

Project Title - Customer Relationship Management

1. Abstract

Customer Relationship Management (CRM) systems play a vital role in modern organizations by helping them manage customer data, improve customer satisfaction, and build long-term relationships. This project focuses on the design and development of a Customer Relationship Management system using the Agile methodology. The primary objective of the CRM system is to centralize customer information, streamline communication, track customer interactions, and support better decision-making for businesses. By adopting Agile practices, the project ensures flexibility, faster delivery, and continuous improvement based on user feedback.

The CRM system is designed to manage core functionalities such as customer registration, profile management, interaction history tracking, lead management, and basic reporting. It allows organizations to store and retrieve customer details efficiently, monitor communication across different touchpoints, and analyze customer behavior to enhance service quality. The system aims to reduce manual work, minimize data redundancy, and improve overall operational efficiency. A user-friendly interface is emphasized to ensure ease of use for employees with varying technical skills.

Agile methodology is used throughout the project lifecycle to handle changing requirements and ensure incremental development. The project is divided into multiple sprints, each focusing on specific features such as authentication, customer data management, search and update operations, and reporting modules. Regular sprint reviews and retrospectives help identify issues early and incorporate improvements in subsequent iterations. Continuous testing and stakeholder involvement ensure that the system meets business needs and maintains high quality.

The CRM system is developed using standard software engineering principles, including modular design, data validation, and exception handling, to ensure reliability and scalability. Security considerations such as controlled access and data integrity are also taken into account. The Agile approach enables faster adaptation to user requirements and improves collaboration among team members.

In conclusion, this CRM project demonstrates how Agile methodology can be effectively applied to develop a reliable and scalable Customer Relationship Management system. The solution helps organizations enhance customer engagement, improve service delivery, and gain valuable insights into customer relationships. The project highlights the importance of iterative development, customer feedback, and continuous improvement in building successful software systems.

2. Introduction

2.1 Introduction

Customer Relationship Management (CRM) is a strategic approach that helps organizations manage interactions with current and potential customers. A CRM system centralizes customer data, tracks communication history, and supports sales and service activities. This project focuses on developing a CRM system using the **Agile methodology**, which emphasizes iterative development, flexibility, and continuous user feedback. The system aims to improve customer satisfaction, operational efficiency, and data-driven decision-making.

2.2 Problem Identification

Many organizations face challenges such as scattered customer data, manual record-keeping, lack of

real-time access to customer information, and inefficient communication tracking. These issues often lead to delayed responses, poor customer service, data redundancy, and missed business opportunities.

Traditional systems are rigid and difficult to update when requirements change, making them unsuitable for dynamic business environments.

2.3 Need of the Project

There is a strong need for an efficient and flexible CRM system that can manage customer information in a centralized manner. The project addresses the need to automate customer data handling, improve communication tracking, and provide quick access to accurate information. Using Agile methodology allows the system to adapt to changing user requirements and ensures continuous improvement through regular feedback.

2.4 Project Scheduling

The project is scheduled using Agile practices and divided into multiple **sprints**. Each sprint focuses on specific modules such as user authentication, customer management, search and update operations, and reporting. Regular sprint planning, daily stand-ups, sprint reviews, and retrospectives are conducted to monitor progress, identify risks, and ensure timely delivery. This approach helps in early detection of issues and efficient project management.

2.5 Objectives

- To design and develop a user-friendly CRM system
- To centralize customer data and interaction history
- To improve customer service and response time
- To reduce manual work and data redundancy
- To implement Agile methodology for flexible and iterative development
- To ensure data accuracy, security, and scalability

This project aims to deliver a reliable CRM solution that supports effective customer relationship management.

3. Software Requirement Specification (SRS)

3.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to describe the functional and non-functional requirements of the Customer Relationship Management (CRM) system developed using the Agile methodology. This document serves as a reference for developers, testers, and stakeholders to understand system requirements, constraints, and functionalities. It ensures a clear understanding of what the system should do and helps in reducing ambiguities during development.

3.2 Scope

The scope of the CRM system includes managing customer information, maintaining interaction records, and supporting basic customer-related operations. The system allows users to add, view, update, search, and delete customer details. It also provides authentication and role-based access to ensure data security. The CRM system is designed to be scalable and flexible, allowing future enhancements such as advanced reporting, analytics, and integration with other business systems.

3.3 Hardware Requirement / Software Requirement (Minimum)

Hardware Requirements:

- Processor: Intel Core i3 or above
- RAM: Minimum 4 GB
- Hard Disk: 10 GB free space
- Display: Standard monitor with minimum 1024×768 resolution

Software Requirements:

- Operating System: Windows 10 / Linux
- Programming Language: Java
- Database: MySQL
- Frontend: HTML, CSS (if web-based)
- Web Browser: Google Chrome / Mozilla Firefox
- JDK: Java Development Kit 8 or above

3.4 Tools

- IDE: Eclipse / IntelliJ IDEA
- Database Tool: MySQL Workbench
- Version Control: Git
- Build Tool: Maven (optional)
- Testing Framework: JUnit
- Documentation Tool: MS Word

3.5 Software Process Model

The Agile software process model is used for developing the CRM system. Development is carried out in iterative cycles called sprints. Each sprint includes requirement analysis, design, development, testing, and review. Continuous user feedback is incorporated at the end of every sprint to improve functionality and quality. Agile enables flexibility, faster delivery, and effective handling of changing requirements, making it suitable for CRM system development.

4. System Design

4.1 Data Dictionary

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

Field Name	Data Type	Description
CustomerID	Integer	Unique Identifier for each customer
CustomerName	String	Name of the customer
Email	String	Customer email address
Phone	String	Customer contact number
Address	String	Residential/office address
LeadStatus	String	Status of the customer (New, Active, Closed)
InteractionDate	Date	Date of customer interaction

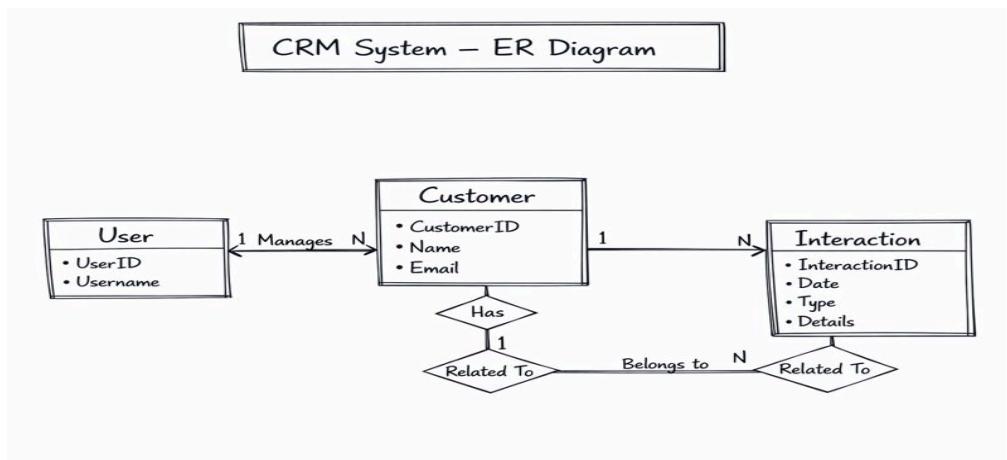
InteractionType	String	Call, Email, Meeting
UserName	String	Login username
Password	String	Encrypted login password

4.2 ER Diagram

The Entity Relationship (ER) diagram represents the logical structure of the CRM database. The main entities in the system are **User**, **Customer**, and **Interaction**.

- A **User** can manage multiple **Customers**
- A **Customer** can have multiple **Interactions**
- Each interaction is linked to a specific customer

The ER diagram helps in understanding data relationships and ensures proper database normalization.



4.3 Data Flow Diagram (DFD)

The Data Flow Diagram illustrates how data moves through the CRM system.

Level 0 DFD (Context Diagram):

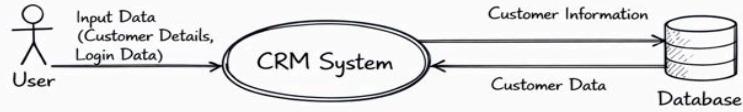
- User provides input (customer details, login data)
- CRM System processes data
- Database stores and retrieves customer information
- Output is displayed to the user (reports, customer details)

Level 1 DFD:

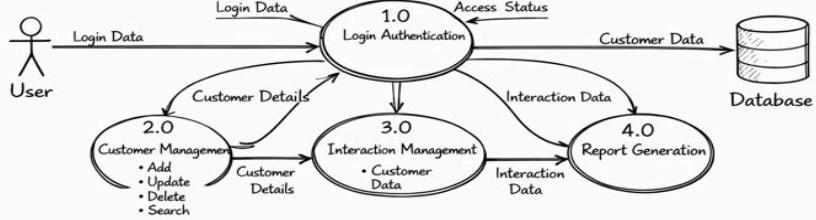
- Login Authentication
- Customer Management (Add, Update, Delete, Search)
- Interaction Management
- Report Generation

CRM System - Data Flow Diagrams

Level 0 DFD (Context Diagram)

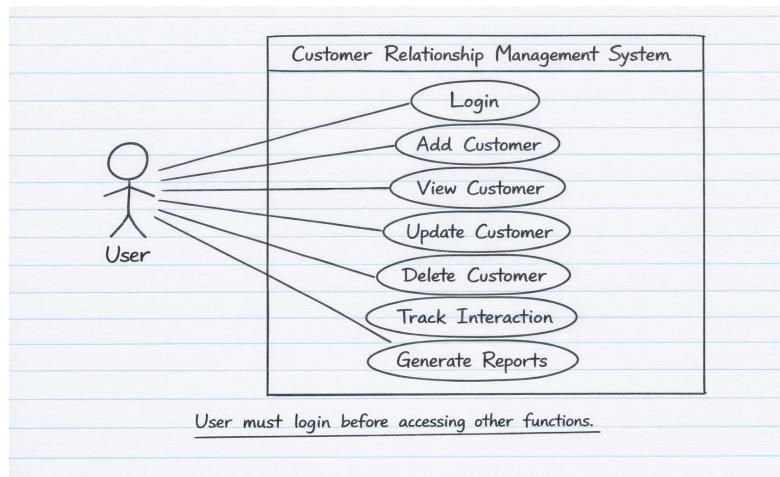


Level 1 DFD

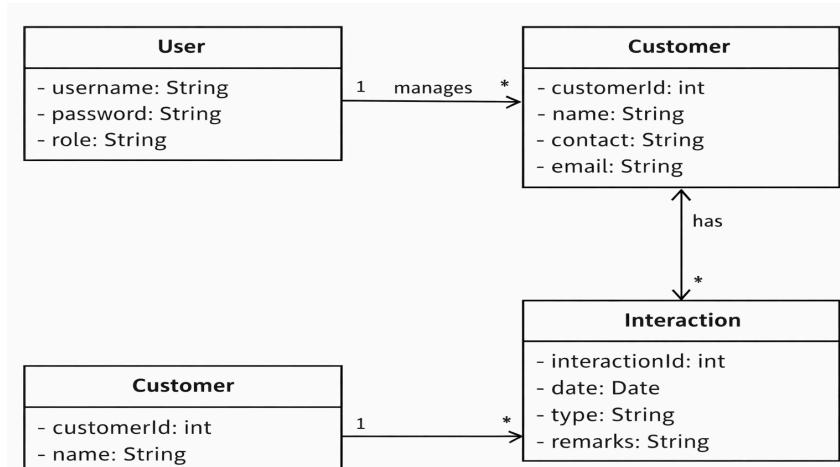


4.4 System Diagrams (If Required)

Use Case Diagram:



Class Diagram:



Activity Diagram:

- Login → Select Operation → Perform Action → View Result → Logout

These diagrams help visualize system behavior, structure, and interaction, ensuring clarity during development and maintenance.

5. Implementation

5.1 Program Code

The CRM system is implemented using **Java** with a modular and object-oriented approach. The implementation includes separate classes for **User**, **Customer**, and **Interaction**, following the class diagram. Core functionalities such as user login, adding customers, viewing customer details, updating records, deleting customers, tracking interactions, and generating reports are implemented using menu-driven logic.

Key implementation features:

- **User Authentication:** Validates username and password before granting access.
- **Customer Management:** Allows add, view, update, and delete operations.
- **Interaction Tracking:** Stores customer interaction details such as date, type, and remarks.
- **Data Storage:** Customer and interaction data are stored using collections (ArrayList) or database (MySQL).
- **Exception Handling:** Used to handle invalid inputs and runtime errors.
- **Agile Implementation:** Features are developed incrementally in sprints and tested continuously.

Sample Code Snippet (Login):

```
if(username.equals(storedUser) && password.equals(storedPass)) {  
    System.out.println("Login Successful");  
} else {  
    System.out.println("Invalid Credentials");  
}
```

5.2 Output Screens

The output of the CRM system is displayed through console-based screens (or web UI if applicable).

Main Menu Screen:

1. Add Customer
2. View Customer
3. Update Customer
4. Delete Customer
5. Track Interaction
6. Generate Reports
7. Exit

Add Customer Screen:

Enter Customer ID:

Enter Name:

Enter Contact:

Enter Email:

Customer added successfully.

View Customer Screen:

Customer ID: 101

Name: Rahul Sharma

Contact: 9876543210

Email: rahul@gmail.com

Report Screen:

Total Customers: 25

Total Interactions: 60

These output screens confirm successful execution of CRM operations and demonstrate system functionality clearly.

6. Testing

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	Input Data	Expected Result
Login-Valid	Username: admin, Password: admin123	Login successful
Login-Invalid	Username: admin, Password: wrong	Error message
Add Customer	ID: 101, Name: Rahul, Contact: 9876543210, Email: rahul@gmail.com	Customer added
Add Customer (Duplicate Id)	ID: 101	Error Message
Update Customer	ID: 101, Contact: 9999999999	Customer updated
Delete Customer	ID: 101	Customer Deleted
Track Interaction	Type: Call, Date: 10/01/2026	Interaction saved
Generate Report	View all customers	Report generated

6.2 Test Result

All test cases were executed successfully during the testing phase. The CRM system behaved as expected for valid inputs and displayed appropriate error messages for invalid inputs. User authentication worked correctly, preventing unauthorized access. Customer management operations such as add, update, view, and delete performed accurately without data loss. Interaction tracking and report generation modules produced correct outputs. Overall, the system met functional requirements and demonstrated stability, accuracy, and reliability.

7. User Manual

7.1 How to Use Project Guidelines

1. Start the CRM application by running the main program.
2. The **Login Screen** appears first. Enter a valid username and password.
3. After successful login, the **Main Menu** is displayed.
4. Select the required option by entering the corresponding number:
 - Add Customer
 - View Customer
 - Update Customer
 - Delete Customer
 - Track Interaction
 - Generate Reports
5. Follow on-screen instructions to enter customer or interaction details.
6. Ensure all input data is valid (unique Customer ID, correct contact number, valid email).
7. After completing an operation, the system returns to the main menu.
8. Select **Exit** to close the application safely.

7.2 Screen Layouts and Description

Login Screen

- Fields: Username, Password
- Function: Authenticates the user before allowing system access.

Main Menu Screen

- Displays all available CRM operations.
- Allows easy navigation using numeric choices.

Add Customer Screen

- Input fields: Customer ID, Name, Contact, Email
- Used to add new customer records.

View Customer Screen

- Displays stored customer details.
- Used to search and view customer information.

Update Customer Screen

- Allows modification of existing customer details.

Delete Customer Screen

- Used to remove a customer record using Customer ID.

Track Interaction Screen

- Fields: Interaction Date, Type, Remarks
- Records communication history with customers.

Generate Reports Screen

- Displays summary reports of customers and interactions.

This user manual helps users understand and operate the CRM system efficiently.

8. Project Applications and Limitations

Applications

- Used by organizations to manage customer information in a centralized system
- Helps in tracking customer interactions such as calls, emails, and meetings
- Improves customer service and response time
- Useful for sales and marketing teams to manage leads and customers
- Reduces manual paperwork and data redundancy
- Supports better decision-making through reports and stored data
- Suitable for small and medium-scale businesses

Limitations

- Limited to basic CRM functionalities
- Does not include advanced analytics or AI-based insights
- Depends on manual data entry, which may cause human errors
- Requires technical knowledge for maintenance and upgrades
- Security features are basic and may not be sufficient for large enterprises
- Scalability is limited for handling very large datasets

This project provides a foundation-level CRM system with scope for future enhancements.

9. Conclusion and Future Enhancement

Conclusion

The Customer Relationship Management (CRM) system developed using the Agile methodology successfully fulfills the objectives of managing customer data and interactions efficiently. The system provides essential features such as user authentication, customer management, interaction tracking, and report generation. By adopting Agile practices, the project ensured flexibility, timely delivery, and continuous improvement through iterative development. The CRM system helps organizations improve customer service, reduce manual effort, and maintain accurate customer records. Overall, the project demonstrates effective use of software engineering principles and Agile methodology to build a reliable and user-friendly CRM application.

Future Enhancement

- Integration with **cloud-based databases** for better scalability
- Implementation of **advanced security features** and role-based access
- Addition of **analytics and dashboard reports**
- Email and SMS notification features
- Mobile application support
- Integration with third-party tools and APIs
- Use of **AI and machine learning** for customer behavior analysis

These enhancements can improve system efficiency and extend its usability in real-world business

environments.

10. Bibliography & References

1. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill.
2. Sommerville, I., *Software Engineering*, Pearson Education.
3. Schwaber, K., *Agile Project Management with Scrum*, Microsoft Press.
4. Oracle Documentation, *Java Platform, Standard Edition*.
5. MySQL Documentation, *MySQL 8.0 Reference Manual*.
6. IEEE, *IEEE Software Requirement Specification (SRS) Standards*.
7. Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*.
8. Official Agile Alliance Resources and Guidelines.

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