|  |  |  |
| --- | --- | --- |
| iublogo | The Islama University of Bahawalpur  **Department of Information Technology**  **Software Requirement Specification (SRS)**  **ElevatedCRM: A Customer Relationship Management System** | itlogo |

**Project Title:**

ElevatedCRM: An Opensource Customer Relationship Management Solution

**Submitted by:**

Noor Ul Huda (F22BINFT1M01090)

Muneeba Arshad (F22BINFT1M01093)

**Submitted to:**

Sir Muzammil Ur Rehman

**Department of Information Technology**

**Faculty of Computing**

**The Islamia University of Bahawalpur**

### Meeting Details

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No** | **Details** | **Date** | **Supervisor Signature** |
| 1 | Initial SRS Draft Submitted | 04-12-2025 |  |
| 2 | Flowchart and Use Case Review |  |  |
| 3 | Final SRS Approval |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |

## Summary

This document details the Software Requirements Specification (SRS) for **ElevatedCRM**, a Free & Opensource Customer Relationship Management system built upon the Laravel (PHP) and JavaScript frameworks. The project is designed to provide Small and Medium Enterprises (SMEs) and larger Enterprises with a comprehensive toolset for managing the entire customer lifecycle, from initial lead capture and qualification through opportunity management and post-sale customer engagement.

The core functionalities include a real-time Dashboard for sales monitoring, a drag-and-drop Kanban view for pipeline management, detailed Contact/Organization tracking, integrated Activities (calls, meetings, notes), and robust administrative features like Access Control Lists (ACL) and Unlimited Custom Field configuration. The system's modular design ensures high maintainability and extensibility, making it a viable and future-proof solution for customer relationship management.

## Table of Contents

1. Introduction

1.1. Purpose

1.2. Scope

1.3. Product Perspective

1.4. User Characteristics

1.5. Similar apps and systems/Literature Review

1.6. Proposed Technologies

2.Requirements  
2.1. Function Requirements  
 2.1.1. FR001: User Sign Up/Registration  
 2.1.2. FR002: Dashboard Management  
 2.1.3. FR003: Lead & Opportunity Management  
 2.1.4. FR004: Contact Management  
 2.1.5. FR005: Interaction & Activities  
 2.1.6. FR006: Customization and Administration  
 2.1.7. FR007: Product Catalog Management  
 2.1.8. FR008: Reporting & Analytics

2.2. Non-Functional Requirements

3.Use Cases and Flow of Processes

3.1. Use Case 1: Lead Creation and Assignment

References

## 1. Introduction

The Department of Information Technology at The Islamia University of Bahawalpur supports the development of modern, open-source enterprise solutions. This project, **ElevatedCRM**, addresses the critical need for SMEs and enterprises to centralize and streamline their customer interaction and sales processes. Effective customer relationship management is pivotal for sustainable business growth, requiring a system that can track customer data, manage communication, and provide clear pipeline visibility. **ElevatedCRM** is developed as a modular, full-stack application to meet these demands.

### 1.1. Purpose

The primary goal of the **ElevatedCRM** project is to deliver a functional, scalable, and open-source CRM platform. This SRS serves to clearly define the functional and non-functional requirements necessary to achieve this goal, providing a verifiable roadmap for development and successful deployment.

### 1.2. Scope

The project encompasses all features required for complete customer lifecycle management. Items included are the **CRM Dashboard**, **Lead and Opportunity Management** (including Kanban view), **Activities Management** (Calls, Meetings, Notes), **Contact Management** (Persons and Organizations), **Product Catalog**, and **System Administration** (Users, Roles, ACL, Custom Fields). Features explicitly excluded from this scope are deep integration with external proprietary accounting systems and advanced Machine Learning features (e.g., predictive sales forecasting).

### 1.3. Product Perspective

**ElevatedCRM** is a standalone, web-based system built on a modern three-tier (Presentation, Logic, Data) architecture. It functions as the authoritative source for all customer and sales data within the enterprise. It will interface with external services like email servers for automated tracking and logging, but it operates independently of other internal HR or financial systems.

### 1.4. User Characteristics

|  |  |  |
| --- | --- | --- |
| **User Role** | **Description** | **Key System Interactions** |
| **Administrator** | System configuration, user management, security, and customization. | ACL configuration, User CRUD operations, Custom Field setup, full data oversight. |
| **Sales Manager** | Team oversight, pipeline monitoring, performance tracking, and lead assignment. | Dashboard viewing, Kanban analysis, Opportunity funnel reporting. |
| **Sales Agent** | Daily customer interaction, data input, and activity execution. | Lead/Opportunity creation, updating contact records, Activities scheduling, email tracking. |

### 1.5. Similar apps and systems/Literature Review

The CRM market features established commercial and open-source systems. **Salesforce** and **HubSpot** offer comprehensive but proprietary solutions, often with high recurring costs. Open-source alternatives like **SuiteCRM** or **Odoo** provide flexibility but can have steeper learning curves or legacy codebases. ElevatedCRM aims to combine the modern, elegant user experience of systems like HubSpot with the flexibility and transparency of an open-source Laravel/Vue.js stack, offering superior customization (Unlimited Custom Fields) and developer-friendliness (Modular Design).

### 1.6. Proposed Technologies

The system will be developed using a robust and contemporary technology stack to ensure performance, maintainability, and security.

|  |  |  |
| --- | --- | --- |
| **Technology** | **Role** | **Version/Standard** |
| **Laravel** | Backend Web Framework (PHP) | Latest Stable Version |
| **JS** | Frontend JavaScript Framework (SPA) | Latest Stable Version |
| **Database** | Relational Database Management System | MySQL or PostgreSQL |
| **Styling** | Utility-First CSS Framework | Tailwind CSS |
| **Security** | Authentication/Authorization | Laravel Sanctum / ACL Implementation |

## 2. Requirements

The functions of the **ElevatedCRM** project primarily revolve around managing the customer journey. This includes the initial capturing and qualification of leads, visually managing these leads through a sales pipeline via the Kanban view, detailed tracking of all customer interactions (calls, emails, meetings), and ensuring that administrators have granular control over data access and system customization. All these functions culminate in real-time reporting provided on the CRM Dashboard.

### 2.1. Function Requirements

#### 2.1.1. FR001: User Sign Up/Registration

|  |  |
| --- | --- |
| **Name** | **FR001** |
| **Purpose** | Sign up process is required for a user to become a verified member of the system with an assigned role. |
| **User(s)** | Administrator (Creates and approves), New Sales Agent, Sales Manager |
| **Input** | **Name:** Actual name of user. **Email:** Unique email address. **Password:** Must be 8 characters long and include numbers and special characters. **Address:** Actual residence address. **Phone:** Working phone number. **Display Picture:** Latest face picture. |
| **Output** | User record is created with a 'Pending' status. The user will be notified of pending approval. |
| **Approval from Super User** | A designated Superuser (Administrator) will verify the information and manually set the status of the user to 'Active' and assign a specific Role (e.g., 'Sales Agent'). |

|  |  |
| --- | --- |
| **User(s)** | Sales Agent, Sales Manager |
| **Input** | **Lead Name, Email, Phone, Initial Stage, Expected Close Date, Expected Revenue.** Custom fields as configured by the Administrator. |

|  |  |
| --- | --- |
| **Name** | **FR003** |
| **Purpose** | To transition a qualified lead into a final outcome, updating financial metrics and converting the lead into a permanent customer. |
| **User(s)** | Sales Agent |
| **Input** | **Opportunity Status** (Won/Lost). If **Won**: **Final Sale Amount**. |
| **Output** | If **Won**: Lead record is closed, a new permanent **Contact/Customer** and **Organization** record is created, and the Dashboard's revenue metrics are incremented. If **Lost**: Lead record is archived, and pipeline metrics are updated. |

#### **2.1.2. FR002: Dashboard Management**

The system must provide real-time reporting and visual summaries of key sales metrics and activities.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-D.01** | The system shall display a real-time dashboard showing key metrics (e.g., total sales, pipeline value). | High |
| **FR-D.02** | The dashboard shall include widgets for new Leads, upcoming Activities, Customers, and Products. | High |
| **FR-D.03** | The dashboard shall reflect updated metrics upon lead conversion (won/lost) in real-time. | High |

#### **2.1.3. FR003: Lead & Opportunity Management**

The system shall facilitate the capturing, qualification, and progression of potential customers through the sales process.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-L.01** | The system shall allow users to create, view, edit, and delete Lead records. | High |
| **FR-L.02** | The system shall provide a **Kanban View** to visually organize Leads based on their sales stages. | High |
| **FR-L.03** | Users shall be able to drag-and-drop Leads between stages in the Kanban View. | High |
| **FR-L.04** | The system shall allow Leads to be marked as "Won" or "Lost," converting them into Opportunities. | High |
| **FR-L.05** | Each Opportunity shall be associated with an **Expected Revenue** value. | High |

#### **2.1.4. FR004: Contact Management**

The system must maintain a structured repository for all customer and organizational data.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-C.01** | The system shall allow the creation and management of two primary entities: **Persons** and **Organizations**. | High |
| **FR-C.02** | Person records shall include contact details, associated Organization, and a history of all interactions (calls, emails, meetings). | High |
| **FR-C.03** | Organization records shall list all associated Person contacts and a consolidated history of interactions. | High |

#### **2.1.5. FR005: Interaction & Activities**

The system shall track all sales activities and customer communications for a complete interaction history.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-A.01** | The system shall allow users to create and manage Activities (Meetings, Calls, Notes). | High |
| **FR-A.02** | Activities shall be linked to a specific Lead, Person, or Organization. | High |
| **FR-A.03** | The system shall integrate with an email service to track and log all customer email interactions against the corresponding contact record. | High |
| **FR-A.04** | The system shall provide a central view of all upcoming and past Activities. | High |

#### **2.1.6. FR006: Customization and Administration**

The system must provide administrative tools to control user access and customize data fields.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-X.01** | The system shall allow the Administrator to define and assign multiple **User Roles**. | High |
| **FR-X.02** | The system shall implement an **ACL** to control which roles can **edit, create, and delete** records across all modules. | High |
| **FR-X.03** | The system shall allow the Administrator to create **Unlimited Custom Fields** for Leads, Persons, and Organizations. | High |
| **FR-X.04** | The system shall allow developers to customize and extend functionality via a **Modular Design** architecture. | High |

#### **2.1.7. FR007: Product Catalog Management**

The system shall allow Administrators and Sales Managers to manage the product and service catalog used in sales opportunities.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-P.01** | The system shall allow creation, viewing, editing, and deletion of **Product** records. | High |
| **FR-P.02** | Each Product record shall store Name, SKU (Stock Keeping Unit), Unit Price, and Description. | High |
| **FR-P.03** | The system shall allow a Sales Agent to associate one or more Products with an **Opportunity**. | High |
| **FR-P.04** | When Products are associated with an Opportunity, the system shall automatically calculate the total revenue based on unit price and quantity. | High |

#### **2.1.8. FR008: Reporting & Analytics**

The system shall provide capabilities for generating structured reports to analyze sales performance and pipeline health.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| **FR-R.01** | The system shall provide a dedicated Reporting module for generating reports beyond the main dashboard. | High |
| **FR-R.02** | Users shall be able to generate a **Sales Pipeline Report** detailing the value and count of Opportunities in each stage. | High |
| **FR-R.03** | Users shall be able to generate an **Activities Report** filtered by user, date range, and activity type (call, meeting, note). | Medium |
| **FR-R.04** | All reports must be exportable in common formats such as CSV and PDF. | High |

### 2.2. Non-Functional Requirements

#### 2.2.1. Performance

* **NFR-P.01 (Response Time):** The API response time for all primary list views (e.g., Leads, Contacts) shall be less than $500\text{ms}$.
* **NFR-P.02 (Scalability):** The application must efficiently handle and display $100,000+$ records across core modules while supporting a minimum of $100$ concurrent users.

#### 2.2.2. Security

* **NFR-S.01 (Authentication):** User authentication must be secured using token-based mechanisms (e.g., Laravel Sanctum or similar).
* **NFR-S.02 (Authorization):** Data visibility and modification must be strictly enforced via the **ACL** defined in FR004. Users must not be able to access data they are not authorized for.
* **NFR-S.03 (Input Validation):** All user inputs must be sanitized and validated at the backend to prevent common web vulnerabilities like SQL Injection and Cross-Site Scripting (XSS).

#### 2.2.3. Maintainability and Portability

* **NFR-M.01 (Code Standard):** The codebase must adhere to established Laravel and Vue.js coding standards (e.g., PSR-2) to facilitate ease of maintenance and future feature development.
* **NFR-M.02 (Portability):** The system must utilize Laravel's ORM (Eloquent) to maintain database-agnosticism, allowing deployment on different RDBMS platforms (MySQL, PostgreSQL.).

## 3. Use Cases and Flow of Processes

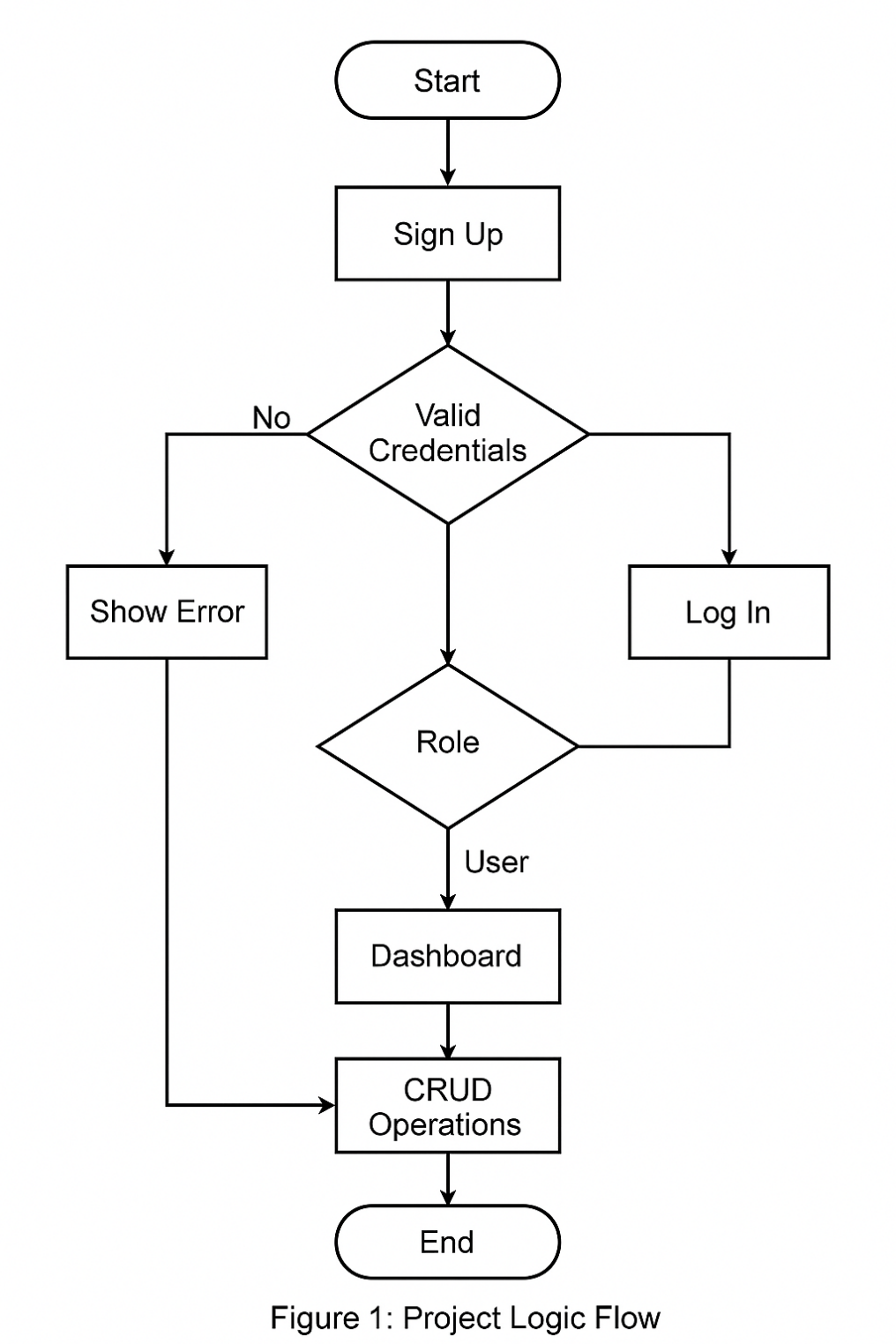
### System Level Use Case Diagram

### 3.1. Use Case 1: Lead Creation and Assignment

|  |  |
| --- | --- |
| **ID** | **UC002** |
| **Name** | Lead Creation and Assignment |
| **Description** | This use case describes the flow for a Sales Agent to input new prospect data and place it into the appropriate initial stage of the sales pipeline. |
| **Requirement(s)** | FR002, FR-L.01 |
| **Actor(s)** | Sales Agent |
| **Precondition** | Sales Agent is authenticated and has 'Create' permission for the Lead module. |
| **Postcondition** | A new Lead record is successfully created and appears in the Kanban view. |
| **Basic Flow** | 1. Actor (Sales Agent) navigates to the 'Create Lead' screen. 2. Actor enters required fields (Name, Email, Phone, Expected Revenue). 3. Actor selects the default 'Prospect' stage and confirms submission. 4. System validates the inputs and checks for duplicate emails. 5. On successful validation, the system creates the new Lead record and assigns it to the Actor. 6. System displays a confirmation message to the Actor. |
| **Alternative Flow** | **Duplicate Email:** System displays an error message, "A Lead with this email already exists," and redirects the Actor to the existing Lead's page for review. |
| **Exceptions** | **Invalid Inputs:** System highlights the required/invalid fields (e.g., missing phone number) and prevents record creation. **System Unable to Respond:** Connection error is logged, and the Actor is notified to try again. |

### Diagrams

#### Activity Diagram for Lead Creation and Assignment (UC002)



## 4. References

[1] Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

[2] IEEE. (1998). IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).

[3] Laravel Documentation (Online Resource), retrieved December 2025.

[4] Vue.js Documentation (Online Resource), retrieved December 2025.