

**Batch No. :FDJ B129**

**Nilesh Tayade(TL)**

**Ajinkya Deshmukh**

**Mahesh Sukalkar**

**Aniket Dhaigude**

**Ghansham**

Index Page :

|  |  |  |
| --- | --- | --- |
| Sr. No. | **Title** | **Page No.** |
| Chapter 1 | **Introduction** |  |
| 1.1 | **Operating Environment – Hardware and Software** |  |
| 1.2 | **Detail Description of Technology Used** |  |
| Chapter 2 | **Proposed System** |  |
| 2.1 | **Proposed System** |  |
| 2.2 | **Objectives of System** |  |
| 2.3 | **User Requirements** |  |
| Chapter 3 | **Analysis And Design** |  |
| 3.1 | **Project Architecture Diagram** |  |
| 3.2 | **Data Flow Diagram** |  |
| 3.3 | **Entity Relationship Diagram** |  |
| 3.4 | **Use Case Diagram** |  |
| 3.4 | **Activity Diagram** |  |
| 3.5 | **Sequence Diagram** |  |
| 3.6 | **Collaboration Diagram** |  |
| 3.7 | **Table Specifications** |  |

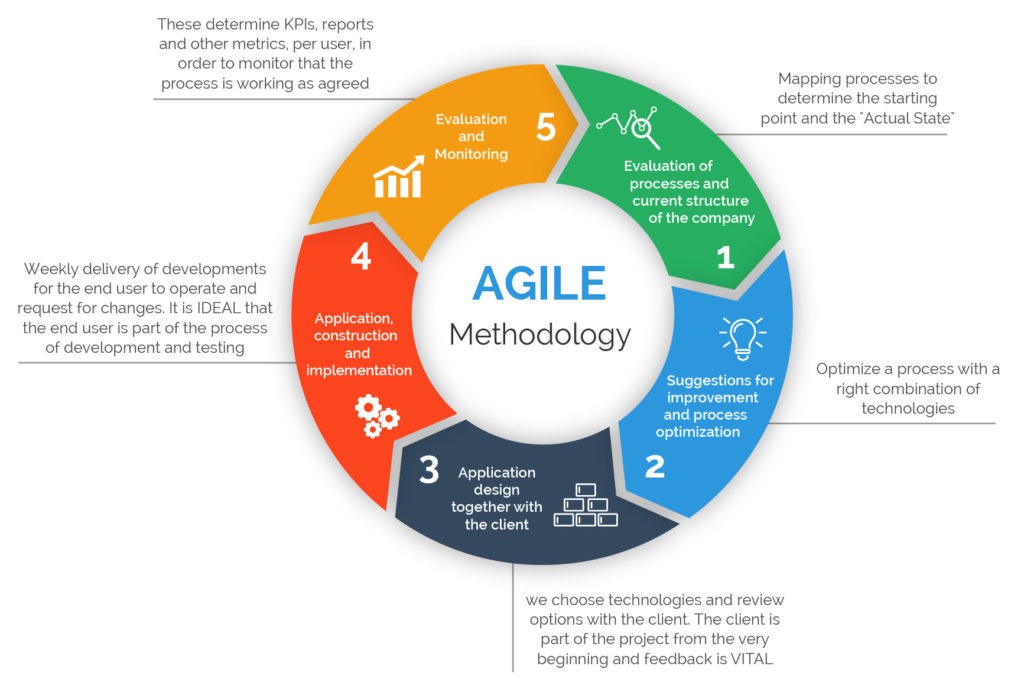
**Chapter 1. Introduction**

* 1. **What is Micro Finance?**

**Microfinance** is a banking service provided to unemployed or low-income individuals or groups who otherwise would have no other access to **financial** services. **Microfinance** allows people to take on reasonable small business loans safely, and in a manner that is consistent with ethical **lending** practices.

* **Concept And Features of Micro-finance:**

1. It is a tool for empowerment of the poorest.
2. Delivery is normally through self Help Group (SHGs).
3. It is essential for promoting self-employment, generally used for:
   1. Direct income generation
   2. Rearrangement of assets and liabilities for the household to participate in future opportunities and
   3. Consumption smoothing.
4. It is not just a financing system, but a tool for social change, especially for women.
5. Because micro credit is aimed at the poorest. micro-finance lending technology needs to mimic the informal lenders rather than the formal sector lending. It has to:
   1. Provide for seasonality
   2. Allow repayment flexibility
   3. Fix a ceiling in loan sizes.
   4. **Methodology Used For Web Site Development:** Agile Methodology



* 1. **Modules:**
* Relationship Executive
* Operational Executive
* Account Head
* Branch Manager
* Credit Manager
* Loan Disbursement
* Master Module
* Ledger Generate
* Telecaller
* Applicant Module
* Admin Module
* **Relationship Executive:**

Relationship executives identify and pursue business relationships with corporate and business unit executives. They create new business opportunities through customer relationships, while also providing value to customers through professional services agreements and solution sales. Relationship executives also establish strategic relationships with internal departments, including sales, marketing and business development teams. They use their internal and external relationships to develop sales strategies and service offerings that enable new opportunities for business and revenue growth.

* **Operational Executive:**

The Operations Executive is responsible for the part of the workplace team that directly manages the operations and maintenance of facilities. The Operations Executive can report to various parts of the company such as the Facilities Executive, Chief Financial Officer or Chief Operations Officer, but usually has direct access to senior management.

* **Admin:**

The Admin is responsible for important operation like adding new electronic items , new loan type , he can manage applicant and loan type , electronic Items and users and create new branch.

* **Applicant:**

The applicant can view his account his active files as well his previous loan file.

He can pay the installment and get the loan details .

**ABSTRACT**

The project entitled “**CJC Microfinance Bank**” is to be developed for maintaining the bank activities like, customer preferences, customer enquiry, interest rates, customer EMI,  personal loan, mortgage loan, Customer follow ups details, customer feedback form entry and employee details. The system is efficient in generating reports which will help in the maintaining records of the customer.

Microfinance is a project which has become a mainstream instrument for providing access to formal financial services for helpless people. This project is developed to maintain all the details of the users and to develop online portal.

**1.2 OPERATING ENVIRONMENT – HARDWARE AND SOFTWARE**

* **Hardware Requirements for Server :**

**Processor**  : Pentium IV 1.8 GHz

**Memory size** : 2 GB RAM

**Storage**  : 40 GB Hard Disk

**Display** : EGA/VGA Color Monitor, 600 x 800 Pixels Resolution, High Color

**Internet Connection** : Required

**Key Board** : Any with minimum required keys

**Mouse** : Any

* **Software Requirements :**

**Operating System** : Windows 7 and above

**Front-End Tool** : ANGULAR 8.0, HTML

J2EE (Java, Spring 3.0, Hibernate 3.0)

**Web Server**  : Apache Tomcat 6.0.

**Back-End Tool** : MySQL 5.0

* **Hardware Requirements for Client :**

**Processor** : Pentium III 800 MHz

**Memory size** : 2 MB RAM

**Storage** : 40 GB Hard Disk

**Display** : EGA/VGA Color Monitor, 600 x 800

Pixels Resolution, High Color

**Key Board** : Any with minimum required keys

**Internet Connection** : Required

**Mouse** : Any

**1.3 DETAIL DESCRIPTION OF TECHNOLOGY USED**

* **JAVA:**

JAVA 8 is a major feature release of JAVA programming language development. Its initial version was released on 18 March 2014. With the Java 8 release, Java provided supports for functional programming, new JavaScript engine, new APIs for date time manipulation, new streaming API, etc.

## FEATUES OF JAVA:

* **Lambda expression** − Adds functional processing capability to Java.
* **Method references** − Referencing functions by their names instead of invoking them directly. Using functions as parameter.
* **Default method** − Interface to have default method implementation.
* **New tools** − New compiler tools and utilities are added like ‘jdeps’ to figure out dependencies.
* **Stream API** − New stream API to facilitate pipeline processing.
* **Date Time API** − Improved date time API.
* **Optional** − Emphasis on best practices to handle null values properly.
* **APACHE TOMCAT SERVER:**

Apache Tomcat is an open-source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Web Socket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

**Apache Tomcat** is **used** to deploy your Java Servlets and JSPs. So in your Java project you can build your WAR (short for Web Archive) file, and just drop it in the deploy directory in **Tomcat**. So basically **Apache** is an HTTP **Server**, serving HTTP. **Tomcat** is a Servlet and JSP **Server** serving Java technologies.

* **HIBERNATE:**

**Hibernate** is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

**Following are the advantages of hibernate framework:**

1) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

### 2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.

### 3) Database Independent Query

HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.

### 4) Automatic Table Creation

Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

### 5) Simplifies Complex Join

Fetching data from multiple tables is easy in hibernate framework.

### 6) Provides Query Statistics and Database Status

Hibernate supports Query cache and provide statistics about query and database status.

* **MySQL:**

**MySQL** is a relational database management system based on SQL – Structured Query Language. The application is **used** for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for **MySQL** however, is for the purpose of a web database.

**MySQL** is not a **programming language**. Instead, it is a relational database management system (RDBMS). It is used to store data, not to write programs. The SQL **programming language** can be used to program a **MySQL** database.

* **MAVEN:**

**Maven** is an automation and management tool developed by Apache Software Foundation. ... In Yiddish language the meaning of **Maven** is "accumulator of knowledge". It is written in Java Language and used to build and manage projects written in C#, Ruby, Scala, and other languages.

**Maven** is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. ... In short terms we can tell maven is a tool that can be used for building and managing any Java-based project

* **SPRING BOOT:**

**Spring Boot** is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

**Spring Boot** is basically an extension of the spring framework which eliminated the boilerplate configurations required for setting up a spring application.

**Spring Boot** is a lightweight framework that takes most of the work out of configuring Spring-based applications. In this tutorial, you'll learn how to use Spring Boot's starters, opinions, and executable JAR file structure to quickly create Spring-based applications that “just run”.

**Spring Boot Rest API** Example. ... Writing **Restful** services in **Spring Boot** is no-different than Spring MVC. If you are a **REST** Client [**Rest** Consumer], **Spring Boot** provides Rest Template Builder that can be used to customize the Rest Template before calling the **REST** endpoints.

* **ANGULAR:**

Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.These Angular docs help you learn and use the Angular framework and development platform, from your first application to optimizing complex single-page apps for enterprises.

**FEATURES AND BENEFITS:**

**CROSS PLATFORM**

**1. Progressive Web Apps**

Use modern web platform capabilities to deliver app-like experiences. High performance, offline, and zero-step installation.

**2. Native**

Build native mobile apps with strategies from Cordova, Ionic, or Native Script.

**3. Desktop**

Create desktop-installed apps across Mac, Windows, and Linux using the same Angular methods you've learned for the web plus the ability to access native OS APIs.

* **PRODUCTIVITY**

**1. Templates**

Quickly create UI views with simple and powerful template syntax.

**2. Angular CLI**

Command line tools: start building fast, add components and tests, then instantly deploy.

**3. IDE’s**

Get intelligent code completion, instant errors, and other feedback in popular editors and IDEs.

* **SPEED AND PERFORMANCE**

**1. Code Generation**

Angular turns your templates into code that's highly optimized for today's JavaScript virtual machines, giving you all the benefits of hand-written code with the productivity of a framework.

**2. Code Splitting**

Angular apps load quickly with the new Component Router, which delivers automatic code-splitting so users only load code required to render the view they request.

**Chapter 2. Proposed System**

2.1 Proposed System

The proposed software will solve all the problems they are facing now. Loan is still perceived as a huge burden for many countries. While loans are known to take off that burden off, it still gives nightmare to people and they avoid availing a personal loan. However, a loan can be a solution to many of your financial problems.

A Microfinance helps you purchase a Vehicle ,Personal loan or Goods of your choice and pay for it in equated monthly installments (EMIs). Personal loans are available for both salaried and self-employed individuals. Bank provides Home loans on easy terms. You will come to us where our representative will help you complete the formalities and inform you if you are eligible for a loan on the spot. Once your loan application has been approved you can get your loan amount in short period of time

**2.1.1 Purpose of proposed system**

* Interest rates and the loan details are also available at the click of a mouse.
* Customer can apply for a loan and after approved it they can track their details from online.
* This system provides detail about the customers, their loan details, EMI details and its rate details.
* System provides download option to download different type of loan form in MS word document.
* Using with this system admin can find customer easily and it’s a paperless system so workload is reduced.
* The decision process becomes faster and more consistent.
* After registration and login customer can use the system easily and also customer can view any query about loan details as well as EMI details in their profile. So this system saves time.
* Provides good communication for the customer.
* In this system there are used EMI (Equated Monthly instalment) calculators.
* Provides a facility to generate the reports very.

**2.1.2 Advantages of the proposed System**

* Entire activities of the show room are recorded through the system.
* Customer Data is maintained.
* Reports generated will be more useful for management to take the quick business decisions.
* Customer database is maintained which will be helpful for intimating the service completion details and new offers
* Customer follow-ups are maintained which will be an added advantage of this system.
* It helps with your CIBIL Score as CIBIL or Credit Scores are a summary of customer history in loan credits and repayments over a period of time.

**2.2 Objectives of System**

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to calculate monthly statements of Customer Record which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record updating, maintenance, and searching. The searching of record has been made quite simple as all the details of the Customer can be obtained by simply keying in the identification of that Customer. Similarly, record maintenance and updating can also be accomplished by using the identification of Employee with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the database or Files and whoever wants to retrieve can’t retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file. The main objective of the entire activity is to automate the process of day to day activities of pay.

* **Major Functionalities:**

The system is very simple in design and to implement. The system requires very less system resources and the system will work in almost all configurations. It has got

* **Following features :-**

1) Ensure data accuracy’s.

2) Proper control of the higher officials.

3) Minimize manual data entry.

5) Minimum time needed for the various processing.

6) Greater efficiency.

7) Better service.

8) User friendliness and interactive.

9) Minimum time required.

**2.3 User Requirements**

The system after careful analysis has been identified to be presented with the following modules:

* Relationship Executive
* Operational Executive
* Account Head
* Branch Manager
* Credit Manager
* Loan Disbursement
* Master Module
* Ledger Generate
* Telecaller.
* **Relationship Executive:**

Relationship executives identify and pursue business relationships with corporate and business unit executives. They create new business opportunities through customer relationships, while also providing value to customers through professional services agreements and solution sales. Relationship executives also establish strategic relationships with internal departments, including sales, marketing and business development teams. They use their internal and external relationships to develop sales strategies and service offerings that enable new opportunities for business and revenue growth.

* Admin manage loan application which is sent by customer using the system.

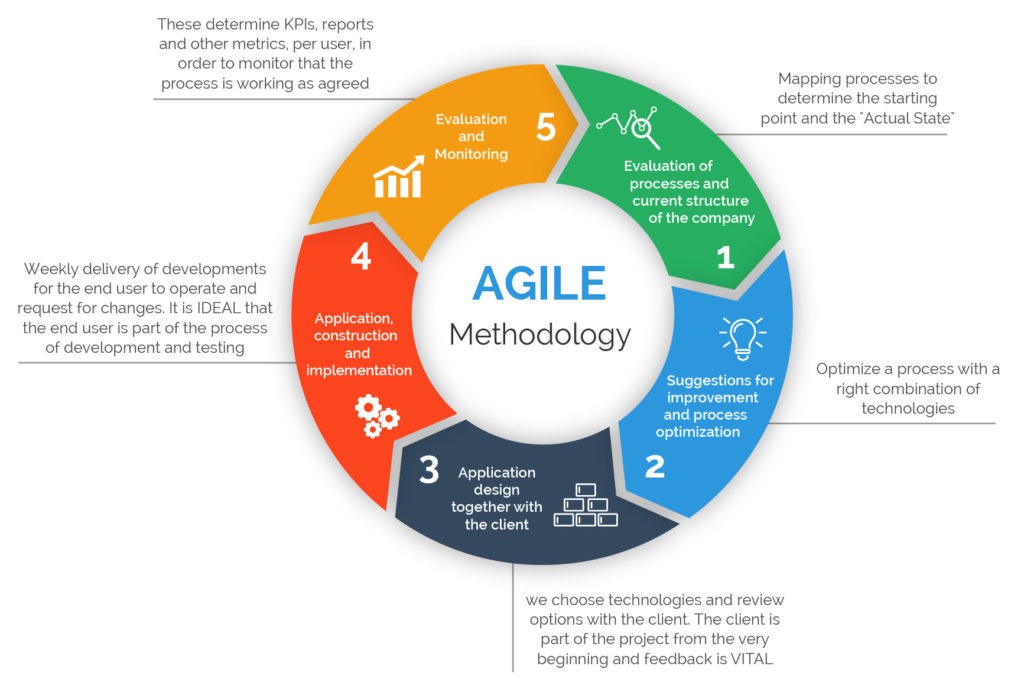
**Operational Executive:**

The Operations Executive is responsible for the part of the workplace team that directly manages the operations and maintenance of facilities. The Operations Executive can report to various parts of the company such as the Facilities Executive, Chief Financial Officer or Chief Operations Officer, but usually has direct access to senior management.

On the basis of the submission of the required details & documents, the financial institution will analyze the application. From existing residential address to CIBIL score, complete information is thoroughly checked. Once the bank has validated all the details, loan amount is sanctioned.

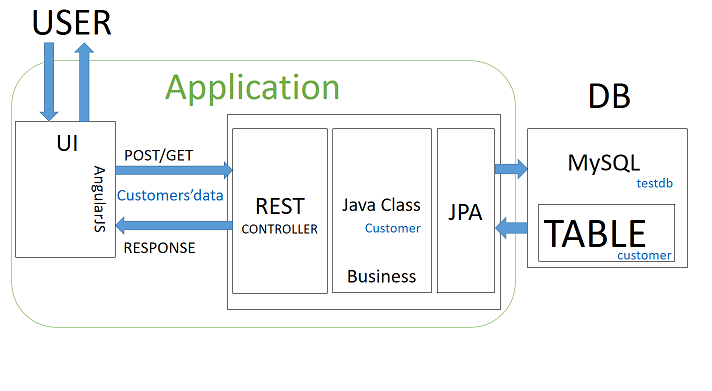
* **Methodology Used For Web Application Development:**

Agile Methodology

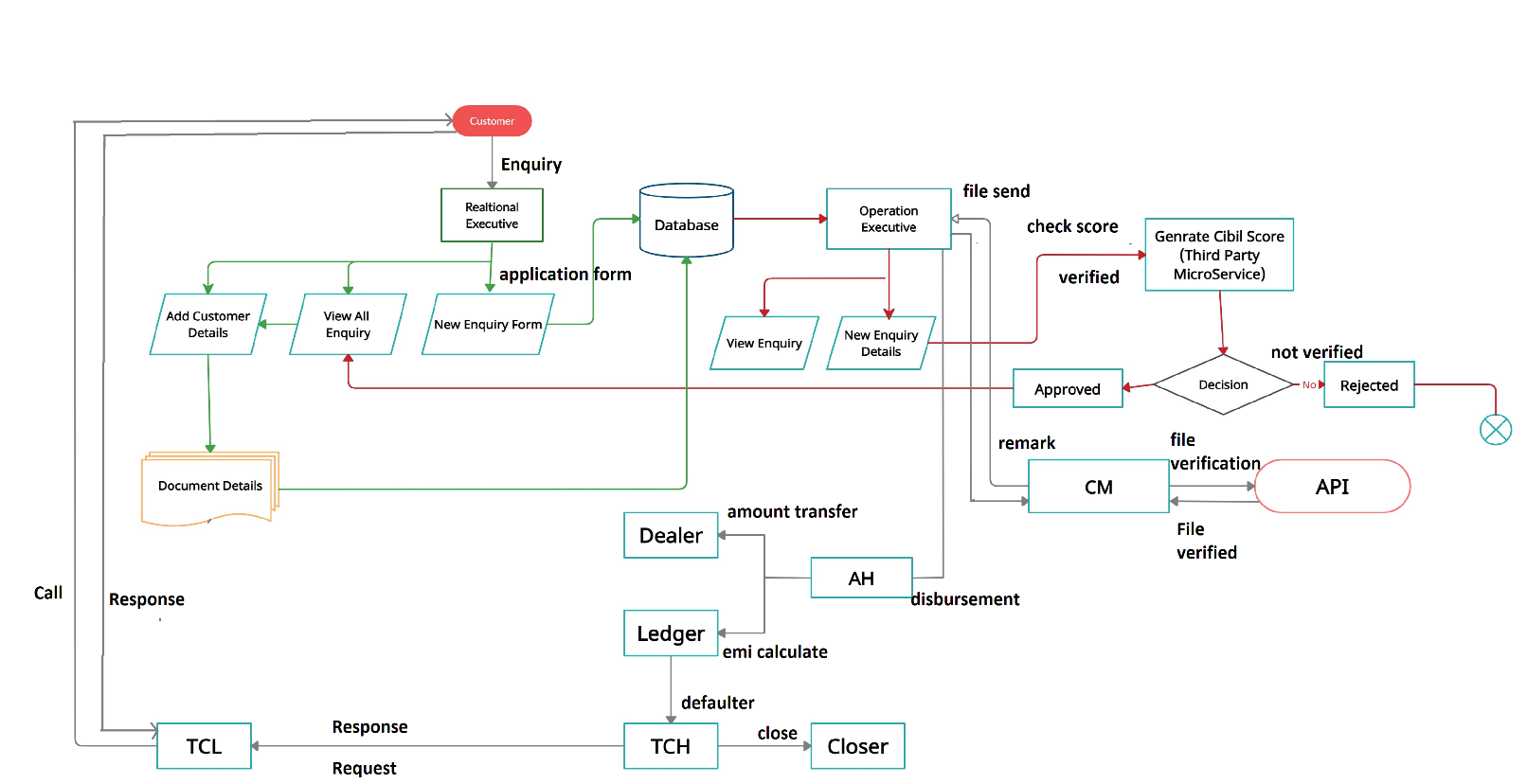


**Chapter 3. Analysis And Design**

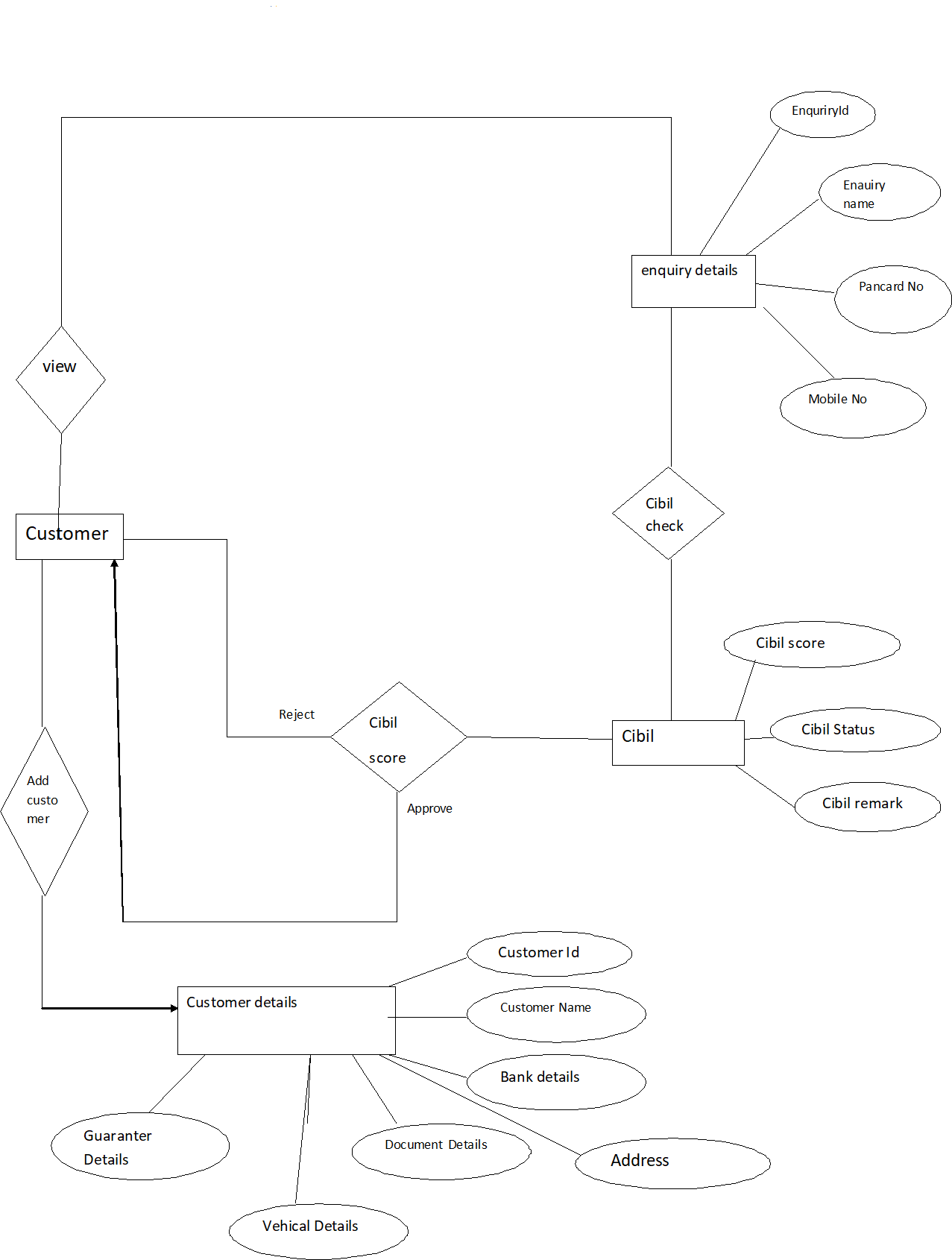
**3.1 Project Architecture :**



**3.2 Data Flow Diagram :**

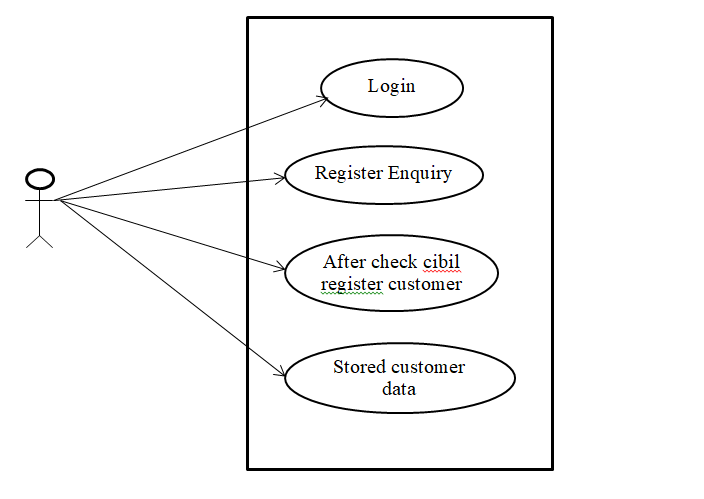


**3.3** **Entity Relationship Diagram :**

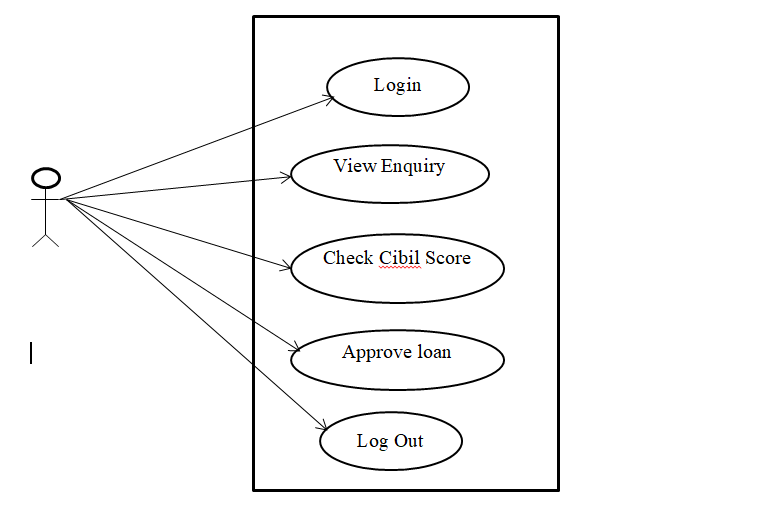
****

**3.4 USE CASE DIAGRAM**

**RE :**



**OE :**



**3.5 Activity Diagram :**

**Admin side :**

Log in

No

Yes

Log out

Manage Loan Type Details

Manage EMI Calculation

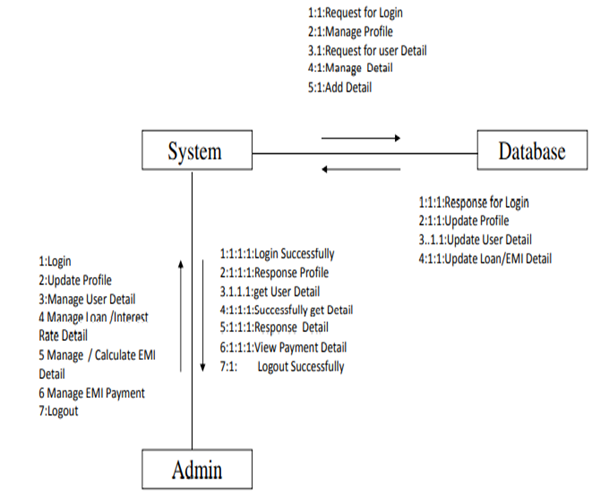
Manage EMI Payment

Manage all User

Manage Report

**3.6 Collaboration Diagram :**

**Admin Side :**



**3.7 Table Specification :**

**Table Name : Enquiry\_Details**

**Primary Key :** enq\_id

**Description :**These table manage the Enquiry(Customer’s) Details Information

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| enq\_id | Integer | Primary Key | Stores enquiry ID |
| enq\_name | String | Not Null | Stores name. |
| enqpancardno | String | Not Null | Stores Pancard Number. |
| enq\_mobileno | String | Not Null | Stores Mobile number. |
| enq\_age | Integer | Not Null | Stores age of customer. |
| enq\_status | String | Not Null | Stores status. |
| enq\_email | String | Not Null | Stores email address. |
| enq\_vehicle\_name | String | Not Null | Stores name of vehicle. |

**Table Name : Cibil Score**

**Primary Key :** CibilId

**Description :**These table manage the Cibil’s(Customer’s) Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| cibilId | Integer | Primary Key | Stores Cibil ID |
| customerPanNo | String | Not Null | Stores Pancard no. |
| Score | Integer | Not Null | Stores cibil score. |
| Cibil\_status | String | Not Null | Stores cibil status. |
| Remark | String | Not Null | Stores remark. |

**Table Name : Customer Details**

**Primary Key :** CustomerId

**Description :**These table manage the Customer’s Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| customerId | Integer | Primary Key | Stores Customer id. |
| customerName | String | Not Null | Stores customer name. |
| customerMobileno | Integer | Not Null | Stores mobile no. |
| customerLoanAmount | Integer | Not Null | Stores loan amount. |
| customerAge | Integer | Not Null | Stores customer age. |
| customerDateofBirth | String | Not Null | Stores Dob of customer. |
| customerEmail | String | Not Null | Stores email address. |
| customerPancardno | String | Not Null | Stores pancard number |
| Address | Address | Foreign Key | It gives reference to the addressID. |
| Vehical | VehicalDetails | Foreign Key | It gives reference to the vehicleID. |
| Bank | Bank | Foreign Key | It gives reference to the bankId. |
| Guaranter | GuaranterDetails | Foreign Key | It gives reference to the guaranterId. |
| loandetails | Previousloandetails | Foreign Key | It gives reference to the preloanId. |

**Table Name : Address Details**

**Primary Key :** addressId

**Description :**These table manage the Address Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| addressId | Integer | Primary Key | Stores address id. |
| localHouseNo | String | Not Null | Stores local house no. |
| localArea | String | Not Null | Stores local area. |
| localLandmark | String | Not Null | Stores local landmark. |
| Localcity | City | Foreign Key | It gives reference to the cityId. |
| permanantHouseNo | String | Not Null | Stores permanent houseno |
| permanantArea | String | Not Null | Stores permanent area. |
| permanantLandmark | String | Not Null | Stores permanent landmark. |
| permananatCity | City | Foreign Key | It gives reference to the cityId. |
| Country | Country | Foreign Key | It gives reference to the countryId. |

**Table Name : Guaranter Details**

**Primary Key :** guaranterId

**Description :**These table manage the Guaranter’s Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| guaranterId | Integer | Primary Key | Stores guaranterId. |
| guranterName | String | Not Null | Stores guaranter name. |
| guaranterAddress | String | Not Null | Stores guaranter address |
| guaranterRealtionshipwithcustomer | String | Not Null | Stores relationship with customer. |
| guaranterMobileNo | Long | Not Null | Stores mobile no. |
| aadharCardNo | Long | Not Null | Stores aadhar card number. |
| jobDetails | String | Not Null | Stores job description. |

**Table Name : Previous Loan Details**

**Primary Key :** previousloanId

**Description :**These table manage the Previous loan Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| previousloanId | Integer | Primary Key | Stores loan id. |
| previousLoanamount | Integer | Not Null | Stores previous loan amount. |
| previousLoanStatus | String | Not Null | Stores loan status. |
| Tenure | Integer | Not Null | Stores tenaure. |
| paidAmount | Integer | Not Null | Stores paid amount. |
| remainingAmount | Integer | Not Null | Stores remaining amount. |
| bank | Bank | Foreign Key | It reference to the bankId. |

**Table Name : Bank Details**

**Primary Key :** bankId

**Description :** These table manage the Bank’s Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| bankId | Integer | Primary Key | Stores bank id. |
| bankName | String | Not Null | Stores bank name. |
| accountNumber | Long | Not Null | Stores account number. |
| bankIfsc | String | Not Null | Stores bank ifsc code. |
| bankMicr | String | Not Null | Stores bank micr code. |
| bankAddress | String | Not Null | Stores bank address. |

**Table Name : Document Details**

**Primary Key :** documented.

**Description :**These table manage the Document’s Details Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| documentId | Integer | Primary Key | Stores document id. |
| customerId | Integer | Not Null | Stores customer id. |
| Pancard | Byte[] | Not Null | Stores pancard image. |
| Aadharcard | Byte[] | Not Null | Stores aadharcard image. |
| Phote | Byte[] | Not Null | Stores photo. |
| Signature | Byte[] | Not Null | Stores signature. |
| Postdatedcheque | Byte[] | Not Null | Stores postdated cheque. |
| Thumb | Byte[] | Not Null | Stores thumb image. |
| Bankstatement | Byte[] | Not Null | Stores bank statement image. |
| itrfile | Byte[] | Not Null | Stores itr file image. |
| Salaryslip | Byte[] | Not Null | Stores salary slip image. |

**Table Name : branch**

**Primary Key :**branchId

**Description :**This table manage the Branchs Information

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type(Size) | Constraint | Description |
| branchId | Integer | Primary Key | Store Branch ID |
| name | String | Not Null | Stores Branch Name |

**Table Name : elesctronicItem**

**Primary Key :**electronicItemId

**Description :**This table manage the All State Related Information

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| electronicItemId | Integer | Primary Key | Stores electronic Item ID |
| product | String | Not Null | Stores Product Name |
| price | Integer | Not null | It stores price |

**Table Name : loanType**

**Primary Key :**loanTypeId

**Description :**This table manage the Loan Types Information

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| loanTypeId | Integer | Primary Key | Stores loan Type ID(Auto increment) |
| loanTitle | Varchar(40) | Not Null | Stores loan title Name |