# CRM APPLICATION

*Software Requirements Specification*

# Revision History

| Date | Revision | Description | Author |
|---|---|---|---|
| 6/15/2021 | 1.0 | Initial Version | GROUP 3 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

# 1. Purpose

This document outlines the requirements for the Customer Relationship Management (CRM) system

## 1.1. Scope

This document will catalog the user, system, and hardware requirements for the CRM system. It will not, however, document how these requirements will be implemented.

## 1.2. Definitions, Acronyms, Abbreviations

CRM - Customer Relationship Management.

Admin - Administrator with total control over the CRM system.

User - A typical user with no admin level privileges.

Manager - A higher level user with some higher level privileges.

Client - A customer.

Record - A record or log of data.

Group - A collection of clients belonging to a user.

## 1.3. References

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | System startup |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | System is up and running |
| **Basic Flow or Main Scenario** | 1. Bind port<br>2. Load previous data<br>3. Start internal systems |
| **Extensions or Alternate Flows** | |
| **Exceptions** | 1. Port fails to bind<br>2. Data corruption or not found |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | Admin creates user |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | New user is created |
| **Basic Flow or Main Scenario** | 1. Admin supplies a new user and user-type for new user<br>2. System validates username and user-type<br>3. System creates a new entry for a new user<br>4. System reports username and password |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● User already exists<br>● Username and password does not fit requirement |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User logs in |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | User is not logged in |
| **Post-Conditions** | User is logged in |
| **Basic Flow or Main Scenario** | 1. User enters username and password<br>2. System validates credentials<br>3. System returns |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● User credentials invalid |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User creates record |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | New record created by user |
| **Basic Flow or Main Scenario** | 1. Enter new record<br>2. System validates record entry<br>3. System saves record |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Record already exist<br>● Invalid record parameters |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User reads record |

| | |
|---|---|
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Record read by user |
| **Basic Flow or Main Scenario** | 1. User request record<br>2. System searches for record<br>3. System returns record contents to user |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Record not found |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User deletes record |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Record deleted by user |
| **Basic Flow or Main Scenario** | |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Record not found |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User updates record |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Record is updated |

| | |
|---|---|
| **Basic Flow or Main Scenario** | 1. User requests record to be updated<br>2. User passes updated record parameters<br>3. System searches for record<br>4. System validates new record parameters<br>5. System updates record old parameters with new parameters<br>6. System confirms success |
| **Extensions or Alternate Flows** | ● Record not found<br>● Record parameters invalid |
| **Exceptions** | |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User creates group |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Group is created |
| **Basic Flow or Main Scenario** | 1. User request to create a new group with group name<br>2. System checks that group does not already exist<br>3. System creates new group<br>4. System confirms success |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Users not found |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User adds client to group |
| **Relevant Requirements** | |
| **Primary Actor** | System |

| | |
|---|---|
| **Pre-Conditions** | None |
| **Post-Conditions** | Client added to group |
| **Basic Flow or Main Scenario** | 1. User requests client be added to a group<br>2. System checks that client and group exists<br>3. System adds client to group<br>4. System confirms success |
| **Extensions or Alternate Flows** | 1. System can ask User if System should create group should it not exist already |
| **Exceptions** | ● Client not found<br>● Group does not exist |
| **Related Use Cases** | |

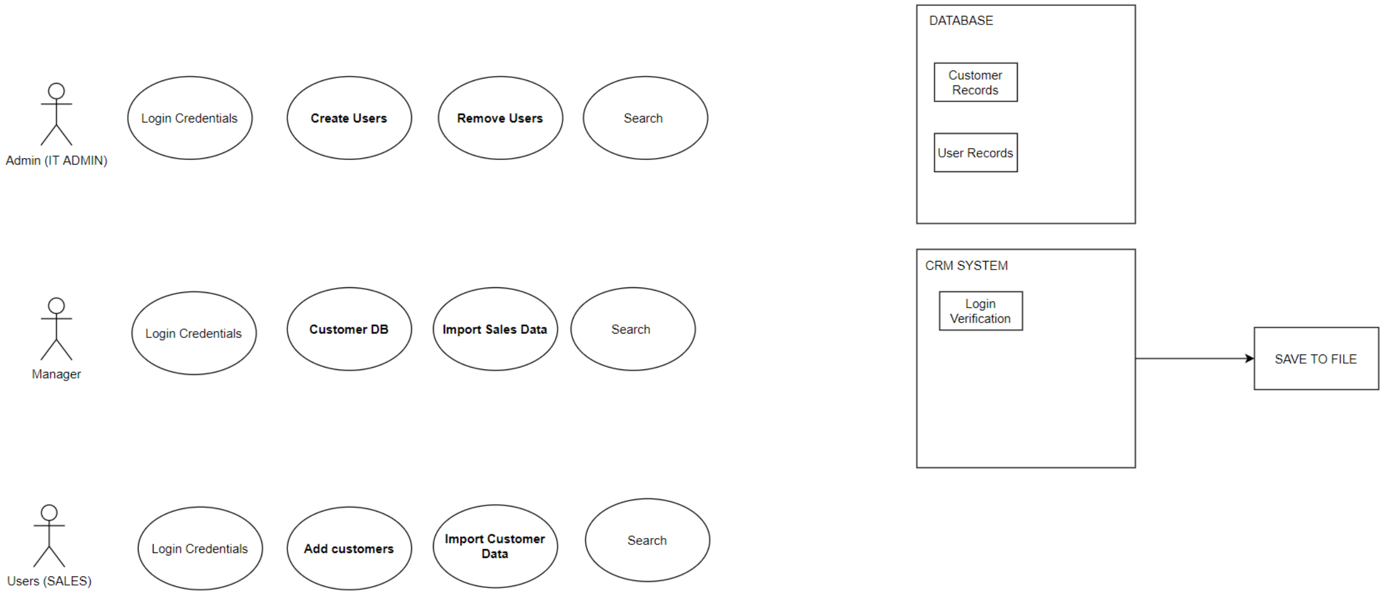
| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User removes client from group |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Client removed from group |
| **Basic Flow or Main Scenario** | 1. User requests client be removed from a group<br>2. System checks that client and group exists<br>3. System removes client from group<br>4. System confirms success |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Client not found<br>● Group not found |
| **Related Use Cases** | |

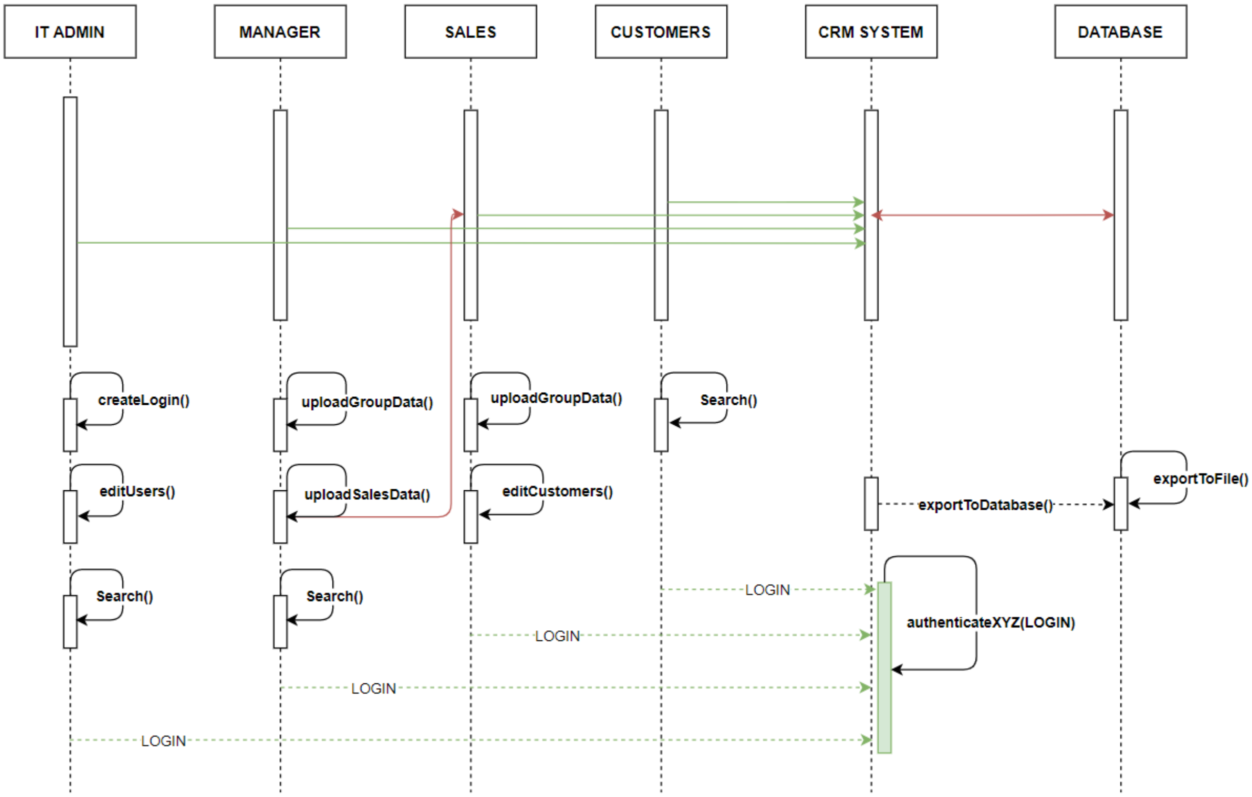
| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | User prints report |

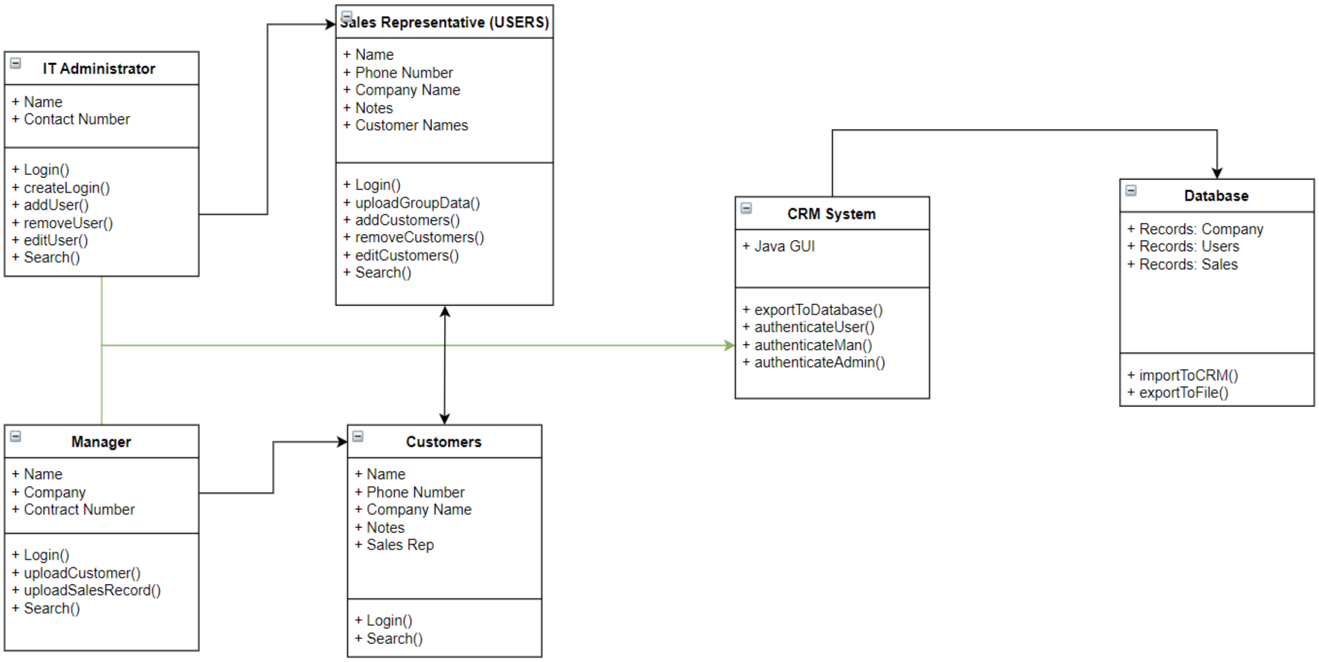
| | |
|---|---|
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | User record data printed |
| **Basic Flow or Main Scenario** | 1. User requests report based on parameters<br>2. System checks for records that match criteria<br>3. System collates all matching records<br>4. System returns results to User |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Search could yield no results |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | System saves data to database |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Data saved to database |
| **Basic Flow or Main Scenario** | 1. System collects all data of all clients<br>2. System sends data to database for storage |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Data corruption |
| **Related Use Cases** | |

| | |
|---|---|
| **Use Case ID** | |
| **Use Case Name** | System loads data from database |
| **Relevant Requirements** | |
| **Primary Actor** | System |
| **Pre-Conditions** | None |
| **Post-Conditions** | Data saved to database |
| **Basic Flow or Main Scenario** | 1. Upon system initialization, system will fetch data from storage<br>2. Data will be read from storage and loaded by system |
| **Extensions or Alternate Flows** | |
| **Exceptions** | ● Data corruption |
| **Related Use Cases** | |


Admin (IT ADMIN)
Login Credentials
Create Users
Remove Users
Search
Manager
Login Credentials
Customer DB
Import Sales Data
Search
Users (SALES)
Login Credentials
Add customers
Import Customer Data
Search
DATABASE
Customer Records
User Records
CRM SYSTEM
Login Verification
SAVE TO FILE


IT ADMIN
MANAGER
SALES
CUSTOMERS
CRM SYSTEM
DATABASE
createLogin()
uploadGroupData()
uploadGroupData()
Search()
editUsers()
uploadSalesData()
editCustomers()
exportToFile()
exportToDatabase()
Search()
Search()
LOGIN
authenticateXYZ(LOGIN)
LOGIN
LOGIN
LOGIN



## 1.4. Overview

The CRM is a Customer Relationship Management application that is used to maintain records of client relationships.

# 2. Overall Description

## 2.1. Product Perspective

## 2.2. Product Architecture

The system will be organized into 3 major modules: the controller module, storage module and the client module.

## 2.3. Product Functionality/Features

The high-level features of the system are as follows (see section 3 of this document for more detailed requirements that address these features):

## 2.4. Constraints

2.4.1. The system does not use any off-the-shelf software for data storage

2.4.2. Administrators are responsible for creating subsequent users of the system

2.4.3. Users cannot manipulate or edit the data of other users

## 2.5. Assumptions and Dependencies

2.5.1. It is assumed that the system will have 3 levels of users: Administrator, Manager, User.

2.5.2. It is assumed that Administrator-level users will create all users.

2.5.3. It is assumed that data will persist for all connected users such that upon restarting or reaccessing the application, all their data will be present from the last session.

2.5.4. It is assumed that User-level users can create/read/update/delete data pertaining to their customers.

2.5.5. It is assumed that User-level users cannot interact with or manipulate data of other users

2.5.6. of the system.

2.5.7. It is assumed that each user has their own account and password.

2.5.8. Groups depend on the the user that created the group

# 3. Specific Requirements

## 3.1. Functional Requirements

### 3.1.1. Common Requirements:

3.1.1.1. Users log into the system using a username and password.

3.1.1.2. Users access the system using a GUI to connect to system

3.1.1.3. Administrator-level users create subsequent users.

3.1.1.4. User-level users can create/read/update/delete records

3.1.1.5. User-level users can create groups and add clients to these groups

3.1.1.6. All user data persists in external data storage

### 3.1.2. Storage Module Requirements:

3.1.2.1. Storage will be custom implementation and does not use any sort of off-the-shelf software (Cannot be a MYSQL db for instance)

3.1.2.2. Data for all users should be stored and loaded when the application is started

### 3.1.3. Controller Module Requirements:

3.1.3.1. Controls the initial startup of the application.

3.1.3.2. Binds port for clients to connect to.

3.1.3.3. Saves and loads all user data.

3.1.3.4. Processes user requests

### 3.1.4. Client Module Requirements:

3.1.4.1. The client module will help facilitate the connection between the client and the CRM controller.

3.1.4.2. A user will provide a login and password to the client which will send a connection request to the server.

3.1.4.3. The client will pass user requests to the server.

## 3.2. External Interface Requirements:

3.2.1. None

## 3.3. Internal Interface Requirements:

3.3.1. None

# 4. Non-Functional Requirements

## 4.1. Security and Privacy Requirements

4.1.1. The system requires a username and password for each user of the system.

## 4.2. Environmental Requirements

4.2.1. System is required to run as a Java based application

## 4.3. Performance Requirements

4.3.1. None