# Chapter 3: Design

# 3. Design:

Designing is the process or the early phase of the project development where’s the project works/key features, structure as well as the success of the project criteria are well explained and figure out. Some of the criteria are mentioned in the design phase of my project such as structural modelling, behavioral modelling, database modeling and UI modeling which requires certain diagram. All of the phases included in design are well explained how it is applied in the project and they are:

## 3.1 Structural Modeling

This modeling represents the framework for the system and this is the framework or the place where all other components exist. Structural modeling captures the static features of a system, it consists of class diagram, objects diagram, deployment diagram, component diagram etc. but this model never describes the dynamic behavior of the system and looking to the requirement for the design phases I have asked to provide details on class diagram as well as the final class diagram of the project.

### 3.1.1 class diagram

It is the general-purpose modeling language used to visualize the system. It is a graphical language that is standard to the software industry for specifying, visualizing, constructing and documenting the artifacts of the software systems, as well as for business modeling. Class diagrams are useful in all forms of object-oriented programming (OOP).

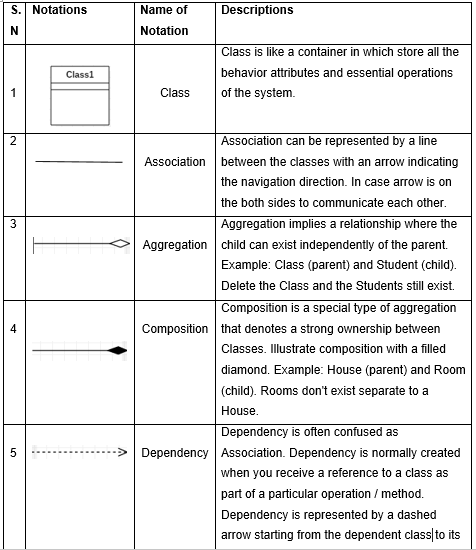
#### Justification: why class diagram is made

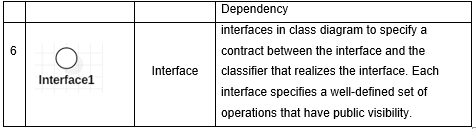
Reasons of including class diagram in my project are explained/described in points below:

* As my project is OOP object-oriented program this is the only UML which can appropriately depict various aspects of OOPS concept.
* It can be used for the base for deployment and component diagram.
* As the classes are building block of an application based on OOPs, class diagram has appropriate structure to represent the classes, inheritance, and everything that OOPs have in its context.
* It represents or describes the overall project on a single diagram.
* It illustrates the proper design and analysis of application can be faster and efficient.

In the process of drawing class diagram many notations representing the process are included and it uses or notations explanation used in process of drawing class diagram are:

**Notations used in class diagram:**





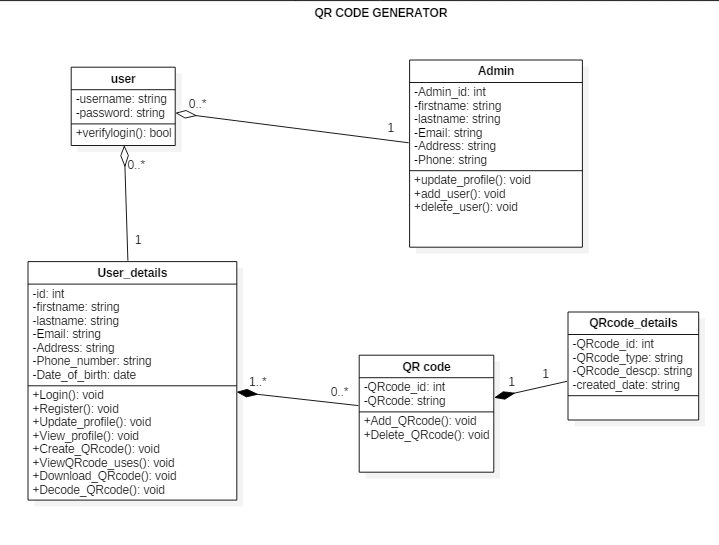


Fig: Class diagram

The above shown class diagram is for the QR code generator system. This class diagram clearly describes the attributes and the methods of the classes which helps in explaining the system in easy way. While developing the class diagram I have identified some classes and their attributes as well as the methods and the relations have been also shown in diagram with multiplicity. Such as user can be one or many and QR code be zero to many as one user may not have QR code created or can have created multiply times in numbers and one QR code can have one description and specify date of creation so they are shown in one to one.

## 3.2 Behavioral Modeling

Behavioral Diagrams depict the elements of a system that are dependent on time and that convey the dynamic concepts of the system and how they relate to each other. The elements in these diagrams resemble the verbs in a natural language and the relationships that connect them typically convey the passage of time.

### 3.2.1 Activity Diagram

Activity diagram explains the dynamic aspects of the system/project. It is important diagram in UML as the part of Behavioral modeling. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

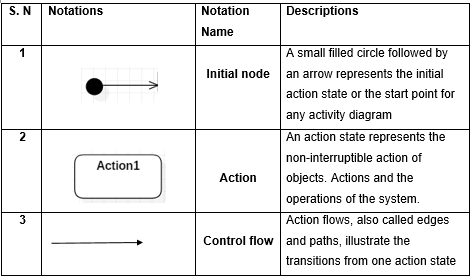
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

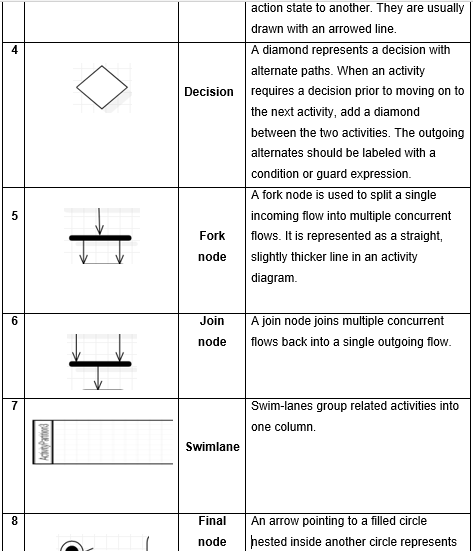
#### Justification:

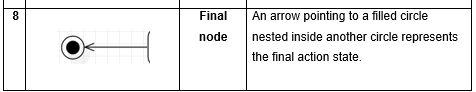
Activity diagram captures the dynamic behavior of the system and other purpose of an activity diagram are:

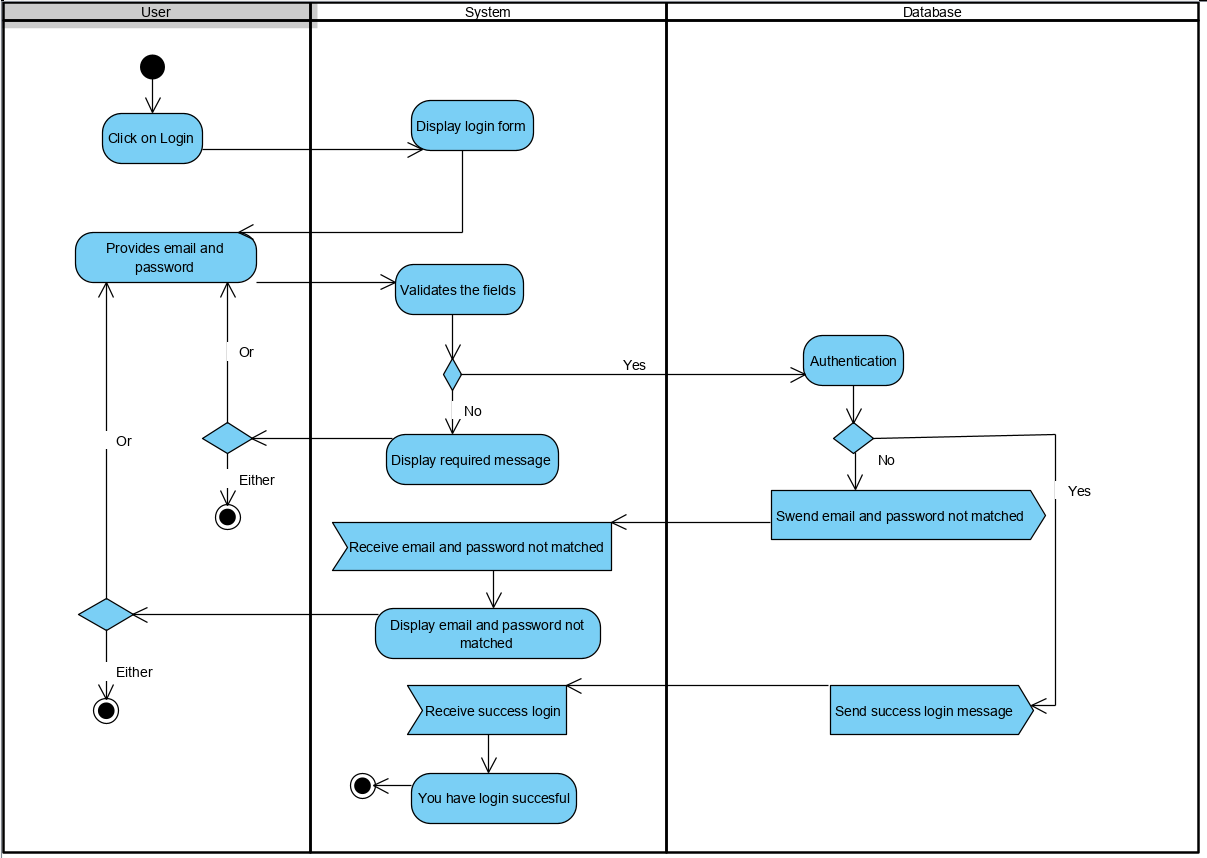
* It draws the activity flow of a system on high level.
* This diagram describes the parallel, branched and concurrent flow of the system.
* Diagram describes the sequence from one activity to another.
* An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

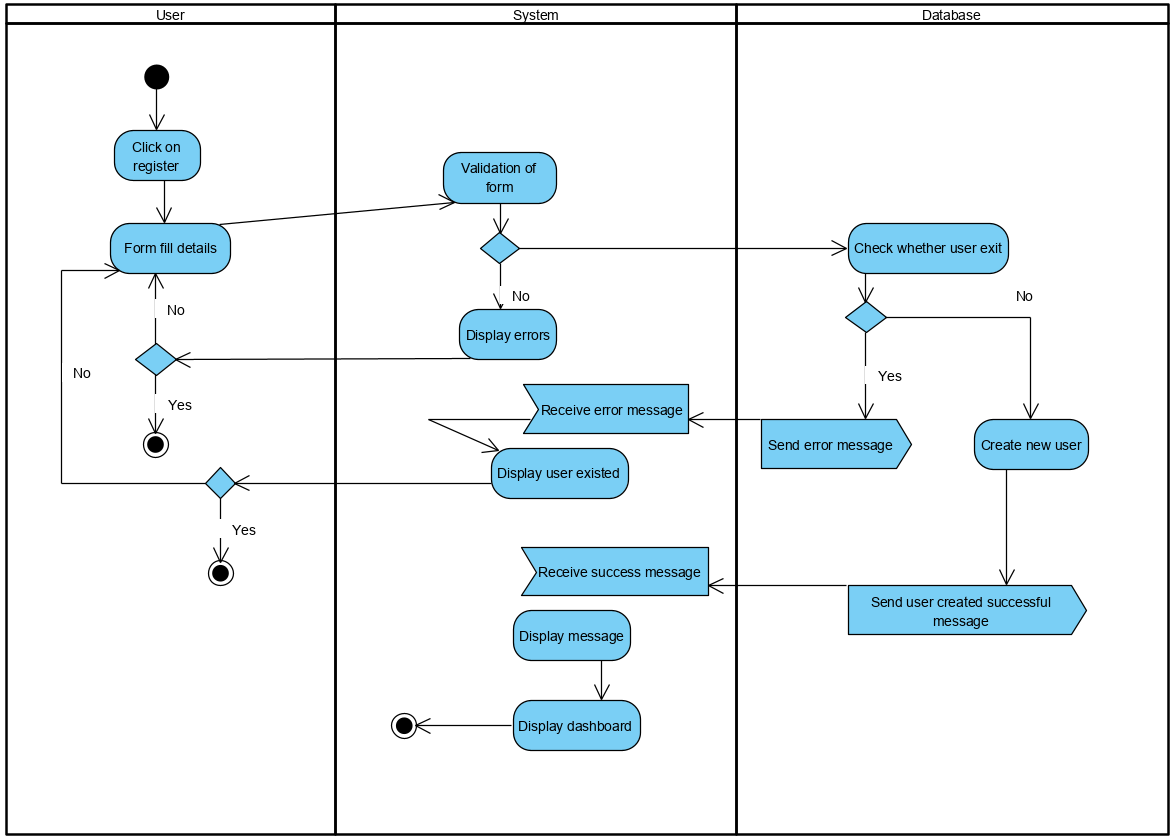
**Notations used in Activity diagram:**

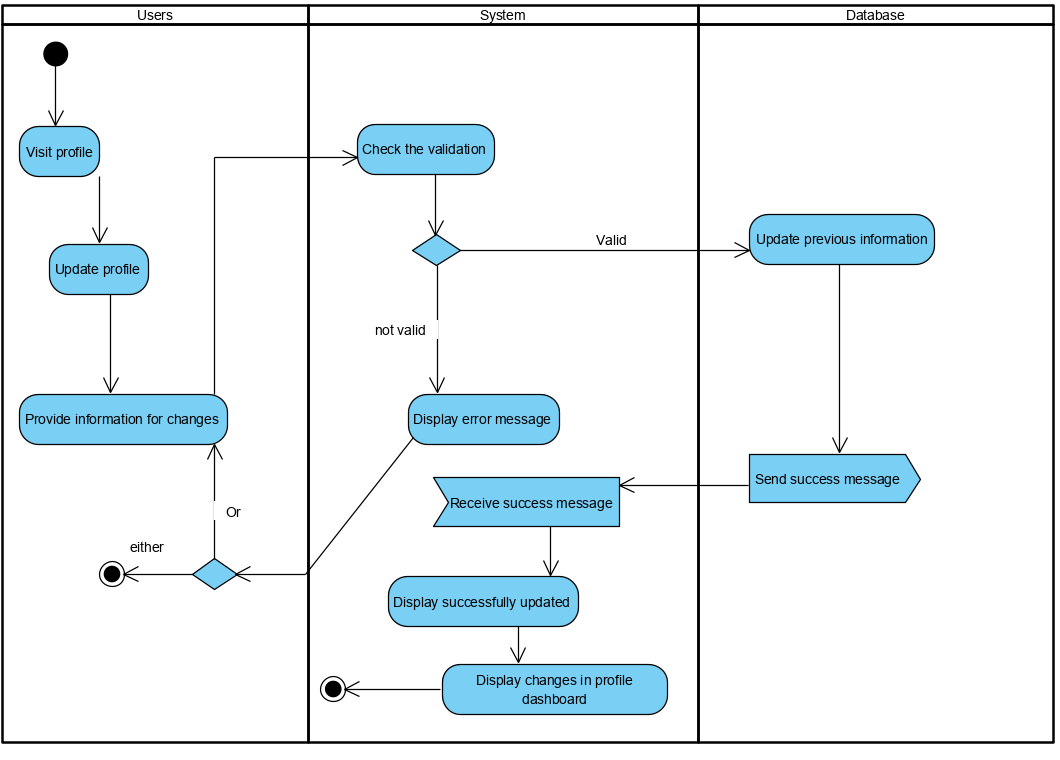


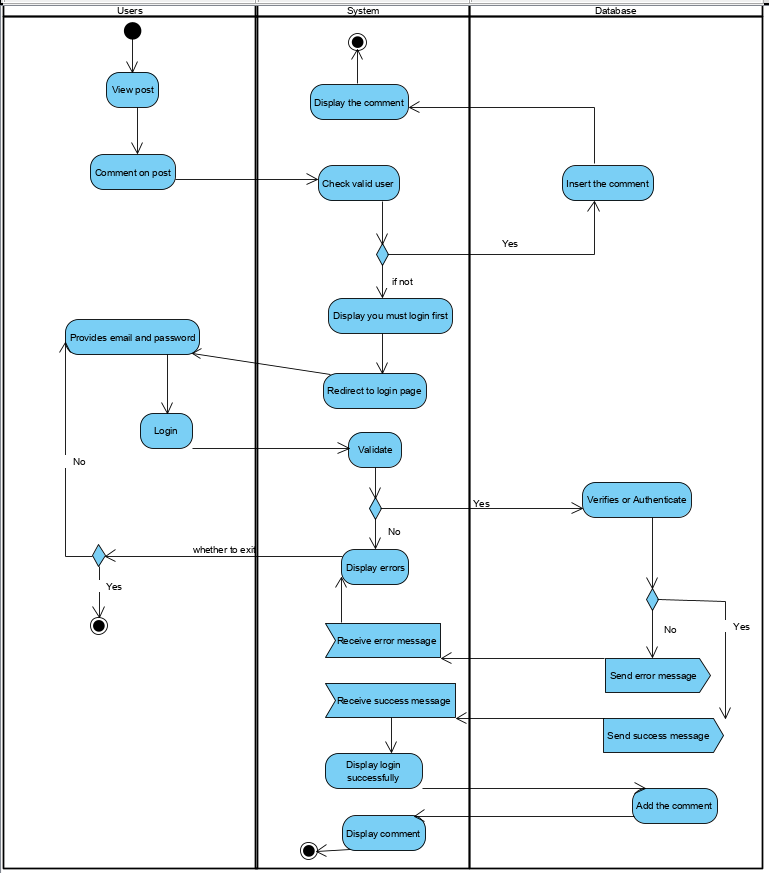


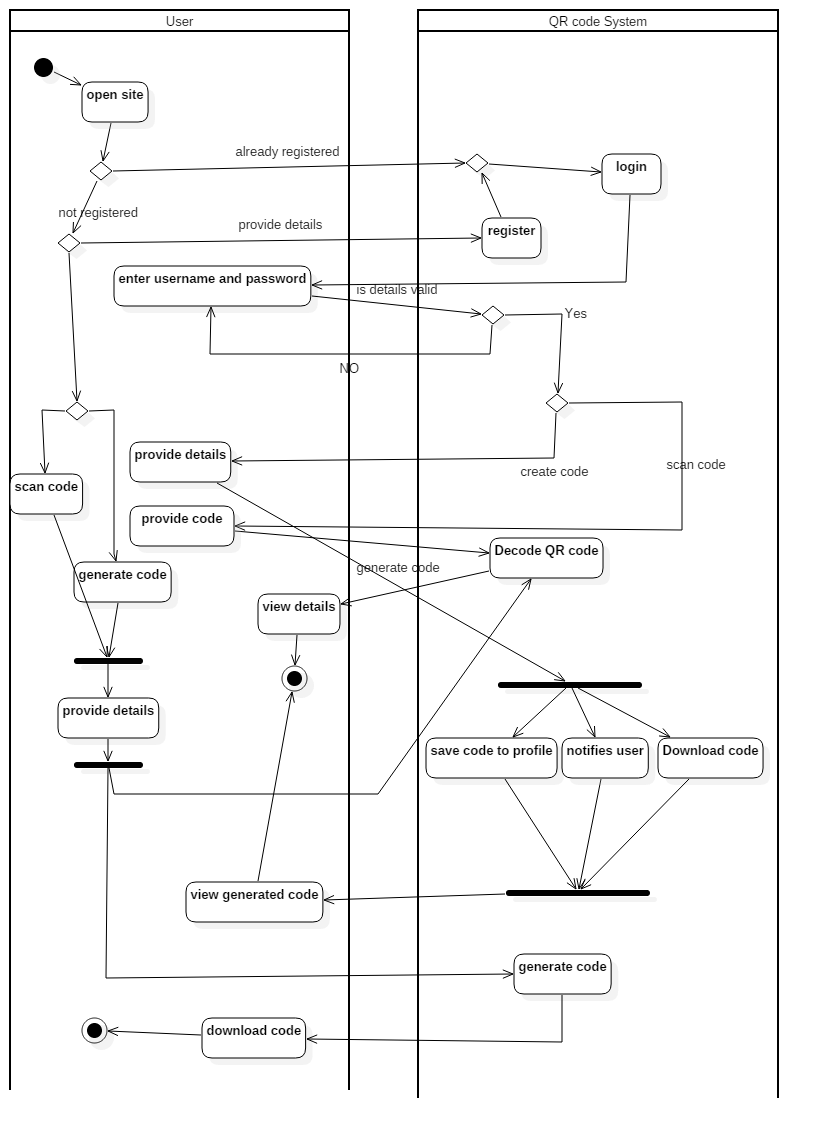












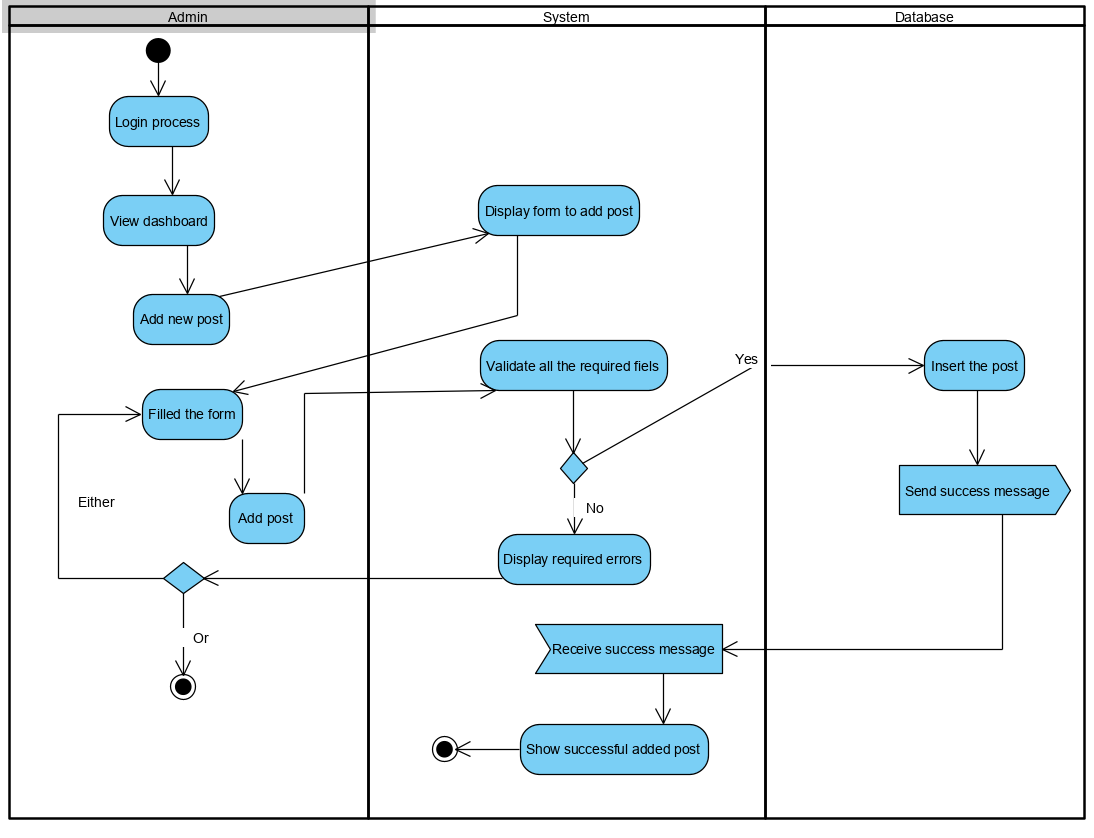


Fig: Activity Diagram

The above activity diagram shows the user interaction to our system how it reacts or acts. And going through the diagram we would know that user who open site can go through the easy process of creating code with registration or login process or can go through without process and create as well as decode code.

The diagram is developed in swim lane using a tool named StarUML. how the user interact with system in every aspect is shown from starting to ending point.

### 3.2.2 Sequence Diagram

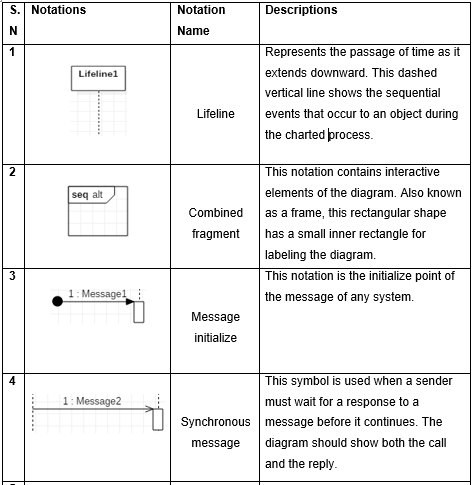
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

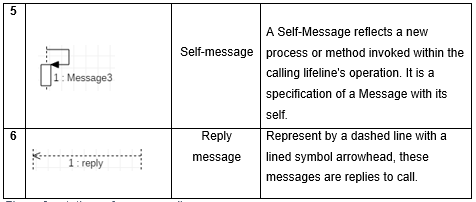
#### Justification:

In the design phase of my project I have made sequence diagram and the justification of making this diagram is:

* Sequence diagram show the interactions between objects in the sequential order that those interactions occur.
* Besides documenting an organization current affair, a business-level sequence diagram can be used as requirement document to communicate requirements for a future system implementation.
* In the design phase, architects and developers can use the diagram to force out the system’s object interactions, thus fleshing out overall system design.

**Notations used in Sequence diagram:**





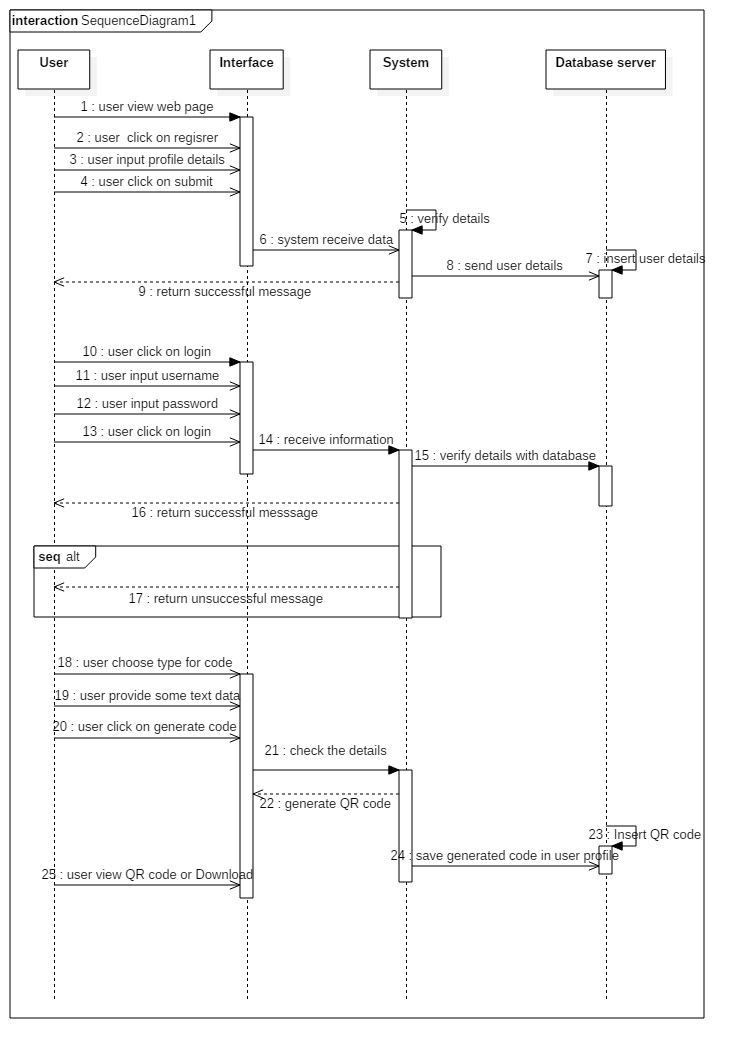
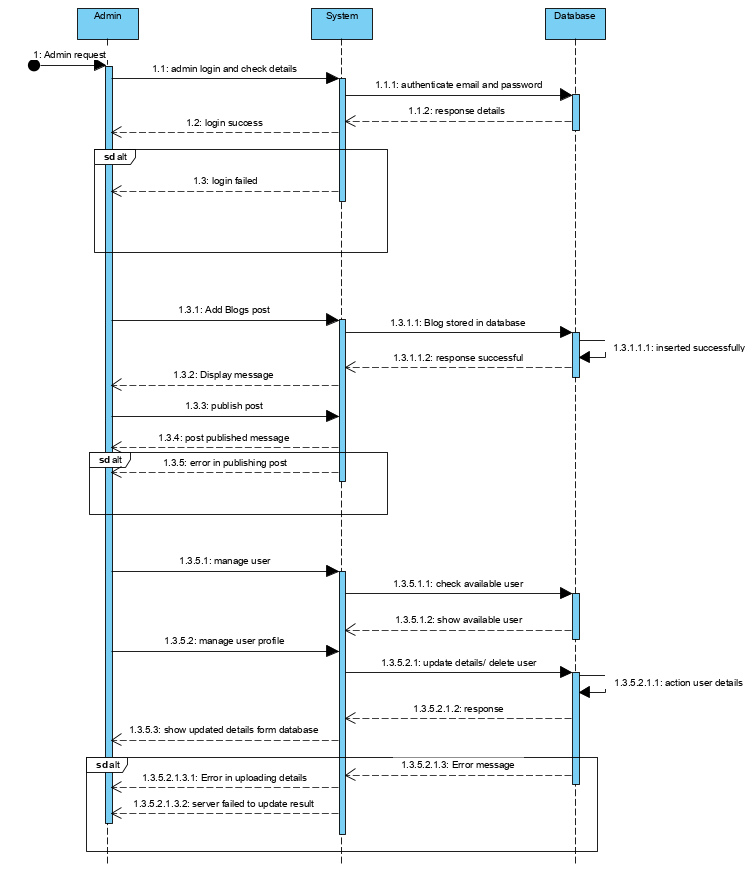


Fig: Sequence Diagram



The designed sequence diagram explains how the objects interactions occur in sequential order and the functionality. The above diagram says the how user registered in our system, how the exchange of information and message is going on through the system in order. In diagram I have user four lifeline user, interface, system, database server.

User interact with the interface and the system receive info as the user provide it to the interface and provide return message and check the login functionality in case of user login.

## 3.3 Database Modeling

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. A data model helps design the database at the conceptual, physical and logical levels.

### 3.3.1 Data dictionary

To a data dictionary we can say a file or a set of files containing the metadata i.e, data of the data. It is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users only database administrators interact with the data dictionary. The metadata in the data dictionary includes the following:

- Names of all tables in the database and their owners.

- Names of all indexes and the columns to which the tables in those indexes relate.

- Constraints defined on tables, including primary keys, foreign-key relationships to other tables, and not-null constraints.

Some of the tables were made for storing information of user, admin and QR code as well as posts and images used they are:

**Table for user:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Length | Null | Constraint |
| User\_id | Integer | 10 | No | Primary key |
| Firstname | Varchar | 200 | Yes | - |
| Lastname | Varchar | 200 | Yes | - |
| Email | Varchar | 200 | Yes | - |
| Address | Varchar | 200 | Yes | - |
| Phone\_number | Varchar | 200 | Yes | - |
| Date\_of\_birth | Date | - | Yes | - |
| Registered\_date | Date | - | Yes | - |
| Admin\_id | Integer | 10 | No | Foreign key |

**Table for admin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Length | Null | Constraint |
| Admin\_id | Integer | 10 | No | Primary key |
| Firstname | Varchar | 200 | Yes | - |
| Lastname | Varchar | 200 | Yes | - |
| Email | Varchar | 200 | Yes | - |
| Address | Varchar | 200 | Yes | - |
| Phone\_number | Varchar | 200 | Yes | - |

**Table for QR code:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Length | Null | Constraint |
| QRcode\_id | Integer | 10 | No | Primary key |
| QRcode\_type | Varchar | 100 | Yes | - |
| Descrption | Varchar | 200 | Yes | - |
| QRcode | Varchar | 200 | Yes | - |
| Created\_date | date |  | Yes | - |
| User\_id | Integer | 10 | No | Foreign key |

**Table for post:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Length | Null | Constraint |
| Post\_id | Integer | 10 | No | Primary key |
| Title | Varchar | 100 | Yes | - |
| Descrption | Varchar | 200 | Yes | - |
| Image\_name | Varchar | 200 | Yes | - |
| Published\_date | date |  | Yes | - |
| Admin\_id | Integer | 10 | No | Foreign key |

**Table for slider:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Length | Null | Constraint |
| Id | Integer | 10 | No | Primary key |
| Title | Varchar | 100 | Yes | - |
| Descrption | Varchar | 200 | Yes | - |
| Image\_name | Varchar | 200 | Yes | - |
| Published\_date | date |  | Yes | - |

### 3.3.2 ER diagram

An Entity Relationship Diagram (ERD) is a snapshot of data structures. An Entity Relationship Diagram shows entities (tables) in a database and relationships between tables within that database.

an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

And there are three basic elements in ER-Diagrams:

* Entities are the “things” for which we want to store information. An entity is a person, place, things or event.
* Attributes are the data we want to collect for an entity.
* Relationships describe the relations between the entities.

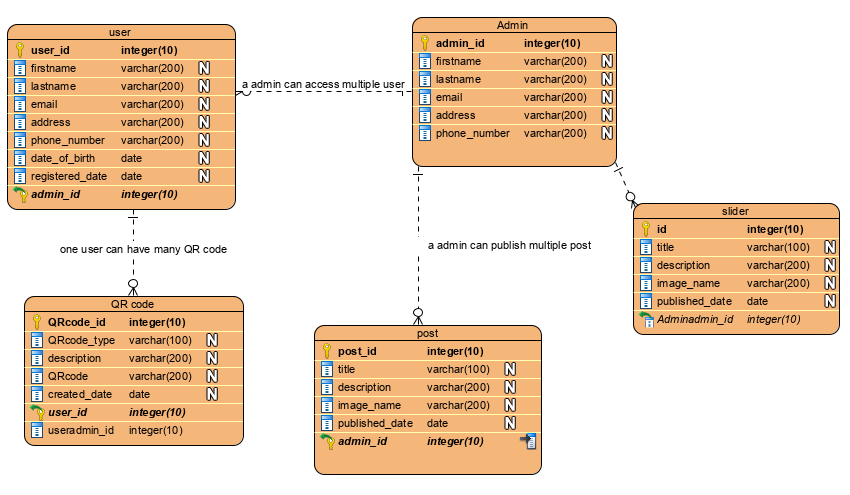
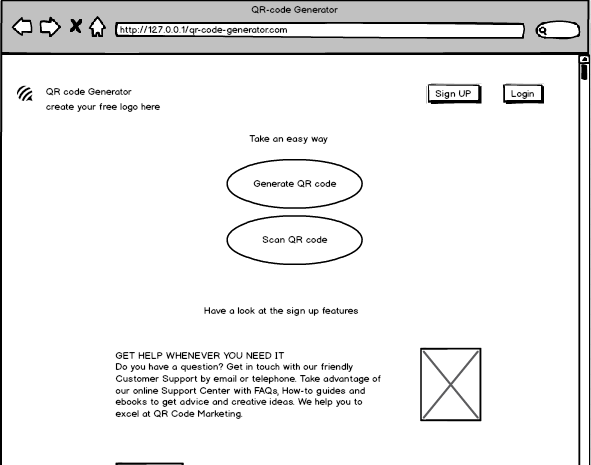


Fig: ER- Diagram

## 3.4 UI Modeling

UI modeling is the process of making interfaces in software which focus on design or looks. It typically refers to graphical user interface but also includes others, such as voice-controlled ones. The design for my web app as a proto-types are

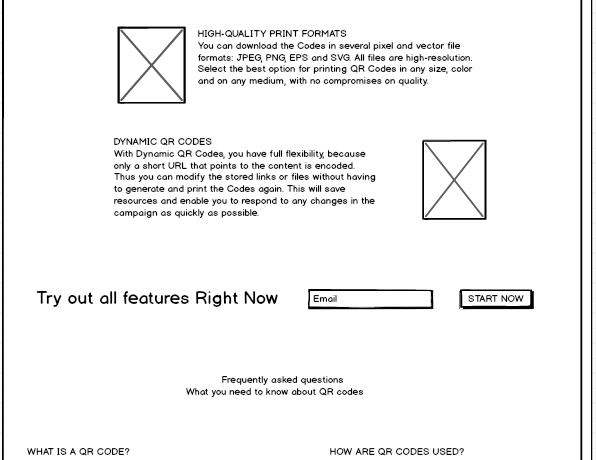
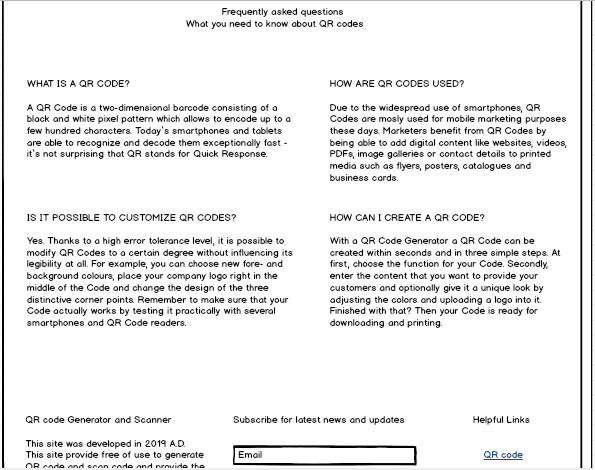


Fig: home page



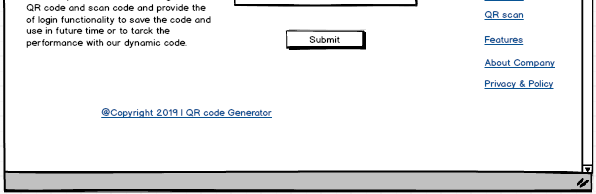


Fig: home page

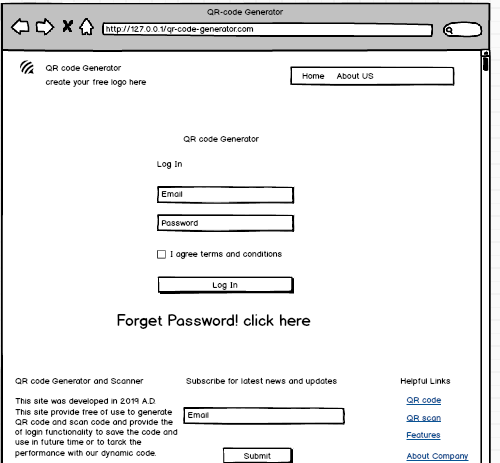


Fig: login page/ form

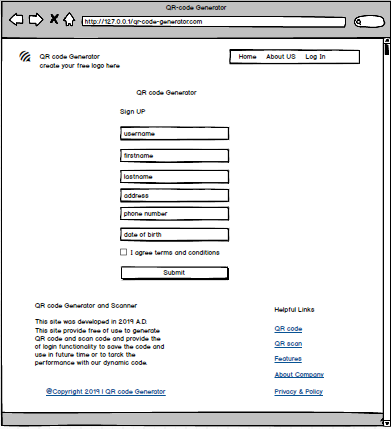


Fig: registration page

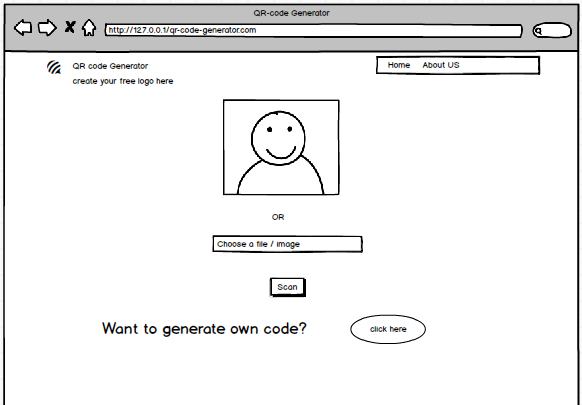


Fig: scan code

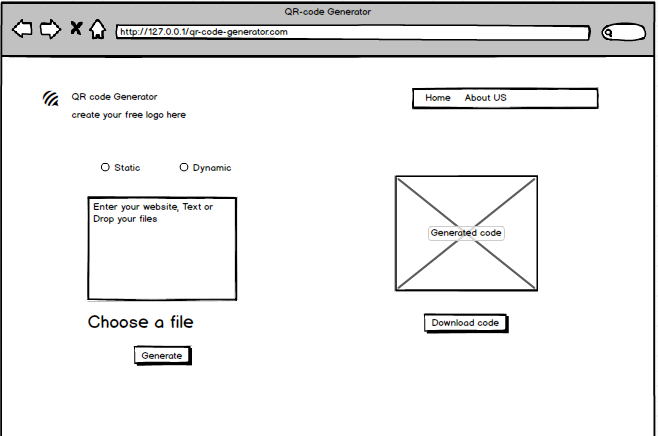


Fig: generate code

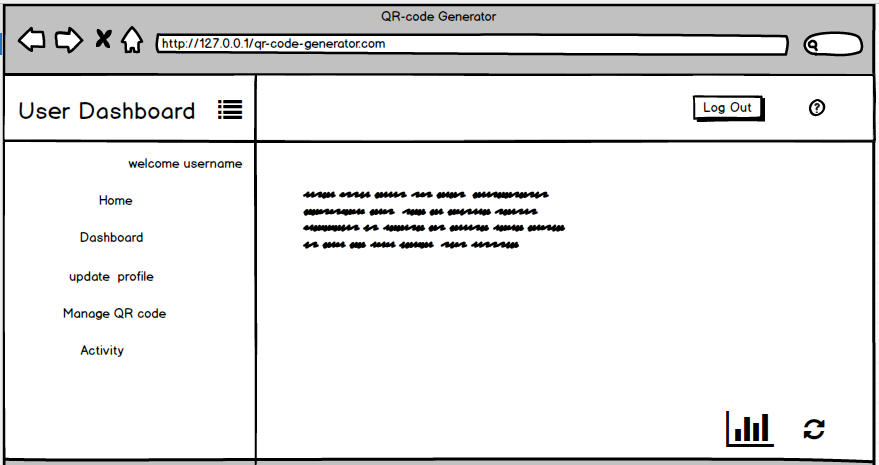


Fig: user dashboard

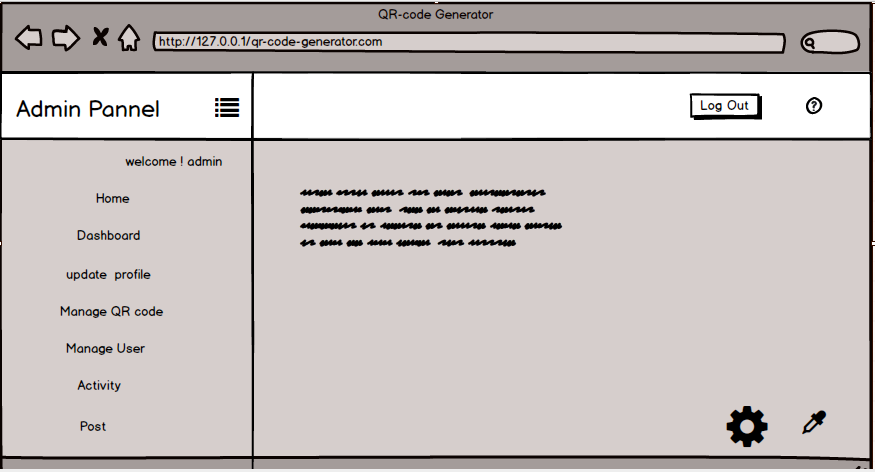


Fig: admin panel