

Software Requirement Specification
for
Customer Relationship Management

Version 1.0 approved

Prepared by: Ayush Singh

HCL Training

Date: 13/01/2026

Submitted in partial fulfillment
of the requirements of
Software Engineering (Agile Methodology)

Table of Contents

1. Abstract
2. Introduction
 - 2.1 Introduction
 - 2.2 Problem identification
 - 2.3 Need of the project
 - 2.4 Project Scheduling
 - 2.5 Objectives
3. Software Requirement Specification
 - 3.1 Purpose
 - 3.2 Scope
 - 3.3 Hardware Requirement / Software Requirement (minimum)
 - 3.4 Tools
 - 3.5 Software Process Model
4. System Design
 - 4.1 Data Dictionary
 - 4.2 ER diagram
 - 4.3 DFD
 - 4.4 System Flow Chart/ Object Diagram/Class Diagram/State Diagram/Activity Diagram/Use Case Diagram etc.
5. Implementation
 - 5.1 Program code
 - 5.2 Output screens
6. Testing
 - 6.1 Test data
 - 6.2 Test result
7. User Manual
 - 7.1 How to use project guidelines
 - 7.2 Screen Layouts and Description
8. Project applications and Limitations
9. Conclusion and Future Enhancement
10. Bibliography & References

List Of Figures

| Figure | Page No. |
|--|-----------------|
| fig. 4.1 - Entity Relationship Diagram | 12 |
| fig. 4.2 - Data Flow Diagram | 13 |
| fig. 4.3 - System Flow Chart | 13 |

List of Tables

| Table | Page No. |
|--|-----------------|
| Table 2.1 Project Scheduling Sprint Plan | 7 |
| Table 4.1 Data Dictionary | 11 |
| Table 6.1 Test Data | 15 |
| Table 6.2 Test Result | 16 |

1. Abstract

Customer Relationship Management (CRM) is a strategic approach that enables organizations to build strong relationships with customers by effectively managing interactions, data, and communication across various touchpoints. The primary objective of this project is to design and develop a comprehensive CRM system that helps businesses streamline customer-related processes, improve service quality, and enhance overall customer satisfaction.

The proposed CRM system provides a centralized platform to store and manage customer information such as personal details, contact information, purchase history, and communication records. By maintaining a well-organized customer database, organizations can easily track customer interactions and gain valuable insights into customer behavior and preferences. This enables businesses to deliver personalized services, anticipate customer needs, and implement targeted marketing strategies.

This project is developed using Agile methodology, which allows incremental development through iterative sprints. The Agile framework ensures continuous feedback from stakeholders, enabling the team to make improvements at every stage of development. The CRM system is built using modern technologies including Java, MySQL, and web-based interfaces to ensure reliability, scalability, and security. The system supports essential modules such as customer registration, lead management, sales tracking, complaint handling, and reporting.

One of the key features of the system is automation. Routine tasks such as follow-up reminders, email notifications, and data updates are automated, reducing manual effort and improving operational efficiency. Additionally, role-based access control ensures data security by restricting access based on user roles such as administrator, sales executive, and customer support staff.

The CRM system also provides analytical tools and dashboards that generate real-time reports on sales performance, customer retention rates, and service efficiency. These insights help management make data-driven decisions and identify areas for improvement. The integration of feedback mechanisms enables organizations to capture customer opinions and measure satisfaction levels.

In conclusion, the developed Customer Relationship Management system serves as an effective solution for managing customer relationships in a structured and efficient manner. By adopting this system, organizations can enhance customer engagement, improve service quality, and achieve long-term business growth. The project demonstrates how technology-driven CRM solutions can transform traditional business operations into customer-centric models, ensuring competitive advantage in today's dynamic market environment.

2. Introduction

2.1 Introduction

In today's competitive business environment, building strong customer relationships is essential for organizational success. Customer Relationship Management (CRM) helps businesses manage customer interactions and improve service quality through technology. This project focuses on developing a CRM system that provides a centralized platform for storing and managing customer information. The system helps track customer interactions, sales activities, and support requests efficiently. Automation features reduce manual work and improve productivity. The project follows Agile methodology to ensure continuous improvement through iterative development. Modern technologies such as Java and MySQL are used to ensure scalability and security. Overall, the system aims to enhance customer satisfaction and support business growth.

2.2 Problem identification

Many organizations still rely on traditional manual or semi-digital methods to manage customer information and interactions. These methods often result in data duplication, inconsistency, and difficulty in retrieving accurate customer records. As customer data increases, managing it becomes time-consuming and inefficient.

Lack of a centralized system makes it hard to track customer history, follow-ups, complaints, and sales activities. This leads to poor customer service, missed opportunities, and reduced customer satisfaction. Manual processes are also prone to human errors and security risks.

Therefore, there is a need for an automated Customer Relationship Management system that can securely store, manage, and analyze customer data in a centralized platform to improve efficiency, accuracy, and overall business performance.

2.3 Need of the project

In today's competitive business environment, organizations must focus on building and maintaining strong relationships with their customers to achieve long-term success. Traditional methods of managing customer information are often manual, time-consuming, and prone to errors, which can lead to data duplication, loss of important records, and inefficient decision-making. As customer data continues to grow, it becomes increasingly difficult to track interactions, follow-ups, complaints, and sales activities without a proper system in place. Therefore, there is a strong need for a centralized and automated Customer Relationship Management system that securely stores and manages customer information in an organized manner. The proposed project addresses these challenges by improving data accuracy, enhancing customer service, and enabling businesses to analyze customer behavior effectively. This system helps organizations personalize services, improve customer satisfaction, and support sustainable business growth.

2.4 Project Scheduling

Sprint-wise project scheduling divides the project into short development cycles called sprints. Each sprint focuses on completing prioritized tasks from the product backlog within a fixed time frame. Regular meetings help track progress and address challenges. This approach ensures continuous improvement and timely project delivery.

| Sprint No | Duration | Sprint Goal | Activities | Deliverables |
|-----------|----------|------------------------|---|---------------------------------------|
| Sprint 1 | Week 1 | Requirement Analysis | Requirement gathering, stakeholder discussion, backlog creation | Product backlog, requirement document |
| Sprint 2 | Week 2 | System Design | Architecture design, database design, UI mockups | Design diagrams, database schema |
| Sprint 3 | Week 3 | Module Development | Customer module, login system, data validation | Working modules |
| Sprint 4 | Week 4 | Feature Implementation | Sales tracking, complaint handling, notifications | Integrated features |
| Sprint 5 | Week 5 | Testing | Unit testing, integration testing, bug fixing | Tested system |
| Sprint 6 | Week 6 | Deployment | Final deployment, user training, documentation | Live system, user manual |

Table 2.1

2.5 Objectives

The main Objectives of the Customer Relationship Management are:

- To design and develop an efficient Customer Relationship Management system
- To provide a centralized platform for storing and retrieving customer information securely.
- To automate routine tasks such as follow-ups, notifications, and report generation.
- To improve customer service by tracking interactions and resolving issues effectively.
- To analyze customer behavior and generate reports for better decision-making.
- To ensure data security through role-based access control.
- To implement the system using Agile methodology for continuous improvement.

3. Software Requirement Specification

3.1 Purpose

The purpose of this project is to design and develop an effective Customer Relationship Management system that helps organizations manage customer information in a centralized and organized manner. This system aims to improve customer interaction, enhance service quality, and increase customer satisfaction. By automating routine processes such as follow-ups, notifications, and report generation, the project reduces manual effort and improves operational efficiency. The CRM system also supports data analysis to help management make informed decisions. Overall, the project focuses on building a reliable, secure, and user-friendly solution to strengthen customer relationships and support business growth.

3.2 Scope

The scope of this project includes the design and development of a Customer Relationship Management system that supports efficient management of customer information and interactions. The system will allow organizations to store, update, and retrieve customer details through a centralized database. It covers key functionalities such as customer registration, lead management, sales tracking, complaint handling, and report generation. The project also includes role-based access control to ensure data security and restricted access for different users. Integration of automated notifications and reminders is included to improve communication and follow-up processes. The system is designed to be scalable, allowing future enhancements such as mobile application support, cloud integration, and advanced analytics. This project does not cover third-party integrations or advanced AI features in the current phase but provides a strong foundation for future upgrades.

3.3 Hardware Requirement and Software Requirement

Hardware Requirement

The following hardware components are required for the successful execution of the Customer Relationship Management (CRM) system:

- Processor: Intel Core i3 or above
- RAM: Minimum 4 GB (8 GB recommended)
- Hard Disk: Minimum 250 GB free space
- System Type: 64-bit architecture
- Input Devices: Keyboard, Mouse
- Output Devices: Monitor, Printer (optional)
- Internet Connection: Required for cloud access and updates

Software Requirement

The following software components are required for the development and execution of the Customer Relationship Management (CRM) system:

- Operating System: Windows 10 / Linux / macOS
- Programming Language: Java
- Database: MySQL
- IDE: Eclipse / IntelliJ IDEA
- Web Technologies: HTML, CSS, JavaScript
- Server: Apache Tomcat
- Version Control: Git & GitHub
- Testing Tool: Selenium
- Documentation Tool: MS Word / Google Docs

3.4 Tools

The following tools are required for the development, testing, and deployment of the Customer Relationship Management (CRM) system:

- Eclipse IDE – For writing and managing Java code
- MySQL Workbench – For database design and management
- Apache Tomcat Server – For deploying and running the web application
- Git & GitHub – For version control and project collaboration
- Selenium – For automated testing of the application
- Postman – For API testing (if applicable)
- MS Word / Google Docs – For documentation
- Draw.io – For creating system diagrams and flowcharts
- Google Chrome / Firefox – For testing the application in browsers

3.5 Software Process Model

The Agile Software Process Model is used for the development of this Customer Relationship Management (CRM) system. Agile is an iterative and incremental approach that focuses on continuous improvement, customer feedback, and flexible development. The entire project is divided into small development cycles called sprints, where each sprint delivers a functional part of the system. Requirements are gathered in the form of user stories and maintained in a product backlog, which is prioritized based on business needs. Regular meetings such as sprint planning, daily stand-ups, sprint reviews, and retrospectives are conducted to monitor progress and improve team collaboration. This model allows quick adaptation to changes, ensures timely delivery, and enhances overall product quality. By following Agile methodology, the project achieves better efficiency, transparency, and customer satisfaction.

4. System Design

4.1 Data Dictionary

The Data Dictionary defines the data elements used in the CRM system along with their descriptions.

| Table | Field | Type | Description |
|----------|-------------|--------------|---------------|
| Customer | customer_id | INT (PK) | Customer ID |
| Customer | name | VARCHAR(100) | Customer name |
| Customer | email | VARCHAR(100) | Email |
| Customer | phone | VARCHAR(15) | Contact |
| User | user_id | INT (PK) | System user |
| User | username | VARCHAR(50) | Login name |
| User | role | VARCHAR(20) | User role |
| Leads | lead_id | INT (PK) | Lead ID |
| Leads | status | VARCHAR(20) | Lead status |
| Sales | sale_id | INT (PK) | Sales ID |
| Sales | amount | DECIMAL | Amount |
| Support | ticket_id | INT (PK) | Ticket |
| Support | status | VARCHAR(20) | Ticket Status |
| Feedback | feedback_id | INT (PK) | Feedback ID |
| Feedback | rating | INT | Rating (1–5) |

Table 4.1

4.2 ER Diagram

The ER diagram shows the main entities of the CRM system such as Customer, User, Leads, Sales, Support, and Feedback along with their relationships. It explains how customer data is connected to sales, support tickets, and feedback records. One-to-many relationships ensure proper data organization and integrity. This diagram helps in designing an efficient and structured database.

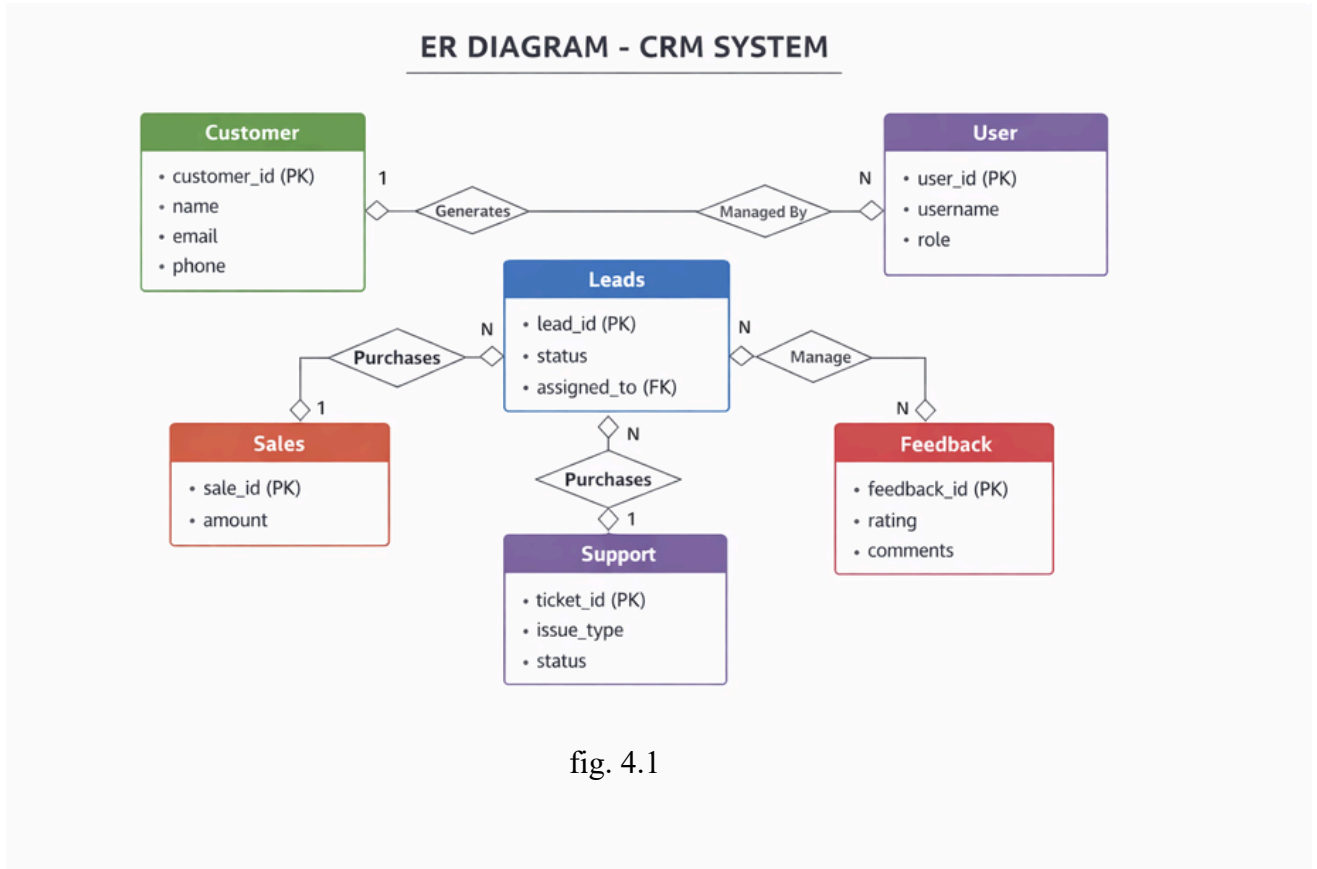


fig. 4.1

4.3 DFD

A Data Flow Diagram (DFD) is a graphical tool used to represent how data moves through a system. It shows data inputs, outputs, data stores, and the processes that transform the data. DFDs provide a high-level overview of system functionality and are widely used in structured analysis due to their simplicity and clarity for both technical and non-technical users.

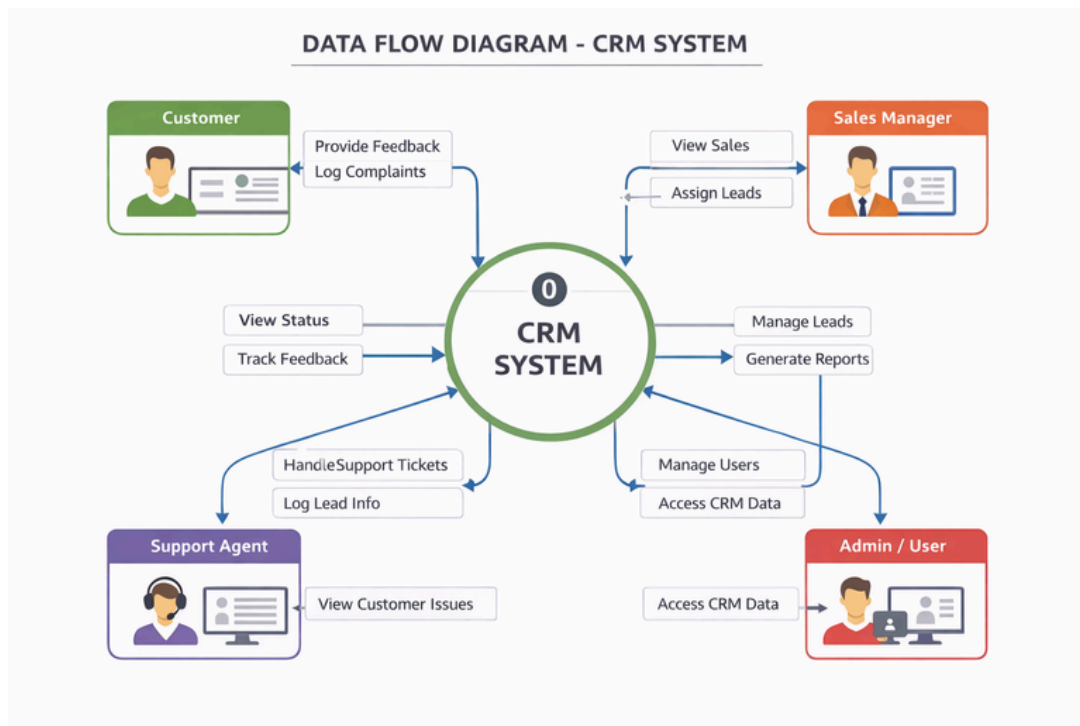


fig. 4.2

4.4 System Flow Chart

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events.

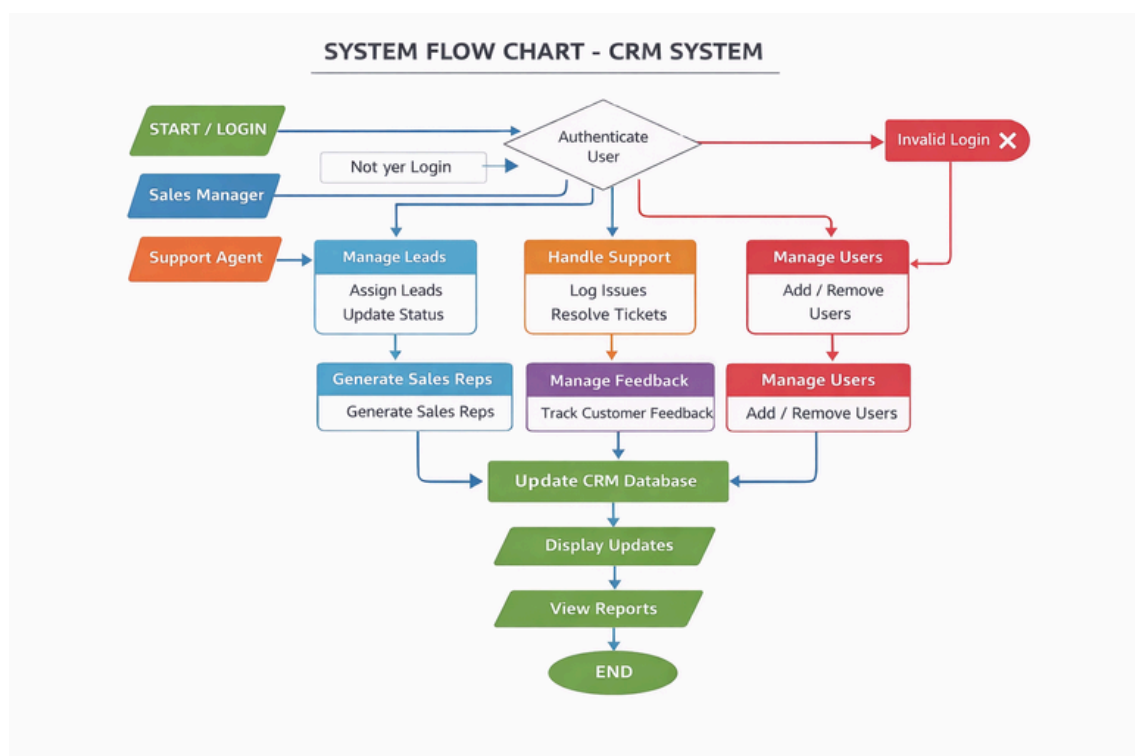


fig. 4.3

5. Implementation

5.1 Program code

The program code is developed to implement the required functionalities of the system using appropriate programming concepts and logic. It follows a modular and well-structured approach to ensure readability, reusability, and easy maintenance. Each module is tested to verify correct output and error handling. Proper comments are added to improve understanding and future enhancements.

Small Java code snippet for a CRM project that adds a new customer record:

```
class Customer {
    int id;
    String name, email;

    Customer(int id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    void display() {
        System.out.println(id + " " + name + " " + email);
    }

    public static void main(String[] args) {
        Customer c1 = new Customer(101, "Rahul", "rahul@gmail.com");
        c1.display();
    }
}
```

5.2 Output screens

The output screen displays a confirmation message after successfully adding a customer record to the system. It shows the customer's ID, name, and email details in a clear and structured format. This helps users verify that the entered information is correct. The screen ensures smooth interaction and proper system feedback.

CUSTOMER RELATIONSHIP MANAGEMENT

Status : Customer Record Added Successfully!

----- Customer Details -----

Customer ID : 101

Customer Name : Rahul

Email ID : rahul@gmail.com

Message : Operation completed successfully.

6. Testing

Testing is performed to ensure that the system functions correctly and meets all specified requirements. Different test cases are executed to validate input handling, system output, and error management. This process helps identify bugs and improve system reliability. Proper testing ensures smooth and accurate system performance.

6.1 Test data

Test data is used to verify the correctness and reliability of the CRM system. Both valid and invalid inputs are tested to ensure proper system behavior.

| Test Case | Custm_ID | Name | Email | Phone | Expected Result |
|-----------|----------|--------|-------------------|------------|---------------------------|
| 1 | 101 | Rahul | rahul@gmail.com | 8569874529 | Record added successfully |
| 2 | 102 | Anjali | anjali@gmail.com | 6987452569 | Record added successfully |
| 3 | 103 | Karan | karan@gmail.com | 7412596325 | Record added successfully |
| 4 | 104 | Priya | priyaap@gmail.com | 985236541 | Record added successfully |
| 5 | 105 | Amit | amit@gmail.com | 9632451036 | Record added successfully |

Table 6.1

6.2 Test result

The test results show that all test cases were executed successfully with expected outputs matching the actual outputs. Each customer record was added correctly without any errors. All test cases passed, confirming the system works as intended. This ensures the reliability and accuracy of the CRM application.

Test Results

| Test Case | Input Data (ID, Name, Email) | Expected Result | Actual Result | Status |
|-----------|-------------------------------|----------------------------|----------------------------|--------|
| 1 | 101, Rahul, rahul@gmail.com | Record added successfully. | Record added successfully. | Pass |
| 2 | 102, Anjali, anjali@yahoo.com | Record added successfully. | Record added successfully. | Pass |
| 3 | 103, Karan, karan@outlook.com | Record added successfully. | Record added successfully. | Pass |
| 4 | 104, Priya, priya@gmail.com | Record added successfully. | Record added successfully. | Pass |
| 5 | 105, Amit, amit@gmail.com | Record added successfully. | Record added successfully. | Pass |

Table 6.2

7. User Manual

The user manual provides clear instructions on how to use the CRM system effectively. It guides users through basic operations such as adding and viewing customer records. Step-by-step procedures and screenshots are included for easy understanding. This helps users operate the system without technical difficulty.

7.1 How to use project guidelines

- Start the application and log in using valid user credentials.
- Navigate to the Customer Management section from the main menu.
- Click on Add Customer to enter customer details such as ID, name, and email.
- Save the information to store it in the system database.
- Use the View/Search option to find existing customer records.
- Update or delete customer information when required.
- Log out safely after completing your work.

7.2 Screen Layouts and Description

The CRM system consists of simple and user-friendly screens designed for easy navigation. The main screen displays menu options such as Add Customer, View Customer, Update Customer, and Delete Customer. The Add Customer screen allows users to enter customer ID, name, and email details. The View Customer screen displays stored customer records in a tabular format. Each screen includes clear buttons like Save, Search, Update, and Exit to help users perform actions smoothly.

8. Project Applications and Limitations

Applications

- Used to store and manage customer information in a centralized system
- Helps track customer interactions and history
- Supports quick search and retrieval of customer records
- Allows easy updating and deletion of customer details
- Improves customer service and communication
- Useful for sales teams and support departments
- Applicable in small and large business organizations
- Enhances data accuracy and reduces manual work

Limitations

- Supports only basic customer information storage
- Limited security features and user authentication
- Does not support multi-user access simultaneously
- No advanced analytics or reporting features
- Manual data entry may cause human errors
- Limited scalability for large databases
- Lacks integration with external systems
- No automated backup and recovery option

9. Conclusion and Future Enhancement

Conclusion

The Customer Relationship Management (CRM) project successfully meets its objectives by providing an efficient way to store and manage customer information. The system simplifies data entry, retrieval, and updates through a user-friendly interface. Testing confirms that all features work correctly and reliably. Overall, this project helps improve customer management and supports better business operations.

Future Enhancement

- Add user login and role-based access control
- Implement advanced search and filter options
- Integrate email and SMS notification features
- Develop graphical reports and analytics dashboard
- Enable cloud storage for remote access
- Support bulk data upload and export options
- Add mobile application support
- Implement data backup and recovery system

10. Bibliography & References

- Pressman, R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill
- Sommerville, I., Software Engineering, Pearson Education
- Oracle Java Documentation – Official Website
- MySQL Reference Manual – MySQL Official Documentation
- W3Schools – www.w3schools.com (for programming concepts)
- GeeksforGeeks – www.geeksforgeeks.org (for code examples)
- TutorialsPoint – www.tutorialspoint.com (for CRM and Java tutorials)

These references were used for understanding Agile methodology, CRM concepts, and software development practices.