**Project Document: Customer Management System**

Table of Contents

1. Introduction

1.1. Purpose of the Document

1.2. Project Overview

1.3. Scope

2. System Requirements

2.1. Functional Requirements

3. Architecture

3.1. Functional Requirements

3.2. Class Diagram

3.3. Sequence Diagrams

4. Technologies Used

5. Conclusion

6. References

**1. Introduction**

**1.1. Purpose of the Document**

The purpose of this document is to provide a comprehensive overview of the Customer Management System project, outlining its requirements, architecture, technologies used, and testing methodologies. It serves as a guide for developers, testers, and stakeholders involved in the project.

**1.2. Project Overview**

The Customer Management System is a core Java-based application designed to help businesses efficiently manage their customer information. It allows users to perform CRUD (Create, Read, Update, Delete) operations on customer data and provides a user-friendly interface to interact with the system.

**1.3. Scope**

The scope of the project includes the following functionalities:

1. Add a new customer to the system with relevant information such as name, contact details, and address.

2. View customer details and search for specific customers based on various criteria.

3. Update existing customer information.

4. Delete customer records from the system.

5. Integration with an Oracle SQL database to persist customer data.

6. Integration with JUnit for unit testing.

7. Integration with Jenkins for continuous integration and automated builds.

8. Integration with Docker for containerization and deployment.

**2. System Requirements**

**2.1. Functional Requirements**

1. User Registration: The system shall allow users to register with a unique username and password.

2. User Login: Registered users shall be able to log in to the system using their credentials.

3. Customer Creation: Users shall be able to add new customer details to the system.

4. Customer Retrieval: Users shall be able to view and search for customer information based on different criteria (e.g., name, ID).

5. Customer Update: Users with appropriate privileges shall be able to update customer information.

6. Customer Deletion: Users with appropriate privileges shall be able to delete customer records.

7. Data Validation: The system shall validate input data to ensure accuracy and consistency.

8. Security: Access to sensitive functionalities shall be restricted to authorized users only.

9. Error Handling: The system shall handle errors gracefully and provide meaningful error messages.

1. **Architecture**

**3.1.Layered Architecture**

The Customer Management System will follow a layered architecture:

- Presentation Layer: Responsible for user interface and interaction with users.

- Business Logic Layer: Handles business rules and processes.

- Data Access Layer: Manages interactions with the Oracle SQL database.

3.2. Class Diagram

(TODO: Insert your class diagram here.)

3.3. Sequence Diagrams

(TODO: Insert sequence diagrams for important use cases.)

**5. Technologies Used**

The project will be developed using the following technologies:

- Core Java: For the application's business logic and functionality.

- Oracle SQL Database: For storing and managing customer data.

- Jenkins: For continuous integration and automated builds.

- Docker: For containerizing the application and simplifying deployment.

**6. Conclusion**

In conclusion, the Customer Management System is an essential tool for businesses to manage their customer information efficiently. By adopting a layered architecture and leveraging technologies like Core Java, Oracle SQL, JUnit, Jenkins, and Docker, we aim to deliver a reliable and robust application.

**7. References**

(TODO: List any references used in the development of this document.)