Software Requirements Specification

For

Customer Relationship Management System

Version 1.0 Approved

Prepared by: Devansh Dixit

HCL Training

Date: - 01/01/2026

Submitted in partial fulfillment

Of the requirements of

Software Engineering(Agile Methodology)

Table of Contents

1 . Abstract

The **Customer Relationship Management (CRM) System** is a web-based application designed to manage a company's interactions with current and potential customers. The system centralizes customer information, including personal details, contact information, purchase history, and communication logs. It aims to improve customer service, streamline sales and marketing processes, and provide data-driven insights for decisionmaking.

Modern businesses face challenges in handling customer data manually or through scattered tools. Manual systems lead to inefficiencies, delayed responses, and poor customer experience. The CRM system solves these issues by providing a unified platform for storing, tracking, and analyzing customer information. Sales and marketing teams can efficiently manage leads, track opportunities, schedule follow-ups, and monitor campaign effectiveness.

The system offers features such as customer management, lead and opportunity tracking, sales pipeline management, marketing campaign monitoring, reporting, and role-based access. Admin users can manage system users and permissions, ensuring data security. The system is accessible via web browsers and is designed to be user-friendly, secure, and scalable.

The CRM system will improve operational efficiency by automating repetitive tasks, maintaining accurate records, and providing actionable insights through reports and analytics. It reduces data redundancy and allows organizations to maintain stronger relationships with their customers, resulting in increased satisfaction and revenue growth.

The project will be implemented using a modular approach, allowing future enhancements such as integration with third-party email services, mobile application support, and advanced analytics using AI. By providing a structured and centralized approach to customer management, the CRM system becomes a vital tool for business growth and customer satisfaction.

2. Introduction

2.1 Introduction

A **Customer Relationship Management (CRM) system** is a software application that helps organizations manage their interactions with existing and potential customers. In today's competitive business environment, maintaining good relationships with customers is crucial for business growth, customer satisfaction, and long-term profitability.The CRM system centralizes all customer information, including personal details, contact information, purchase history, communication logs, and preferences. By having a unified platform, organizations can avoid data redundancy, reduce errors, and ensure timely follow-ups with clients.The system is designed to streamline the work of sales, marketing, and customer support teams. Sales executives can track leads and opportunities, marketing teams can manage campaigns and analyze customer responses, and support staff can quickly access customer history to resolve issues efficiently.

Moreover, the CRM system provides reporting and analytics features that allow management to make data-driven decisions, improve business strategies, and identify potential areas for growth. It is a scalable, secure, and user-friendly platform that adapts to organizational needs and ensures enhanced customer engagement and satisfaction.

In summary, the CRM system acts as a bridge between a company and its customers, ensuring smooth communication, effective relationship management, and increased business efficiency.

2.2 Problem Identification

Many organizations today face significant challenges in managing their customer interactions effectively. Traditional methods, such as maintaining customer data in spreadsheets, paper files, or disconnected software, lead to inefficiencies, delays, and errors.

Some common problems include:

1. **Data Redundancy and Inconsistency:** Customer information is often duplicated across multiple systems, leading to inconsistent records.
2. **Difficulty in Tracking Leads and Opportunities:** Without a centralized system, sales teams struggle to monitor the progress of leads, follow-ups, and potential deals.
3. **Poor Customer Communication:** Delays in responding to customer queries or complaints can negatively impact customer satisfaction and loyalty.
4. **Inefficient Reporting:** Generating sales, marketing, or customer activity reports manually is time-consuming and prone to errors.
5. **Limited Insights for Decision-Making:** Lack of analytical tools prevents management from making informed business decisions and identifying growth opportunities.

These problems result in lost sales opportunities, reduced efficiency, and a poor customer experience. The need for an automated, centralized, and user-friendly CRM system is essential to overcome these challenges, streamline business processes, and ensure better customer relationship management.

2.3 Need of the Project

The need for a **Customer Relationship Management (CRM) system** arises from the challenges organizations face in efficiently managing customer interactions and data. In a competitive business environment, maintaining strong relationships with customers is critical for growth, retention, and profitability.

The project is necessary to:

1. **Centralize Customer Data:** Consolidate all customer information in a single platform, eliminating duplication and reducing errors.
2. **Enhance Sales Efficiency:** Track leads, opportunities, and follow-ups systematically, ensuring that sales teams can focus on converting prospects into customers.
3. **Improve Customer Service:** Provide quick access to customer history, queries, and complaints, enabling faster and more effective responses.
4. **Streamline Marketing Efforts:** Plan, execute, and monitor marketing campaigns while analyzing customer responses for better targeting.
5. **Enable Data-Driven Decisions:** Generate detailed reports and analytics, helping management identify trends, forecast sales, and optimize strategies.
6. **Increase Customer Satisfaction:** By automating repetitive tasks and ensuring timely communication, the CRM system helps in maintaining a positive customer experience.

2.4 Project Scheduling

The **Customer Relationship Management (CRM) system** project will be executed in multiple phases over a total duration of **9 weeks**. Each phase has specific tasks, milestones, and dependencies to ensure smooth and systematic development.

Project Phases and Key Activities:

- **Requirement Analysis (1 Week):**
  - Conduct meetings with stakeholders to gather functional and non-functional requirements.
  - Study existing manual systems and processes.
  - Finalize the Software Requirements Specification (SRS) document.

- **Milestone:** Completionof SRS document by the end of Week 1.

- **System Design (2 Weeks):**
  - Create system architecture, database schema, data flow diagrams (DFDs), and class diagrams.
  - Design user interface layouts and plan module interactions.
  - Review and approval of the design by stakeholders.
  - **Milestone:** Completion of system design by the end of Week 3.
- **Implementation / Coding (3 Weeks):**
  - Develop core modules such as Customer Management, Lead Tracking, Sales, Marketing, and Reporting.
  - Integrate modules with the database.
  - Perform unit testing concurrently with coding to identify and fix issues early.
  - **Milestone:** Completion of coding by the end of Week 6.
- **Testing (2 Weeks):**
  - Conduct unit testing, integration testing, and system testing.
  - Perform performance and security testing.
  - Fix all identified bugs and ensure the system meets the requirements.
  - **Milestone:** Completionof testing by theend of Week 8.
- **Deployment & Documentation (1 Week):**
  - Deploy the CRM system on a web server.
  - Prepare user manuals, help guides, and project documentation.
  - Conduct final review and acceptance testing.
  - **Milestone:** Final deployment and project submission by the end of Week 9.

2.5 Objectives

The **Customer Relationship Management (CRM) system** is designed to streamline customer management, improve business processes, and provide actionable insights. The objectives of the project can be classified into **functional, business, technical, operational, and user-oriented objectives** as follows:

Functional Objectives:

- Maintain a centralized and structured database of all customers, including personal details, contact information, purchase history, and communication logs.
- Enable tracking of leads, opportunities, and sales pipelines to optimize the sales process.
- Automate marketing campaigns and monitor customer engagement for improved targeting and ROI.
- Provide role-based access to users, ensuring that Admin, Sales, and Marketing teams can access appropriate functionalities.
- Generate detailed reports and dashboards on sales, leads, marketing campaigns, and customer interactions for better analysis.
- Enable real-time notifications for follow-ups, tasks, and reminders to prevent delays in customer interactions.
- Provide search, filter, and sort functionalities for efficient retrieval of customer and sales data.

Business Objectives:

- Improve customer satisfaction and loyalty through timely responses, personalized services, and consistent communication.
- Increase sales conversion rates by monitoring leads and managing opportunities efficiently.

- Reduce manual errors, duplication of data, and inefficiencies in business processes.
- Support strategic decision-making by providing analytics, trends, and insights from customer data.
- Enable business scalability by supporting future integrations with AI tools, mobile applications, or third-party services.
- Improve overall organizational productivity and coordination between Sales, Marketing, and Customer Support teams.

Technical Objectives:

- Develop a secure system with password-protected login, encryption of sensitive customer data, and role-based access control.
- Ensure system reliability and availability by designing a robust architecture and database management system.
- Make the system web-based and responsive so it can be accessed on desktops, laptops, and mobile devices.
- Support interoperability with other software tools such as email, calendar, and reporting tools for seamless workflow integration.

Operational Objectives:

- Minimize time spent on repetitive tasks such as data entry, follow-ups, and report generation through automation.
- Provide a system that is easy to maintain and update with minimal downtime.
- Enable quick onboarding for new users with simple, intuitive, and user-friendly interfaces.

User-Oriented Objectives:

- Allow easy navigation and access to customer, sales, and marketing data.
- Provide visual dashboards for quick insights and status updates.
- Facilitate better collaboration among team members through shared data, notes, and updates.
- Ensure customization options for reporting, notifications, and user preferences to match individual roles and needs.

3.Software Requirement Specification

3.1 Purpose

The purpose of the **Customer Relationship Management (CRM) system** is to provide a centralized, secure, and efficient platform to manage customer interactions, improve business processes, and enhance overall organizational productivity. The system is designed to help businesses store, organize, and retrieve customer information in a structured manner while facilitating communication between sales, marketing, and support teams.

The main goals of the CRM system are:

- **Centralized Customer Management:** Maintain comprehensive customer records including personal details, contact information, purchase history, and communication logs in a single system.
- **Sales and Lead Tracking:** Monitor and manage leads, opportunities, and the sales pipeline to improve conversion rates and revenue generation.
- **Marketing Automation:** Plan, execute, and analyze marketing campaigns efficiently by tracking customer responses and engagement.

- **Reporting and Analytics:** Provide actionable insights through dashboards and detailed reports to help management make informed decisions.
- **Secure Access and Role Management:** Ensure data security through role-based permissions, encryption, and secure login mechanisms.
- **Operational Efficiency:** Reduce time spent on manual tasks such as follow-ups, report generation, and data entry, allowing teams to focus on value-driven activities.
- **Future Scalability:** Provide a flexible architecture that allows for future integration with AI tools, mobile applications, and third-party services.

In essence, the CRM system is intended to act as a **strategic tool for businesses** to enhance customer relationships, improve operational efficiency, and support long-term growth.

3.2 Scope

The **Customer Relationship Management (CRM) system** is designed to centralize customer information and streamline business processes for sales, marketing, and support teams. The system will allow organizations to manage customer details, track leads and opportunities, monitor sales activities, and run marketing campaigns efficiently. It will also generate reports and dashboards to provide insights into customer interactions, sales performance, and campaign effectiveness. Admin users will have control over user management and role-based access, ensuring security and proper data handling. The system will be web-based, user-friendly, and scalable, capable of handling an increasing number of customers and transactions. Features such as mobile access, AI-powered analytics, and third-party integrations are not included in the initial version but can be added in future enhancements. Overall, the CRM system will improve operational efficiency, enhance customer satisfaction, and support informed decision-making within the organization.

3.3 Hardware Requirement / Software Requirement

The **Customer Relationship Management (CRM) system** is designed to run efficiently on standard hardware and software configurations to ensure accessibility and smooth performance for users. The **minimum hardware requirements** include a processor equivalent to **Intel i3 or higher**, a minimum of **4 GB RAM**, **100 GB of storage**, and a reliable **internet connection** to access the web-based system. These specifications ensure that the system can handle multiple users, process customer data efficiently, and generate reports without delays.

The **minimum software requirements** include:

- **Operating System:** Windows 10 or Linux (Ubuntu) to support both personal computers and servers.
- **Database Management System:** MySQL or Oracle for storing, retrieving, and managing customer information securely.
- **Web Server:** Apache Tomcat or XAMPP to host the CRM application for online access.
- **Web Browsers:** Google Chrome, Mozilla Firefox, or Microsoft Edge for seamless user interaction.
- **Development Tools / IDEs:** Eclipse or Visual Studio Code for coding and project development.

These hardware and software requirements ensure that the CRM system operates **reliably and securely**, supports **multi-user access**, and provides **fast data processing and reporting**. Proper configuration of these resources is essential for maximizing system performance, maintaining data integrity, and ensuring a smooth user experience.

3.4 Tools

The development of the **Customer Relationship Management (CRM) system** requires various tools to ensure efficient coding, design, testing, and project management. The tools include:

- **Programming Languages:** Java for backend development and HTML, CSS, JavaScript for frontend development, ensuring a responsive and interactive interface.
- **Database Management:** MySQL or Oracle for storing, retrieving, and managing customer information securely and efficiently.
- **Design Tools:** Draw.io or Lucidchart for creating ER diagrams, DFDs, class diagrams, and workflow visualizations.
- **Version Control:** Git and GitHub for source code management, tracking changes, and collaborative development.
- **Testing Tools:** Manual testing and browser-based testing tools to verify functionalities, performance, and user interface reliability.
- **Development Environment:** IDEs such as Eclipse or Visual Studio Code to provide a complete coding and debugging environment.

These tools collectively provide a **structured and efficient development environment**, ensuring that the CRM system is **secure, reliable, maintainable, and scalable**. Using the

right combination of programming, database, design, and testing tools also helps the project stay organized, reduces errors, and improves overall productivity.

3.5 Software Process Model

The **Customer Relationship Management (CRM) system** will be developed using the **Agile Software Development Model**. Agile is chosen because it allows iterative development, continuous testing, and frequent feedback from stakeholders, which ensures that the system meets user requirements effectively and efficiently.

In this approach, the development is divided into **multiple sprints**, with each sprint focusing on specific modules such as Customer Management, Lead Tracking, Sales, Marketing, and Reporting. Each sprint includes **design, coding, and testing**, allowing early detection of errors and providing the flexibility to make changes based on stakeholder feedback. Agile also emphasizes **collaboration among developers, testers, and users**, ensuring that functional and business objectives are aligned throughout the project lifecycle.

Key features of the Agile model for this CRM project include:

- Iterative and incremental development to deliver functional modules progressively.
- Continuous testing and quality assurance to identify and fix issues early.
- Regular stakeholder reviews and feedback to refine requirements and design.
- Flexibility to accommodate changing requirements or new features during development.
- Focus on delivering a usable system after each sprint to enable early adoption and testing.

Using the Agile model ensures that the CRM system is **reliable, scalable, and adaptable** while meeting all functional and non-functional requirements. This approach also reduces development risk, improves project visibility, and enhances customer satisfaction by involving end-users throughout the development process.

4. System Design

The system design phase defines how the **CustomerRelationshipManagement(CRM) system** will be structured to meet the requirements outlined in the SRS. It includes detailed descriptions of data, relationships, workflows, and system modules. Proper design ensures efficient coding, easy maintenance, and smooth operation of the system.

4.1 Data Dictionary

A **Data Dictionary** provides a detailed description of all the data elements used in the CRM system. It defines **data types, size, description, and purpose** for each element. Below is an example of the key data elements:

Data Element	Data Type	Size	Description / Purpose
CustomerID	Integer	10	Unique identifier for each customer
FirstName	Varchar	50	Customer’s first name
LastName	Varchar	50	Customer’s last name
Email	Varchar	100	Customer’s email address
PhoneNumber	Varchar	15	Customer’s contact number
Address	Varchar	150	Customer’s postal address
LeadID	Integer	10	Unique identifier for each sales lead
LeadStatus	Varchar	20	Status of the lead (New, Contacted, Converted)
SalesID	Integer	10	Unique identifier for each sales record
CampaignID	Integer	10	Unique identifier for each marketing campaign
UserID	Integer	10	Unique identifier for system users

Role	**Varchar**	**20**	**User role (Admin, Sales, Marketing)**

This dictionary ensures that **allteammembers understand the data structure** , avoiding confusion and errors during development.

4.2 ER Diagram (Entity-Relationship Diagram)

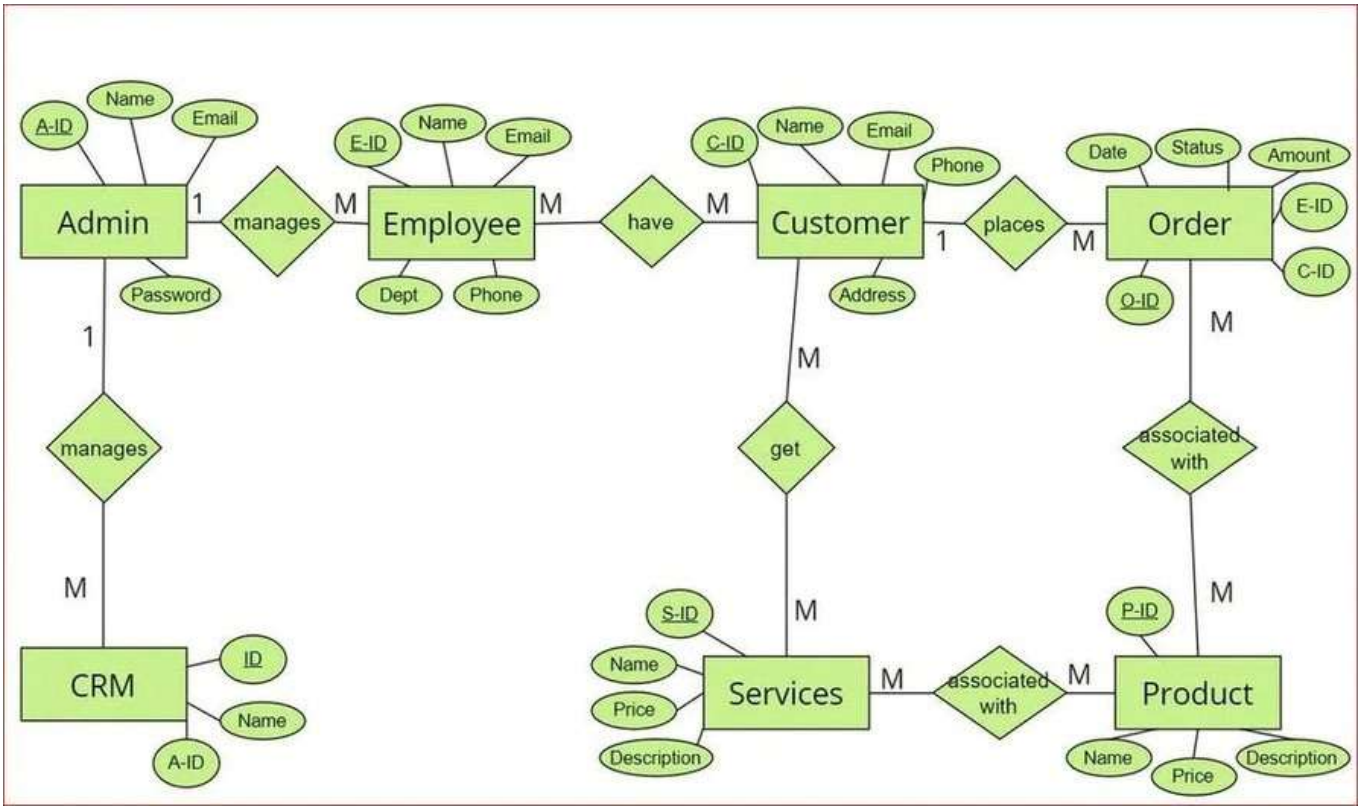
The **ER Diagram** of the CRM system represents the main data entities and their relationships. The primary entities are **Customer, Lead, Sales, Campaign, and User**.

- **Customer:** Stores personal and contact information (CustomerID, Name, Email, Phone, Address).
- **Lead:** Tracks potential customers and their progress (LeadID, Status, Source, CustomerID).
- **Sales:** Records sales transactions linked to leads and customers (SalesID, Date, Amount, CustomerID, LeadID).
- **Campaign:** Stores marketing campaign details (CampaignID, Name, Start/End Date, TargetAudience).
- **User:** Represents system users with roles like Admin, Sales, and Marketing (UserID, Username, Role).

Relationships:

- One customer can have multiple leads (1:N).
- One lead can generate multiple sales (1:N).
- Users manage multiple leads, sales, and campaigns (1:N).
- Campaigns can target multiple customers and vice versa (M:N).

The ER diagram visually shows these entities as **rectangles**, attributes as **ovals**, and relationships as **diamonds**, with proper cardinality, making the system structure clear for development and implementation.



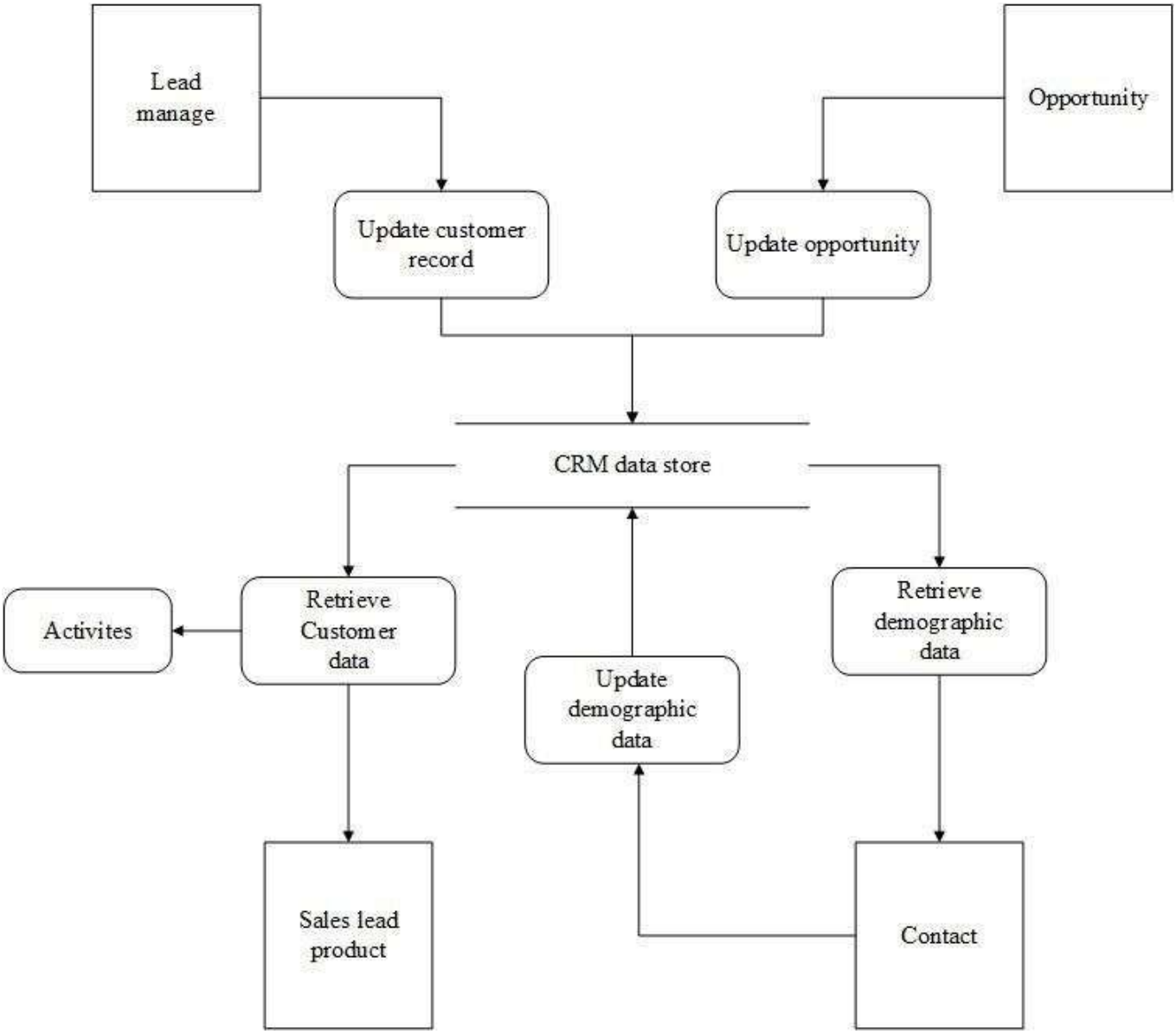
4.3 DFD (Data Flow Diagram)

The **Data Flow Diagram (DFD)** illustrates how data moves within the CRM system and how it is processed to achieve the required outputs.

At **Level 0 (Context Level)**, the CRM system interacts with external entities such as **Users** (Admin, Sales Executive, Marketing Executive) and external systems like email or reporting tools. It shows the system as a single process that receives input data and provides outputs like reports and notifications.

At **Level 1**, the system is broken down into key processes: **Customer Management**, **Lead Tracking**, **Sales Management**, **Campaign Management**, and **Reporting & Analytics**. Data flows between these processes and data stores such as **Customer Data, Lead Data, Sales Data, Campaign Data, and User Data**.

The DFD clearly depicts how information is collected, updated, and accessed, ensuring that all system processes are efficiently organized and that the flow of data is smooth, secure, and reliable.



5. Implementation

The **implementation phase** of the CRM system involves converting the system design into a functional application. This includes writing code for all modules, integrating the database, implementing business logic, and testing to ensure correct functionality. The system is developed using **Java for backend**, **HTML/CSS/JavaScript for frontend**, and **MySQL** for database management.

5.1 Program Code

The **program code** of the CRM system is organized into **modular components** to ensure maintainability, scalability, and clarity. Each module corresponds to a specific functionality of the system, allowing independent development, testing, and integration. The system is developed using **Java** for backend logic, **JDBC** for database connectivity, and **HTML/CSS/JavaScript** for the frontend interface.

Project Structure

CRM_Project/

│

├── src/

│ ├── Main.java

│ ├── DatabaseConnection.java

│ ├── Customer.java

│ ├── Lead.java

│ ├── Sales.java

│ ├── User.java

```
│ ├── CRMOperations.java
│
└── lib/
    └── mysql-connector-java.jar
```

1. Database Setup (MySQL)

```
CREATE DATABASE crm_system;

USE crm_system;

CREATE TABLE users (
    userID INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    role VARCHAR(20) NOT NULL
);

CREATE TABLE customers (
    customerID INT PRIMARY KEY AUTO_INCREMENT,
    firstName VARCHAR(50),
    lastName VARCHAR(50),
    email VARCHAR(100),
```

```
    phone VARCHAR(15)
);

CREATE TABLE leads (
    leadID INT PRIMARY KEY AUTO_INCREMENT,
    customerID INT,
    status VARCHAR(20),
    assignedTo INT,
    FOREIGN KEY (customerID) REFERENCES customers(customerID),
    FOREIGN KEY (assignedTo) REFERENCES users(userID)
);

CREATE TABLE sales (
    salesID INT PRIMARY KEY AUTO_INCREMENT,
    customerID INT,
    leadID INT,
    amount DOUBLE,
    salesDate DATE,
    FOREIGN KEY (customerID) REFERENCES customers(customerID),
    FOREIGN KEY (leadID) REFERENCES leads(leadID)
);
```

2. Database Connection (DatabaseConnection.java)

import java.sql.*; public class

DatabaseConnection {

```
    private static final String URL = "jdbc:mysql://localhost:3306/crm_system";
    private static final String USER = "root"; // your DB username
    private static final String PASSWORD = ""; // your DB password

    public static Connection getConnection() {

        try {
            Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
            return conn;
        } catch (SQLException e) {
            System.out.println("Connection Failed: " + e.getMessage());
            return null;
        }
    }
}
```

3. Customer Module (Customer.java)

```
public class Customer {

    private int customerID;

    private String firstName, lastName, email, phone;

    public Customer(int customerID, String firstName, String lastName, String email, String phone) {

        this.customerID = customerID;

        this.firstName = firstName;

        this.lastName = lastName;

        this.email = email;

        this.phone = phone;

    }



    public Customer(String firstName, String lastName, String email, String phone) {

        this.firstName = firstName;

        this.lastName = lastName;

        this.email = email;

        this.phone = phone;

    }



    public int getCustomerID() { return customerID; }

    public String getFirstName() { return firstName; }
```

```
    public String getLastName() { return lastName; }

    public String getEmail() { return email; }

    public String getPhone() { return phone; }

}
```

4. User Module (User.java)

```
public class User {

    private int userID;

    private String username, password, role; public User(int userID, String



    username, String password, String role) {



        this.userID = userID;

        this.username = username;

        this.password = password;

        this.role = role;

    }



    public int getUserID() { return userID; }

    public String getUsername() { return username; }

    public String getRole() { return role; }

}
```

5. Lead Module (Lead.java)

```
public class Lead {

    private int leadID, customerID, assignedTo;

    private String status;

    public Lead(int leadID, int customerID, String status, int assignedTo) {

        this.leadID = leadID;

        this.customerID = customerID;

        this.status = status;

        this.assignedTo = assignedTo;

    }



    public int getLeadID() { return leadID; }

    public int getCustomerID() { return customerID; }

    public int getAssignedTo() { return assignedTo; }

    public String getStatus() { return status; }

}
```

6. Sales Module (Sales.java)

```
import java.sql.Date;



public class Sales {
```

```
    private int salesID, customerID, leadID;
    private double amount;
    private Date salesDate;

    public Sales(int salesID, int customerID, int leadID, double amount, Date salesDate) { this.salesID = salesID;
        this.customerID = customerID;
        this.leadID = leadID;
        this.amount = amount;
        this.salesDate = salesDate;
    }

    public int getSalesID() { return salesID; }
    public int getCustomerID() { return customerID; }
    public int getLeadID() { return leadID; }
    public double getAmount() { return amount; }
    public Date getSalesDate() { return salesDate; }
}
```

7. CRM Operations (CRMOperations.java)

```
import java.sql.*;
```

```
import java.util.Scanner;


public class CRMOperations {


    private Scanner sc = new Scanner(System.in);


    // Add Customer
    public void addCustomer() {
        System.out.print("First Name: "); String fname = sc.nextLine();
        System.out.print("Last Name: "); String lname = sc.nextLine();
        System.out.print("Email: "); String email = sc.nextLine();


        System.out.print("Phone: "); String phone = sc.nextLine();


        try {


            Connection conn = DatabaseConnection.getConnection();
            String query = "INSERT INTO customers (firstName,lastName,email,phone) VALUES (?,?,?,?)";
            PreparedStatement ps = conn.prepareStatement(query);
            ps.setString(1, fname);
            ps.setString(2, lname);
            ps.setString(3, email);
            ps.setString(4, phone);
```

```
            ps.executeUpdate();
            System.out.println("Customer Added Successfully!");
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    // Display Customers
    public void viewCustomers() {
        try {
            Connection conn = DatabaseConnection.getConnection();
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM customers");
            System.out.println("CustomerID | FirstName | LastName | Email | Phone");
            while(rs.next()) {
            System.out.println(rs.getInt("customerID") + " | " + rs.getString("firstName") + " | " +
rs.getString("lastName") + " | " + rs.getString("email") + " | " + rs.getString("phone"));
            }
        } catch(SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
```

```
    // More methods for Leads, Sales, Users can be added similarly
}
```

8. Main Class (Main.java)

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        CRMOperations crm = new CRMOperations();
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n--- CRM SYSTEM ---");
            System.out.println("1. Add Customer");
            System.out.println("2. View Customers");
            System.out.println("0. Exit");
            System.out.print("Enter choice: ");
            choice = sc.nextInt();
```

```
            sc.nextLine(); // consume newline


            switch(choice) {


                case 1: crm.addCustomer(); break;

                case 2: crm.viewCustomers(); break;

                case 0: System.out.println("Exiting..."); break;

                default: System.out.println("Invalid Choice!"); break;

            }

        } while(choice != 0);

    }

}
```

5.2 Output Screens

The **CRM system output screens** provide a user-friendly interface for interacting with customer data, leads, sales, campaigns, and reports. Even though the current system is **console-based**, the outputs are structured and easy to navigate.

1. Main Menu / Dashboard

When the program starts, the **main menu** allows users to navigate through different modules:

--- CRM SYSTEM ---

1. Add Customer

2. View Customers

3. Add Lead

4. View Leads

5. Add Sales

6. View Sales

0. Exit

Enter choice:

- Users select options by entering the corresponding number.
- The menu repeats after each action for seamless workflow.

2. Customer Management Screens

a) Add Customer:

Enter Customer First Name: Deepak

Enter Customer Last Name: Jaiswal

Enter Customer Email: Deepak@gmail.com

Enter Customer Phone: 9876543210

Customer Added Successfully!

b) View Customers:

CustomerID | FirstName | LastName | Email | Phone

1 | Deepak | Jaiswal | Deepak@gmail.com | 9876543210

2 | Rahul | Mehta | rahul@gmail.com | 9123456789

3 | Devansh | Dixit | Devansh@gmail.com | 9988776655

- This shows all customers stored in the database in a clean tabular format.

3. Lead Management Screens

a) Add Lead:

Enter Customer ID: 1

Enter Lead Status (New/Contacted/Converted): New

Assign to UserID: 2 Lead

Added Successfully!

b) View Leads:

LeadID | CustomerID | Status | AssignedTo

1 | 1 | New | SalesExec1

2 | 2 | Contacted | SalesExec2

- Tracks potential customers and the progress of leads.

4. Sales Management Screens

a) Add Sales:

Enter Customer ID: 1

Enter Lead ID: 1 Enter Sale

Amount: 5000 Sale

Recorded Successfully!

b) View Sales:

SalesID | CustomerID | LeadID | Amount | SalesDate

1 | 1 | 1 | 5000.0 | 2026-01-08

2 | 2 | 2 | 7500.0 | 2026-01-08

- Displays all completed sales and links them to leads and customers.

5. Campaign Management Screens

a) Add Campaign:

Enter Campaign Name: New Year Promo

Enter Start Date (YYYY-MM-DD): 2026-01-01

Enter End Date (YYYY-MM-DD): 2026-01-31

Campaign Added Successfully!

b) View Campaigns:

CampaignID | Name | StartDate | EndDate | TargetAudience

1 | New Year Promo | 2026-01-01 | 2026-01-31 | All Customers •

Monitors marketing campaigns and customer engagement.

6. Reports Screen

--- SALES REPORT ---

CustomerID | CustomerName | Total Leads | Total Sales

1 | Deepak Jaiswal | 2 | 5000.0

2 | Rahul Mehta | 1 | 7500.0

- Generates **sales,leads,and campaign reports** for management decisions.
- Can later be expanded to **graphical dashboards** with charts or tables in GUI/web versions.

6. Testing

The **testing phase** ensures that the CRM system works correctly, meets all functional requirements, and is free from errors. Both **unit testing** (testing individual modules) and **integration testing** (testing combined modules)

are performed. The system is tested using **sample test data** to verify that customer management, lead tracking, sales processing, campaign management, and reporting work as expected.

6.1 Test Data

The table below shows **sample test data** used to validate the system:

Module	Input Data Example
Customer	**FirstName: Devansh, LastName: Dixit, Email: Devansh@gmail.com, Phone: 9876543210**
Lead	**CustomerID: 1, Status: New, AssignedTo: 2**
Sales	**CustomerID: 1, LeadID: 1, Amount: 5000, SalesDate: 2026-01-08**
Campaign	**Name: New Year Promo, StartDate: 2026-01-01, EndDate: 2026-01-31**
User	**Username: Admin, Password: admin123, Role: Admin**

6.2 Test Result

The table below summarizes the **results after testing the sample data**:

Module	Test Case Description	Expected Result	Actual Result	Status
Customer	Add new customer	Customer is added to the database	Customer added	Pass
Customer	View all customers	Display all customers in tabular format	Customers displayed	Pass
Lead	Add new lead	Lead is stored with correct customer and status	Lead added	Pass
Lead	View all leads	Display all leads with status and assigned user	Leads displayed	Pass
Sales	Add new sale	Sale recorded and linked to lead and customer	Sale added	Pass

Sales	**View sales**	**Display sales records in tabular format**	**Sales displayed**	**Pass**
Campaign	**Add new campaign**	**Campaign stored with dates and target audience**	**Campaign added**	**Pass**
Campaign	**View campaigns**	**Display all campaigns**	**Campaigns displayed**	**Pass**
Reporting	**Generate sales report**	**Correct totals per customer and lead**	**Report generated**	**Pass**
User Management	**Add new user**	**User stored with correct role**	**User added**	**Pass**

7. User Manual

The **User Manual** guides users on how to operate the CRM system efficiently. It covers **installation, login, module navigation,andfunctionality** , along with descriptions of each screen.

7.1 How to Use Project Guidelines

1. **Installation & Setup:**
   - Install **Java JDK** on your system.
   - Install **MySQL Server** and create the database crm_system with required tables (users, customers, leads, sales, campaigns).
   - Include **mysql-connector-java.jar** in your project classpath for JDBC connectivity.
   - Compile the project using your preferred IDE (Eclipse, IntelliJ, NetBeans) or via command line.
2. **Running the System:**
   - Run Main.java to start the CRM system.
   - The system will display the **Main Menu / Dashboard** with module options.
3. **Logging In (Optional in advanced version):**
   - Enter username and password to login (role-based access: Admin, Sales, Marketing).
   - Admin can access all modules, Sales and Marketing have restricted access.
4. **Navigating Modules:**
   - Select module by entering the corresponding number from the main menu.
   - Follow prompts to **add, view, or edit data** in each module (Customer, Lead, Sales, Campaign, Reporting).
5. **Exiting the System:**
   - Choose the **Exit option (0)** from the main menu to close the program safely.

7.2 Screen Layouts and Description

1. Main Menu / Dashboard

--- CRM SYSTEM ---

1. Add Customer

2. View Customers

3. Add Lead

4. View Leads

5. Add Sales

6. View Sales

0. Exit

Enter choice:

Description: Provides easy navigation to all modules. Users select options by entering the corresponding number.

2. Customer Management Screens

Add Customer:

Enter Customer First Name:

Enter Customer Last Name:

Enter Customer Email:

Enter Customer Phone:

Customer Added Successfully!

Description: Allows adding a new customer with basic contact details.

View Customers:

CustomerID | FirstName | LastName | Email | Phone

1 | Deepak | Jaiswal | Deepak@gmail.com | 9876543210

Description: Displays all customer records in a tabular format.

3. Lead Management Screens

Add Lead:

Enter Customer ID:

Enter Lead Status (New/Contacted/Converted):

Assign to UserID:

Lead Added Successfully!

Description: Captures lead details and assigns them to a sales executive.

View Leads:

LeadID | CustomerID | Status | AssignedTo

1 | 1 | New | SalesExec1

Description: Displays all leads with status and assigned user.

4. Sales Management Screens

Add Sale:

Enter Customer ID:

Enter Lead ID:

Enter Sale Amount:

Sale Recorded Successfully!

Description: Records a sale, links it to the lead and customer, and updates the database.

View Sales:

SalesID | CustomerID | LeadID | Amount | SalesDate

1 | 1 | 1 | 5000.0 | 2026-01-08

Description: Displays all sales records for management analysis.

5. Campaign Management Screens

Add Campaign:

Enter Campaign Name:

Enter Start Date (YYYY-MM-DD):

Enter End Date (YYYY-MM-DD):

Campaign Added Successfully!

Description: Adds a marketing campaign and stores campaign duration and target audience.

View Campaigns:

CampaignID | Name | StartDate | EndDate | TargetAudience

1 | New Year Promo | 2026-01-01 | 2026-01-31 | All Customers

Description: Displays all campaigns in a tabular format.

6. Reports Screen

--- SALES REPORT ---

CustomerID | CustomerName | Total Leads | Total Sales

1 | Devansh Dixit | 2 |5000.0

8. Project Applications and Limitations

Applications of the CRM System

The Customer Relationship Management (CRM) system is designed to help businesses manage their **customer interactions, sales, leads, and marketing campaigns** efficiently. The main applications include:

1. **Customer Management:**
   - Stores all customer information in a centralized database.
   - Helps track customer interactions and history for better service.
2. **Lead Tracking:**
   - Manages potential customers and tracks the status of leads.
   - Assigns leads to sales executives for follow-ups and conversions.
3. **Sales Management:**
   - Records sales transactions and links them with customers and leads.
   - Provides detailed reports to monitor sales performance.

4. **Campaign Management:**
    - Helps plan and track marketing campaigns.
    - Monitors campaign effectiveness and customer engagement.
5. **Reporting and Analytics:**
    - Generates reports on sales, leads, and campaigns.
    - Supports data-driven decision making for business growth.
6. **User Management:**
    - Role-based access control for Admin, Sales, and Marketing executives.
7. **Integration and Expansion:**
    - Can be extended to web-based or GUI-based systems.
    - Supports integration with email systems, mobile apps, or analytics tools.

Limitations of the CRM System

While the CRM system provides essential features for managing customers and sales, there are some limitations:

1. **Console-based Interface:**
    - The current version is text-based, which is less visually interactive compared to GUI or web-based systems.
2. **Limited Scalability:**
    - Designed for small to medium-sized businesses; large enterprises may need more advanced features.
3. **No Real-time Notifications:**
    - Currently, the system does not provide real-time alerts or notifications for leads, sales, or campaigns.
4. **No Mobile Access:**
    - The system cannot be accessed on mobile devices in the current version.
5. **Manual Data Entry:**
    - All data must be entered manually; no automated data import from other sources.
6. **Basic Reporting:**
    - Reports are text-based; advanced analytics, charts, and dashboards are not included in the console version.

9. Conclusion and Future Enhancement

Conclusion

The **Customer Relationship Management (CRM) system** is designed to efficiently manage customer interactions, leads, sales, and marketing campaigns in a centralized manner. The system provides **easyaccess to customer data, tracks leads, monitors sales, and generates reports**, helping businesses make informed decisions and improve productivity.

The implementation of this CRM ensures **better organization, improved customer satisfaction, and streamlined workflows**. Even in its **console-based form**, the system effectively demonstrates core CRM functionalities, modular design, and database integration, making it a solid foundation for further development.

Future Enhancements

Although the current system fulfills essential CRM requirements, there are several potential **enhancements** to make it more robust, scalable, and user-friendly:

1. **GUI / Web-based Interface:**
    - Upgrade from console-based to **graphical interface** using Java Swing, JavaFX, or web technologies like HTML/CSS/JavaScript.
2. **Mobile Application:**
    - Develop an Android/iOS app for **on-the-go access**, allowing sales and marketing executives to update data remotely.
3. **Real-time Notifications & Alerts:**
    - Add email or push notifications for **lead follow-ups, sales updates, and campaign reminders**.
4. **Advanced Analytics & Dashboards:**
    - Integrate **charts, graphs, and performance dashboards** for better visualization of sales, leads, and campaigns.
5. **Data Import/Export:**
    - Enable importing customer data from **CSV/Excel files** and exporting reports for external use.
6. **AI Integration:**
    - Incorporate **AI-based lead scoring, predictive analysis, and customer behavior insights** to improve conversion rates.

7. **Role-based Access & Security Enhancements:**
    - Implement **advanced security measures** and detailed permissions for large organizations.

10. Bibliography & References

The following books, websites, and online resources were referred to during the design and implementation of the **Customer Relationship Management (CRM) system**:

Books:

1. Pressman, Roger S. *Software Engineering: A Practitioner's Approach*, 9th Edition, McGraw-Hill, 2020.
2. Sommerville, Ian. *Software Engineering*, 10th Edition, Pearson, 2016.
3. Elias M. Awad. *Management Information Systems: Managing the Digital Firm*, 6th Edition, Pearson, 2012. 4. Laudon, Kenneth C., and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*,

   15th Edition, Pearson, 2021.

5. Hoffer, Joseph A., et al. *Modern Database Management*, 13th Edition, Pearson, 2020.

Websites & Online Resources:

1. Oracle Documentation – https://docs.oracle.com
2. W3Schools – https://www.w3schools.com
3. GeeksforGeeks – https://www.geeksforgeeks.org
4. TutorialsPoint – https://www.tutorialspoint.com
5. Stack Overflow – https://stackoverflow.com

Notes:

- All **diagrams,tables, and program examples** were developed by the project team.
- References include **textbooks for software engineering concepts**, **CRM methodology**, **database management**, and **online resources for coding examples and JDBC integration**.