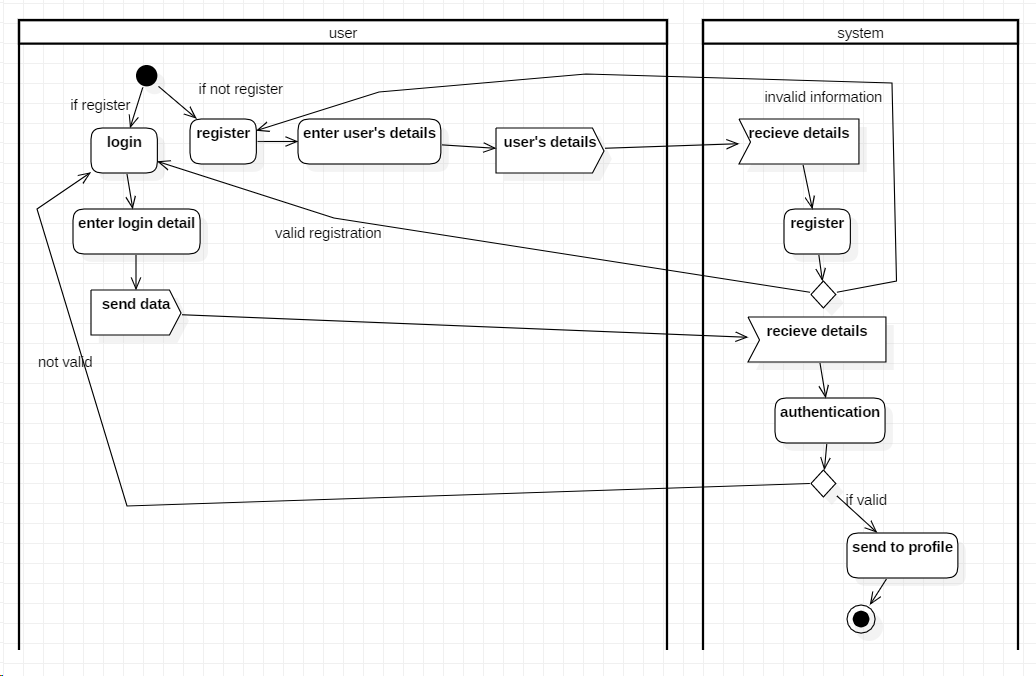


Above given picture is the flowchart design for this project which show the flow of process in which user will operate the system. At first login for two type of user admin and general user. They have their own accessible function for themselves.

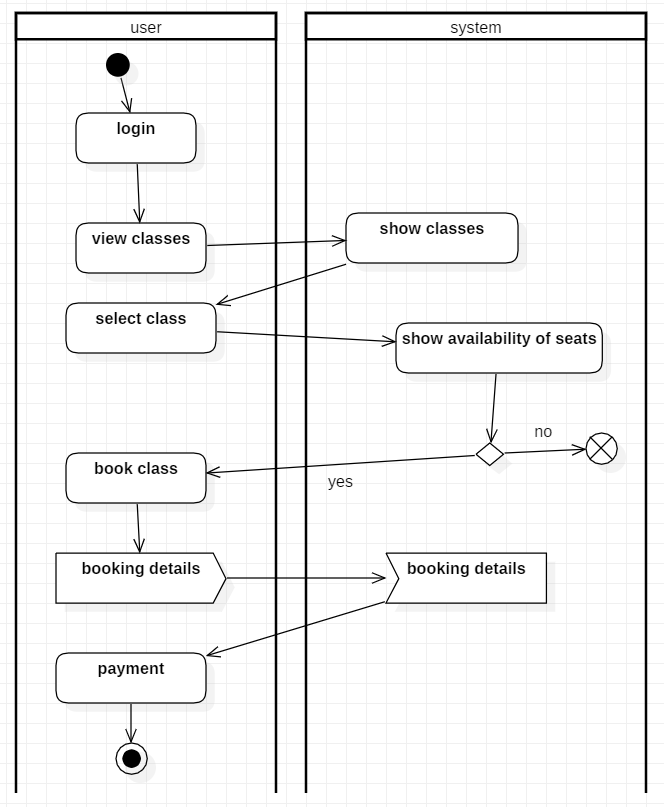
# **3.2 Behavior modelling**

## **3.2.1 Activity diagram**

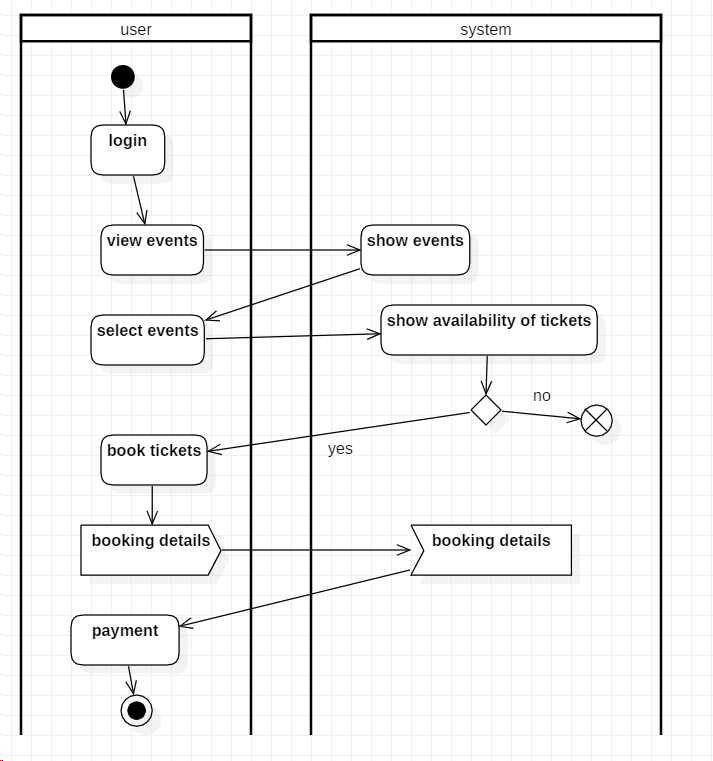
Activity diagram is basically a flowchart to represent the flow of one activity to another activity.



Above activity diagram show login and register system of the project. It means user need to get login to use some functions. It shows that user need to login into the system if they don’t have account in the system, they will require to register their account.



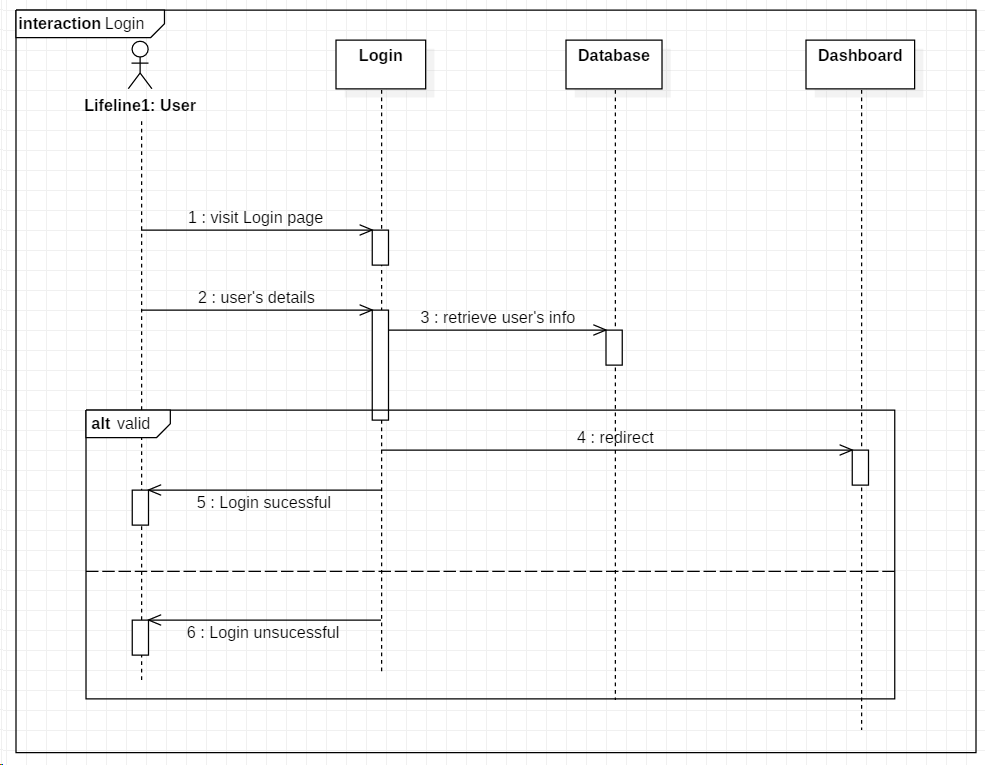
Above activity diagram show the activity process on booking dance class. Only login user can use this function. At first, user need to be logged in the system then they will view class after choosing class they check for seats availability of that class if seats are not available process gets terminate else they will book class.



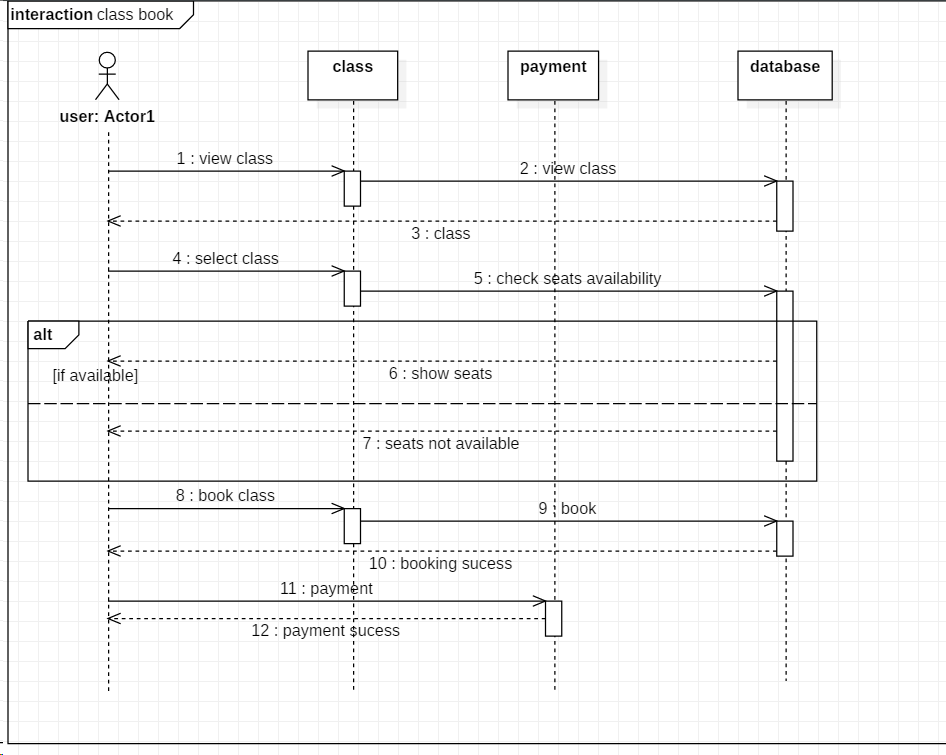
Above activity diagram show the activity process on booking tickets of events. Only login user can use this function. At first, user need to be logged in the system then they will view events after choosing events they will checks if tickets are available or not for this events if tickets are not available process gets terminate else they will book class by submitting some booking details.

## **3.2.2 Sequence diagram**

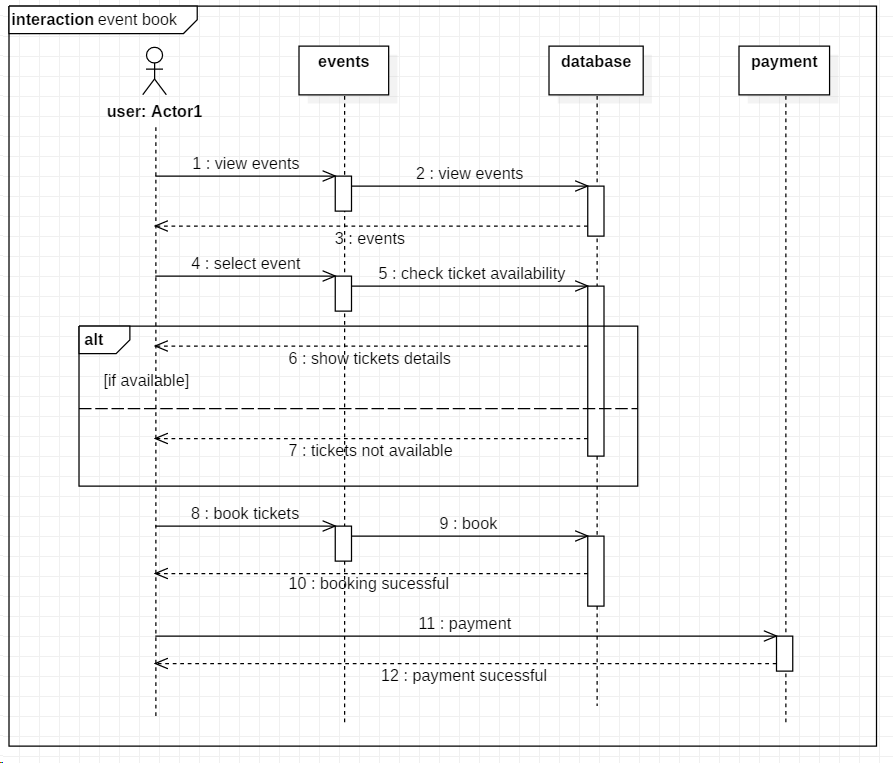
A sequence diagram is a pictorial representation of object interactions arranged in time sequence. It shows the objects and classes involved in the scenario and sequence of message exchanged between the object.



Above given diagram is login sequence diagram. Its checks authentication of user by asking some user information like email and password. If the information given by user is correct then they will get access to system else get redirect to failed page.



This sequence diagram shows the flow in which dance class is booked. User asks for class information from database and decide to either book the class or not then if they want to book the class, they will do payment.



This sequence diagram shows the flow in which events’ tickets is booked. User asks for events’ information from database and decide to either book the tickets or not then if they want to book the tickets, they will do payment.

# **3.3 Database modelling**

## **3.3.1 Data dictionary**

The table containing with many sets of information that explains how database is designed is known as data dictionary. It consists of contents like names, functions of data elements and their explanation, etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| user\_id | Integer(10) | 10 | pk\_user\_id | No | primary |
| first\_name | Varchar(20) | 20 | - | No | - |
| last\_name | Varchar(20) | 20 | - | No | - |
| address | Varchar(20) | 20 | - | No | - |
| gender | Varchar(10) | 10 | - | No | - |
| dob | date | - | - | No | - |
| email | Varchar(30) | 30 | - | No | - |
| phone | Varchar(20) | 20 | - | No | - |
| password | Varchar(20) | 20 | - | No | - |
| type | Varchar(10) | 10 | - | No | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| class\_id | integer | 10 | pk\_class\_id | No | Primary |
| class\_name | varchar | 20 | - | No | - |
| description | varchar | 255 | - | No | - |
| date | date | - | - | No | - |
| time | time | 7 | - | No | - |
| no\_of\_seats\_available | integer | 10 | - | Yes | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| events\_id | integer | 10 | pk\_events\_id | No | Primary |
| title | varchar | 30 | - | No | - |
| description | varchar | 255 | - | No | - |
| date | date | - | - | No | - |
| time | time | 7 | - | No | - |
| venue | integer | 30 | - | Yes | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| tickets\_id | integer | 10 | pk\_tickets\_id | No | primary |
| tickets\_available | integer | 10 | - | Yes | - |
| events\_id | integer | 10 | fk\_events\_id | No | foreign |

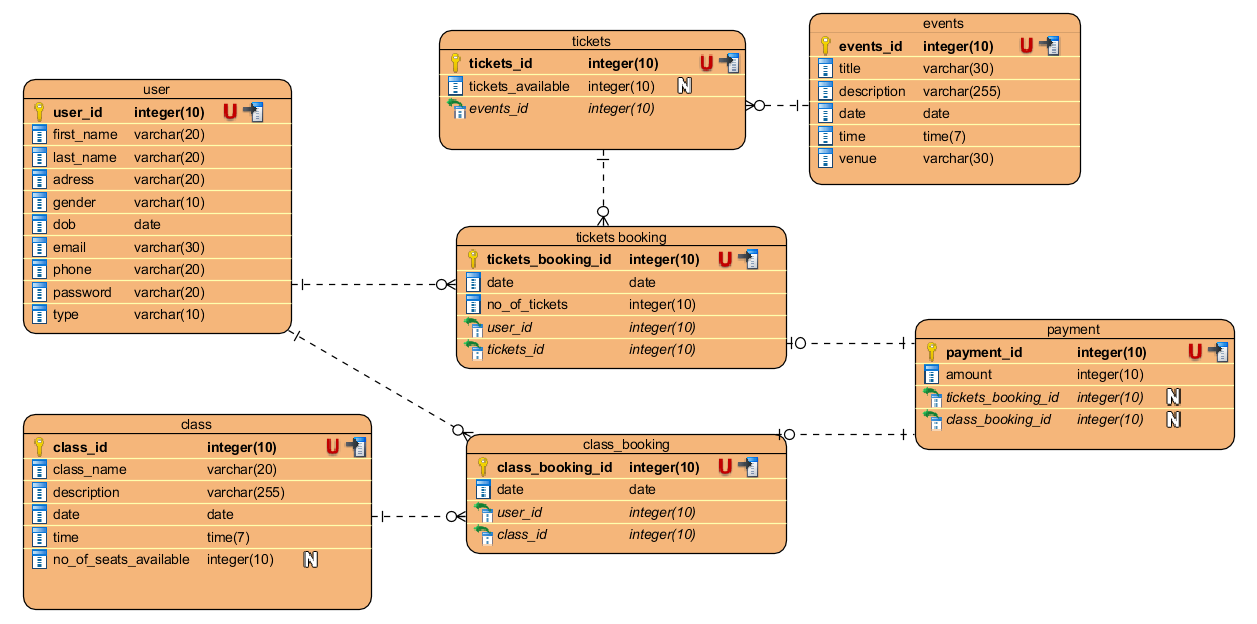
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| class\_booking\_id | Integer | 10 | Pk\_class\_booking\_id | No | primary |
| date | Date | - | - | No | - |
| user\_id | integer | 10 | fk\_user\_id | No | foreign |
| class\_id | integer | 10 | fk\_class\_id | No | foreign |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| tickets\_booking\_id | integer | 10 | pk\_tickets\_booking\_id | No | primary |
| date | date | - | - | No |  |
| no\_of\_tickets | integer | 10 | - | No |  |
| user\_id | integer | 10 | fk\_user\_id | No | foreign |
| tickets\_id | integer | 10 | fk\_ticket\_id | No | foreign |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Length** | **Constraint** | **Nullable** | **Key** |
| payment\_id | integer | 10 | pk\_payment\_id | No | primary |
| amount | integer | 10 | - | No | - |
| tickets\_booking\_id | integer | 10 | fk\_tickets\_booking\_id | Yes | foreign |
| class\_booking\_id | integer | 10 | fk\_class\_booking\_id | Yes | foreign |

## **3.3.2 ER-Diagram**

Entity-relationship (ER) diagram is a graphical representation of entities and their relationship with each other, typically used in database or information system.



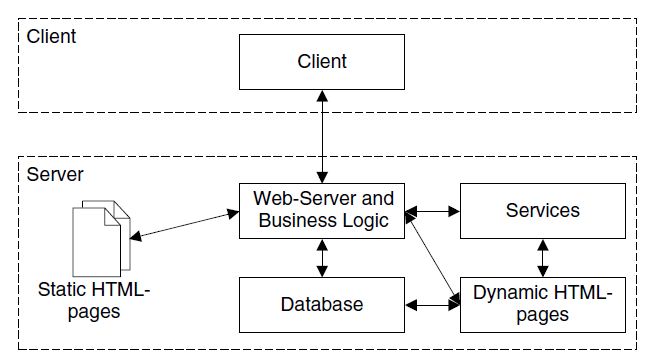
Above presented ER-diagram is designed based on data dictionary given prior to this design. Here the entities are shown along with the relationship that they have with each other.

# **3.4 Architectural Model**

For network architecture I am using client server architecture. Client server architecture is a network architecture in which each computer on the network is either a client or a server. Clients rely on servers for resources.

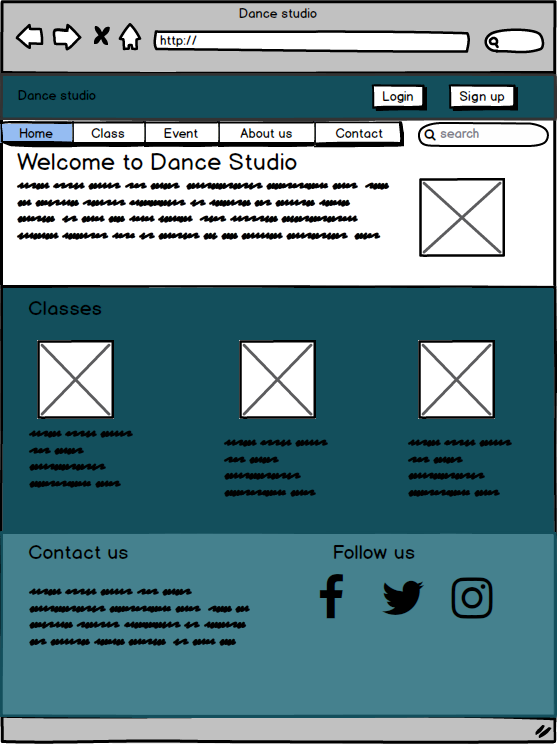
Because of the following reasons I have chosen client server architecture:

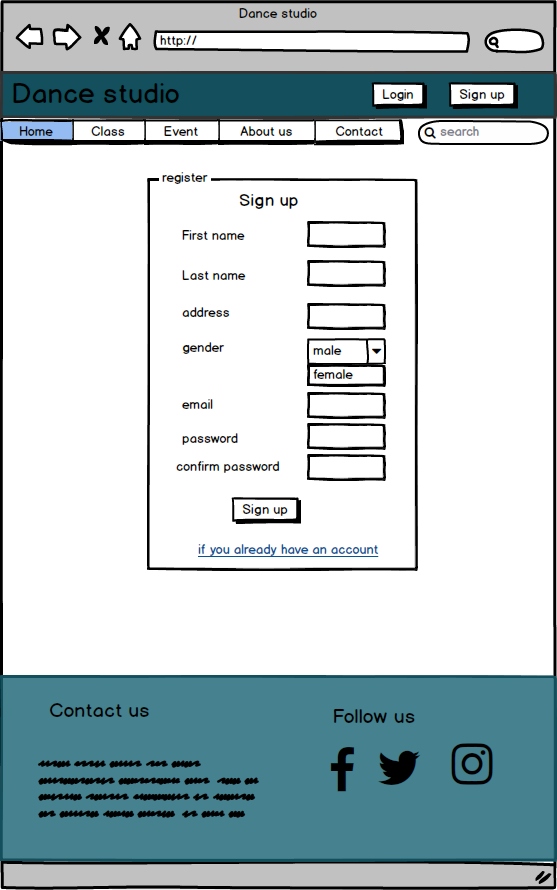
* More secure
* Data integrity
* Scalability
* High performance
* Easy maintenance and modification.

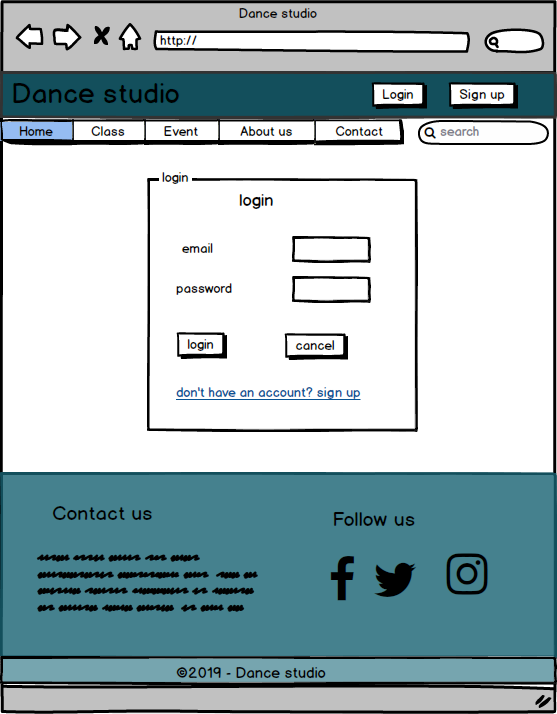


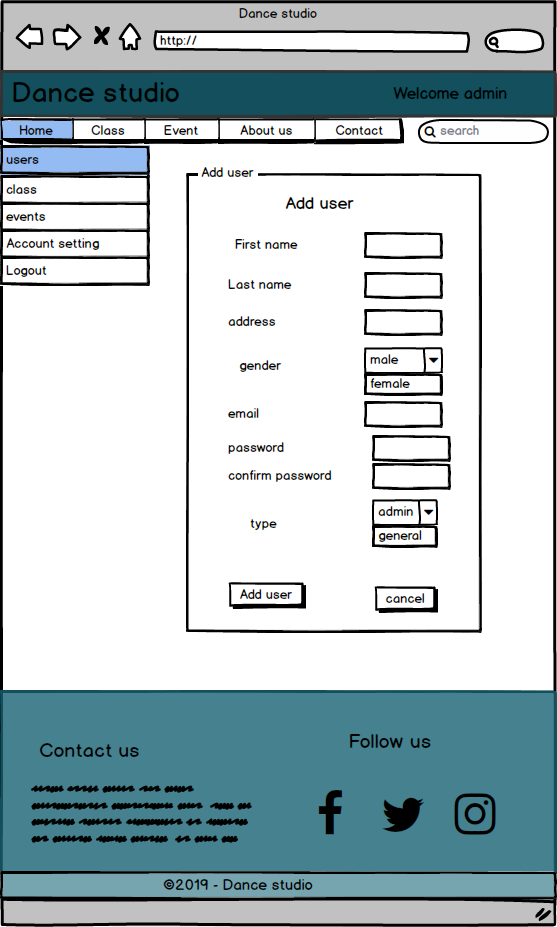
# **3.5 Prototype design**

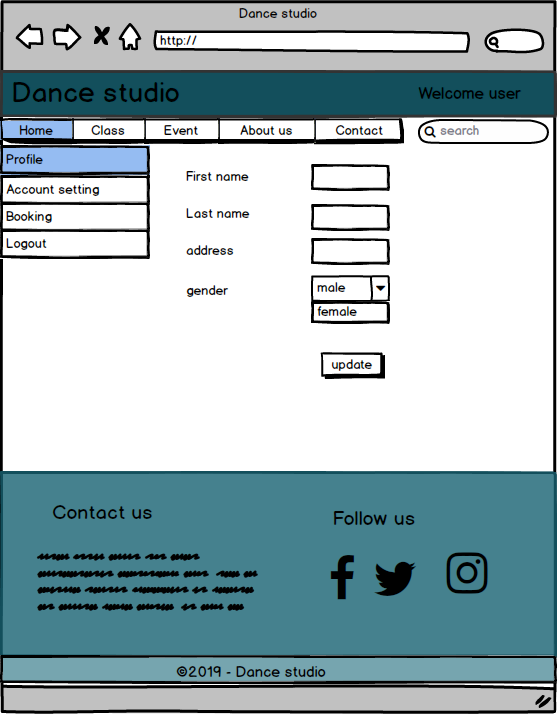
Prototype design is a way of presenting your idea in a buildable blueprint. It is an early sample or model of a product built.

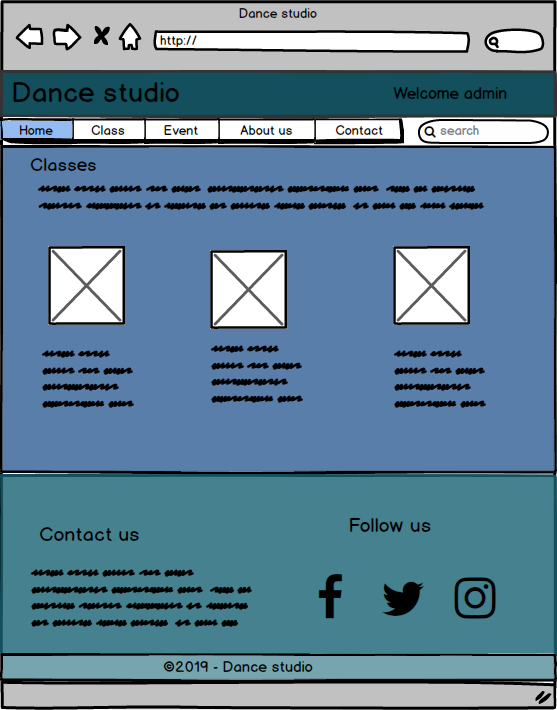


****

****

****

****

****