VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", BELAGAVI – 590018

A MINI PROJECT REPORT

ON

"LOAN MANAGEMENT SYSTEM"

Submitted in partial fulfillment of requirements for the *course*
DBMS Laboratory with Mini Project [18CSL58] *of Fifth Semester of Bachelor of Engineering in Computer Science & Engineering during the academic year 2022-23.*

Submitted By

Dhyan Medappa B
4MH20CS027

Mohammed Huzaifa Baig
4MH20CS062

Under the Guidance of

Prof.Sahana Thrilok
Assistant Professor,
Dept. of CS&E,

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

2022 - 2023

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project work entitled **"LOAN MANAGEMENT SYSTEM"** is a bonafide work carried out by **Dhyan Medappa B [4MH20CS027]** and **Mohammed Huzaifa Baig [4MH20CS062]** in partial fulfillment for the **DBMS Laboratory with Mini Project (18CSL58)** prescribed by the Visvesvaraya Technological University, Belagavi during the year 2022-2023 for the fifth semester B.E in Computer Science and Engineering. The mini project report has been approved as it satisfies the academic requirements.

Signature of Guide	Signature of HOD
(Prof.Sahana Thrilok)	**(Dr.Shivamurthy R C)**
Assistant Professor, Dept. of CS&E	Professor & Head, Dept. of CS&E
MIT Mysore	MIT Mysore

Name of the Examiners	Signature with date
1……	……
2……	……

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the people who helped and guided us in completing this mini project work.

We are thankful to **Dr. B.G. Naresh Kumar, Principal, Maharaja Institute of Technology Mysore**, for having supported us in our academic endeavors.

We are extremely thankful to **Dr.Shivamurthy R C, Professor & Head, Department of Computer Science and Engineering,** for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide **Prof.Sahana Thrilok, Assistant Professor, Department of Computer Science and Engineering,** for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members** of the **Department of Computer Science and Engineering** for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

Dhyan Medappa B (4MH20CS027)

Mohammed Huzaifa Baig [4MH20CS062]

ABSTRACT

The Loan management system is important and helps to ensure success or failure of any credit institution.

Mortgage loan problems have always been a key note on the risk of loan loss.

The scope of this project is to provide good communication and communication between the customer and the manager.

The current system can be a user-friendly system, which does not store data in the proper security and can easily track information and contains the operation of fast-recovery information, such as customer data, all loan details and includes many documents.

The Financial Management System is designed to perform the functions of the back offices of a bank and a non-cash financial institution offers any sort of loan. The system can make daily operations more efficient and provide faster response. Including adding, editing, retrieving customer information, maintaining and issuing new loans, change the loan rate.

~~~~~ TABLE OF CONTENTS ~~~~~

Chapter 1

INTRODUCTION

1.1 Aim of the Project

The project aims at managing and maintaining the finances of a group member. The usage of a reliable Database Management System, ensures security, allowing access only to authorized members.

✓ The project is totally built at the administrative end and thus only the administrator is guaranteed access.

✓ The main objective of the project on Financial Management System is to manage the details of lending loans & total transaction of a group.

✓ The purpose of the project is to build an application program to reduce the manual work for managing the transactions, maintaining a book like in traditional way, saves time

1.2 Overview of the Project

The main objective of developing this project is to handle all details of Loans and Investments in the bank. The project has been developed to smoothen the processing of Loans in banks. Our proposed project automates the loan process from both bankers as well as the customer's side. Customers can apply for a loan and after approval they can track their details online. It is a very efficient process to handle all loan related transactions in a very accurate and convenient way. Bank loan management system is an interface which facilitates a customer to apply for a loan online and to track the status from time-to-time. This system provides details about the customers, their loan details,EMI details and its rate details.Getting a loan is a very tiring and complicated process in India. It may take weeks or even months for loans to get approved and people have to visit the loan office again and again for documents and verification. Using this system admin can find customer details easily and it’s a paperless system so workload is reduced. The front-end, or the user interface is coded in PHP, CSS, JS. which gives the user a necessary platform to interact, and modify contents of the database (MySQL), which assures reliability.

1.3 Outcome of this Project

The outcome of all the hard work done for the Loan management system is here. It is software that helps the user to work with the different banks and their branches easily. This software reduces the amount of manual data entry and gives greater efficiency.

The User Interface of it is very friendly and can be easily used by anyone. It also decreases the amount of time taken to write customer details and other modules. In the end, we can say that the software is performing all the tasks accurately and is doing the work for which it is made.

1.4 Software Requirements

+ Name of Project: Car Rental Management System
+ Language Used: PHP
+ Database Used: My SQL
+ Design Interface: JavaScript, HTML, Ajax, jQuery
+ Browser: Opera Mozilla Google Chrome
+ Software: WAMP/ XAMPP/ LAMP/MAMP

Chapter 2

DESIGN

2.1 Schema Diagram

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database.

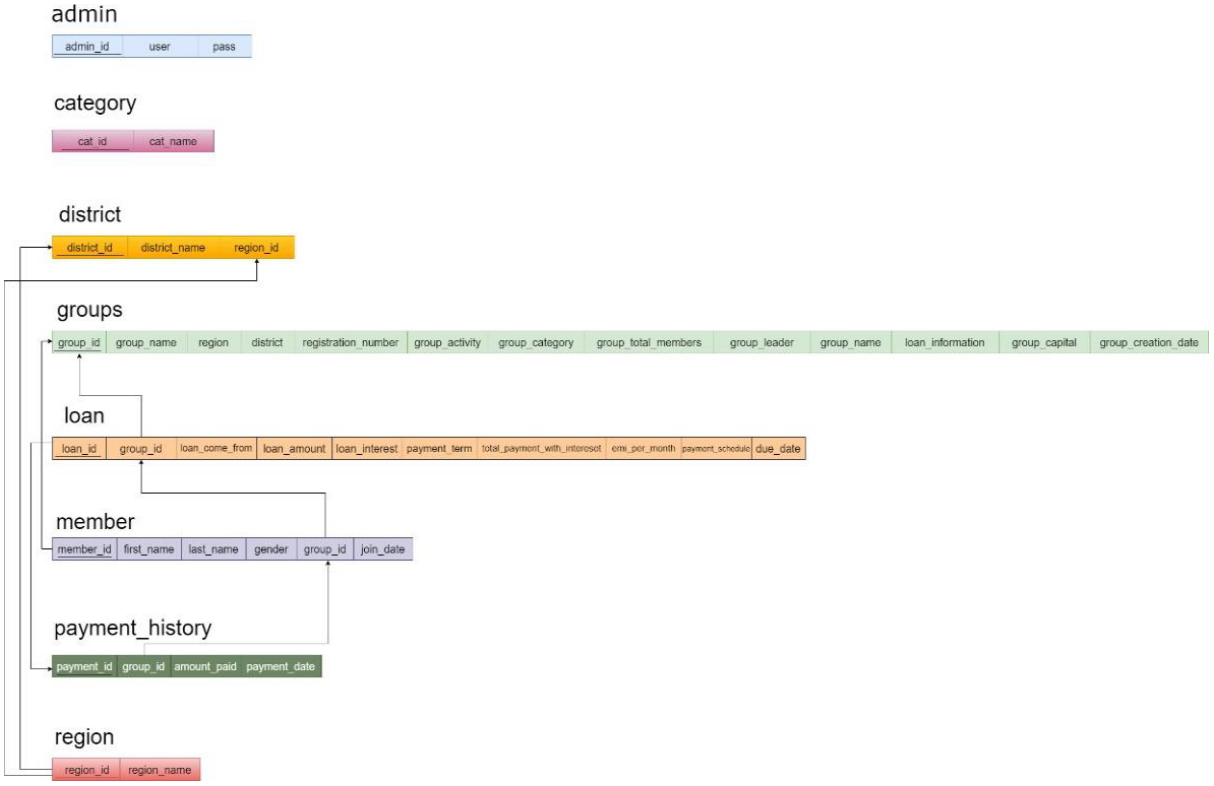
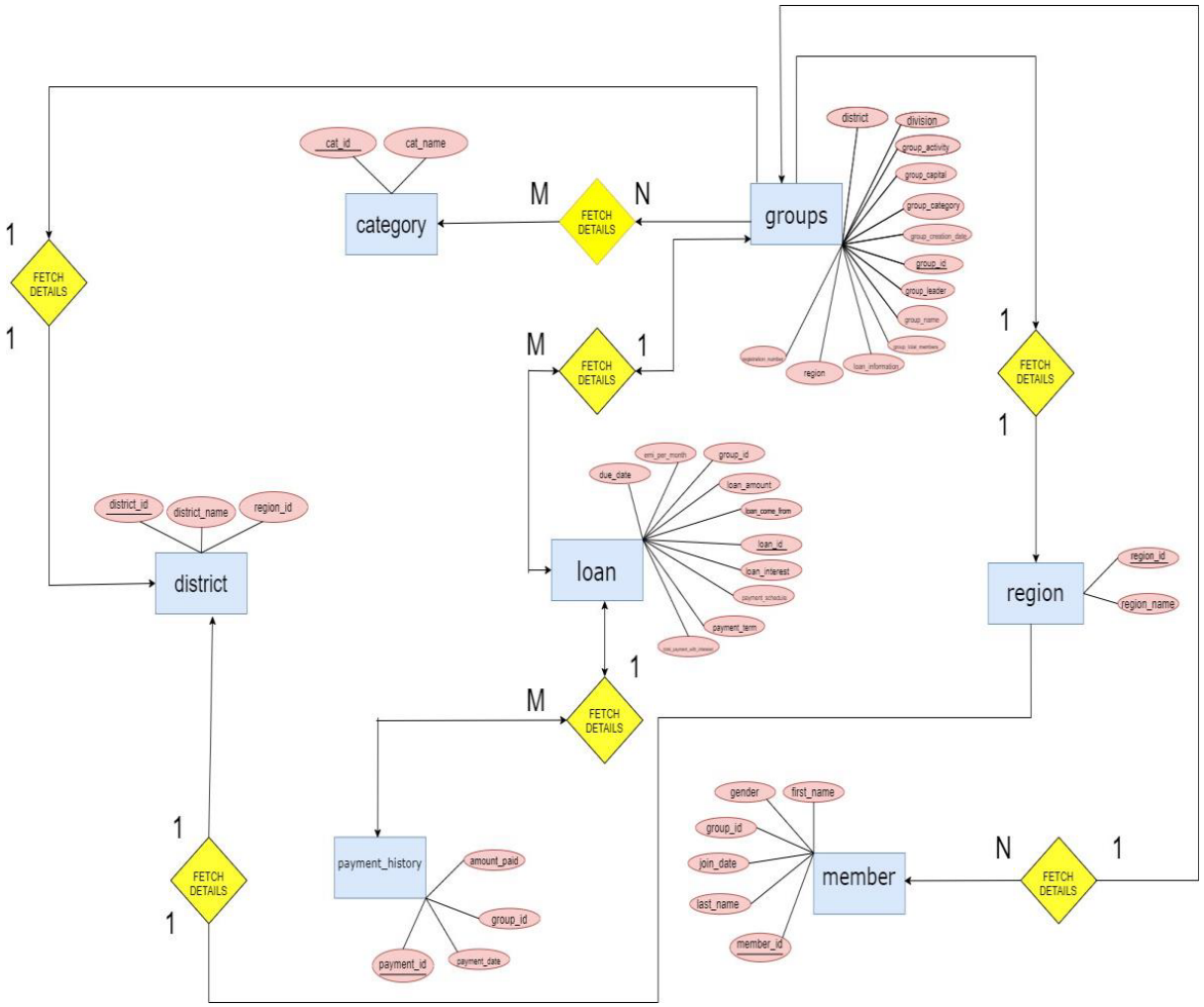


Fig 2.1 Schema Diagram

2.2 Entity Relationship Diagram

ER-Diagram is a pictorial representation of data that describes how data is communicated and related to each other. Any object, such as entities, attributes of an entity, sets of relationship, and other attributes of relationship, can be characterized with the help of the ER diagram.



2.3 Use Case Diagram

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

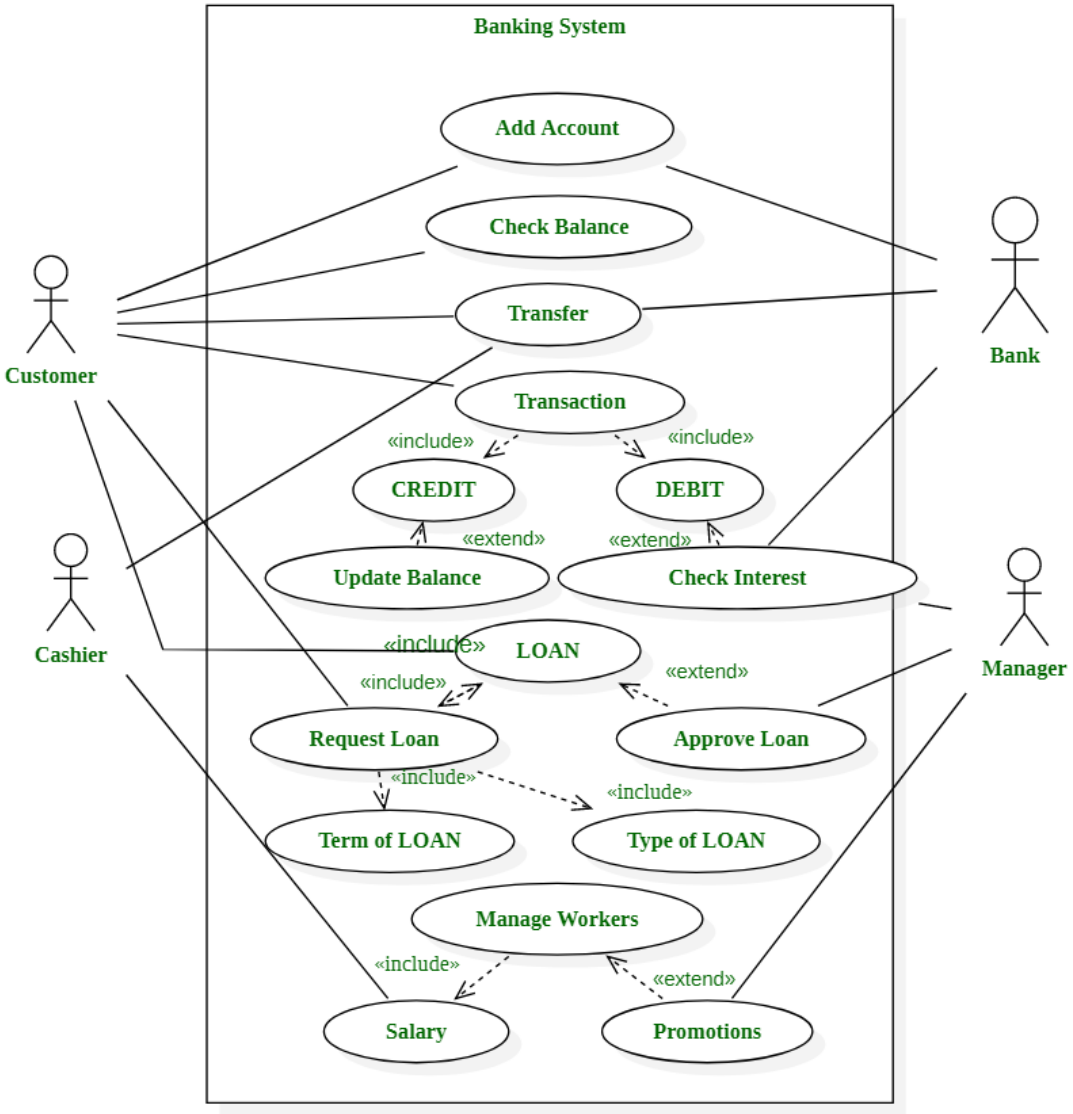


Fig 2.3 Use Case Diagram

2.4 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called a data flow graph or bubble chart.

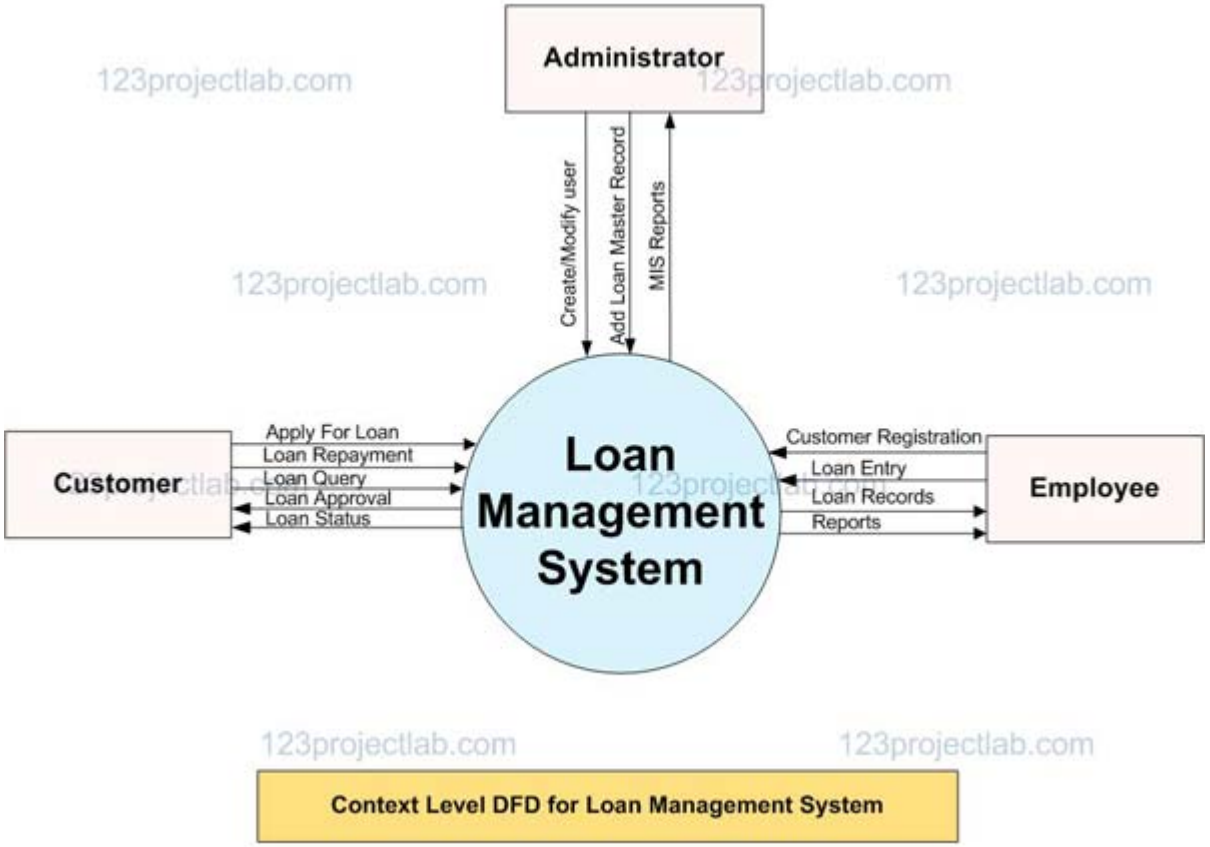


Fig 2.4 Data Flow Diagram

2.5 Sequence Diagram

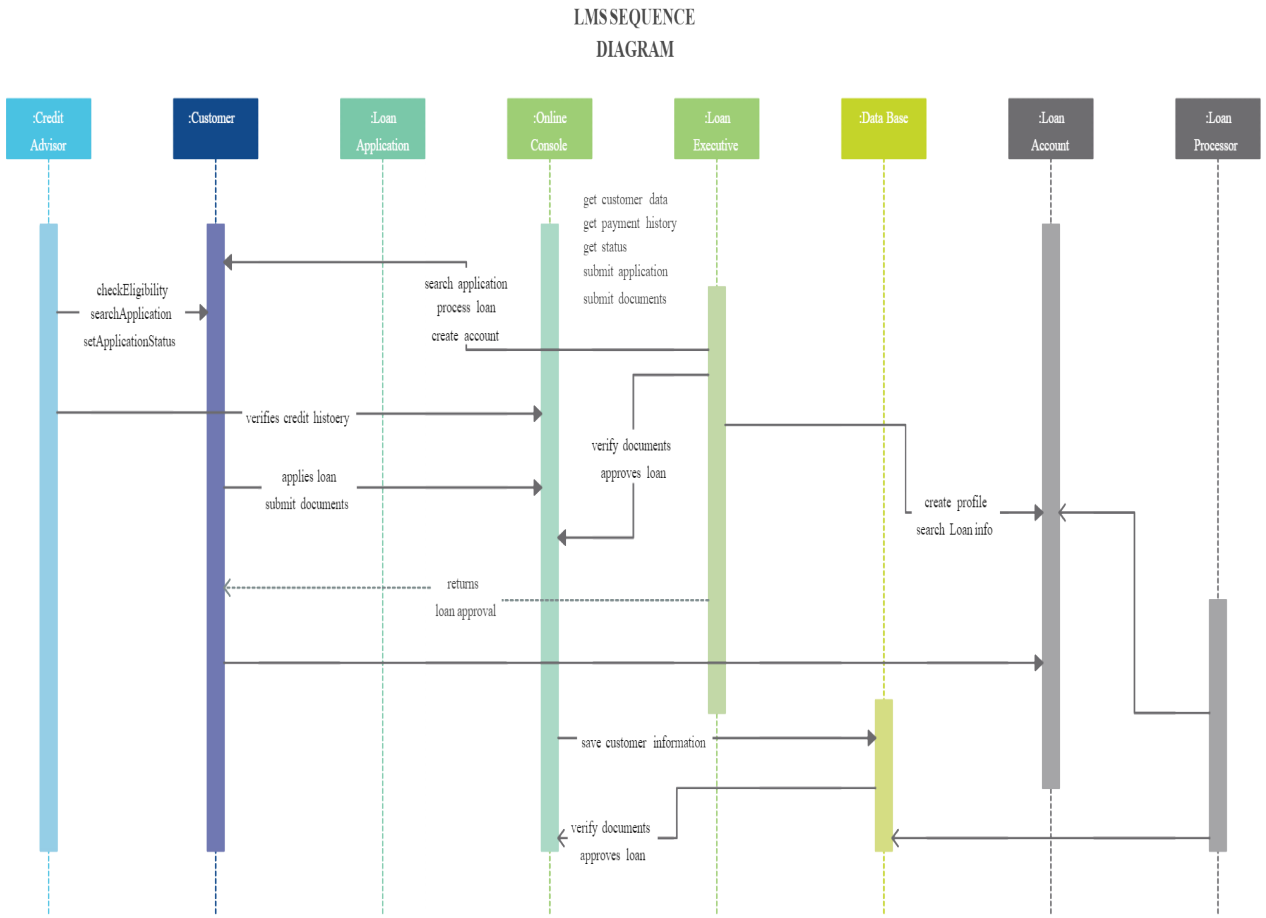


Fig 2.5 Sequence Diagram

Chapter 3

IMPLEMENTATION

3.1 Description of tables

3.1.1 Borrowers

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	firstname	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	middlename	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	lastname	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	contact_no	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
☐	6	address	text	utf8mb4_general_ci		No	None			Change Drop More
☐	7	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	8	tax_id	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	9	date_created	int(11)			No	None			Change Drop More

Fig 3.1.1

3.1.2 Loan_List

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change
☐	2	ref_no	varchar(50)	utf8mb4_general_ci		No	None			Change
☐	3	loan_type_id	int(30)			No	None			Change
☐	4	borrower_id	int(30)			No	None			Change
☐	5	purpose	text	utf8mb4_general_ci		No	None			Change
☐	6	amount	double			No	None			Change
☐	7	plan_id	int(30)			No	None			Change
☐	8	status	tinyint(1)			No	0	0= request, 1= confrimed,2=released,3=compleated,4=denied		Change
☐	9	date_released	datetime			No	None			Change
☐	10	date_created	datetime			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change

Fig 3.1.2

3.1.3 Loan_Plan

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	months	int(11)			No	None			Change Drop More
☐	3	interest_percentage	float			No	None			Change Drop More
☐	4	penalty_rate	int(11)			No	None			Change Drop More

Fig 3.1.3

3.1.4 Loan_Schedules

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	loan_id	int(30)			No	None			Change Drop More
☐	3	date_due	date			No	None			Change Drop More

Fig 3.1.4

3.1.5 Loan_Types

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	type_name	text	utf8mb4_general_ci		No	None			Change Drop More
☐	3	description	text	utf8mb4_general_ci		No	None			Change Drop More

Fig 3.1.5

3.1.6 Payments

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	loan_id	int(30)			No	None			Change Drop More
☐	3	payee	text	utf8mb4_general_ci		No	None			Change Drop More
☐	4	amount	float			No	0			Change Drop More
☐	5	penalty_amount	float			No	0			Change Drop More
☐	6	overdue	tinyint(1)			No	0	0=no , 1 = yes		Change Drop More
☐	7	date_created	datetime			No	current_timestamp()			Change Drop More

Fig 3.1.6

3.1.7 Users

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	doctor_id	int(30)			No	None			Change Drop More
☐	3	name	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	address	text	utf8mb4_general_ci		No	None			Change Drop More
☐	5	contact	text	utf8mb4_general_ci		No	None			Change Drop More
☐	6	username	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	7	password	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
☐	8	type	tinyint(1)			No	2	1=admin , 2 = staff		Change Drop More

Fig 3.1.7

3.2 Constraints on Tables

- In table Borrowers, id is a primary key.
- In table Loan_List, id is a primary key.
- In table Loan_Plan, id is a primary key.
- In table Loan_Schedules , id is a primary key.
- In table Loan_Types, id is a primary key.
- In table Payments, id is a primary key.
- In table Users, id is a primary key.

3.3 Back End Implementations

Table structure for table `borrowers`
--

```
CREATE TABLE `borrowers` (
  `id` int(30) NOT NULL,
  `firstname` varchar(100) NOT NULL,
  `middlename` varchar(100) NOT NULL,
  `lastname` varchar(100) NOT NULL,
  `contact_no` varchar(30) NOT NULL,
  `address` text NOT NULL,
  `email` varchar(50) NOT NULL,
  `tax_id` varchar(50) NOT NULL,
  `date_created` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `borrowers`
--

INSERT INTO `borrowers` (`id`, `firstname`, `middlename`, `lastname`, `contact_no`,
`address`, `email`, `tax_id`, `date_created`) VALUES
(1, 'John', 'C', 'Smith', '+16554 454654', 'Sample address', 'jsmith@sample.com', '789845-23',
0);

-- --------------------------------------------------------

--
-- Table structure for table `loan_list`
--

CREATE TABLE `loan_list` (
  `id` int(30) NOT NULL,
  `ref_no` varchar(50) NOT NULL,
  `loan_type_id` int(30) NOT NULL,
  `borrower_id` int(30) NOT NULL,
  `purpose` text NOT NULL,
```

```
  `amount` double NOT NULL,
  `plan_id` int(30) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0= request, 1=
confrimed,2=released,3=complteted,4=denied\r\n',
  `date_released` datetime NOT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `loan_list`
--

INSERT INTO `loan_list` (`id`, `ref_no`, `loan_type_id`, `borrower_id`, `purpose`, `amount`,
`plan_id`, `status`, `date_released`, `date_created`) VALUES
(3, '81409630', 1, 1, 'Sample Only', 100000, 1, 2, '2020-09-26 09:06:00', '2020-09-26
15:06:29');

-- --------------------------------------------------------

--
-- Table structure for table `loan_plan`
--

CREATE TABLE `loan_plan` (
  `id` int(30) NOT NULL,
  `months` int(11) NOT NULL,
  `interest_percentage` float NOT NULL,
  `penalty_rate` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `loan_plan`
--

INSERT INTO `loan_plan` (`id`, `months`, `interest_percentage`, `penalty_rate`) VALUES
(1, 36, 8, 3),
(2, 24, 5, 2),
(3, 27, 6, 2);

-- --------------------------------------------------------

--
-- Table structure for table `loan_schedules`
--

CREATE TABLE `loan_schedules` (
  `id` int(30) NOT NULL,
  `loan_id` int(30) NOT NULL,
  `date_due` date NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `loan_schedules`
--

INSERT INTO `loan_schedules` (`id`, `loan_id`, `date_due`) VALUES
(2, 3, '2020-10-26'),
(3, 3, '2020-11-26'),
(4, 3, '2020-12-26'),
(5, 3, '2021-01-26'),
(6, 3, '2021-02-26'),
(7, 3, '2021-03-26'),
(8, 3, '2021-04-26'),
(9, 3, '2021-05-26'),
(10, 3, '2021-06-26'),
(11, 3, '2021-07-26'),
(12, 3, '2021-08-26'),
(13, 3, '2021-09-26'),
(14, 3, '2021-10-26'),
(15, 3, '2021-11-26'),
(16, 3, '2021-12-26'),
(17, 3, '2022-01-26'),
(18, 3, '2022-02-26'),
(19, 3, '2022-03-26'),
(20, 3, '2022-04-26'),
(21, 3, '2022-05-26'),
(22, 3, '2022-06-26'),
(23, 3, '2022-07-26'),
(24, 3, '2022-08-26'),
(25, 3, '2022-09-26'),
(26, 3, '2022-10-26'),
(27, 3, '2022-11-26'),
(28, 3, '2022-12-26'),
(29, 3, '2023-01-26'),
(30, 3, '2023-02-26'),
(31, 3, '2023-03-26'),
(32, 3, '2023-04-26'),
(33, 3, '2023-05-26'),
(34, 3, '2023-06-26'),
(35, 3, '2023-07-26'),
(36, 3, '2023-08-26'),
(37, 3, '2023-09-26');

-- --------------------------------------------------------

--
-- Table structure for table `loan_types`
--
```

```
CREATE TABLE `loan_types` (
  `id` int(30) NOT NULL,
  `type_name` text NOT NULL,
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `loan_types`
--

INSERT INTO `loan_types` (`id`, `type_name`, `description`) VALUES
(1, 'Small Business', 'Small Business Loans'),
(2, 'Mortgages', 'Mortgages'),
(3, 'Personal Loans', 'Personal Loans');

-- --------------------------------------------------------

--
-- Table structure for table `payments`
--

CREATE TABLE `payments` (
  `id` int(30) NOT NULL,
  `loan_id` int(30) NOT NULL,
  `payee` text NOT NULL,
  `amount` float NOT NULL DEFAULT 0,
  `penalty_amount` float NOT NULL DEFAULT 0,
  `overdue` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0=no , 1 = yes',
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `payments`
--

INSERT INTO `payments` (`id`, `loan_id`, `payee`, `amount`, `penalty_amount`, `overdue`,
`date_created`) VALUES
(2, 3, 'Smith, John C', 3000, 90, 1, '2020-09-26 15:51:01');

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `doctor_id` int(30) NOT NULL,
  `name` varchar(200) NOT NULL,
  `address` text NOT NULL,
```

```
  `contact` text NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(200) NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1=admin , 2 = staff'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `doctor_id`, `name`, `address`, `contact`, `username`,
`password`, `type`) VALUES
(1, 0, 'Administrator', '', '', 'admin', 'admin123', 1);

--
-- Indexes for dumped tables
--

--
-- Indexes for table `borrowers`
--
ALTER TABLE `borrowers`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `loan_list`
--
ALTER TABLE `loan_list`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `loan_plan`
--
ALTER TABLE `loan_plan`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `loan_schedules`
--
ALTER TABLE `loan_schedules`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `loan_types`
--
ALTER TABLE `loan_types`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `payments`
```

```
--
ALTER TABLE `payments`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `borrowers`
--
ALTER TABLE `borrowers`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `loan_list`
--
ALTER TABLE `loan_list`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `loan_plan`
--
ALTER TABLE `loan_plan`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `loan_schedules`
--
ALTER TABLE `loan_schedules`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=38;

--
-- AUTO_INCREMENT for table `loan_types`
--
ALTER TABLE `loan_types`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `payments`
--
ALTER TABLE `payments`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;
```

```
--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

3.4 Front End Implementations

3.4.1 Index.php Front End

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Admin | Loan Management System</title>


<?php
          session_start();
  if(!isset($_SESSION['login_id']))
    header('location:login.php');
 include('./header.php');
 // include('./auth.php');
 ?>

</head>
<style>
          body{
        background: #80808045;
  }
  .modal-dialog.large {
    width: 80% !important;
    max-width: unset;
  }
  .modal-dialog.mid-large {
    width: 50% !important;
    max-width: unset;
  }
</style>

<body>
```

```
        <?php include 'topbar.php' ?>
        <?php include 'navbar.php' ?>
  <div class="toast" id="alert_toast" role="alert" aria-live="assertive" aria-atomic="true">
    <div class="toast-body text-white">
    </div>
  </div>
  <main id="view-panel" >
      <?php $page = isset($_GET['page']) ? $_GET['page'] :'home'; ?>
          <?php include $page.'.php' ?>


  </main>

  <div id="preloader"></div>
  <a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>

<div class="modal fade" id="confirm_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
      <div class="modal-content">
        <div class="modal-header">
        <h5 class="modal-title">Confirmation</h5>
      </div>
      <div class="modal-body">
        <div id="delete_content"></div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" id='confirm'
onclick="">Continue</button>
        <button type="button" class="btn btn-secondary"
data-dismiss="modal">Close</button>
      </div>
      </div>
    </div>
  </div>
  <div class="modal fade" id="uni_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
      <div class="modal-content">
        <div class="modal-header">
        <h5 class="modal-title"></h5>
      </div>
      <div class="modal-body">
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" id='submit' onclick="$('#uni_modal
form').submit()">Save</button>
        <button type="button" class="btn btn-secondary"
data-dismiss="modal">Cancel</button>
      </div>
      </div>
    </div>
```

```
  </div>
</body>
<script>
            window.start_load = function(){
    $('body').prepend('<di id="preloader2"></di>')
  }
  window.end_load = function(){
    $('#preloader2').fadeOut('fast', function() {
        $(this).remove();
      })
  }

  window.uni_modal = function($title = '' , $url='',$size=""){
    start_load()
    $.ajax({
        url:$url,
        error:err=>{
            console.log()
            alert("An error occured")
        },
        success:function(resp){
            if(resp){
                $('#uni_modal .modal-title').html($title)
                $('#uni_modal .modal-body').html(resp)
                if($size != ''){
                    $('#uni_modal .modal-dialog').addClass($size)
                }else{
                    $('#uni_modal .modal-dialog').removeAttr("class").addClass("modal-dialog
modal-md")
                }
                $('#uni_modal').modal('show')
                end_load()
            }
        }
    })
}
window._conf = function($msg='',$func='',$params = []){
     $('#confirm_modal #confirm').attr('onclick',$func+"("+$params.join(',')+")")
     $('#confirm_modal .modal-body').html($msg)
     $('#confirm_modal').modal('show')
  }
   window.alert_toast= function($msg = 'TEST',$bg = 'success'){
      $('#alert_toast').removeClass('bg-success')
      $('#alert_toast').removeClass('bg-danger')
      $('#alert_toast').removeClass('bg-info')
      $('#alert_toast').removeClass('bg-warning')

    if($bg == 'success')
      $('#alert_toast').addClass('bg-success')
    if($bg == 'danger')
```

```
    $('#alert_toast').addClass('bg-danger')
   if($bg == 'info')
    $('#alert_toast').addClass('bg-info')
   if($bg == 'warning')
    $('#alert_toast').addClass('bg-warning')
   $('#alert_toast .toast-body').html($msg)
   $('#alert_toast').toast({delay:3000}).toast('show');
  }
  $(document).ready(function(){
   $('#preloader').fadeOut('fast', function() {
       $(this).remove();
     })
  })
  $('.datetimepicker').datetimepicker({
     format:'Y/m/d H:i',
     startDate: '+3d'
  })
  $('.select2').select2({
   placeholder:"Please select here",
   width: "100%"
  })
</script>
</html>
```

3.4.2 Login.php Front End

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Login | Loan Management System</title>


<?php include('./header.php'); ?>
<?php include('./db_connect.php'); ?>
<?php
session_start();
if(isset($_SESSION['login_id']))
header("location:index.php?page=home");

?>

</head>
<style>
        body{
                width: 100%;
```

```
            height: calc(100%);
            /background: #007bff;/
        }
        main#main{
                width:100%;
                height: calc(100%);
                background:white;
        }
        #login-right{
                position: absolute;
                right:0;
                width:40%;
                height: calc(100%);
                background:white;
                display: flex;
                align-items: center;
        }
        #login-left{
                position: absolute;
                left:0;
                width:60%;
                height: calc(100%);
                background:#59b6ec61;
                display: flex;
                align-items: center;
                background: url(assets/img/loan.png);
            background-repeat: no-repeat;
            background-size: cover;
        }
        #login-right .card{
                margin: auto;
                z-index: 1
        }
        .logo {
    margin: auto;
    font-size: 8rem;
    background: white;
    padding: .5em 0.7em;
    border-radius: 50% 50%;
    color: #000000b3;
    z-index: 10;
}
div#login-right::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: calc(100%);
    height: calc(100%);
    background: #000000e0;
```

```
}

</style>

<body>


  <main id="main" class=" bg-info">
        <h1>Loan Management System</h1>
                <div id="login-left">
                </div>

                <div id="login-right">
                        <div class="card col-md-8 bg-info">
                                <div class="card-body">
                                                <div class="logo">
                                                        <span class="fa
fa-money-check-alt"></span>
                                                </div>
                                        <form id="login-form" >
                                                <div class="form-group">
                                                        <label for="username"
class="control-label">Username</label>
                                                        <input type="text" id="username"
name="username" class="form-control">
                                                </div>
                                                <div class="form-group">
                                                        <label for="password"
class="control-label">Password</label>
                                                        <input type="password" id="password"
name="password" class="form-control">
                                                </div>
                                                <center><button class="btn-sm btn-block
btn-wave col-md-4 btn-primary">Login</button></center>
                                        </form>
                                </div>
                        </div>
                </div>


  </main>

  <a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>


</body>
<script>
        $('#login-form').submit(function(e){
                e.preventDefault()
```

```
		$('#login-form button[type="button"]').attr('disabled',true).html('Logging in...');
		if($(this).find('.alert-danger').length > 0 )
			$(this).find('.alert-danger').remove();
		$.ajax({
			url:'ajax.php?action=login',
			method:'POST',
			data:$(this).serialize(),
			error:err=>{
				console.log(err)
		$('#login-form button[type="button"]').removeAttr('disabled').html('Login');

			},
			success:function(resp){
				if(resp == 1){
					location.href ='index.php?page=home';
				}else if(resp == 2){
					location.href ='voting.php';
				}else{
					$('#login-form').prepend('<div class="alert alert-danger">Username or password is incorrect.</div>')
					$('#login-form button[type="button"]').removeAttr('disabled').html('Login');
				}
			}
		})
	})
</script>
</html>
```

Chapter 4

RESULT ANALYSIS

4.1 Snapshots

4.1.1 Login(Authentication) Page for Admin.

Fig 4.1.1

4.1.2 Dashboard

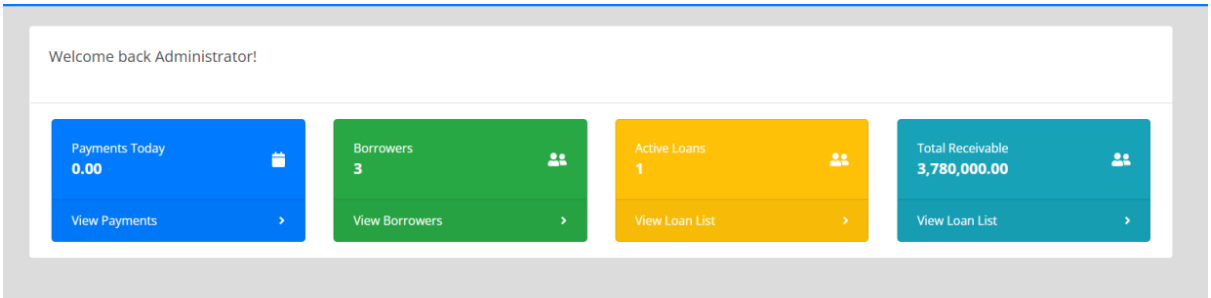


Fig 4.1.2

4.1.3 Loan application list

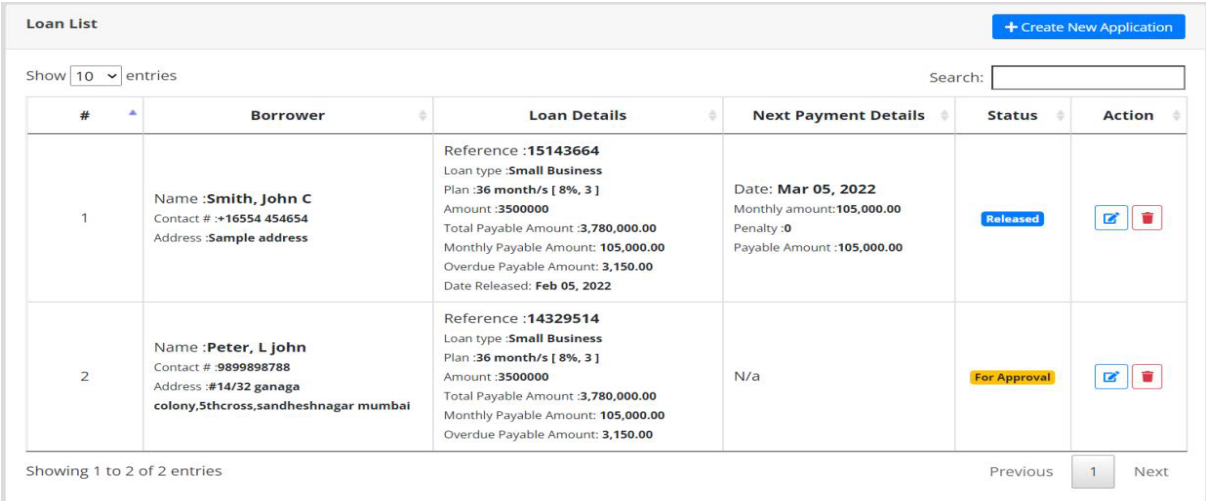


Fig 4.1.3

4.1.4 Adding a New Member

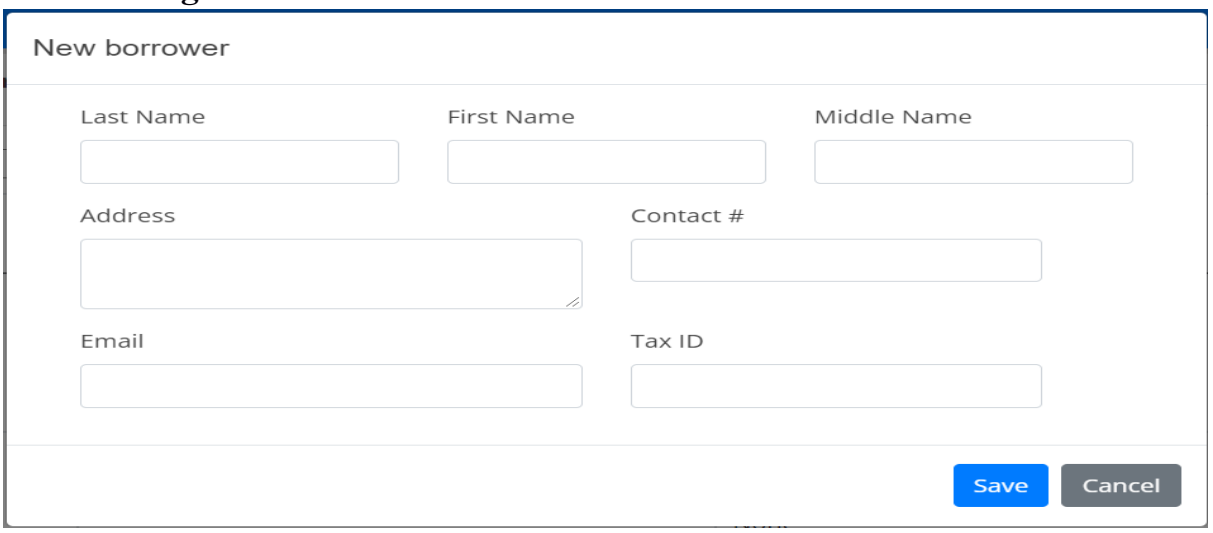
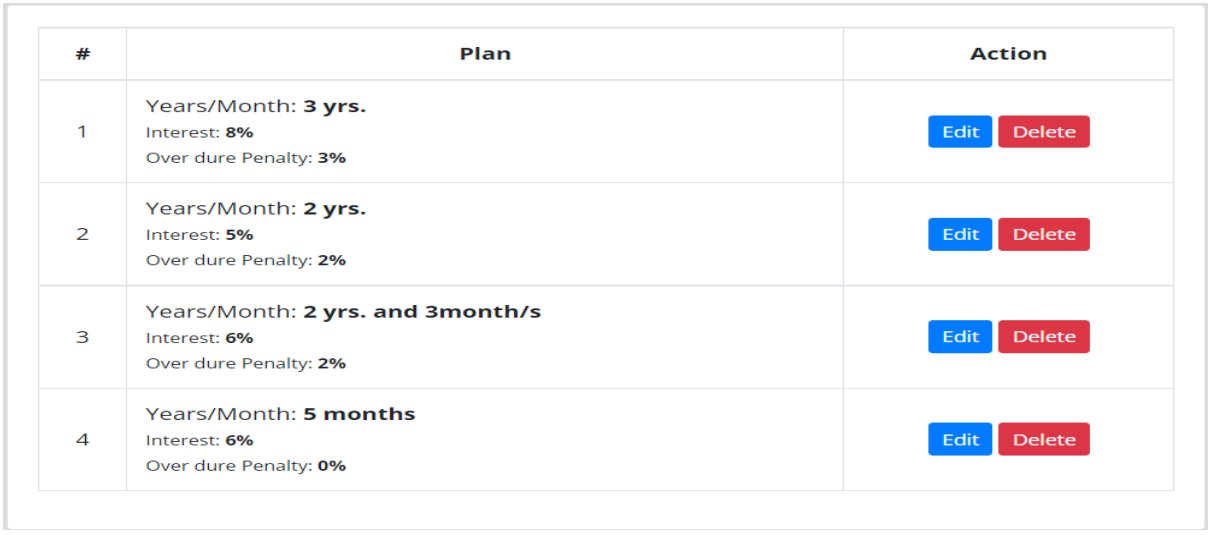


Fig 4.1.4

4.1.5 Loan Plans



#	Plan	Action
1	Years/Month: **3 yrs.** Interest: **8%** Over dure Penalty: **3%**	Edit Delete
2	Years/Month: **2 yrs.** Interest: **5%** Over dure Penalty: **2%**	Edit Delete
3	Years/Month: **2 yrs. and 3month/s** Interest: **6%** Over dure Penalty: **2%**	Edit Delete
4	Years/Month: **5 months** Interest: **6%** Over dure Penalty: **0%**	Edit Delete

Fig 4.1.5

4.1.6 Types of loans

#	Loan Type	Action
1	Type Name: **Small Business** Description: **Small Business Loans**	Edit Delete
2	Type Name: **Mortgages** Description: **Mortgages**	Edit Delete
3	Type Name: **Personal Loans** Description: **Personal Loans**	Edit Delete
4	Type Name: **Home loans** Description: **Home loans**	Edit Delete
5	Type Name: **car loans** Description: **car loans**	Edit Delete

Fig 4.1.6

4.2 Discussion

- **Login Page for Admin**
  Fig 4.1.1 shows a page where the admin needs to put his/her Username and Password.

- **Dashboard**
  Fig 4.1.2 shows a page where the database of Payments done, Borrowers, Active loans and Total receivable amount.

- **Loan application list**
  Fig 4.1.3 shows a page where the admin can see Borrower's Information, Loan Details, Next Payment Details, Status and Action.

- **Adding a New Member**
  Fig 4.1.4 shows a page where a new borrower can be added to the database.

- **Loan Plans**
  Fig 4.1.5 shows a page where Loan duration, Interest(in %) and overdue penalty (in %) can be viewed and/or edited.

- **Types of loans**
  Fig 4.1.6 shows a page where the type of loan can be seen with the description.

4.3 Testing

Sl.no	Test Name	Actual Output	Expected Output	Status
1	Admin Login	Admin is successfully authenticated	Admin is successfully authenticated	Pass
2	Updating the Admin Password	Admin password updated successfully	Admin password updated successfully	Pass
3	Adding a new Group	New group created successfully	New group created successfully	Pass
4	Adding a new Member	New member added successfully	New member added successfully	Pass
5	Creating a new Loan	New loan created successfully	New loan created successfully	Pass
6	Adding a Payment	New payment method added successfully	New payment method added successfully	Pass
7	Creating a Payment History	New payment history based on last payment created successfully	New payment history based on last payment created successfully	Pass

Table 4.3 Test Case

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, the Loan Management System project was a success. The LMS achieved all the expected functions and quality requirements. With each implementation, a considerable effort was made to enhance extensibility. As an outcome, the end-product had high flexibility, scalability, modifiability and maintainability. Deployment and maintenance will be done in the upcoming future. In our opinion, there is always room for potential improvements in functional logic, code structure, user experience design and application performance. We will continue to do code refactoring to reduce complexity, improve readability and maintainability.

At the end it is concluded that we have made effort on following points:-

- A description of the background and context of the project and its relation to work already done in the area.
- Made a statement of the aims and objectives of the project. The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which
- describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interfaces and security issues related to the system.
- Finally the system is implemented and tested according to test cases.

5.2 Future Enhancement

For banks, credit unions, and other lenders, there are three basic loan process improvement ideas in banking on which they can capitalize. These new fintech capabilities help optimize the lending process for both borrowers and lenders.

- Completely replace paper with digital documents.
- Integrate data with loan origination and decisioning.
- Analyze performance of lending processes and portfolio profitability.11

REFERENCES

Websites:-

1. HTML Tutorial - https://www.w3schools.com/html

2. CSS Tutorial - https://www.w3schools.com/css

3. PHP Tutorial - https://www.w3schools.com/php

4. SQL Tutorial - https://www.w3schools.com/sql

Books:-

1. **Database Management System (5th edition) -Ramakrishna and Gehrke.**
2. **A Database System Models, Languages, Design and Application the programming-Ramez Elmasri and Shamkant B Navathe 7 edition**
3. **Pearson 2017. B . Database Management System- Ramakrishnan and Gehrke 3 the edition 2014 McGrawHill**