Software Requirements Specification

For

Customer Relationship Management (CRM) System
Version 1.0 approved

Prepared by Shejal Dhadse

HCL Training

Date: 10/01/2026

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

Abstract

The Customer Relationship Management (CRM) system project focuses on designing and developing a centralized platform that helps organizations manage customer interactions, sales activities, and service processes efficiently using Agile methodology. In today's competitive business environment, maintaining strong customer relationships is critical for business growth and customer retention. Traditional manual methods and disconnected systems make it difficult to track customer data, follow up on leads, and provide consistent service. This project addresses these challenges by providing an integrated CRM solution.

The proposed CRM system allows organizations to store and manage customer information, track leads and sales opportunities, handle customer support requests, and monitor communication history through a single platform. The system provides role-based access for administrators, sales teams, and support staff, ensuring secure and controlled usage. By centralizing customer data, the CRM system improves visibility, collaboration, and decision-making across departments.

Agile methodology plays a key role in the development of this CRM system. The project is divided into short development cycles called sprints, where each sprint delivers a working feature such as customer data management, lead tracking, sales pipeline management, or reporting dashboards. Continuous testing and stakeholder feedback during each sprint help in early identification of issues, faster improvements, and easy adaptation to changing business requirements.

The CRM system is designed with scalability, performance, and security as important non-functional requirements. Secure authentication, role-based authorization, and data protection mechanisms are implemented to safeguard sensitive customer information. The system architecture supports future enhancements such as mobile access, integration with email and messaging tools, analytics dashboards, and AI-based customer insights.

Overall, this project demonstrates how Agile methodology can be effectively applied to develop a practical, enterprise-ready CRM system. The outcome is a reliable, flexible, and efficient solution that enhances customer satisfaction, improves operational efficiency, and supports long-term business growth.

Table of Contents

List of Figures

List of Tables

2. Introduction

2.1 Introduction

Customer Relationship Management (CRM) is a business system used to manage interactions with existing and potential customers. In modern organizations, customer data is spread across multiple departments such as sales, marketing, and customer support.

Without a centralized system, managing customer information becomes difficult and inefficient. The CRM system provides a single platform to store customer details, track communication, manage sales activities, and improve customer service.

This project aims to develop a web-based CRM system using Agile methodology to ensure flexibility, faster delivery, and continuous improvement.

2.2 Problem Identification

Many organizations face challenges due to manual and fragmented customer management processes.

Problems Identified:

- Customer data stored in multiple systems
- Difficulty in tracking sales leads and follow-ups
- Lack of visibility into customer communication history
- Delayed response to customer queries
- Poor coordination between departments

Example:
A sales team may lose potential leads due to missed follow-ups and lack of centralized information.

2.3 Need of the Project

There is a strong need for an automated CRM system that centralizes customer data and streamlines business processes.

Need Includes:

- Centralized customer information management
- Better tracking of leads and opportunities
- Improved customer service and response time
- Data-driven decision making

Example:
Support teams can quickly access customer history and resolve issues efficiently.

2.4 Project Scheduling

The CRM project follows the Agile development approach, where development is divided into short sprints. Each sprint focuses on delivering specific features such as customer management, lead tracking, or reporting. Regular sprint reviews and testing help ensure timely delivery and quality improvement.

2.5 Objectives

- Centralize customer data in a single system
- Improve sales and lead management
- Enhance customer service efficiency
- Ensure data security and access control
- Support continuous improvement using Agile methodology

3. Software Requirement Specification (SRS)

3.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to clearly define the functional and non-functional requirements of the Customer Relationship Management (CRM) system.

This document serves as a common reference for developers, testers, project managers, and stakeholders. It helps ensure that the system is developed according to business needs and reduces misunderstandings during development and testing phases.

3.2 Scope

The CRM system is a web-based application designed to manage customer information, sales activities, and customer support operations in a centralized manner. The system supports multiple user roles such as administrators, sales executives, and support agents. It improves communication, data visibility, and operational efficiency across departments.

The scope of the system includes:

- Customer information management
- Lead and opportunity tracking
- Sales pipeline management
- Customer support and ticket management
- Reporting and analytics
- User and role management

The system is designed to be scalable and can be enhanced with additional features in the future.

3.3 Hardware Requirement / Software Requirement

Hardware Requirements (Minimum):

- Processor: Intel i3 or higher
- RAM: 4 GB
- Storage: 100 GB
- Network: Stable internet connection

Software Requirements:

- Operating System: Windows or Linux
- Frontend Technologies: HTML, CSS, JavaScript
- Backend Technologies: Java / Python / PHP
- Database: MySQL
- Web Browser: Google Chrome or Mozilla Firefox

3.4 Tools

The following tools are used for the development and maintenance of the CRM system:

- Integrated Development Environment (IDE): VS Code or Eclipse
- Database Tool: MySQL Workbench
- Version Control System: Git
- Testing Tools: Manual testing tools / Selenium
- Documentation Tools: MS Word or Canva

These tools help improve development efficiency, collaboration, and code quality.

3.5 Software Process Model

The CRM system is developed using the **Agile Software Development Model**. The project is divided into short development cycles called sprints.

Each sprint delivers a working feature such as customer management, lead tracking, or reporting. Continuous feedback, testing, and review are performed at the end of each sprint to improve quality and adapt to changing requirements.

4. System Design

The System Design phase defines the overall structure of the CRM system, including data organization, component interaction, and information flow. This phase provides a clear blueprint for developers and testers, ensuring that the system meets functional and non-functional requirements efficiently.

4.1 Data Dictionary

The Data Dictionary describes the key data elements used in the CRM system along with their meanings. It helps maintain consistency in data usage across the application.

Data Element	Description
Customer_ID	Unique identifier assigned to each customer
Customer_Name	Full name of the customer
Email	Email address of the customer
Phone_Number	Contact number of the customer
Lead_ID	Unique identifier for sales leads
Lead_Status	Current status of the lead
Opportunity_ID	Identifier for sales opportunity
Ticket_ID	Unique support ticket number
User_Role	Role assigned to the system user
Created_Date	Date when the record was created

4.2 ER Diagram

The Entity Relationship (ER) Diagram represents the database structure of the CRM system. It shows entities and the relationships between them.

Main Entities:

- Customer
- Lead

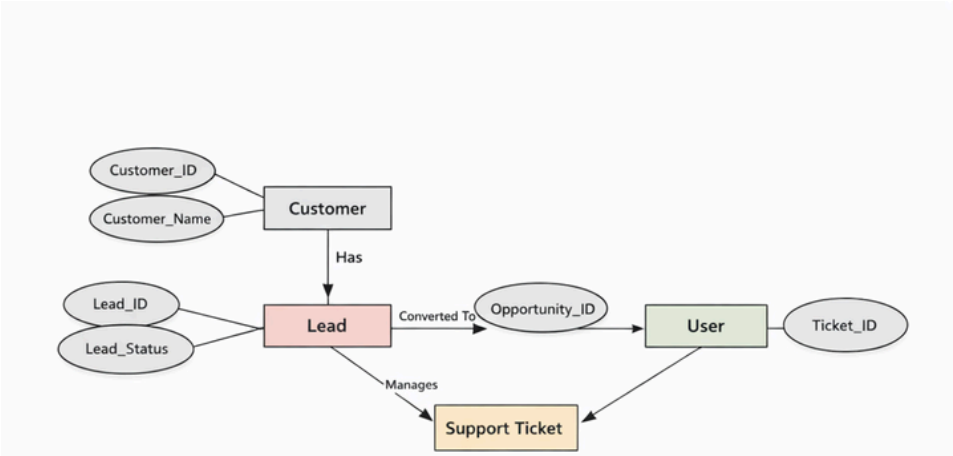
- Opportunity
- User
- Support Ticket

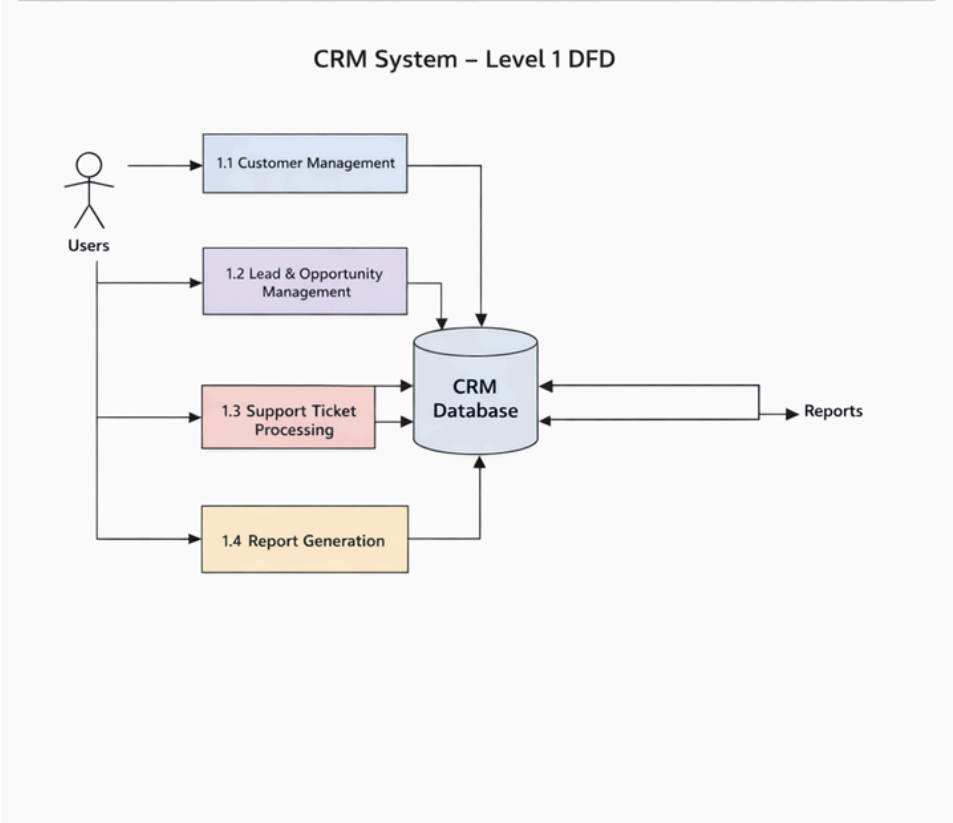
Relationships:

- One customer can have multiple leads
- A lead can be converted into an opportunity
- A user can manage multiple customers and tickets

The ER diagram ensures proper database normalization and efficient data storage.

4.3 Data Flow Diagram (DFD)





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

Level 0 DFD:

- Users interact with the CRM system
- The system processes user requests
- Data is stored and retrieved from the database

Level 1 DFD:

- Customer Management Process
- Lead and Opportunity Management
- Support Ticket Processing
- Report Generation

The DFD helps in understanding system processes and data dependencies.

4.4 Use Case Diagram

The Use Case Diagram shows interactions between users and the CRM system.

Actors:

- Administrator
- Sales Executive
- Support Agent

Use Cases:

- Manage Customer Information
- Create and Update Leads
- Track Sales Opportunities
- Manage Support Tickets
- Generate Reports

This diagram clearly defines system functionality from the user’s perspective.

CRM System Development Process

(Using Agile)

5. Implementation

The Implementation phase focuses on converting the system design into a working CRM application. This phase follows Agile principles, where development is carried out in small iterations (sprints). Each sprint delivers a functional module that is tested and reviewed before moving to the next iteration.

5.1 Program Code

The CRM system is developed using a modular architecture to ensure maintainability and scalability. The application is divided into multiple modules, each responsible for a specific function.

Major Modules Implemented:

- User Authentication and Authorization
- Customer Management Module
- Lead and Opportunity Management
- Support Ticket Management
- Reporting and Dashboard Module

Each module is developed using standard coding practices, with proper validation, error handling, and database interaction. Code reviews are conducted at the end of each sprint to ensure quality and consistency.

5.2 Output Screens

The CRM system provides user-friendly and responsive output screens for different user roles.

Key Output Screens:

- Login and User Registration Screen
- Customer List and Customer Detail Screen
- Lead and Opportunity Tracking Screen
- Support Ticket Management Screen
- Reports and Analytics Dashboard

The screens are designed to be simple and intuitive, allowing users to perform tasks efficiently with minimal training. Role-based access ensures that users only see the screens relevant to their responsibilities.

6. Testing

Testing ensures that the CRM system works as expected and meets business requirements. In Agile, testing is performed continuously in every sprint to detect defects early and improve system quality.

6.1 Test Data

Test data is used to verify different system functions under valid and invalid conditions. Types of Test Data Used:

- Valid customer details (name, email, phone)
- Invalid login credentials
- Existing and non-existing customer IDs
- Different user roles (Admin, Sales, Support)
- Sample leads and tickets
- **Example:** Valid email format is accepted, while invalid email format is rejected by the system.

6.2 Test Result

Test results are recorded to confirm whether the system behaves as expected. All critical modules were tested successfully, and defects identified during testing were fixed in subsequent sprints.

Test Case ID	Module	Test Scenario	Expected Result	Actual Result	Status
TC01	Login	Valid user login	User logged in successfully	As expected	Pass
TC02	Login	Invalid password	Error message displayed	As expected	Pass
TC03	Customer	Add new customer	Customer added successfully	As expected	Pass
TC04	Lead	Update lead status	Lead status updated	As expected	Pass
TC05	Ticket	Close support ticket	Ticket closed successfully	As expected	Pass

7. User Manual

The User Manual provides step-by-step guidance to help users understand and operate the CRM system efficiently. It explains how different users can access and use system features based on their roles.

7.1 How to Use Project Guidelines

1. Open the CRM application in a web browser.
2. Log in using a valid username and password.
3. Based on the assigned role (Admin, Sales, or Support), the system displays relevant modules.
4. Navigate through the dashboard to access customers, leads, tickets, or reports.
5. Save changes after completing any action.
6. Log out after completing work to maintain system security.

7.2 Screen Layouts and Description

Login Screen:
Allows users to enter credentials and access the system securely.

Dashboard Screen:
Displays an overview of customers, leads, tickets, and performance metrics.

Customer Management Screen:
Used to add, update, view, and delete customer records.

Lead Management Screen:
Helps sales users track leads, update status, and convert leads into opportunities.

Support Ticket Screen:
Allows support agents to create, update, and close customer tickets.

Each screen is designed for ease of use, ensuring minimal training is required for users.

8. Project Applications and Limitations

This section describes where the CRM system can be applied in real-world scenarios and highlights the current limitations of the system.

8.1 Project Applications

The CRM system can be effectively used in various business environments to improve customer management and operational efficiency.

Applications:

- Sales organizations for managing leads and opportunities
- Customer support centers for handling service requests
- Marketing teams for tracking customer interactions
- Small and medium enterprises for centralized customer data management
- Service-based companies for improving customer satisfaction

Example:
A sales team can track customer interactions and close deals faster using the CRM system.

8.2 Project Limitations

Despite its benefits, the CRM system has certain limitations.

Limitations:

- Requires stable internet connectivity
- Initial setup and user training may be needed
- Limited advanced analytics in the current version
- Integration with third-party systems is minimal
- Performance depends on server capacity

These limitations can be addressed in future enhancements.

9. Conclusion and Future Enhancement

9.1 Conclusion

The Customer Relationship Management (CRM) system developed using Agile methodology provides an efficient and centralized platform for managing customer data, sales activities, and support services. The system improves coordination between departments, enhances customer satisfaction, and supports informed business decision-making.

Agile development ensured continuous feedback, early issue detection, and flexibility in handling changing requirements. The CRM system successfully meets business needs by delivering a scalable, secure, and user-friendly solution.

9.2 Future Enhancement

The CRM system can be further improved by adding advanced features and technologies.

Future Enhancements:

- Integration with email and messaging platforms
- Advanced analytics and reporting using AI
- Mobile application support
- Third-party payment and ERP system integration
- Improved role-based security and access control

These enhancements will increase system functionality and support long-term business growth.

10. Bibliography & References

1. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill Education.
2. Sommerville, I., *Software Engineering*, Pearson Education.
3. Schwaber, K., and Sutherland, J., *The Scrum Guide*, Scrum.org.
4. Agile Alliance, *Agile Software Development Principles*.
5. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.
6. Salesforce Documentation, *CRM Concepts and Best Practices*.
7. Oracle Documentation, *Customer Relationship Management Overview*.
8. Official MySQL Documentation, *Database Design and Management*.
9. W3Schools, *Web Development Technologies*.
10. Project Management Institute (PMI), *Agile Practice Guide*.