

CS 459: Software Engineering Senior Project

Spring 2025

Requirement Documentation

Project Title	HCAR - Client Database
Sponsoring Company	Humboldt Community Access and Resource Center (HCAR)
Sponsor(s)	Pennie Lee and Wes Patterson
Students	<ol style="list-style-type: none">1. Michael Goodwyn2. Orlando Trujillo-Ortiz3. Justin Crittenden4. Carson Gustafson

Table of Contents

1. Introduction.....	3
2. Requirements.....	3
NON-FUNCTIONAL REQUIREMENTS.....	4
TRACEABILITY.....	4
EVIDENCE OF CONFIGURATION MANAGEMENT.....	4
ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS.....	4
ADDITIONAL REFERENCES.....	4

1. Introduction

This Requirements Document will lay out the Functional / Non-Functional Requirements expected from the primary stakeholders/sponsors of this project. Traceability and working models representing the products expected functionality will be tied to their respective sections of this document.

Functional and Non-Functional Requirements elicited in this document are derived from user stories listed in the Requirements section below. Traceability between such requirements will be expressed through figures and diagrams in the Traceability section. This process of requirement elicitation can be referenced to the IEEE standards for Requirements Engineering.

2. Requirements

2.1. User Stories / Acceptance Criteria

- 2.1.1. As a staff member, I want to be able to view client demographics and non-medical information to better serve our clients.
 - 2.1.1.1. AC1: System will show client information on the home page
 - 2.1.1.2. AC2: Client information is neatly organized by field (name, age, exp date)
- 2.1.2. As a staff member, I want to be able to add new clients to the database so that we can securely record client information.
 - 2.1.2.1. AC1: Creates a new client when user inputs all necessary information to the database
 - 2.1.2.2. AC2: Data is protected by a layer of security for confidentiality
 - 2.1.2.3. AC3: Data is backed up to the server to strengthen data integrity.
- 2.1.3. As a staff member, I'd like to be able to upload my client documents in one location for convenience and so that I can see all my client's information in one place.
 - 2.1.3.1. AC1: Able to import excel documents to autofill a client's information page.
 - 2.1.3.2. AC2: Able to export a client's information page into a formatted excel document.
 - 2.1.3.3. AC3: Should be able to import/export with the click of a button
- 2.1.4. As a staff member, I'd like to be able to view my client's documents in one location for convenience and so that I can see all my client's information in one place.
 - 2.1.4.1. AC1: Staff member is able to upload client's documents to the database.
 - 2.1.4.2. AC2: Documents are securely retrieved from the database for the staff member.
 - 2.1.4.3. AC3: Database is able to accept multiple documents per client.

- 2.1.5. As a staff member, I'd like for the purchase of the service date of expiration to notify me (or turn red) in some way when it is getting close to expiring, so I can avoid missing this date and safely obtain another purchase of services in a timely manner.
 - 2.1.5.1. AC1: Database has a trigger to notify when a purchase of service is close to the expiration date (1 month)
 - 2.1.5.2. AC2: UI reflects the alert with a notification or turning the purchase of service red.
- 2.1.6. As a staff member, I want to be able to write case notes so that others and I can be better informed about our clients
 - 2.1.6.1. AC1: Staff member is able to navigate to a client's "profile."
 - 2.1.6.2. AC2: Staff member is able to add multiple case notes to the client's profile.
 - 2.1.6.3. AC3: Database is able to accept multiple case notes per client.
 - 2.1.6.4. AC4: Case note metadata is stored in the database upon case note creation(User and creation date)
- 2.1.7. As a staff member, I want to be able to review case notes so that myself and others can be better informed about our clients
 - 2.1.7.1. AC1: Staff member is able to navigate to a client's "profile."
 - 2.1.7.2. AC2: When a staff member navigates to the client's profile, a request is sent to grab the client's case notes securely.
- 2.1.8. As a staff member, I want to be able to log into the program so that I can keep client data securely protected.
 - 2.1.8.1. AC1: User is granted access when correct username and password is entered
 - 2.1.8.2. AC2: User is denied access if incorrect username OR password is entered
 - 2.1.8.3. AC3: User is denied access if missing username OR password

2.2. Functional Requirements

- 2.2.1. The system shall allow the system administrators to create or delete clients from the database.
- 2.2.2. The system shall back up database information at the end of every working day.
- 2.2.3. The system shall display client information organized by a data field selected by the user.
- 2.2.4. The system shall only allow users with the proper credentials.
- 2.2.5. The system shall allow system administrators to add or remove user access in the system.
- 2.2.6. The system shall allow users to upload and download multiple documents pertaining to client information.
- 2.2.7. The system shall allow users to connect case note documents to a client's profile.

2.3. Non-Functional Requirements

- 2.3.1. The system shall be simple to use for users with proper credentials.
- 2.3.2. The system shall be efficient and quick to load requested information.
- 2.3.3. The system shall be secure and uphold data integrity.
- 2.3.4. The system shall allow multiple documents to be uploaded / downloaded on user request.

- User Stories and Acceptance Criteria (the primary driver)
For each user story, include:
 - User Story ID & Title
 - Description: (As a type of user, I want some goal so that reason/benefit).
 - Acceptance Criteria: Clear, testable conditions (e.g., “Given X, when Y, then Z”).
 - Priority or Story Points (Optional)
 - Sprint Assigned (Optional, updated each sprint)
- Use Case Model for Functional Requirements (additional helper)
 - Graphic Use Case Model
 - Textual Description
For Each Use Case
 - Use Case Name
 - Participating Actors
 - Entry Condition(s)
 - Normal Flow of Events
 - Exit Condition(s)
 - Exceptions (Alternate Flow of Events)
 - Special Requirements
- Rationale: Explain how these use cases tie into your user stories or if they complement them, mapping each use case to one or more user stories

3. Traceability

- 3.1. Use-Case Model for Functional Requirements:
- 3.2. other diagrams (sequence, action, data flow even) we make later

4. Evidence of Configuration Management

5. Engineering Standards and Multiple Constraints

5.1. IEEE

- 5.1.1. IEEE Std 830-1998: Software Requirements [pdf]
- 5.1.2. IEEE Std 29148: Requirements Engineering [pdf]
- 5.1.3. ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
- 5.1.4. Life Cycle Processes
- 5.1.5. Requirements Engineering [pdf]
- 5.1.6. Additional standards suggested by the sponsor(s)

5.2. Additional References

- 5.2.1. Lamsweerde, A.V., 2009. Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley
- 5.2.2. Additional references suggested by the sponsor(s)