**Acknowledgment:** Online loan management is web-based application created to take loan online by submitting documents online. This system is developed to reduce manual data entry and to save the time of the client and bank as the client does not have to go to bank for loan approve.

**Abstract:** I have developed a web-based system named Online Loan management which allows the user to ask for a loan online. It is PHP Laravel based site. This system saves the time of user as they don’t have to go to the bank and ask loan, they can simply register to the system ask for the type of loan required and also there’s no need to fill form, the user can simply upload all their documents can ask for loan. After all of this if the documents match to the requirements of the bank the loan will be approved. The approved sign will be shown after the user login to the system. After the loan is granted and the user doesn’t pay the debt or interest on time then an email is sent to them as a reminder of loan payment. The user themselves can update their data and delete their data only admin can reject the loan if the documents doesn’t match the criteria of the bank. This system will reduce manual data entry and will provide great efficiency.

# **Chapter -1**

## **Introduction:**

* The platform is referred to as an electronic loan management system. This system is designed specifically to keep the loan customer data. This keeps records of the customers who have borrowed from a bank. So, I have developed PHP Laravel based software which will help the customers and bank. This system will reduce manual data entry and will provide great efficiency.

## **Background of the system:**

* Filling up the form while taking loan from the bank, people find it difficult. Many people find difficult to fill up the form so I have created a software they just they need to upload their documents which will also save time of both the parties. And the loan customer data can’t also be kept in paper in the bank, it might get lost. Manual data entry doesn’t provide great efficiency. So, in order to reduce manual data entry, I will create this project. Also, it will provide confidentiality to the data of customers.

Thus, the main goal of this project is to provide all the features to bank and the user and also reducing the time taken to write the customer details and provide great efficiency.

## **Overview of the project:**

* Here the system will have two main interfaces. One will be for customers and other will be for the user (admin) which will manage all the bank account. Admin will be one who will verify the customer. There will be only one account of admin and other all will be of customers or users. Admin can add or delete the account of users. The admin will have to keep the loan details. Like for example how many instalments has been paid by the user and how many are left, how many installments were not paid by the customer in the past etc. when will the loan be over and what kind of loan is taken by the customer. If the loan is not paid on time then the admin sends mail to the user as a remainder to pay the loan.

## **Justification of project:**

* Here I have proposed this system because with the help of this system people can easily take loan from the bank and data of the loan customers can be backed. It will also decrease the amount of time taken to write customer’s details and other modules. This system will also be helpful in terms of reducing manual data entry and will provide great efficiency. The user interface of the system will be very friendly and can be easily used by anyone. People run to the bank with all their documents to take loan from the bank and this take much time and as my system is online users can ask for loan online by submitting all their documents and can be notified online after their loan is approved. Basically, my system works online and It saves time as everything is done online and people don’t have to go to bank. So, I have developed Online Loan management system.

## **Amis of the project:**

• To reduce the manual data entry and provide great efficiency.

• To develop the software in minimum cost and provide all the features.

• To reduce the time taken to write the customers details and other modules.

• To help the user work with bank and their branches.

• To notify the user (sending mail) about the loan payment according to their duration.

## **Objectives of the project:**

• To make the user interface of the system very friendly so that it can be used by anyone.

• To record the loan customer details in systematically way.

• Backing up the data so that it will never be lost.

• Notifying the user about the loan payment through mail.

• Data of the customer will be kept confidential.

• To keep the website secure and functional.

# **Chapter 2: - Analysis**

* Analysis is the systematic way of examining and evaluating data or information, by breaking it into components parts to uncover their interrelationships. For the development of any system analysis is the most important part. Without analysis the requirements of the system or without knowing anything about the system, if the program starts there is high chance that the system will fail. Different research is done before starting any projects. Basically, analysis is the process of breaking a complex topic or substance into smaller parts in order to gain better understanding of the project.

Analysis is done in order to get better understanding of the project. It helps to keep the biasness away from research conclusion with help of statistical data. It also helps to break information into components parts.

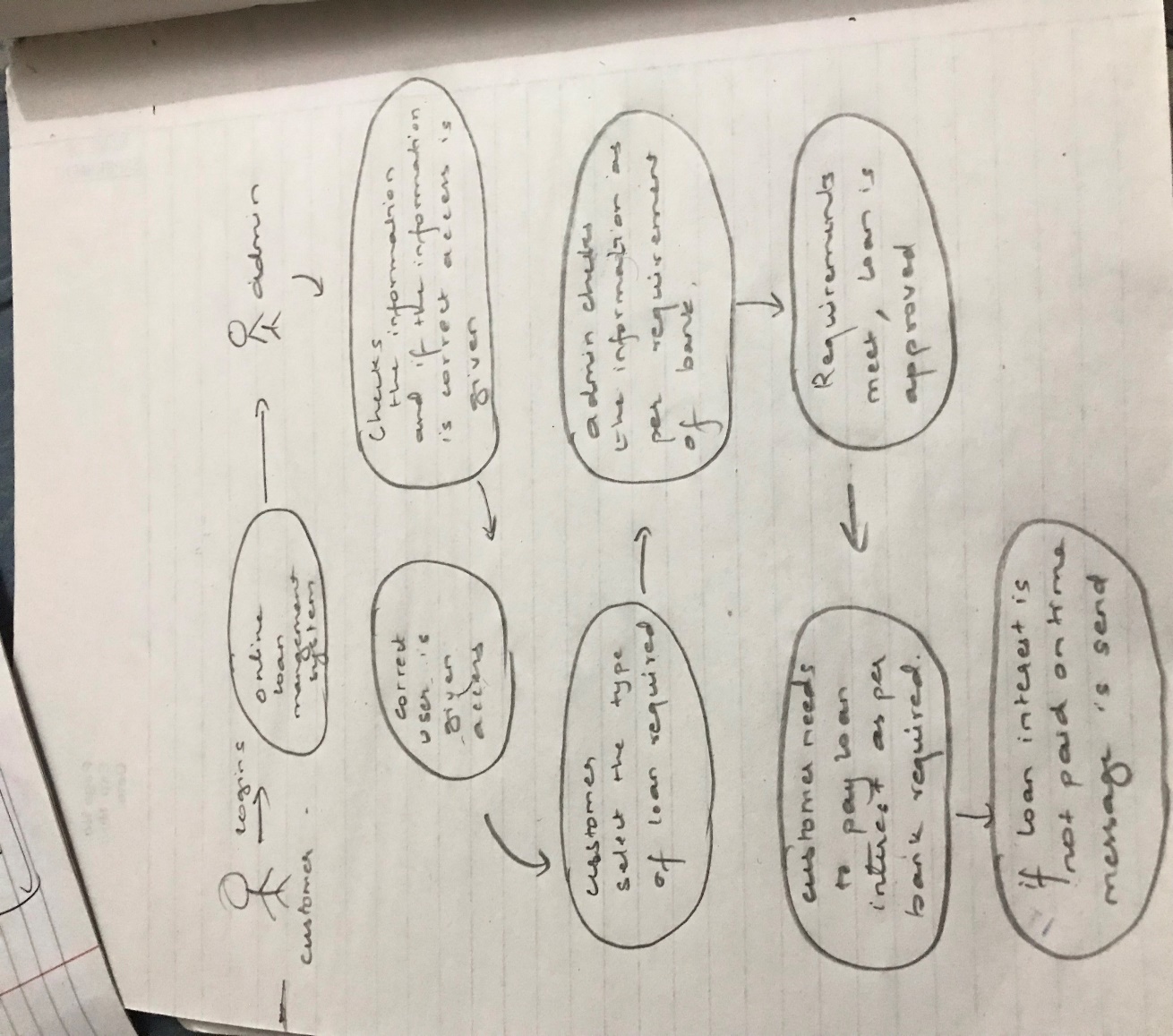
Since my system will be providing loan to people as per their need and requirement so for analysis methodology, I will be using soft approach as it focuses on people need’s and requirement.

There are certain steps that should followed while using soft approach:

1. Analysis of existing information and the generation of rich images.
2. Definition of the root definition of major information system components.
3. Conceptual production of models. Comparing the concept of the system with actual system.
4. Defining an implementing the system.

## **Rich picture:**

Rich picture represents view of the whole system and can enable better planning and understanding of a system. It is basically drawn by hand and includes structures, processes, issues or developments. Here down below is the rich picture of my system to know how the system functions or works.



*Fig 1: Rich picture*

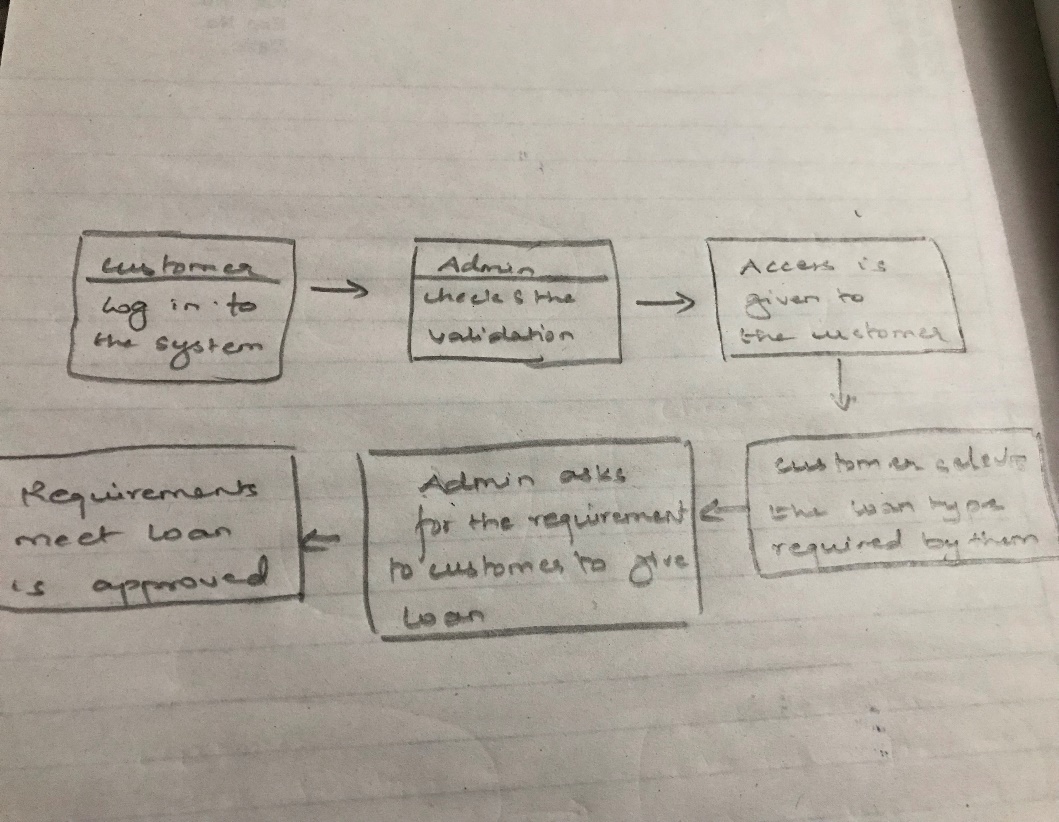
## **Root definition:**

* It helps to clarify the system processes and any problems. It is the short textual statements which describes the aims and functions of the potential system. As per my project admin registers the user information to the system and then only the user’s logins to the system and ask for loan. After all that the bank checks the user information’s and with all the necessary requirements loan approval is given to the user.

## 

## **Conceptual modals:**

* The analyst uses the rich picture and root definition to construct a conceptual ‘ideal’ system that defines. This conceptual model can be used to describe how the system should function and what activities are necessary for the processes to take place.

****

*Fig 2: Conceptual modal*

## **Comparing the concept of the system with actual system**.

* Differences between the actual system and the model are noted and discussed with management. Required developments of the existing system are discussed. Necessary and feasible solutions are agreed and new system is implemented.

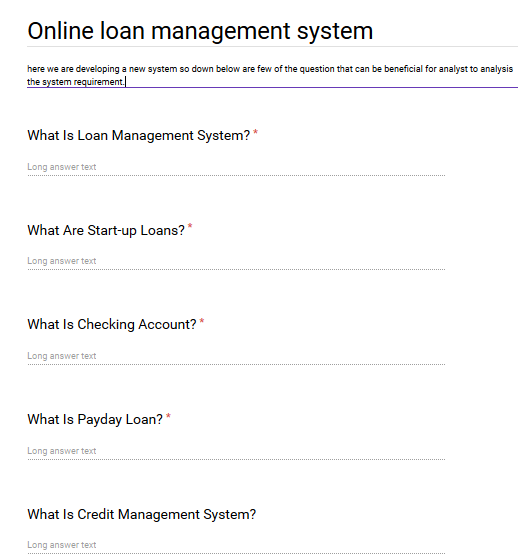
## **Information gathering method:**

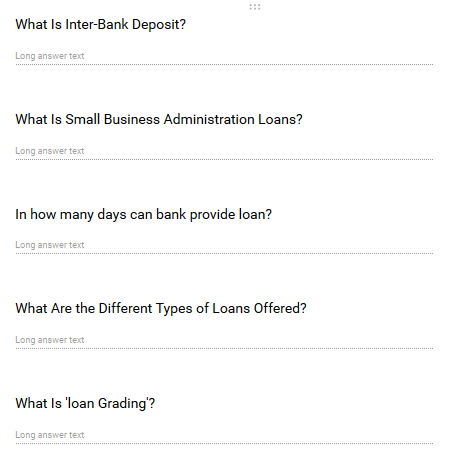
* Information gathering method falls under the analysis. Here we gather the information for the development of the project and as per plan the system is built. Information can be collected from different sources like questionnaire, survey, interview e.tc. And here for my system I will be using questionnaire system for gathering the information.

## **Questionnaire:**

Here in this method different question are prepared and ask to the group of people so that different views/opinions can be collected and system can be built according to it. The question should simple so that the person giving the answer doesn’t feel bore to answer. it should simple so that analyst can get more information.

Here down below are some of the question asked:





* **Feasibility study:** The Studies are known as the Feasibility Study to determine the advantages or disadvantages, practicability, or ability to accomplish a projected plan, study, or project. The outcome of my studies is-
* Technical Feasibility.
* Economic Feasibility.
* Operational Feasibility.
* Legal Feasibility.

## **Technical Feasibility:**

* Technical feasibility is related to the hardware and software specification required by the system to function properly. For my system the basic requirements are That the system needs at least 2 GB of ram to run smoothly and suddenly all the features. It requires a minimum 1.3 GHz processor to run smoothly as small as this can cause problems and computer must be connected to the internet.

## **Economic Feasibility:**

* Economic feasibility determines the cost effectiveness of a system. As my system is not so big and it does not require anything rather than a normal computer which must be connected to the internet, it can be cost effective. It does not require huge capital to build the system.

## **Operational Feasibility:**

## Operation Feasibility decides whether the proposed solution within the current management and organizational structure is desirable. It is quick to use the Loan Management System. To operate this system, the user does not need any expensive training. The new users, however, must have knowledge of computer operation, particularly on the Windows platform. As with any new software at the beginning

## **Legal Feasibility:**

* It is related to the ethics. Before starting or implementing any system ethical things should be kept in mind. Here in case of my project that is loan management system there is no any conflicts or any legal conflicts that affects the ethics and harm the system.

# **System requirement specification**

## **Functional requirement:**

* The functional requirement of the system describes what a software should do. It specifies what function or the components of the system can be able to perform.

Following are the functional requirements of the system:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function ID** | **Title** | **Description** | **Purpose** | **Dependency** |
| F1 | Registration | New users get registered by providing their information | To get the features of the system. | F2 |
| F2 | Login | Registered user only can get the access of the system | To use the system in secure way. | F1 |
| F3 | Selecting loan | Users gets access to choose the loan they want. | To select loan. | F2 |
| F4 | Check loan | Admin checks the loan if they can approve or not | To manage the loan securely. | F3 |
| F5 | Delete | If the requirement does not meet, loan does not gets approve | To maintain the loan system | F4 |
| F6 | Send email | Admin sends email to the user if loan interest is not paid on time | To run smoothly the loan interest | F2 |
| F7 | Remove user | If the user is fake admin can remove that. | To manage the user of system | F2 |
| F8 | Edit | If any mistake occurs while inserting data, it can edit. | To down the mistakes | F2 |
| F9 | Update | To update user information if require. | To manage the user system | F2 |
| F10 | Logout | After completion user or admin can log out | To maintain security. | F1,F2 |

Fig 3: functional requirement table

## **Non-functional requirement of the system:**

* The requirement that specifies the criteria that can be used to judge the operation of a system rather than specific behavior.it is also called quality attributes of the system.

Here down below are the non- functional requirement of the system:

|  |  |  |  |
| --- | --- | --- | --- |
| Nonfunctional requirement ID | title | description | Purpose |
| NFR1 | Usability | The system interface should be easy to use i.e. it should user friendly. | to make the system easy to use for the user. |
| NFR2 | Availability | The system should be able to access as per user required. | User can get access anywhere or at any time they want. |
| NFR3 | Reliability | The system should be reliable so that user can rely on it without any hesitation. | Make more user. |
| NFR4 | Security | System should not allow any vulnerabilities to enter as it may harm the user information | For privacy purpose. |
| NFR5 | Documentation | Explains what the system is capable to do. | To know the capability of the system. |
| NFR6 | Security | Validation of user email and password | To maintain data security. |
| NFR7 | Performance | The performance should be fast whithout altering user. | Insert, update, delete of user Should be performed fast. |
| NFR8 | Efficiency | Measures the speed or performance of the system. | Performance should be functional. |
| NFR9 | Response time | Time taken to perform the task | Response of the system towards the user should be quick |
| NFR10 | Quality | Functional requirement of the system should be fulfilled. | End product should be fully functional. |

*Fig 4: Nonfunctional requirement*

# **Hardware and software specification:**

Hardware components and software components are necessary to run any system without this no system can run.

The basic hardware and software requirement to run the system is mentioned down below:

|  |  |
| --- | --- |
| **Hardware** | Software |
| Minimum of ***2 GB*** of ram to run all the features smooth and sudden. | Operating system: Windows 2007 to Windows 2010 |
| **1.*3 GHz*** processor | Firefox, internet explorer, chrome |
| ***128 GB*** of hard disk. | Database: php MySQL |

### *Fig5: Specification*

# **MoSCoW Prioritization**

* it is one of the best prioritization techniques for managing requirements.

Here the MoSCoW stands for;

M: Must have (without this project cannot run)

S: Should have (must haves over the long run)

C: Could have (low cost tweaking)

W: Won’t have (not necessary)

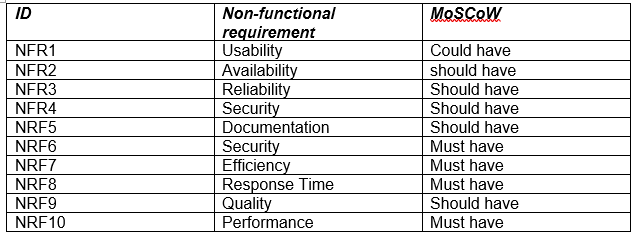
1. **Must have:** it represents the things that a system must have, without this a system cannot be developed properly. A release product becomes useless without work of an initiative and the initiative is most likely a “must have”.
2. **Should have:** the initiative is below must have. They play an important to develop a product but they are not vital. Even if the things in should have are not included in the system the system will work. However, if it is included, they provide significant value to the product.
3. **Could have:** The initiative is not necessary to develop the function of the system. There will be less impact in the system even if it is left out. It is nice to have could have initiative on the system. The initiative in could have are the first one to deprioritized.
4. **Won’t have:** Here the initiative is not necessary while releasing a product. Even if won’t have initiative are included in the system there won’t much impact in the system. But sometime it can be beneficial in the future.

Here down below is the MoSCoW Prioritization for my system:

**Functional requirement:**

### *Fig 6: Moscow functional requirement*

**Non-functional requirement**

*Fig 7: Moscow nonfunctional requirement*

* **System architecture:**  Here the system architecture describes conceptual model which explains about the behavior, structure and view of the system in diagrammatic form. As to develop the system I have choose Laravel with PHP core, this architecture shows the use of PHP script. As my system is web based this architecture is best one to show how the system functions overall. This architecture shows through the use of browser PHP script is executed. Also, in my system all data are stored in database server and even the architecture also explains the same. Architecture is required to see the process architecture, incorporate plan-based layout, and control the system's performance and requirements.

HTML

Browser

Desktop

PHP script

Web server

HTTP SQL

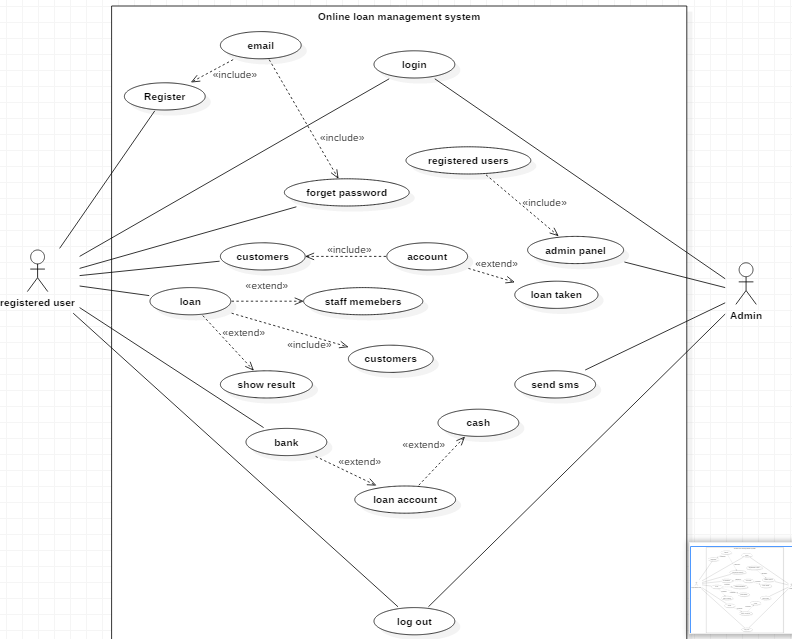
Database

Database server

HTML tables

*Fig 8: System architecture*

# **Use-case diagram:**



*Fig 9: Use-case diagram*

* **Name:** Online loan management system
* **Participating actor**: Registered user and Admin
* **Entry condition:**
* Customer must have an account in the bank database to get the access.
* Customer must have all requirements fulfilled in order to get loan.
* **Event flow**:
* User registers to the system using email and password.
* Admin verifies the user and give access.
* Admin can access his profile and as well as he can search users and can see the detail.
* Admin can only add user and give access.
* Admin can manipulate data as per system requires.
* Customer search for loan they need.
* Admin asks for the information as per bank requires and approves the loan if all the requirements matches.

* **Exit condition**
* Loan is approved by the bank.
* **Exceptional case**
* If bank requirement is not fulfilled by the customer then loan can’t be approved.

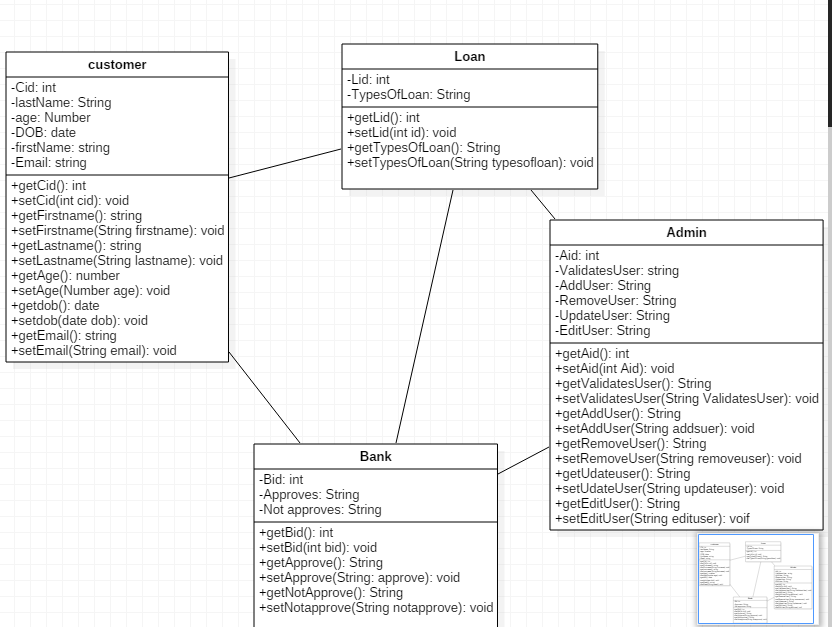
# **Candidate class list and Diagrams:**

**Scenario:**  the system name is online loan management system. The main purpose of this system is to maintain the record of the customers who have taken loan from the bank. this system is designed in way that the data are kept confidentially. Customers must have an account in the bank in order to get loan. customer must register and then only they can get access. Admin can only add update delete the user information. New and unique id is given to customer whoever take the loan from bank. the customer logins in to the system and selects the type of loan want by them after that admin asks for the information and if only the information matches the requirements of the bank the customer can take loan. And if it does not match loan is not approved.

# **NLA/initial class diagram:**

NLA is basically done to find out the possible classes from the given paragraph by following NLA rule. Main goal of NLA is to find out the candidate class / actual class.

|  |  |  |
| --- | --- | --- |
| Noun | Verb | attribute |
| Customer | Add  Remove  Update  Edit | CId  Name  Email  Password  D.O.B  Address  Age |
| Admin | validates | Aid  Add user  Remove user |
| Bank | approve | Bid  Approves  Not approve |
| loan | Loan want | Lid  Types of loan |



*Fig 10: Class diagram*

# **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 models 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.

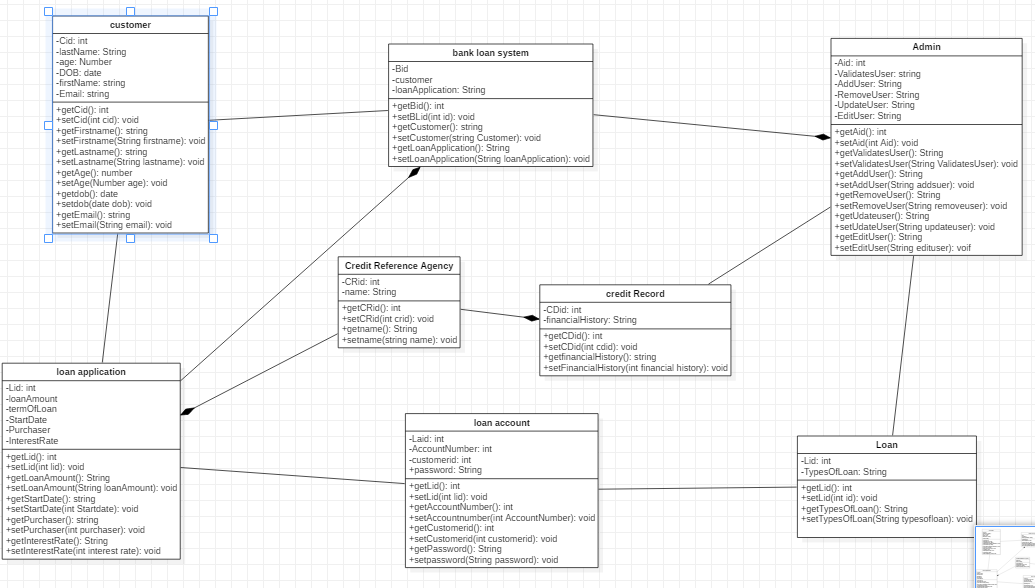
## **Final Class diagram:**

Class diagram is a type of static structure diagram that describes the system structure by showing system classes, attributes, operations or methods as well as the object relationship.

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 11: 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:**

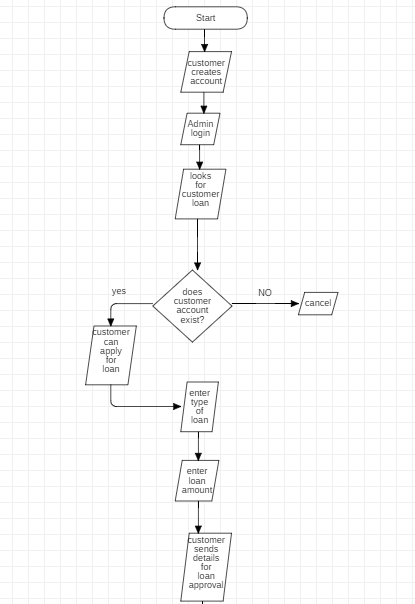
Flow chart shows the system workflow or process. The workflow or process of the software/application is shown diagrammatically. It is also known as the diagrammatical representation of an algorithm. (step-step approach solving a task.)

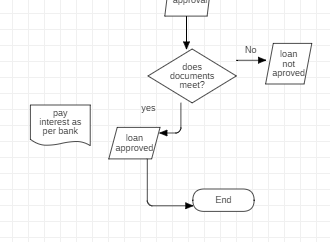
Flowchart helps in the improvement of the processes. It helps the projects teams by identifying the different elements of a process and also helps in understanding the interrelationships among the various steps. Therefore, due this reason I have used flowchart to show the workflow of my system.

* **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 12: 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:**

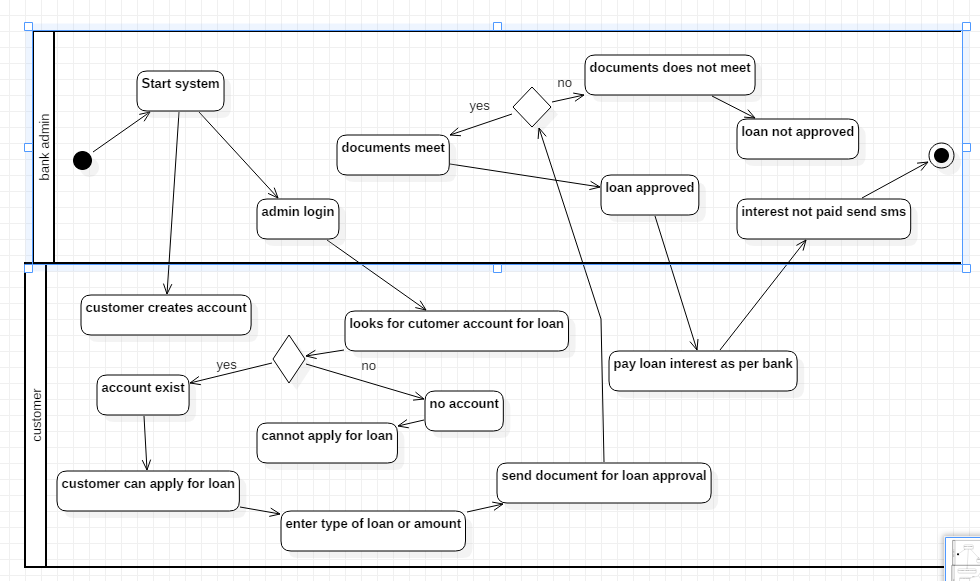
* Activity diagram is a pseudo-code graphic representation. It is a type of behavioral diagram showing dynamic aspects of the system in the UML diagram. It basically shows the series of action and the flow of control in a system. It is also called as the advanced version of the flowchart.

Activity diagram helps show the flow of message from activity to activity. It also helps capture the system's dynamic behavior so I used the diagram of activity.

**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 13: 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.

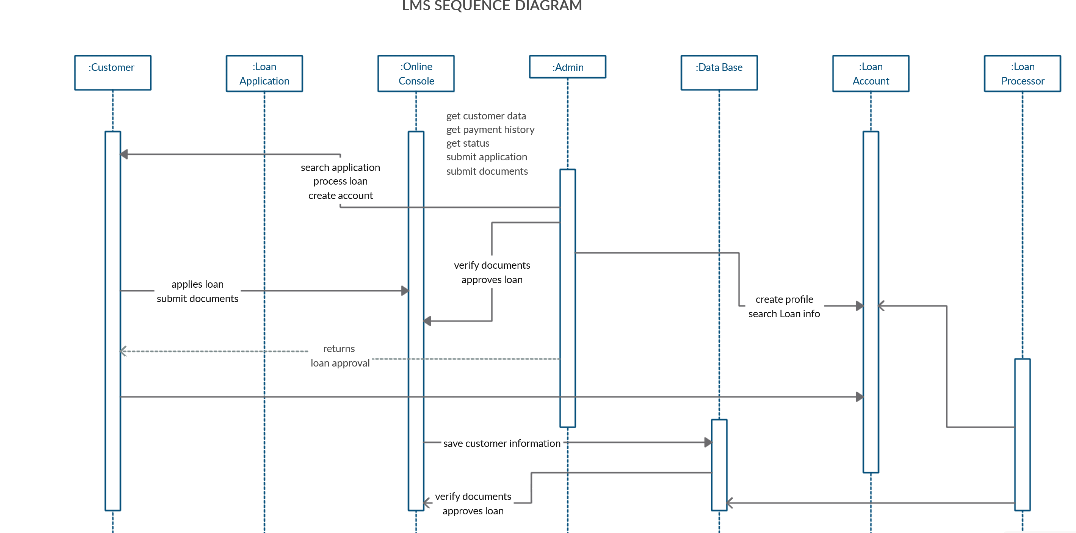
* **Sequence diagram:** This diagram shows the object interaction arranged in time sequence. It is one of the types of UML diagram that shows the interaction and detail in how operations are carried out. They show the interaction visually by using a vertical axis of diagram to represent time like when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



*Fig14: 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.

# **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 |

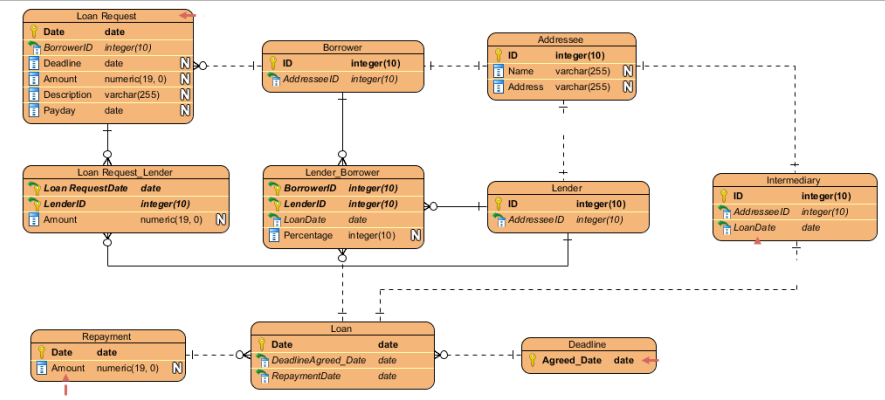
**Table for loan:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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.



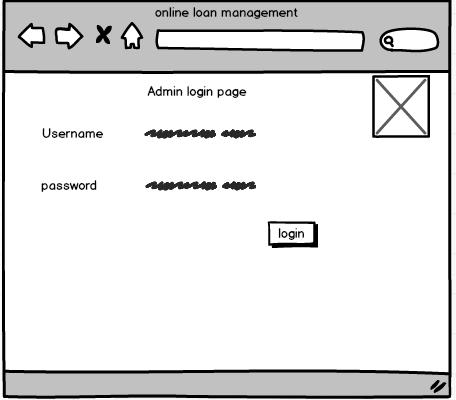
*Fig 15: ER diagram*

# **Prototype:**

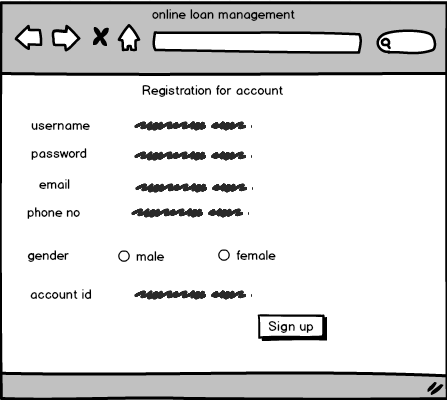
Prototype is basically a model/demo/sample of a project or system. It gives the overview of the system or project like how it is going to look. It helps to know to how the system is going to look after it is built.

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

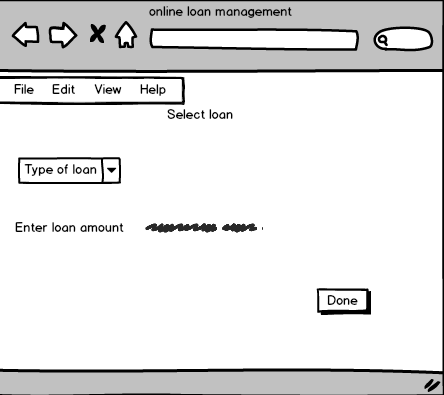
* Down below are samples/prototype of online loan management.



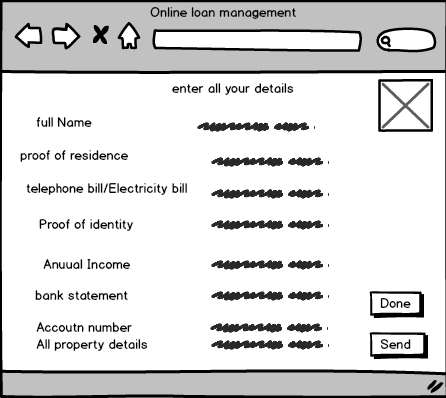
*Fig 16: admin page*



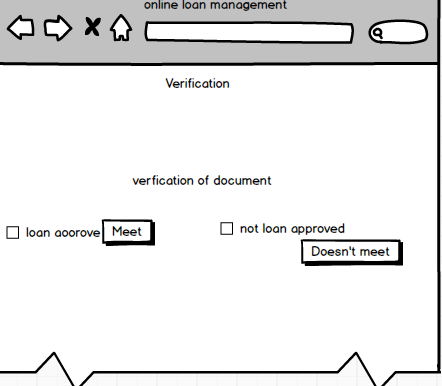
*Fig 17: registration page*



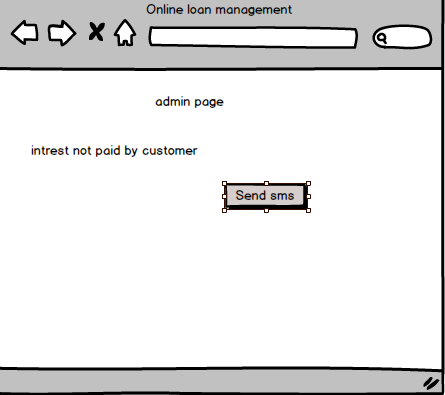
*Fig 18: types of page*

**

*Fig 19: Customer page*



*Fig 20: approve page*



*Fig 21: message page*

# **Chapter 4: Implementation:**

* It processes of putting a plan into work. It includes the sector of front-end user and back end user. Designing of database, user interface, coding and validation lies in this phase of the system. Putting a plan into work is implementation.

# **Chapter 5: Testing**

* Software testing is a set of processes that is done to aim to know the quality of a computer software. It ensures the compliance of a software product. Testing of a software is done basically to know the quality, performance and reliability of a product. It helps to find the errors, gaps or missing requirement in relation to the actual requirements of the software. In simple words software testing means verification of application under test.

There are different types of testing with which the software can be tested like unit testing, integration testing, system testing, sanity testing, smoke testing, interface testing, regression testing, beta/acceptance testing e. t. c and for testing mine software I have used black testing and unit testing.

**Types of testing:**

## 1.Black box testing

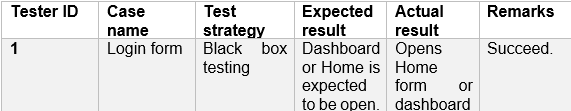
## 2. Unit testing

* **Black box testing:** Black box testing is a software testing where internal structure or design or implementation of a item is tested.it is also known as Behavioral testing. These tests can be functional ore nonfunctional though mainly it is functional. This testing is done to know whether the software functions properly or not.it ensures the functionality of a software. This research lets the user know what to do with the code and what to do after one. It’s success is like model testing.
* **Reasons for doing black box testing:**

1. It ensures the functionality of a software.
2. Tests the design or implementation of an item.
3. Shows behavior or performance errors.
4. Shows interface errors.

Here down below are the black box testing the I have performed to test my system:

1.

 Fig 22: login table

* **After admin login with his/her email and password, dashboard opens where admin can view all the customer:**

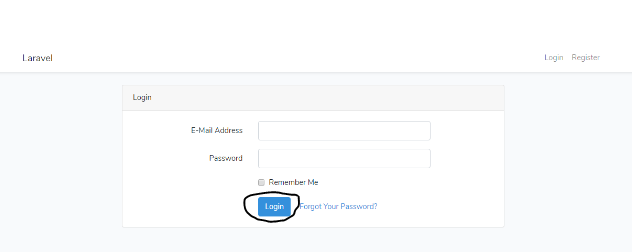


Fig 23: login form

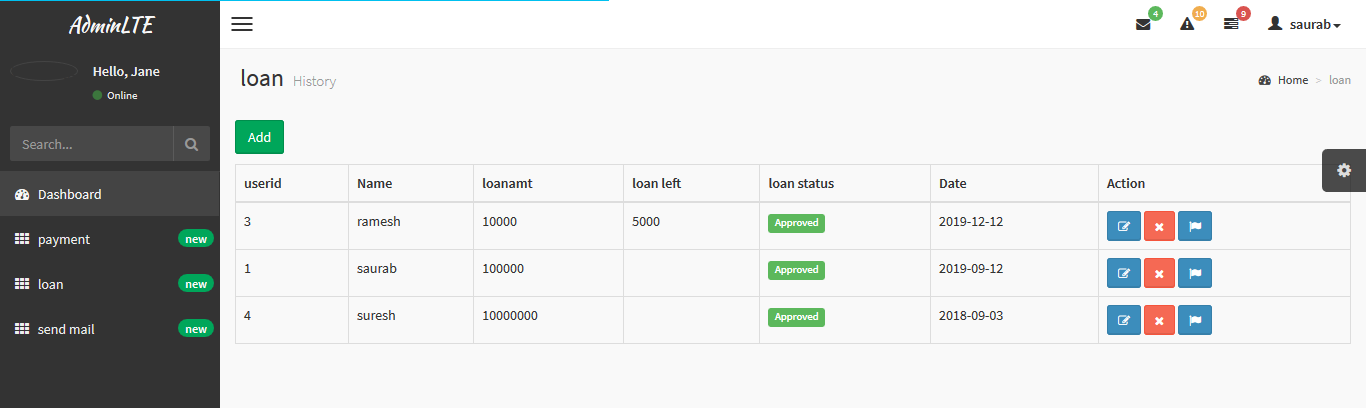


Fig 24: loan form

**2.**

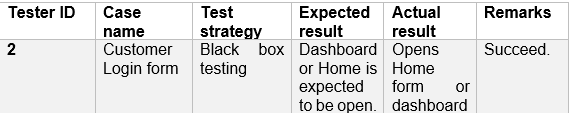
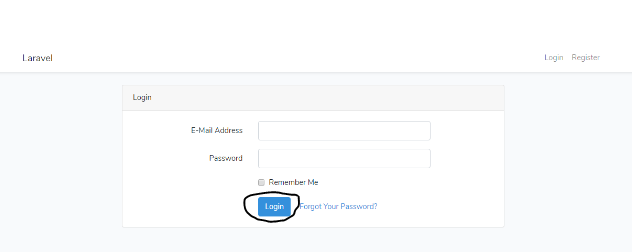
* 
* **After Customer logins a dashboard open where he/she can view his/her records only.**   
  

Fig 25: login form

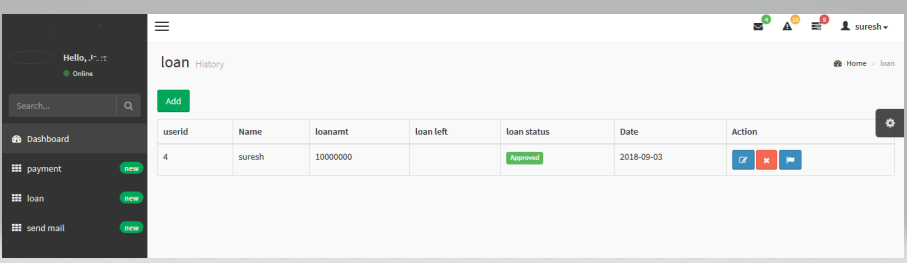


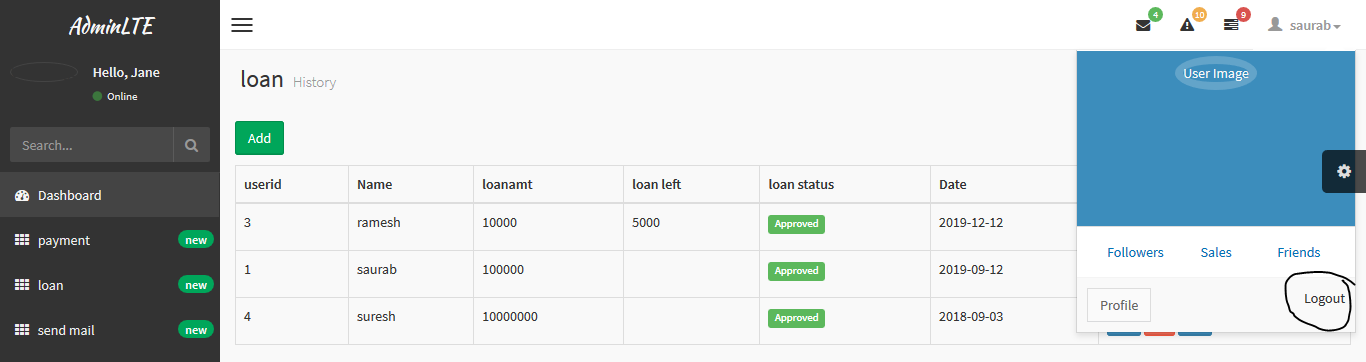
Fig 26: loan form

**3.**

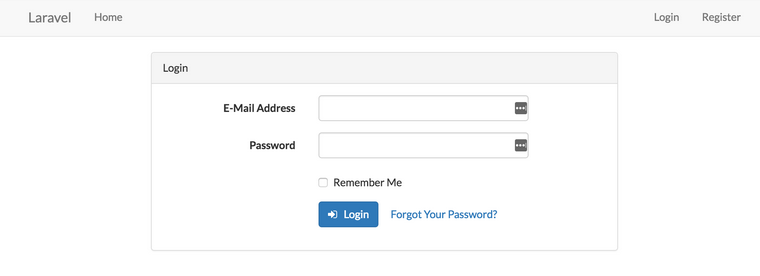
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 3 | Logout | Black box testing | Login form appears | Opens login form | Succeed. |

*Fig 27: logout table*

**When logout button is clicked login form appears:**



*Fig 28: logout form*



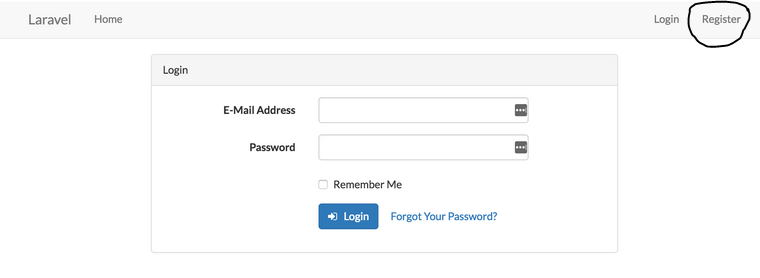
*Fig 29: login form*

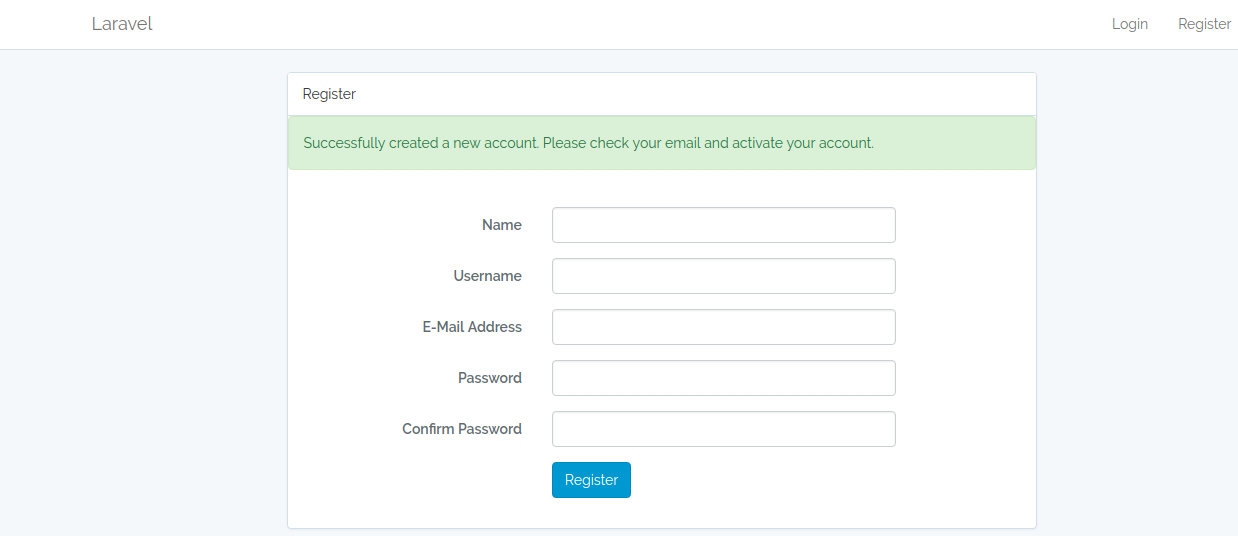
4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 4 | Registration | Black box testing | Registration form appears | Opens registration form | Succeed. |

*Fig 30: registration table*

* **When registration button is clicked, registration form opens:**





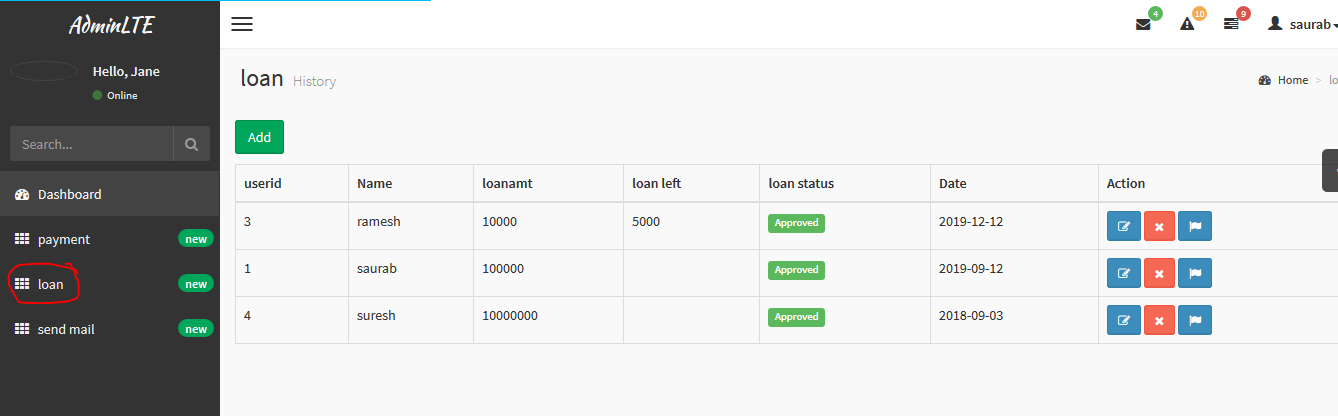
*Fig 31: registration form*

**5.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 5 | Loan | Black box testing | loan form appears | Opens loan form | Succeed. |

*Fig 32: loan table*

* **When loan button is clicked by admin, he/she can view the users who have taken loan.**



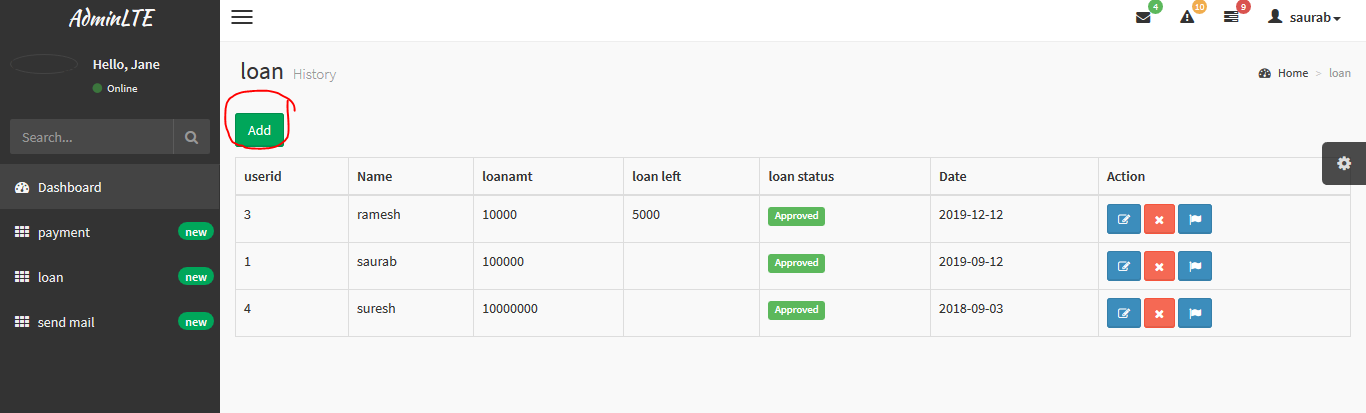
*Fig 33: loan form*

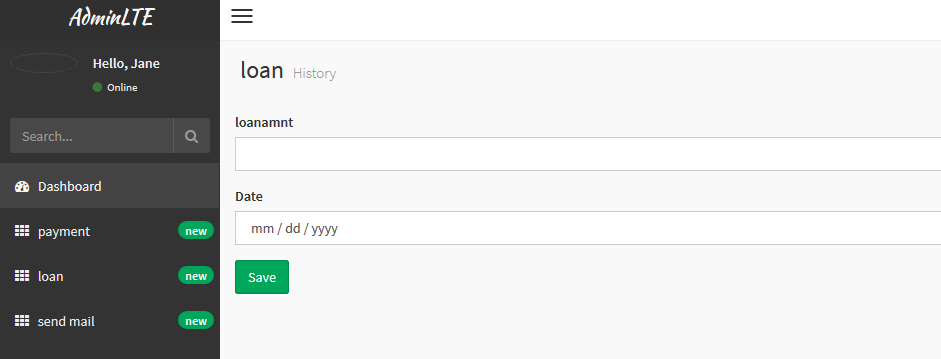
**6.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 6 | Add loan | Black box testing | Loan add form appears | Opens loan add form | Succeed. |

*Fig 34: Add loan table*

* **When add loan is clicked, loan form opens:**





*Fig 34: add loan form*

**7.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 7 | Payment | Black box testing | payment form appears | Opens payment add form | Succeed. |

Fig 35: payment table

* **When payment button is clicked, payment form opens:**

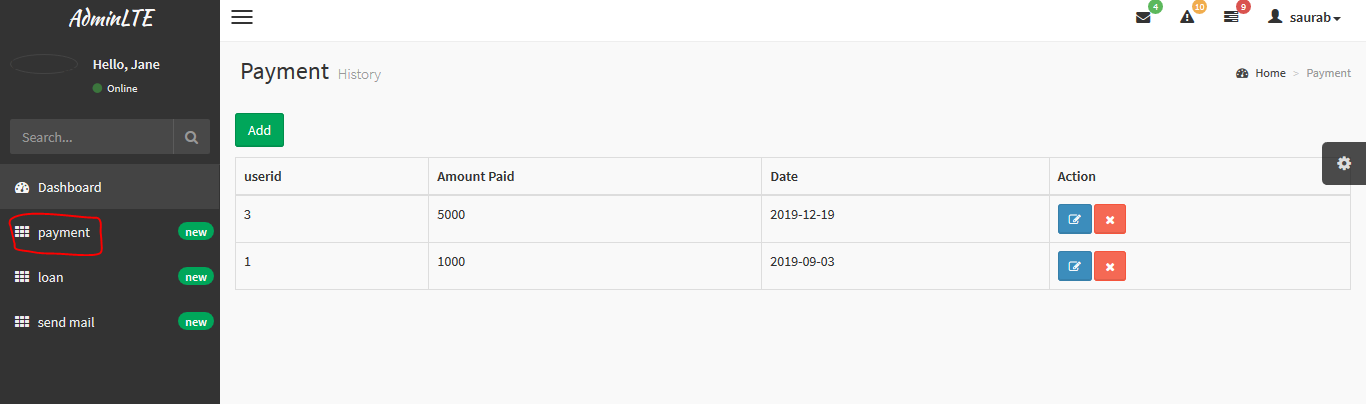
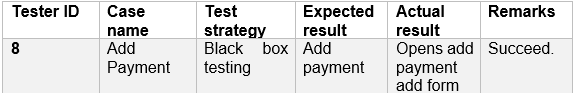


fig 35: payment form

**8.**

Fig 36: add payment table

* **When Add payment button is clicked, add payment forms open where user can pay their loan amount.**

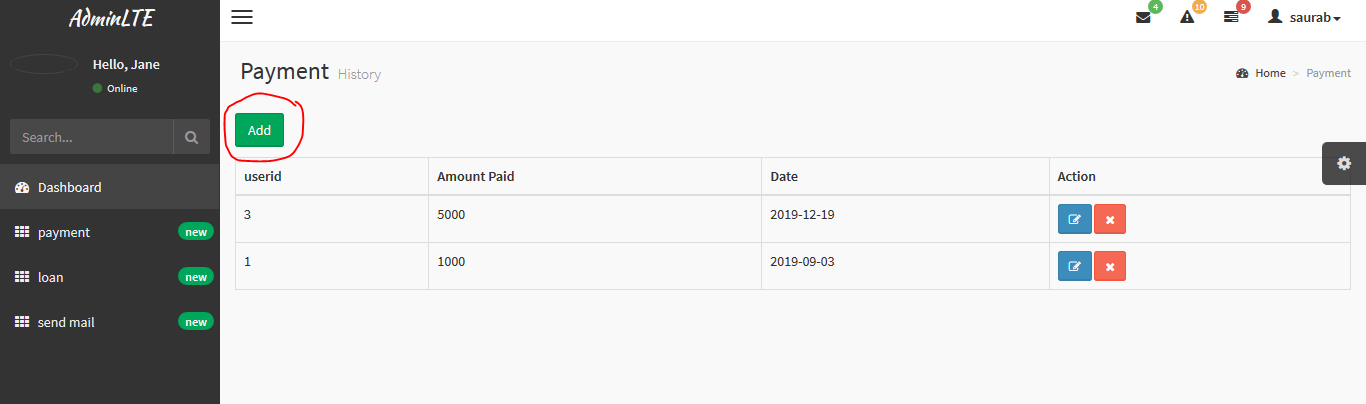


Fig 37: payment form

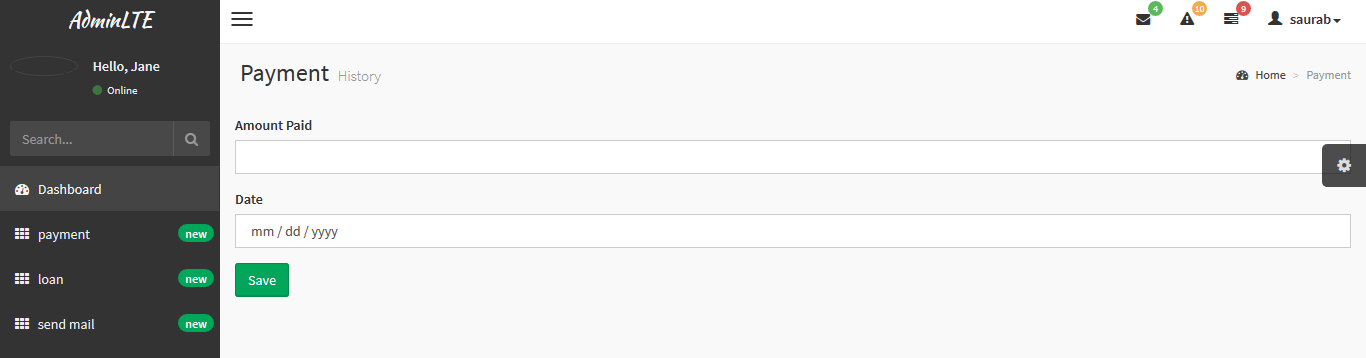


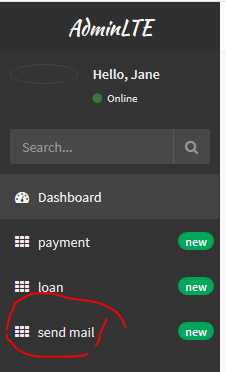
Fig 38: add payment form

**9.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 9 | Email | Black box testing | Email form appears | Opens email form | Succeed. |

Fig 39: email table

* **When send email button is clicked, email form appears where admin sends reminder of loan payment:**



Email form

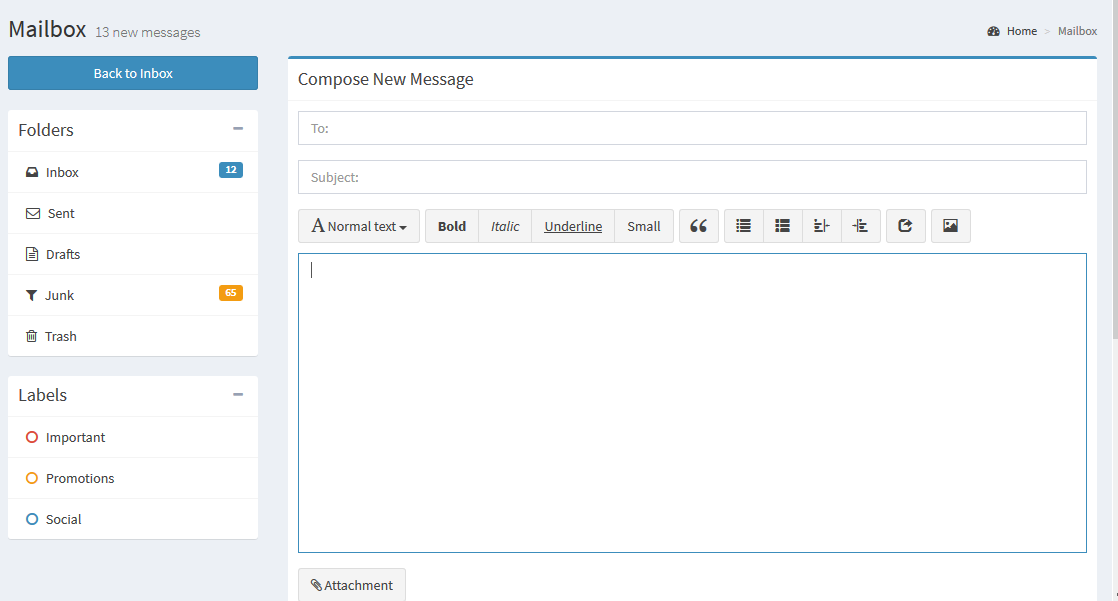
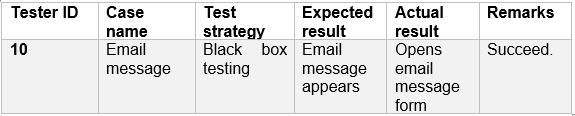
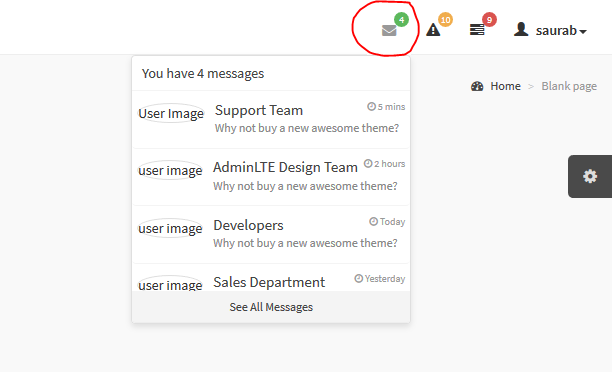


Fig 40: email form

10



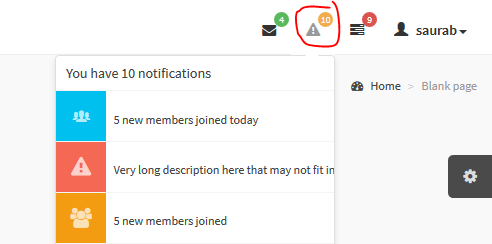
* **When mail icon is clicked admin can view all emails send to him by the customers.**



**11**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 11 | Notification message | Black box testing | Notification message appears | Opens notification message form | Succeed. |

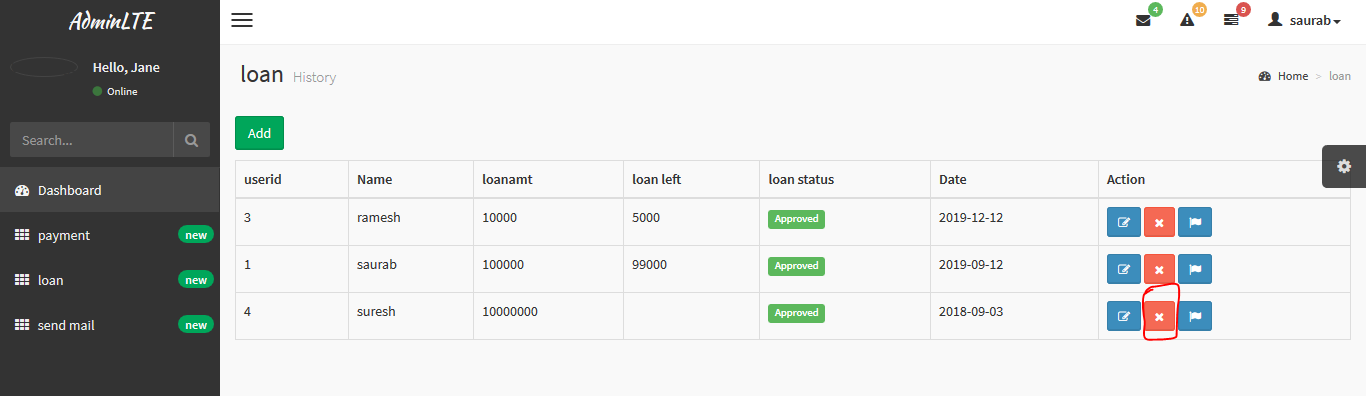
* **When notification icon is clicked admin can be notified what is going on.**

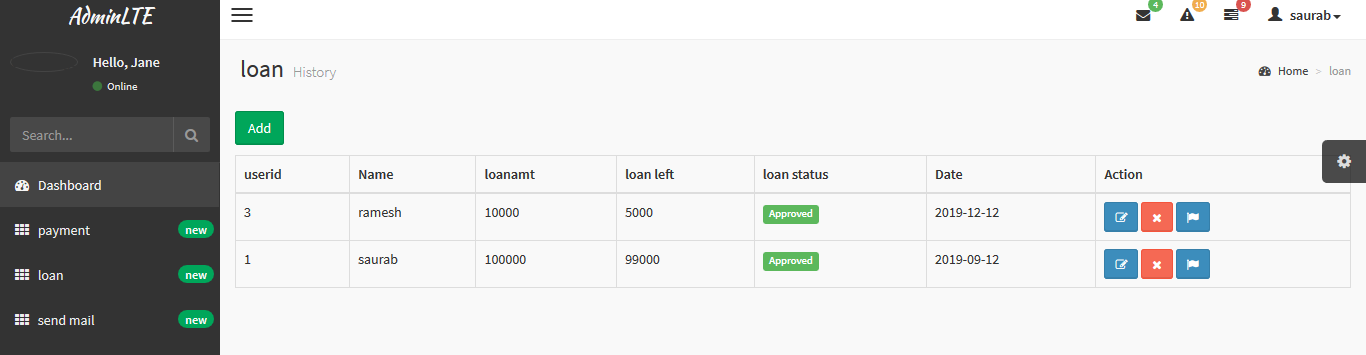


1**2.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 12 | Loan delete | Black box testing | Loan amount deletes | Deletes user loan amount | Succeed. |

* **When admin clicks on delete button the loan amount of user gets deleted.**

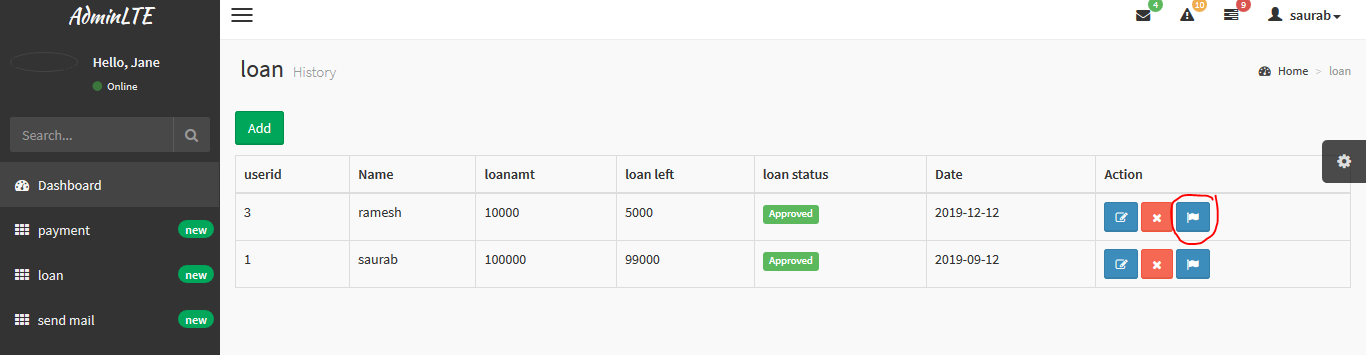




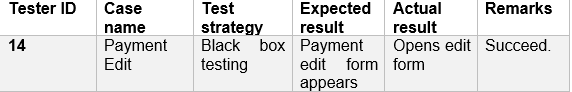
**13**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 13 | Flag | Black box testing | Approved sign comes | Approves displays | Succeed. |

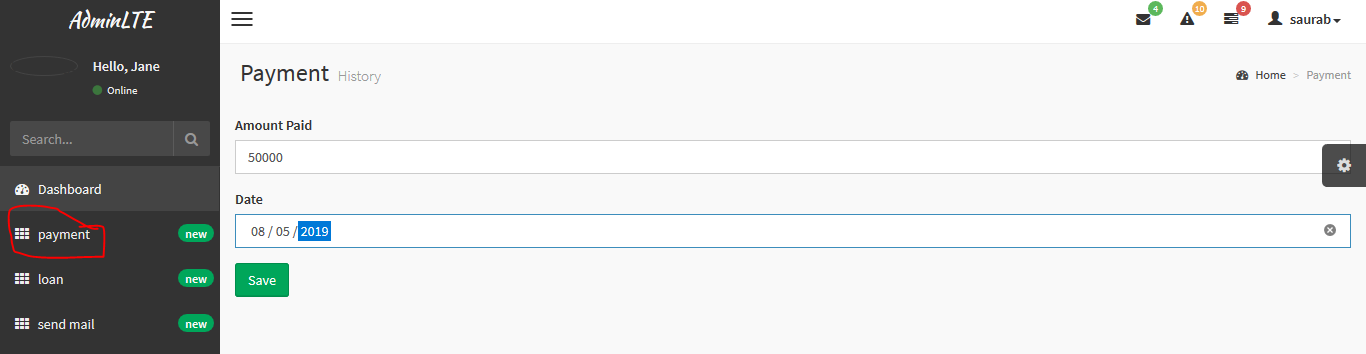
* **When admin flag button is clicked, approved symbol displays:**



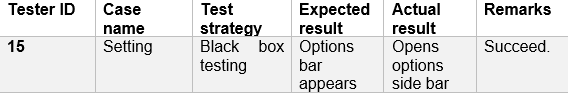
**14**



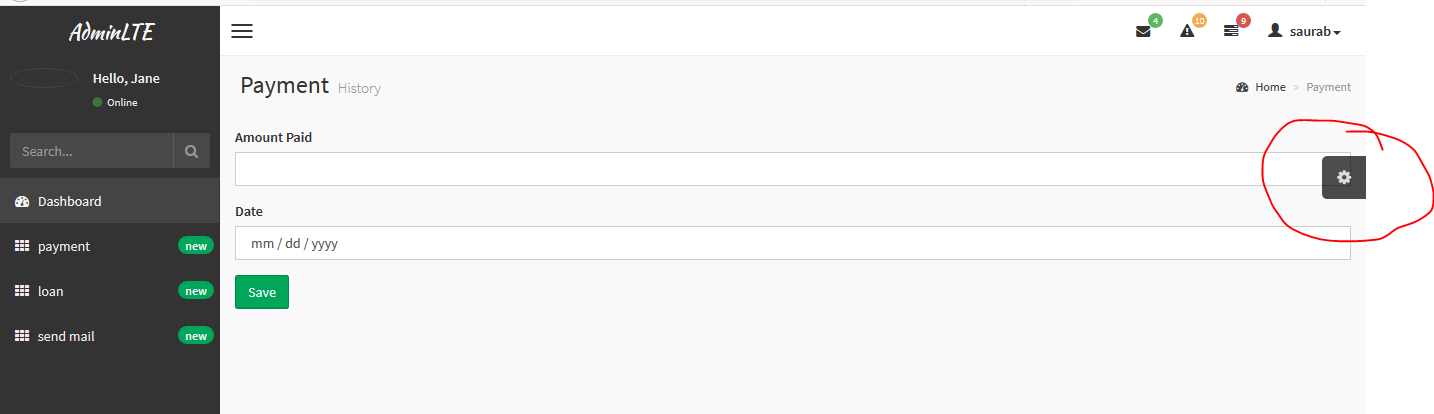
* **When customer click on payment button to pay their loan interest, payment form opens.**

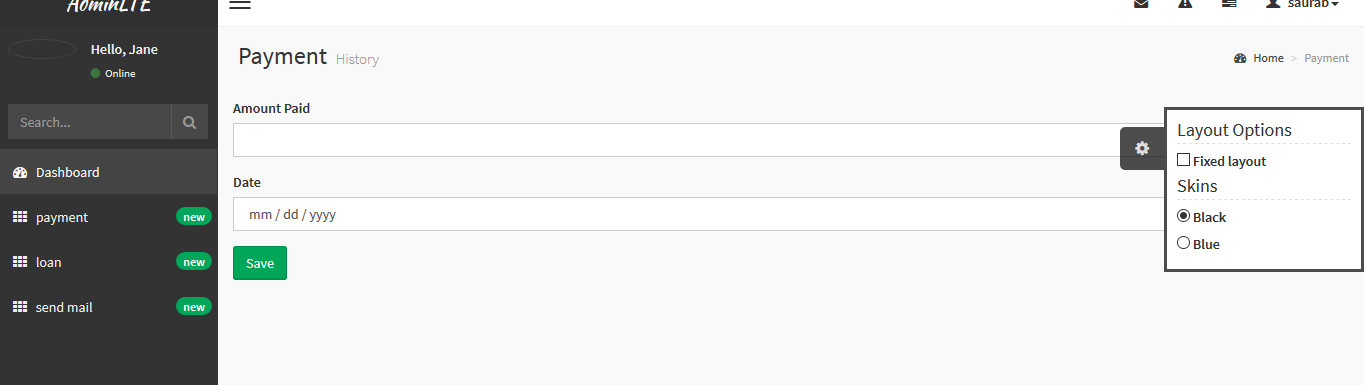


**15.**



* **When option side bar is clicked, options bar displays.**

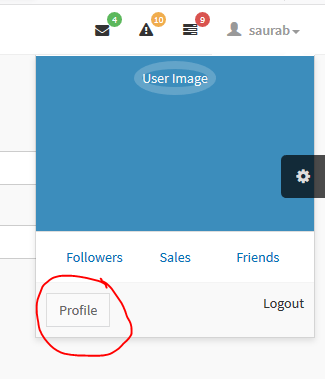


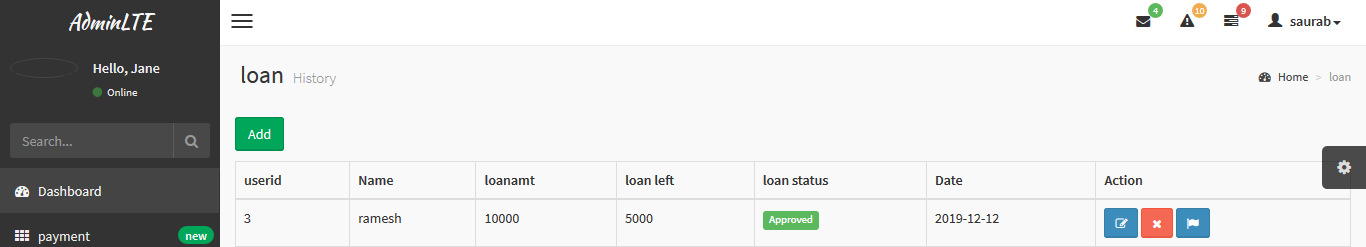


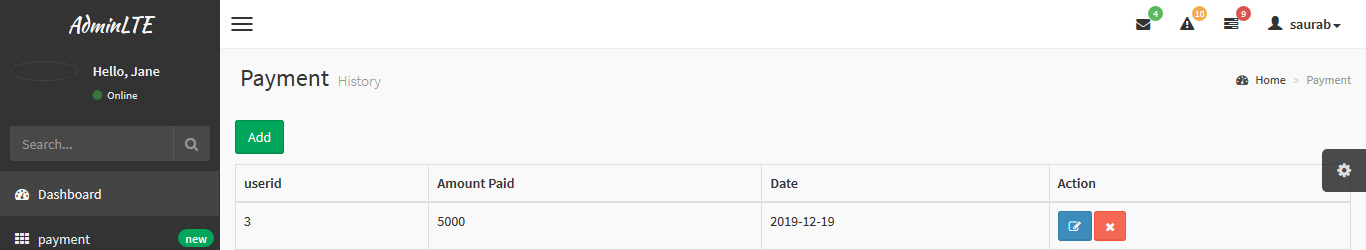
**16.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tester ID | Case name | Test strategy | Expected result | Actual result | Remarks |
| 16 | Profile | Black box testing | User profile opens | Opens user profile | Succeed. |

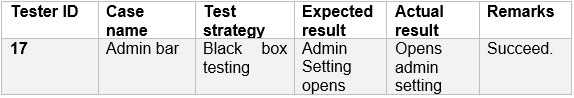
* **When user profile is clicked, the payment and loan taken by the user is displayed.**



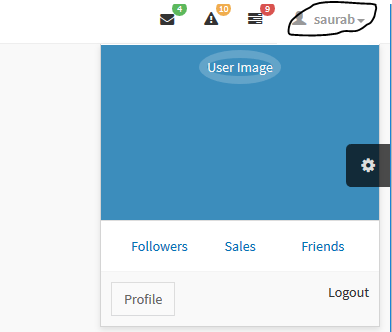




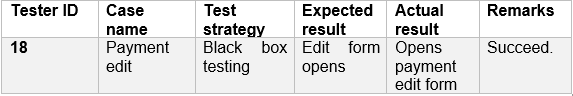
17.



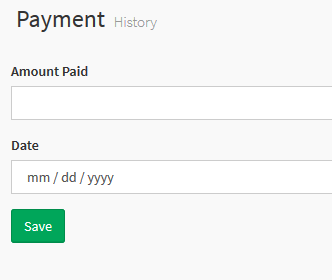
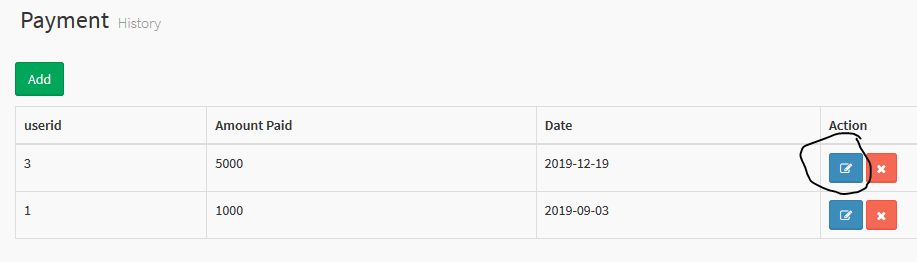
* **When admin side bar is clicked, it opens the options**



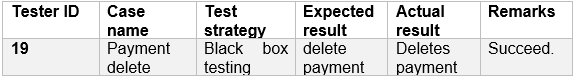
18.



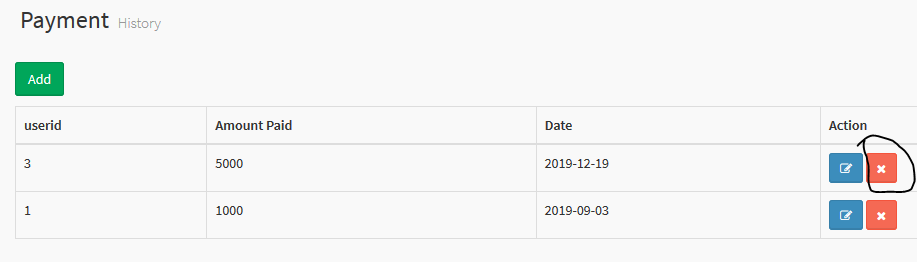
* **When payment edit button is clicked,it opens edit form.**



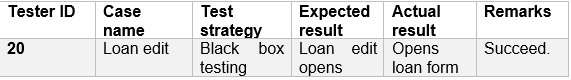
**19.**



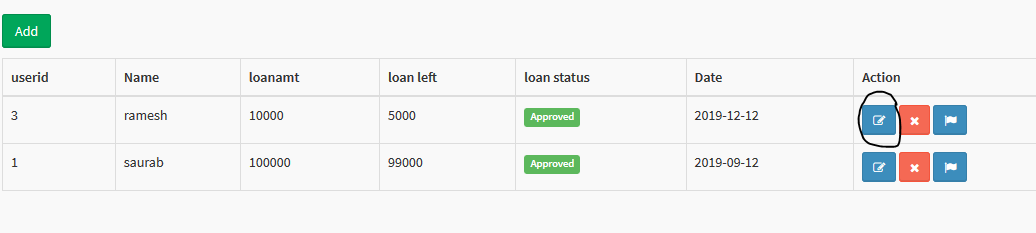
* **When payment delete button is clicked it deletes the data.**

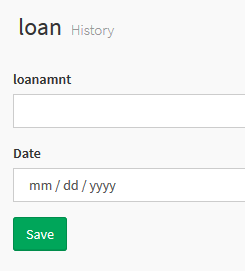


**20.**



* **When edit button is clicked loan edit form opens**



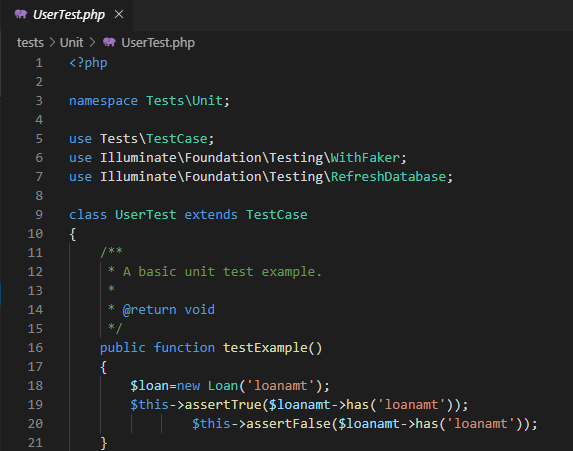


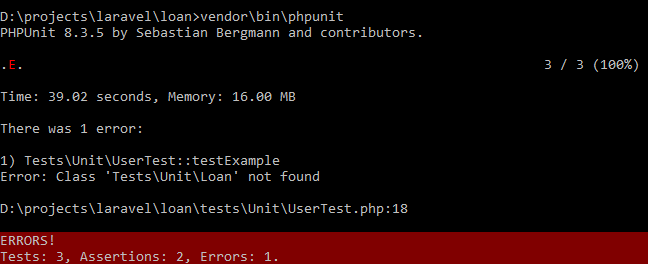
* **Unit testing**: Unit testing is software testing in which each and every unit or software component is tested. Basically, every unit of the software is tested. Its main purpose is to validate as designed each unit of the software. Every components of the software are tested.
* **Reasons to do unit testing:**

1. Ensures that each and every unit of the software works properly.
2. Facilitates changes & simples’ integration
3. Ensures quality of code.
4. Find software bugs early.

Here down below are the unit testing that I have performed for my software:

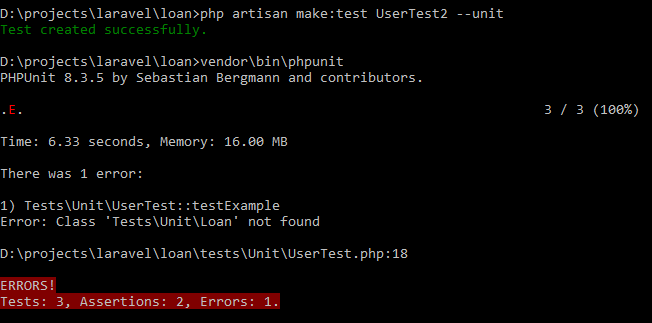
1. Loan amount test:



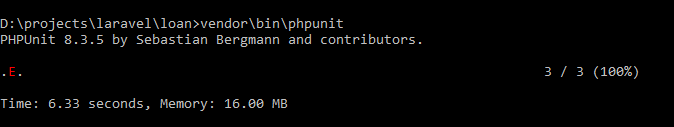


**2**.loan date:



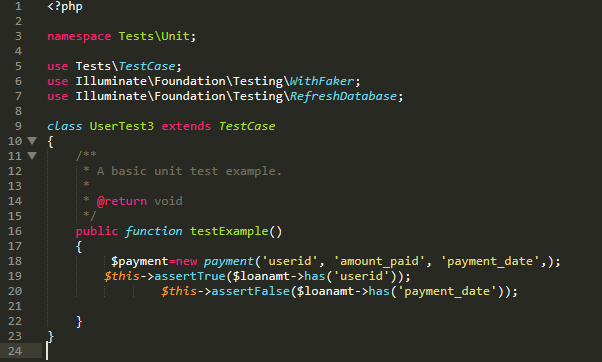


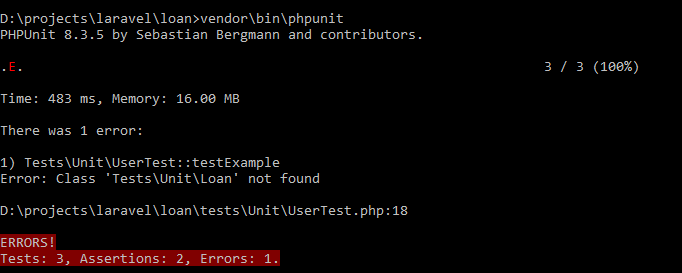
1. loan userid



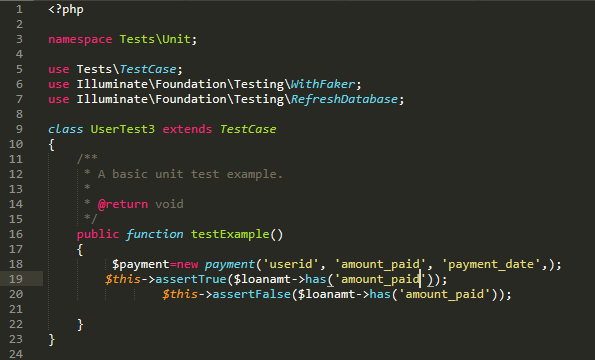


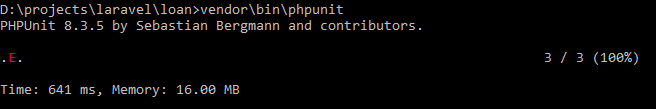
**4**.payment





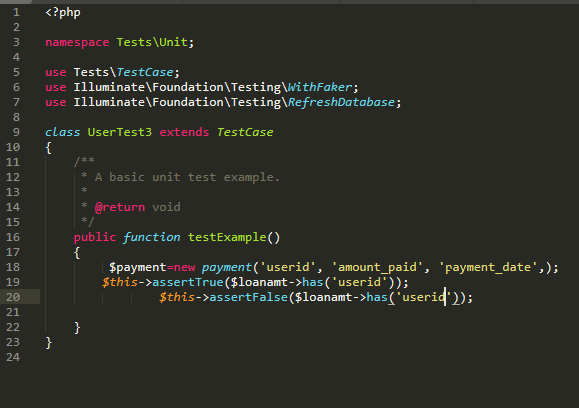
5.payment (amount paid)

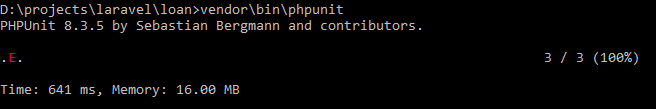






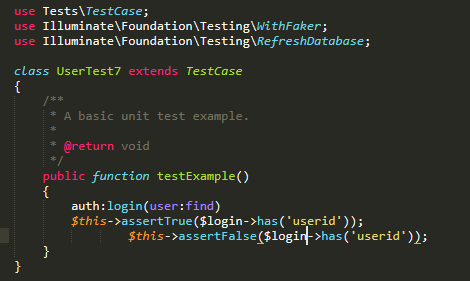
**6.**payment userid

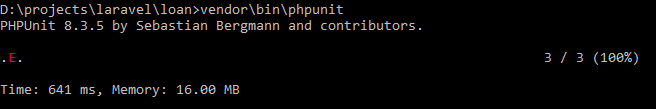






**7**. login

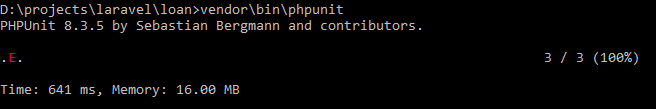






**8**.registration

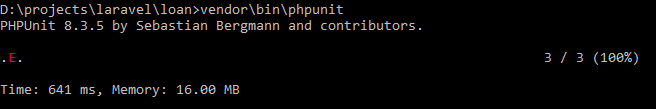






**9**.registration name

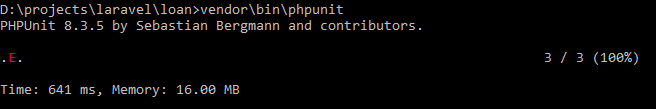






**10**.Registration email







# **Chapter -6**

# **Other project issues:**

## **Limitation:**

There is certain limitation of my system that it cannot perform or cannot function well. Therefore, the limitation of online loan management are as follows:

1. Client and customer cannot have conservation online through chat box.
2. Admin cannot send mails in Gmail as a reminder to the clients to pay their interest.
3. There’s no facility of paying the loan/ interest online.
4. Cannot track the location of the client who took loan from the bank.

## **Future work:**

1. **Chat box**

Here the chat box functionality helps the client to get connected to the admin of the bank where the client can ask the queries that he/she finds difficult understand loan taking criteria. With this function the client can interact with admin anytime. Currently our system lacks this functionality but it will be included in future wok.

1. **Email**

Currently the admin sends the email through mail-trap which is fake kind of email and in future I could like to Gmail services in the system so that the admin can send the client after the loan completion or if the client does not pay interest in time, in that case that reminder of payment can also be send through email. This functionality will be added in future work so that the system can provide its best services.

1. **Online payment**

With the help of this functionality client can pay their loans online. This service will save the time of the client as they don’t have carry cash and go to bank to pay their debts, the transaction can be made online. Here in this context the payment can be made through e-sewa, IME etc. This functionality will be added in future work so that the system can provide its best services.

1. **Tacking location**:

Currently the admin can track the payment only paid the customer but in future I would like to implement this functionality so that it would be secure to know where the person is after taking the loan. This functionality would secure the bank from loan fraud.

## **Risk management:**

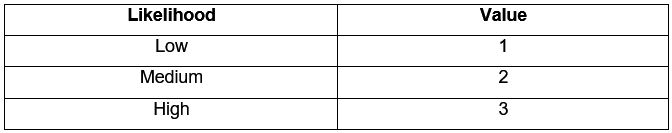
* Risk management is risk recognition, assessment and prioritization accompanied by organized and cost-effective use of resources to mitigate, monitor and control the threat or effects of unfortunate events and optimize opportunities realization. No system is perfect in this world, it certainly contains some defects. The system might get infected or may get harm. It may be affected by the natural disaster, employee theft, system failure, human error etc.

Here down below is some of the risk that might affect my project:

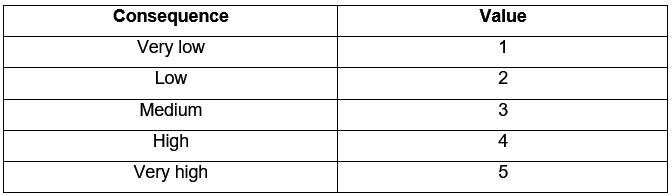
1. Data theft: it’s the act of stealing information from an unknowing victim with the intent of getting confidential information. This risk can cause the loss of data/information of customer can sometimes can be used against them. It is done by the system administrators and office workers to get the confidential information of customers.
2. Update/upgrade: When we update the system in to newer version or upgrade the hardware of the system there’s possibility of system failure, crashes which may lead to loss of data completely and there’s chance of not recovering of information.
3. System hardware failure: Many of the files are stored in the hardware in order to get access offline too and if the storage hardware creates problem than it will be difficult to access the data/information.
4. Disaster caused normally: natural disaster is inevitable. It causes a lot of risk. It may cause loss of life. Human disaster can be control but these natural disasters cannot be controlled.
5. Limited resource: it is also one of the types of risk and is neglected the most. People uses inadequate resource’s and tries to finishes the project which may cause problem in the future.

Risk management table:



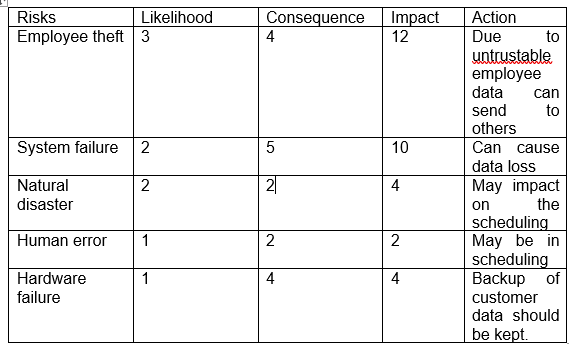


Risk likelihood and values:



Risk Consequence and Values.

Here down below is some of the risk that will affect my project with their impact and actions:



## **Configuration:**

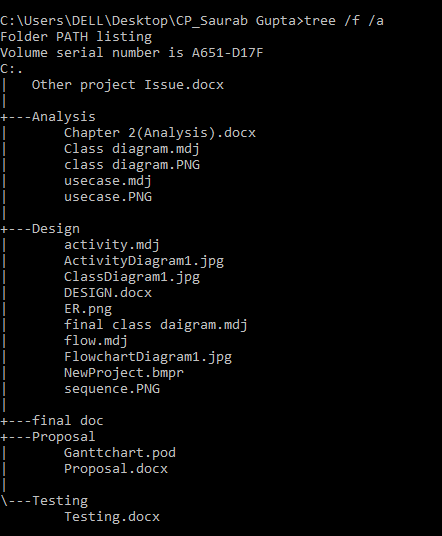
* The management system for software configuration controls basic components such as software objects, program code, test data, test output, design documents and user manuals. It is basically done as it helps in managing the data. Configuration management shows how and where the data/information are store.

For configuration management there is release management which means the process of managing, planning scheduling and controlling a software build through different stages including deployment and testing.

Version control management is known as revision control which manages the changes of documents, computer programs, large website and other information.

For my system I have saved files in the local drive i.e. in the computer and as well as I have pushed my files online i.e. in github.

In drive:



I have also created a GitHub account named saurab906 to store my data within which sub folders are also created. Here down below is the figure that shows how the data of my project are stored in git-hub account.

File saved in github(online):

## **Other project issues:**

**Social factors**: social factor affect the system both in the positive and as well as in negative way. While developing a software social factors should be kept in mind. System should be developed in way that social factors doesn’t get affected.

**Technical factors:** Technical factor is related to the hardware and software specification required by the system to function properly. For my system the basic requirements are that the system needs a minimum of 2 GB of ram to run all the features smooth and sudden. If these criteria do not match then it might affect the software to run smoothly.

**Ethics factors.** Ethics factors are related to the ethics of a people or community. While developing the system the ethics shouldn’t be affected because to make the software popular ethics/ sentimental of the people shouldn’t be hurt.

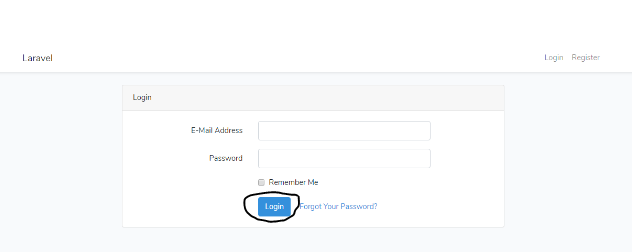
## **User manual:**

* User manual provides the step to use the **Online loan management system,** therefore this manual provides the user guide on how to use the system.

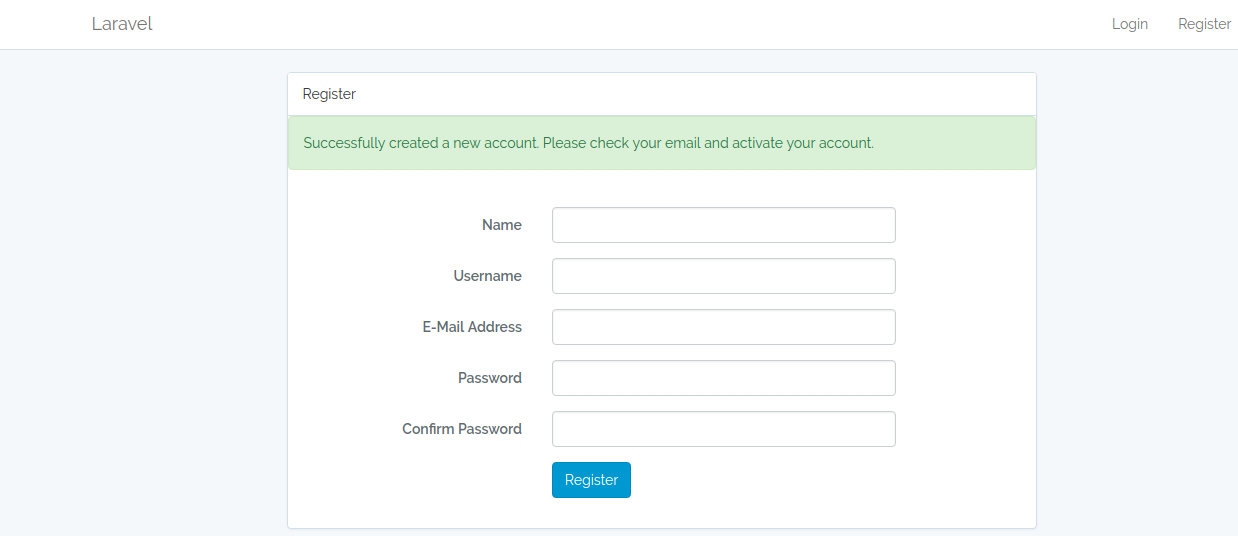
1. **Log in and Registration:**

* Admin can only login in the system and view all the details but in case of user they need to register and login with correct email and password.
* *Login*

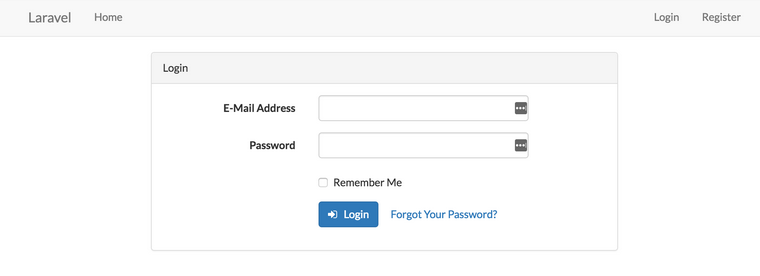
Enter correct email and password:



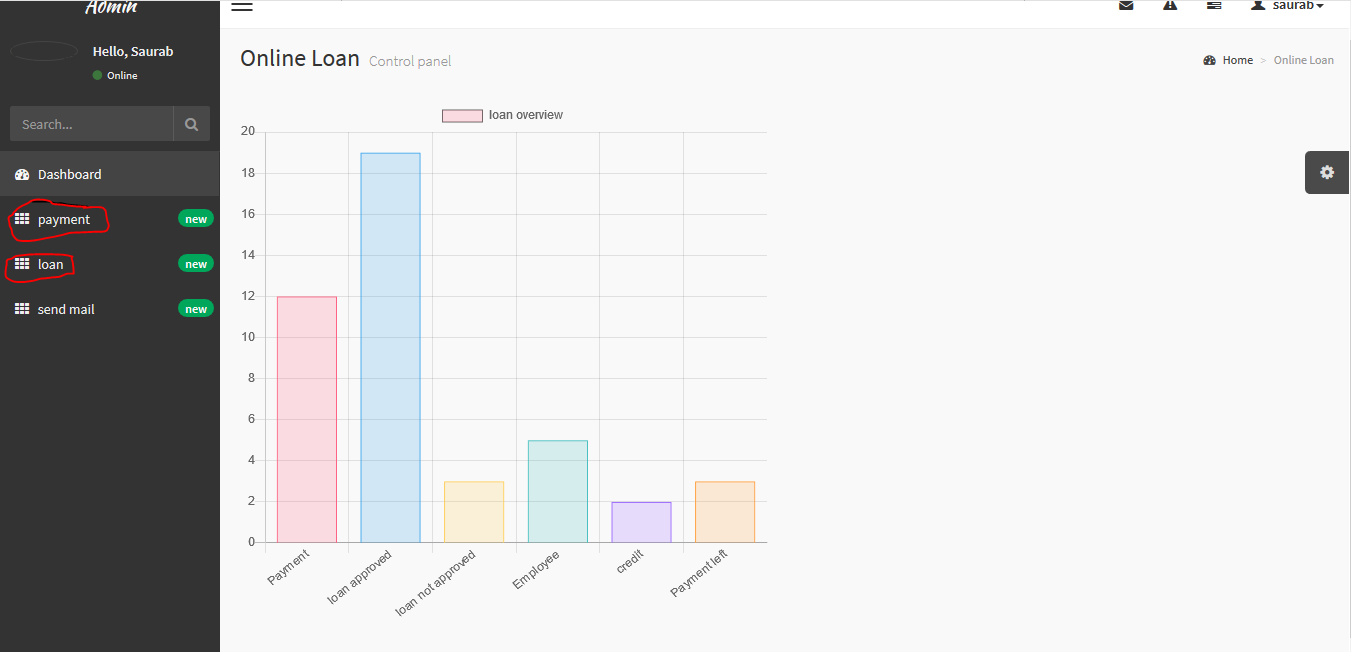
* *Registration page:*

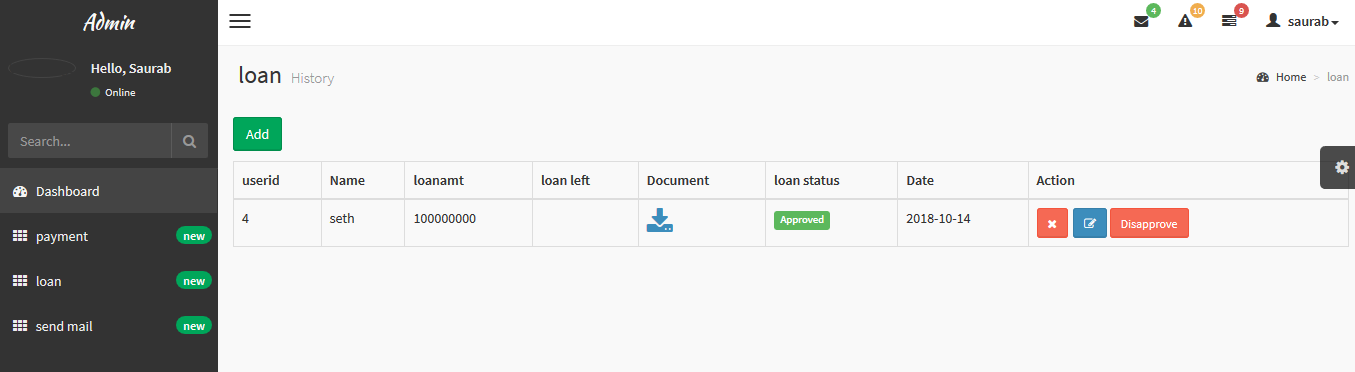


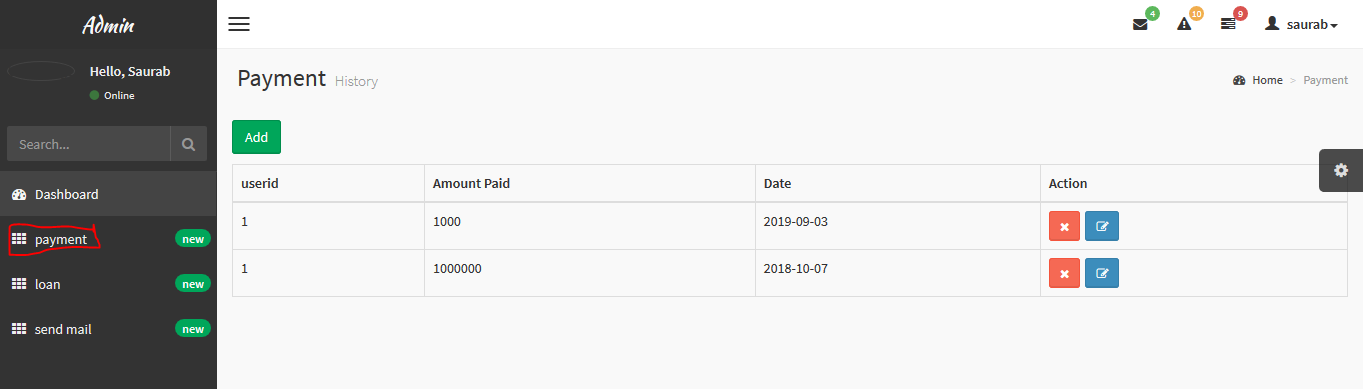
1. Forget password: incase user forgets the password; they can click on it.



1. After logging in a dashboard open for admin where he can view the details of the client and approve loan and see payment made by them.



* By clicking on loan admin can view whose loan are approved and whose are not
* By clicking on payment admin can view who have made payment.



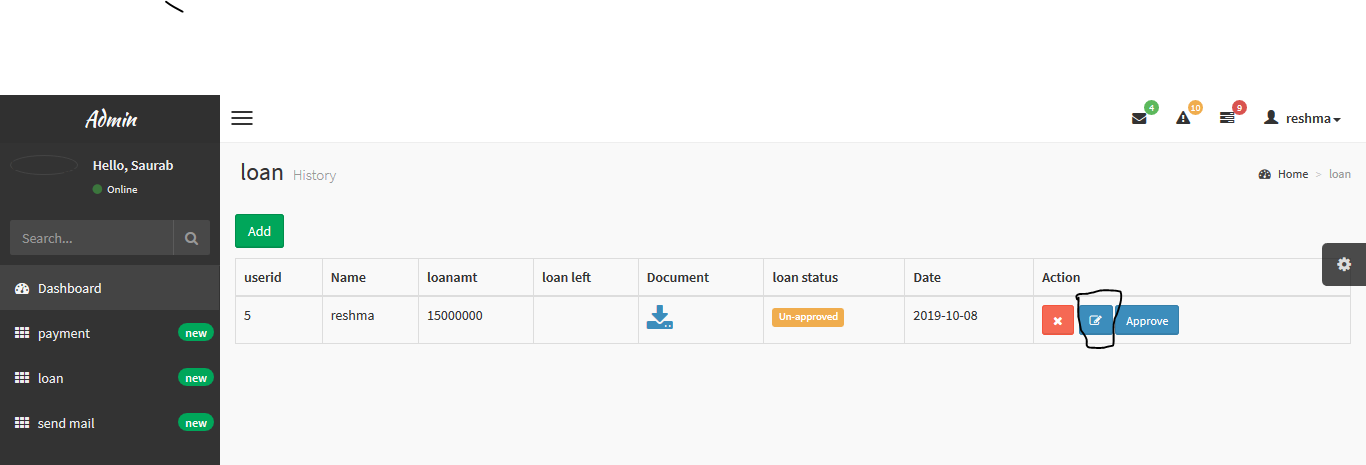
* For client, by clicking on loan they can ask for types of loan and upload their document:

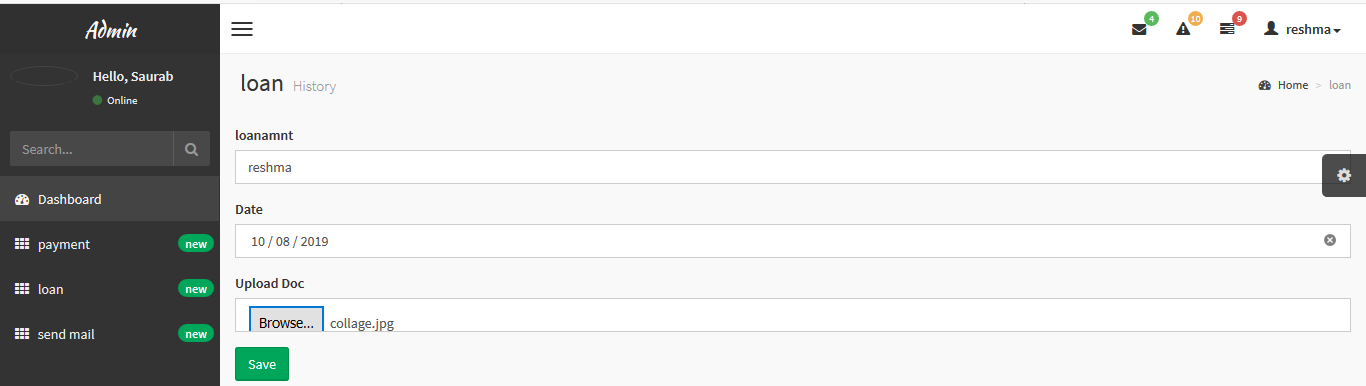


* By clicking in payment, the client can make their payment for the loan that they have taken.

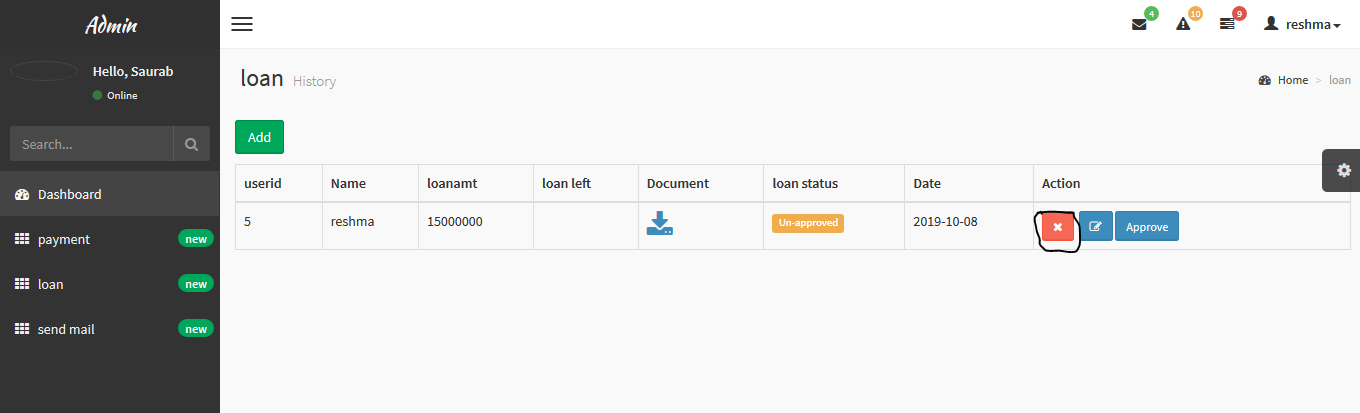


* For updating the data, the client can click on edit button and update their data:

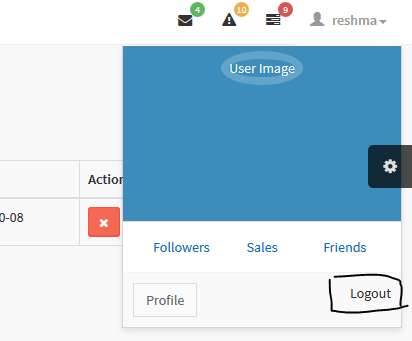




* Only admin can delete the data of the client and for that just click on delete button:



* For signing out they can click on their name on the right-hand side and click on logout:



# **Conclusion**

Finally, after all the work Online Loan management is developed. This system basically helps the user to take loan from the bank and their branches easily. It helps to reduce the manual data entry and provides greater efficiency. the user interface of the software friendly therefore anyone can use it without any difficulties. This software also saves the time of the client and bank as the client does not have to go the bank and ask for loan and wait for the approval, they can just go the site ask for the type of loan and upload their documents and after that it the documents matches the bank criteria than loan will be approved. Basically, it decreases the amount of time between the client and bank during the loan transaction.

At the end of the day, we can say that this software performs all the tasks accurately and does the job it is made for.

# **References:**

1. Scribd, 2018. Scope limitation, definition [Online]

Available at: https://www.scribd.com/presentation/349056812/RESEARCH-PAPER-SCOPE-LIMITATIONS-Reporting

1. Techopedia, 3 tier system architecture[online]

Available at: <https://www.techopedia.com/definition/24649/three-tier-architecture>

1. Wikipedia, waterfall diagram, definition [Online]

Available at: <https://en.wikipedia.org/wiki/Waterfall_chart>

1. Site-point, MVC pattern, defining and diagram2019[online]

Available at: <https://www.sitepoint.com/the-mvc-pattern-and-php-1/>

1. WBS, definition ad diagram,2019 [online]

Available at: <https://www.workbreakdownstructure.com/>

1. Wrike, Risk management ,2019[online]

Available at: <https://www.projectmanager.com/blog/risk-management-process-steps>

1. Misguide, rich picture,2019 [Online]

Available at: <http://www.mspguide.org/tool/rich-picture>

1. Lithopedia, feasibility,2019 [Online]

Available at: <https://www.investopedia.com/terms/f/feasibility-study.asp>

1. Productplan, MosCow,2019[Online]

Available at: <https://www.investopedia.com/terms/f/feasibility-study.asp>

1. Sciencedirect, structural modelling, 2019[online]

Available: <https://www.sciencedirect.com/topics/computer-science/structural-modeling>

1. Wikipedia, behavioral modelling, 2019[Online]

Available at: <https://en.wikipedia.org/wiki/Behavioral_modeling>

1. Wikipedia, UI modelling,2019 [Online]

Available at: <https://en.wikipedia.org/wiki/User_interface_modeling>

1. Techopedia, Prototype, 2019[Online]

Available at: <https://www.techopedia.com/definition/678/prototype>

1. Laravel doc, Testing,2019[Online]

Available at: <https://laravel.com/docs/5.8/testing>

1. SoftwareFundamentals, black box tesing,2019[Online]

Available at: http://softwaretestingfundamentals.com/black-box-testing/

1. Wikipedia, configuration management, definition[online]

Available at: https://en.wikipedia.org/wiki/Configuration management

1. Wikipedia, User interface,2019[Online]

Available at: <https://en.wikipedia.org/wiki/User_interface>

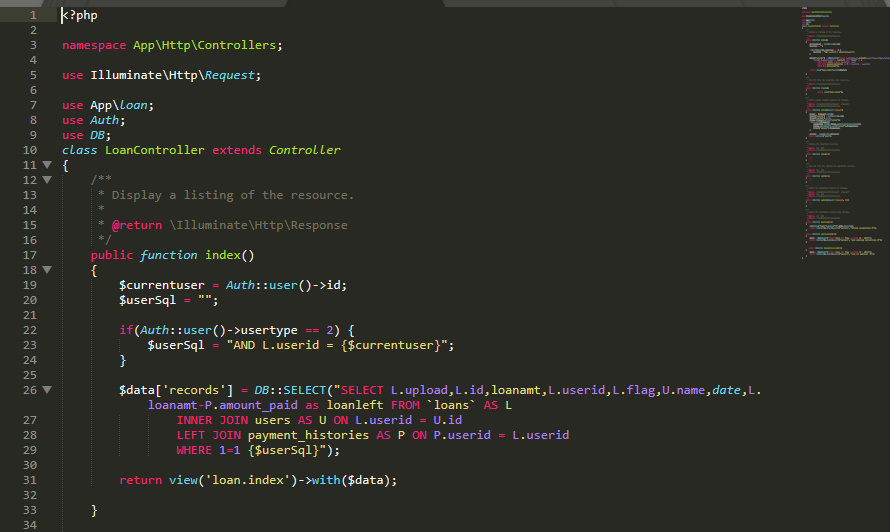
# Appendix:

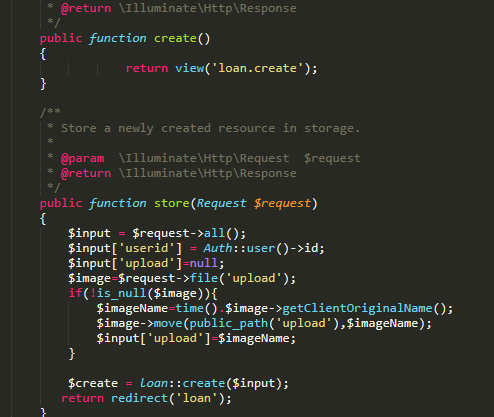
***Code for loan:***

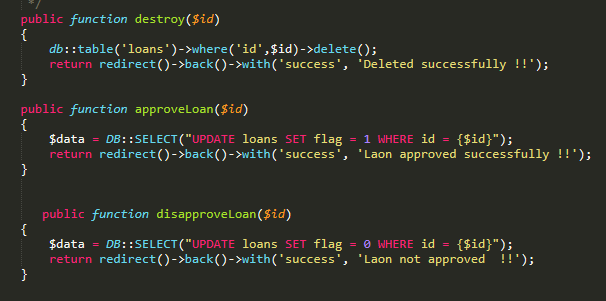
**Loan Module:**



**Loan controller:**







**Loan view:**

