# **Chapter 3: Design**

Design is the process of defining software methods, functions, objects, and the overall structure and interaction of your code so that the resulting functionality will satisfy user requirements.

## Structural Modeling

### Final Class Diagram

A class diagram in the [Unified Modeling Language (UML)](https://en.wikipedia.org/wiki/Unified_Modeling_Language) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**Justification:**

1. It can be easily understand by programmers.
2. Base for component and deployment diagrams.
3. It gives the detail insight into the structure of the systems.

**Notation Used:**

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax | Construct | Explanation | Remarks |
|  | Class | A description of a set of objects that possesses some attributes, operations, methods, relationships etc. |  |
|  | Association | Used to represent the relationship between classes that possesses connections among their instances. |  |
|  | Aggregation | Association of special nature where objects are assembled creating more complex object. |  |

### Data Flow Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Justification:

1. Logical information flow of the system
2. Determination of physical system construction requirements
3. Simplicity of notation

## Behavioral Modeling

### Activity Diagram

Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Activity diagram in [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) diagram is to describe dynamic aspects of the system.

**Justification:**

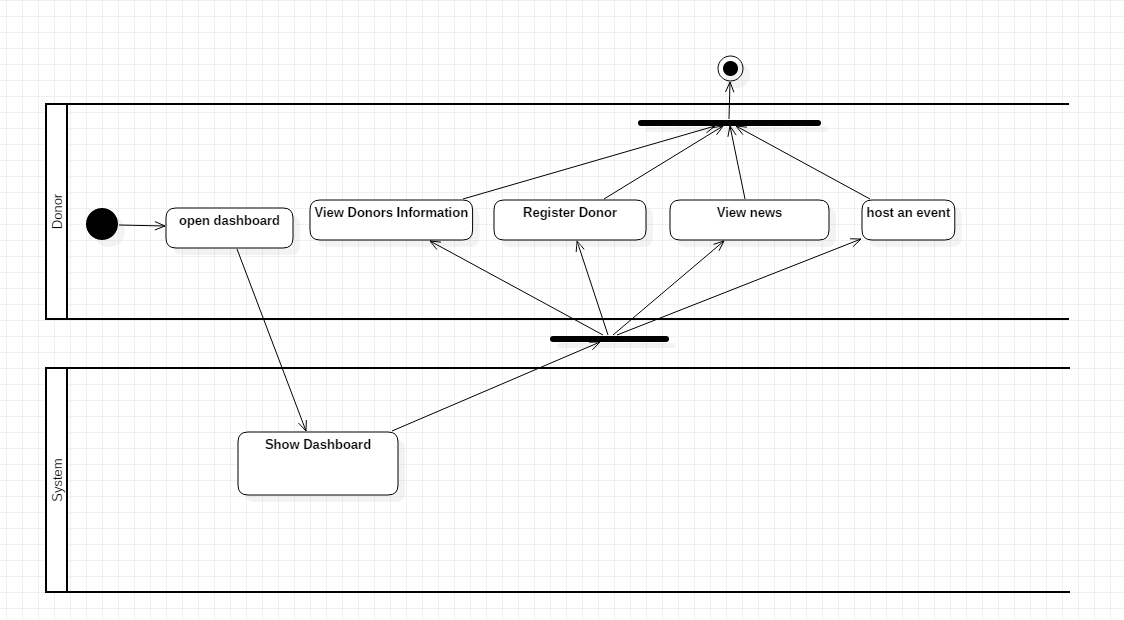
1. Identify candidate use cases, through the examination of business workflows
2. Model workflows between/within use cases.
3. Model complex workflows in operations on objects.
4. Model in detail complex activities in a high level activity Diagram.

**Notation Used:**

|  |  |  |
| --- | --- | --- |
| Notation | Name | Reason |
|  | Start | It shows the starting point of the workflow. |
|  | Action | It represents the action of the object. |
|  | Control flow | It represents the transitions of one action state to another one. |
|  | Decision | It represent a decisions with alternate paths which is represented by diamond shape. |
|  | End | It shows the end of the workflow. |
|  | Fork | Splits the behavior into a set of parallel or concurrent flows of activities. |
|  | Join | Bring back together a set of parallel or concurrent flow of activities. |

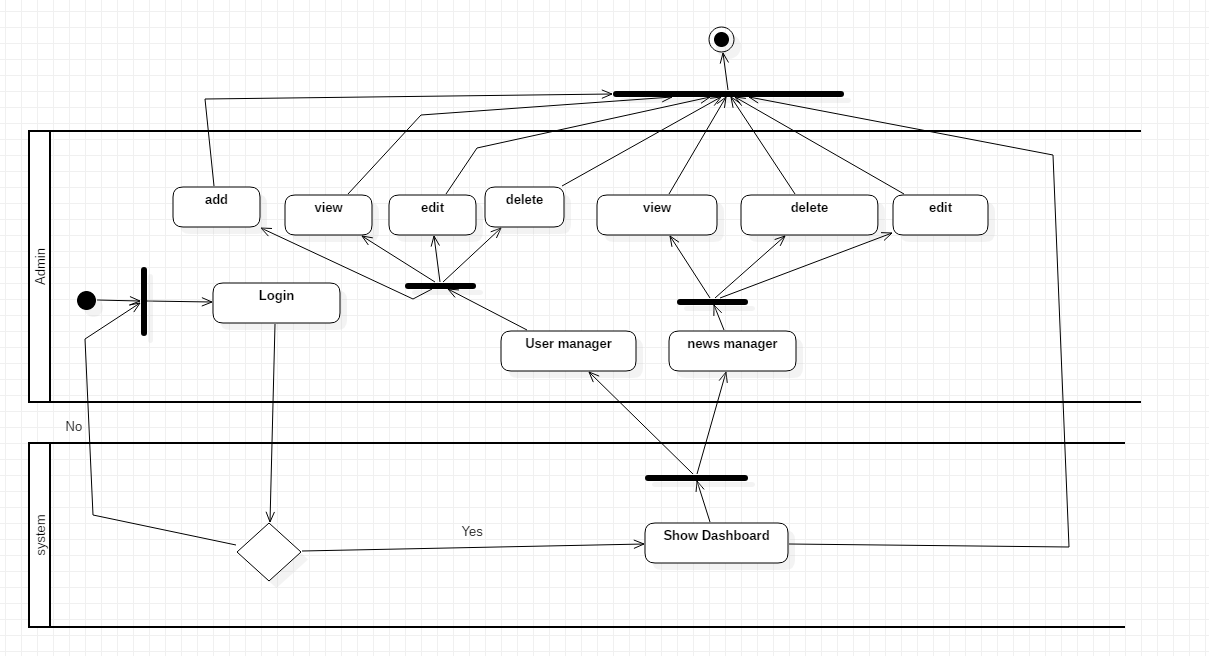
**Activity Diagram:**

**For user:**



In the above diagram, User first login to the system. System show the dashboard only when the credentials provided by the system is correct otherwise it return to login page. When user is logged in, he can check inbox and reply if there is any message, they can post request, view requests and view donors etc. As they view donor and receiver they can select the receivers and donors and get contact and send the message to receivers and donors automatically by system when selected.

**For Admin:**

****

From the above activity diagram for admin, it is clear that admin has login system as we have for the users. Admins can perform CRUD function on all users details and web contents.

### Sequence Diagram

Sequence Diagrams in UML are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

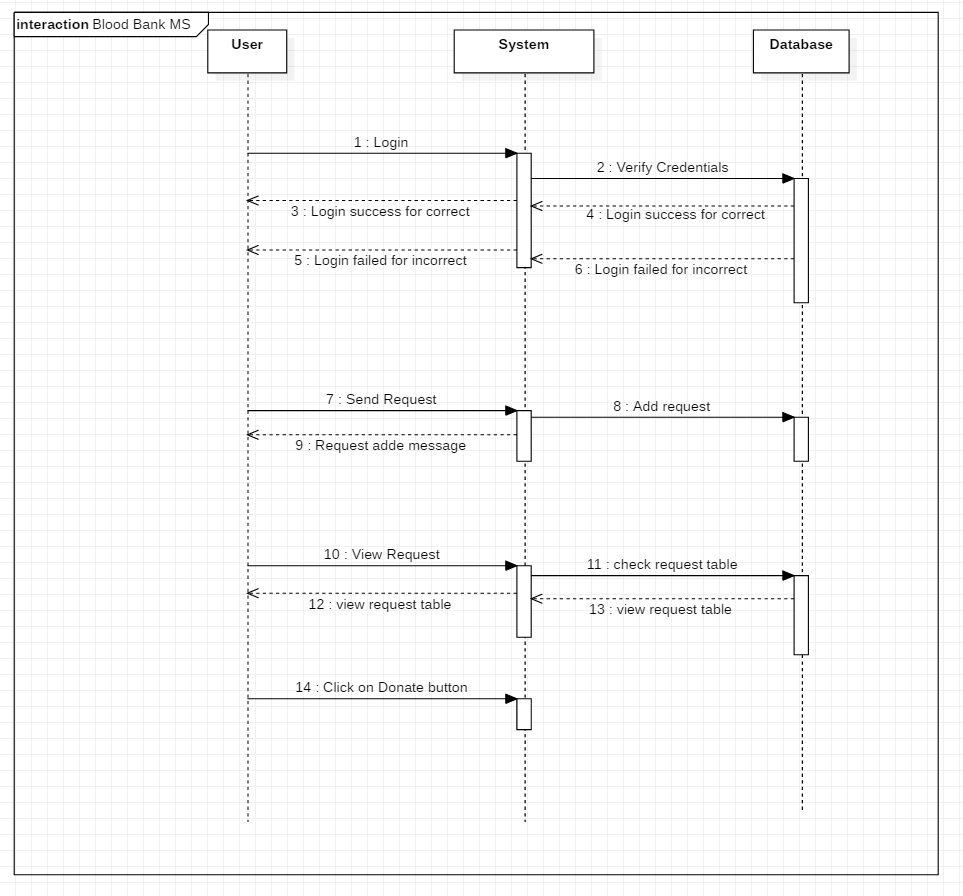
**Justification:**

1. Model high-level interaction between active objects in a system
2. Logic associated with the function or method is modeled.

Notation used:

|  |  |  |
| --- | --- | --- |
| **Notations** | **Name** | **Notation Description** |
|  | Lifeline | A lifeline represents an individual participant in the Interaction. |
|  | Call Message | Message that represents an invocation of operation of target lifeline. |
|  | Return message | Message that represents the pass of information back to the caller of a corresponded former message. |
|  | Self-message | It shows the invocation of message of the same lifeline. |

Diagram:



From the above diagram, it is clear that when user login, they can enter into their accounts only if the credential provide are correct otherwise they cannot. When user send request by filling the form then the system add a request in a request table in database. Users send message to view the requests, system checks in database and respond as database responded and displays the requested table. When user click the button on the request table the status of the request is changed.

## Data Modelling

### Data Dictionary

Donate Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Nullable** | **Key** |
| id | Integer | 10 | Not Null | Primary Key |
| dob | date |  | Not Null |  |
| Email | Varchar | 30 | Not Null |  |
| First\_name | Varchar | 50 | Not Null |  |
| Last\_name | Varchar | 30 | Not Null |  |
| Phone | Varchar | 250 | Not Null |  |
| Occupation | Varchar | 50 | Null |  |
| Provience | Varchar | 200 | Not Null |  |
| Gender | Varchar | 255 | Not Null |  |
| Blood\_type | Varchar | 40 | Not Null |  |
| Zip\_code | Varchar | 255 | Not Null |  |
| Allergies | Varchar | 255 | Null |  |
| Disease | Varchar | 255 | Not null |  |
| Donate before | Varchar | 255 | Not null |  |
| Medication | Varchar | 255 | Not null |  |
| City | Varchar | 255 | Not null |  |
| Created\_at | Date |  | Not null |  |
| Updated\_at | date |  | Not null |  |

Table Host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Nullable** | **Key** |
| id | Integer | 10 | Not Null | Primary Key |
| Org | Varchar | 255 | Not Null |  |
| Title | Varchar | 255 | Not Null |  |
| Email | Varchar | 30 | Not Null |  |
| Time | Varchar | 30 | Not Null |  |
| Dob | Varchar | 50 | Not Null |  |
| Phone | Varchar | 30 | Not Null |  |
| City | Varchar | 50 | Null |  |
| Provience | Varchar | 250 | Not Null |  |
| Zip\_code | Varchar | 250 | Not Null |  |
| Description | Varchar | 250 | Not Null |  |
| Created\_at | Date |  | Not null |  |
| Updated\_at | Date |  | Not null |  |

Table Services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Nullable** | **Key** |
| id | Integer | 10 | Not Null | Primary Key |
| Title | Varchar | 250 | Not Null |  |
| Long\_des | Varchar | 250 | Not Null |  |
| status | Varchar | 30 | Not Null |  |
| Created\_at | Date |  | Not null |  |
| Updated\_at | Date |  | Not null |  |
| Date | Date |  | Not null |  |
| Time | Varchar | 255 | Not null |  |

Table news

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Nullable** | **Key** |
| Id | Integer | 10 | Not Null | Primary Key |
| Created at | Date |  | Not Null |  |
| Updated at | Date |  | Not Null |  |
| Title | Varchar | 250 | Not Null |  |
| Slug | Varchar | 250 | Not Null |  |
| Image | Varchar | 250 | Not Null |  |
| Writer | Varchar | 250 | Null |  |
| Short\_des | Varchar | 250 | Not Null |  |
| List | Varchar | 250 | Not Null |  |
| Sub\_heading | Varchar | 250 | Not null |  |
| About\_title | Varchar | 250 | Not Null |  |
| About\_des | Varchar | 250 | Not null |  |
| Publish\_date | Date |  | Not null |  |
| Status | Varchar | 250 | Not null |  |

Table Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Nullable** | **Key** |
| Id | Integer | 10 | Not Null | Primary Key |
| Username | Varchar | 255 | Not Null |  |
| Email | Varchar | 255 | Not Null |  |
| Email\_verified\_at | Varchar | 255 | Not Null |  |
| Password | Varchar | 255 | Not Null |  |
| Full\_name | Varchar | 255 | Not Null |  |
| Contact\_no | Varchar | 255 | Not Null |  |
| Address | Varchar | 255 | Not null |  |
| Gender | Varchar | 30 | Not Null |  |
| Remember\_token | Varchar | 255 | Null |  |
| Created at | Date | 255 | Not Null |  |
| Updated\_at | Date | 255 | Not null |  |
| Status | Varchar | 255 | Not null |  |

### ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. ER diagrams are used to sketch out the design of a database.

## UI Modelling

### Prototyping

Home Page



Login Form



Register



View Requests

