**Chapter 3: - Design**

The process where user’s need is converted into a suitable form so that it can help the programmer in coding and implementing the software is software design. Designing helps to know the physical and as well as the logical plan of the project. Thus, it helps to develop the software easily and fast.

Online Loan management will be web-based design so that people can surf over the internet for a bank and can provide their information and get loan from the bank.

After completion of analysis we create different types modals and that modals helps to develop the system. Modals helps to make the development process of the software easy and fast. In the same way to develop Online loan management some of the modals are required and they are listed down below:

1. **Structural modelling**
2. **Behavioral modelling**
3. **Database modelling**
4. **UI modelling**
5. **Structural modelling:** The designing of the static part of the system or application is structural modelling. Here the static part of the system or application means the structure of the system which won’t be changed during the development process. Structural modelling contains class diagram and flow chart. Here with the help of class diagram and flowchart I have shown the structure of my system.

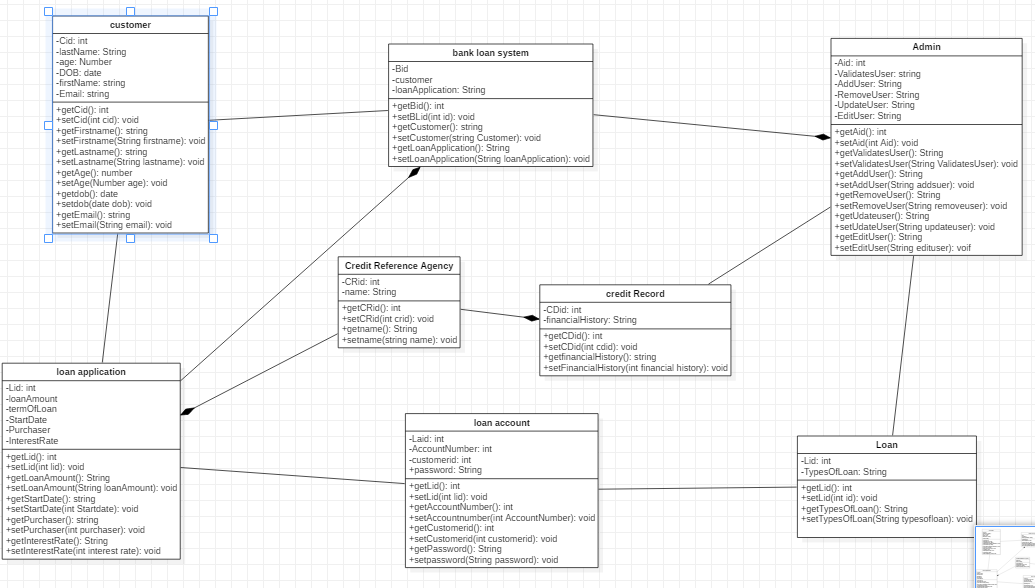
* **Final Class diagram:**

Class diagram is a type of static structure diagram that describes about the structure of a system by showing system classes, attributes, operations or methods and also the relationship among the objects.

As my software is object oriented, classes and interaction between them is most important part and thus class diagram shows the diagrammatical structure of the classes of the system and also shows relationship between them it is important for my system. Class diagram shows the structure of my system by showing their classes, attributes, operation and relationship between them.

**Notation used in class diagram:**

|  |  |  |
| --- | --- | --- |
| **Notation** | **Description** | **Remarks** |
|  | This symbol is used for writing class name, attributes, operations or methods. | Called as class |
|  | This symbol is used for connecting class to one another | Called as composition |
|  | This symbol connects classes | Called as association |



*Fig: Final class diagram*

Here the above diagram shows the structure of a system with the help of the system classes, attributes, operations or methods. The diagram shows the relationship between all the operators. All the classes are according to the class definitions and its properties. There all together ..... classes in the system as shown in the diagram and all the attributes and operations are listed there. the relationship between the classes are shown with the help of aggregation.

* **Flow chart:**

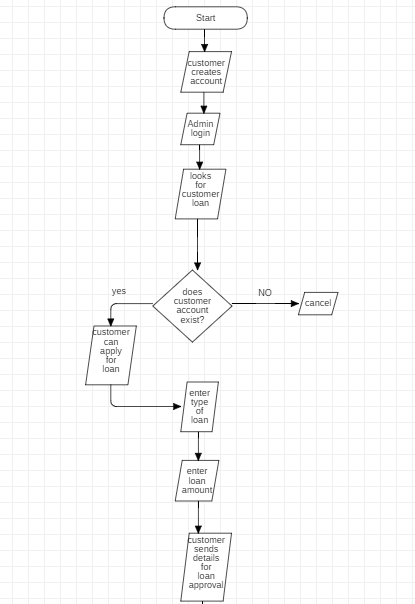
Flow chart shows the system workflow or process. The workflow or process of the software/application is shown diagrammatically. It is also known as the diagrammatical representation of an algorithm. (step-step approach solving a task.)

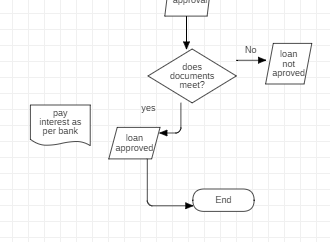
Flowchart helps in the improvement of the processes. It helps the projects teams by identifying the different elements of a process and also helps in understanding the interrelationships among the various steps. Therefore, due this reason I have used flowchart to show the workflow of my system.

* **Notation used:**

|  |  |  |
| --- | --- | --- |
| **Notation** | **Description** | **Remarks** |
|  | This symbol is used for start and end | Called as start/end |
|  | This symbol is used for connecting the step after one another | Called as flow |
|  | This symbol is used for decision making | Called as decision |
|  | This symbol is used for writing data. information. process | Called as data/process |
|  | This symbol is used for writing notes. | Called as notes |

* **Diagram:**





*Fig: flowchart diagram*

Here the above diagram shows the workflow of my system. How it starts and how the process is done and finally the completion of the task. As per the diagram as soon as the system starts the admins logins and then checks the customers who want loan but for that the customer must have an account in the bank. customer selects what type of loan is required by them and enters it. After that admin sends message to the customer and asks for the documents. Customer then provides all the documents asked by the bank to the admin. If the documents match as per the bank request then the admins approve the loan. Then the loan is given as per bank interest rate. The customer sign offs. After that if the customer doesn’t pay interest in time than automatically a message is send to the customer as a reminder. Finally, after that admin can sign off.

1. **Behavioral modal:** this modal shows the behavior of the system through diagrams. In software development behavioral modal mainly focuses on the logical part of the system. Here with the help of the two behavioral modal i.e. Activity diagram and Sequence diagram I have shown the logical flow of my system.

* **Activity diagram:**

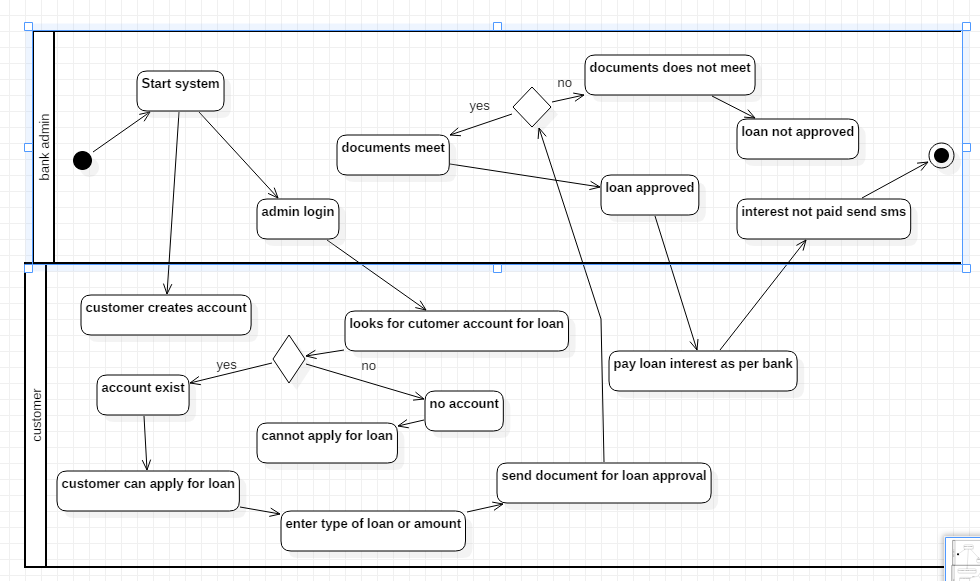
Activity diagram is graphical representation of pseudo codes. It is a type of behavioral diagram in UML diagram that shows dynamic aspects of the system. It basically shows the series of action and the flow of control in a system. It is also called as the advanced version of the flowchart.

Activity diagram helps to show the message flow from one activity to another. Also, it helps to capture the dynamic behavior of the system so I have used activity diagram.

**Notation used:**

|  |  |  |
| --- | --- | --- |
| **Notation** | **Description** | **Remarks** |
|  | This symbol is used for starting system | Called as initial |
|  | This system is used for writing data, information/process | Called as action |
|  | This symbol is used for connecting activity with one another | Called as flow |
|  | This symbol is used for decision making | Called as decision. |
|  | This symbol is used for termination of system | Called as termination |

* **Diagram:**



*Fig: activity diagram*

Here the above diagram shows the message flow of one activity to another. The dark dot represents the start of the system. The arrows show the relations between the activity. As per the diagram as soon as the system starts the admins logins and then checks the customers who want loan but for that the customer must have an account in the bank. customer selects what type of loan is required by them and enters it. After that admin sends message to the customer and asks for the documents. Customer then provides all the documents asked by the bank to the admin. If the documents match as per the bank request then the admins approve the loan. Then the loan is given as per bank interest rate. The customer sign offs. After that if the customer doesn’t pay interest in time than automatically a message is send to the customer as a reminder. Finally, after that admin can sign off.

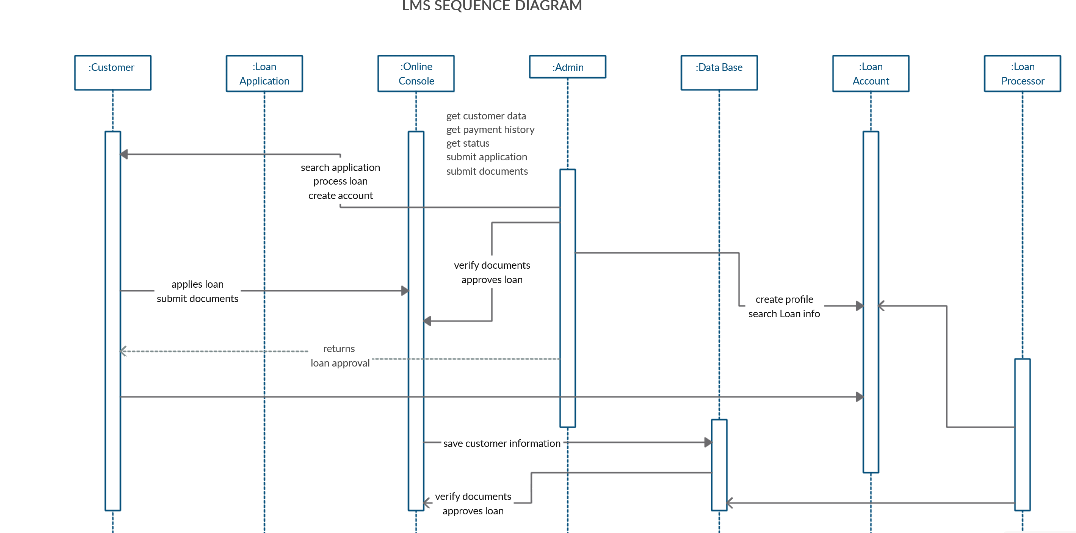
* **Sequence diagram:** This diagram shows the object interaction arranged in time sequence. It is one of the types of UML diagram that shows the interaction and detail in how operations are carried out. They show the interaction visually by using a vertical axis of diagram to represent time like when the messages are sent.

As it shows the high-level interaction between active objects in a system and also interaction between objects with collaboration that realizes a use and an operation it is important and thus, I have used it.

* **Notation used:**

|  |  |  |
| --- | --- | --- |
| **Notation** | **Description** | **Remarks** |
|  | This symbol is used to write actors name | Called as lifeline |
|  | This symbol is used to connect and write message | Called as message |
|  | This symbol is used to reply message. | Called as reply message |

* **Diagram:**



*Fig: sequence diagram*

Here the above diagram shows the object interaction arranged in time. The diagram shows the details on how the operations are carried out and at what time the messages are sent and what messages are sent. Here the objects involved in the operation are listed from left to right according to when they take part in the message sequence. According to diagram the customer applies for loan and sends document and the admin verifies the document, the online console saves the customer data/information and if documents meet loan is granted and interest is paid as per bank.

1. **UI modal**

* **Database dictionary:** Data dictionary shows the set of tables that describes the data which we will be going to use in particular database system. Basically, it is collection of entities with its properties. It also provides the maximum length of data. The database dictionary table consists of attributes, data type, length, constraint and description. It is also called as data of data.

**Down below are the tables that I have made for data dictionary.**

**For customer table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Data type** | **Length** | **Constraint** | **null** | **description** |
| 1. | Customer\_id | Int | 10 | Primary key | No | This is primary key for customer table |
| 2. | Customer\_first\_name | Varchar | 10 |  | Yes | First name of customer |
| 3. | Cutomer\_last\_name | Varchar | 10 |  | Yes | Last name of customer |
| 4. | Customer\_address | Varchar | 10 |  | Yes | Address of customer |
| 5. | Customer\_password | Varchar | 10 |  | Yes | Password for customer |
| 6. | Customer gender | Varchar | 10 |  | Yes | Gender of customer |
| 7. | Customer phn | Varchar | 10 |  | Yes | Phone no. of customer |

**Table for admin:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Data type** | **Length** | **Constraint** | **null** | **description** |
| 1. | Admin\_id | Int | 10 | Primary key | No | This is primary key for admin table |
| 2. | Username | Varchar | 10 |  | Yes | Name for admin |
| 3. | Password | Varchar | 10 |  | Yes | Password of admin |
| 4. | Send SMS | Varchar | 100 |  | Yes | Interest not paid then send SMS |

**Table for bank:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Data type** | **Length** | **Constraint** | **null** | **description** |
| 1. | Bank\_id | Int | 10 | Primary key | No | This is primary key for bank table |
| 2. | Customer\_id | Int | 10 | Foreign key | No | This is foreign key from customer table |
| 3. | Customer\_details | Varchar | 10 |  | Yes | Details of customer from customer table |
| 4. | Customer account | Int | 10 |  | No | Account number of customers |

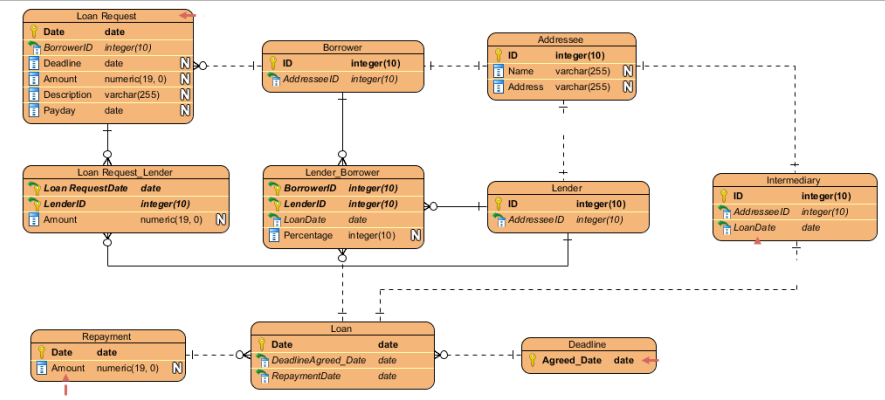
**Table for loan:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Data type** | **Length** | **Constraint** | **null** | **description** |
| **1.** | **Loan\_id** | **Int** | **10** | **Primary key** | **No** | **Primary key for loan table** |
| 2. | Customer\_first\_name | Varchar | 10 |  | Yes | First name of customer |
|  | Cutomer\_last\_name | Varchar | 10 |  | Yes | Last name of customer |
|  | Customer phn | Varchar | 10 |  | Yes | Phone no. of customer |
|  | Customer salary | Varchar | 100 |  | Yes | Salary of customer |
|  | Customer citizenship no. | Int | 10 |  | No | Number of citizenships |
|  | Customer property details | Varchar | 200 |  | Yes | All details of property |

**Table for payment:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Attribute | Data type | Length | Constraint | null | description |
|  | Payment\_id | Int | 10 | Primary key | No | Primary key payment table |
|  | Date | Varchar | 10 |  | Yes | Date of payment |
|  | Amount | Varchar | 100 |  | Yes | Amount paid |

* **ER diagram:** Entity relationship shows the entity types and specifies the relationships that can exist between entities. It shows the relationship of entities stored in database. Entity is composed of object and component of data. Entities can have attributes that describes its properties. Being based on this actual database system is designed in a system during development process.



*Fig: ER diagram*

1. **Prototype:**

Prototype is basically a model/demo/sample of a project or system. It gives the overview of the system or project like how it is going to look. It helps to know to how the system is going to look after it is built.

Here for online loan management I have used balsamiq tool to demonstrate on how my project will look.

* Down below are samples/prototype of online loan management.

