

Technical Documentation – STUDI-desktop-app

Table of Contents

1. Initial Technological Considerations.....	1
2. Work Environment Configuration	1
3. Conceptual Data Model (MCD).....	2
4. Use Case Diagram	2
5. Sequence Diagram	2
6. Plan Explanation	3

1. Initial Technological Considerations

The primary objective for developing the application for SoigneMoi hospital was to enhance efficiency and streamline operations. The key considerations included:

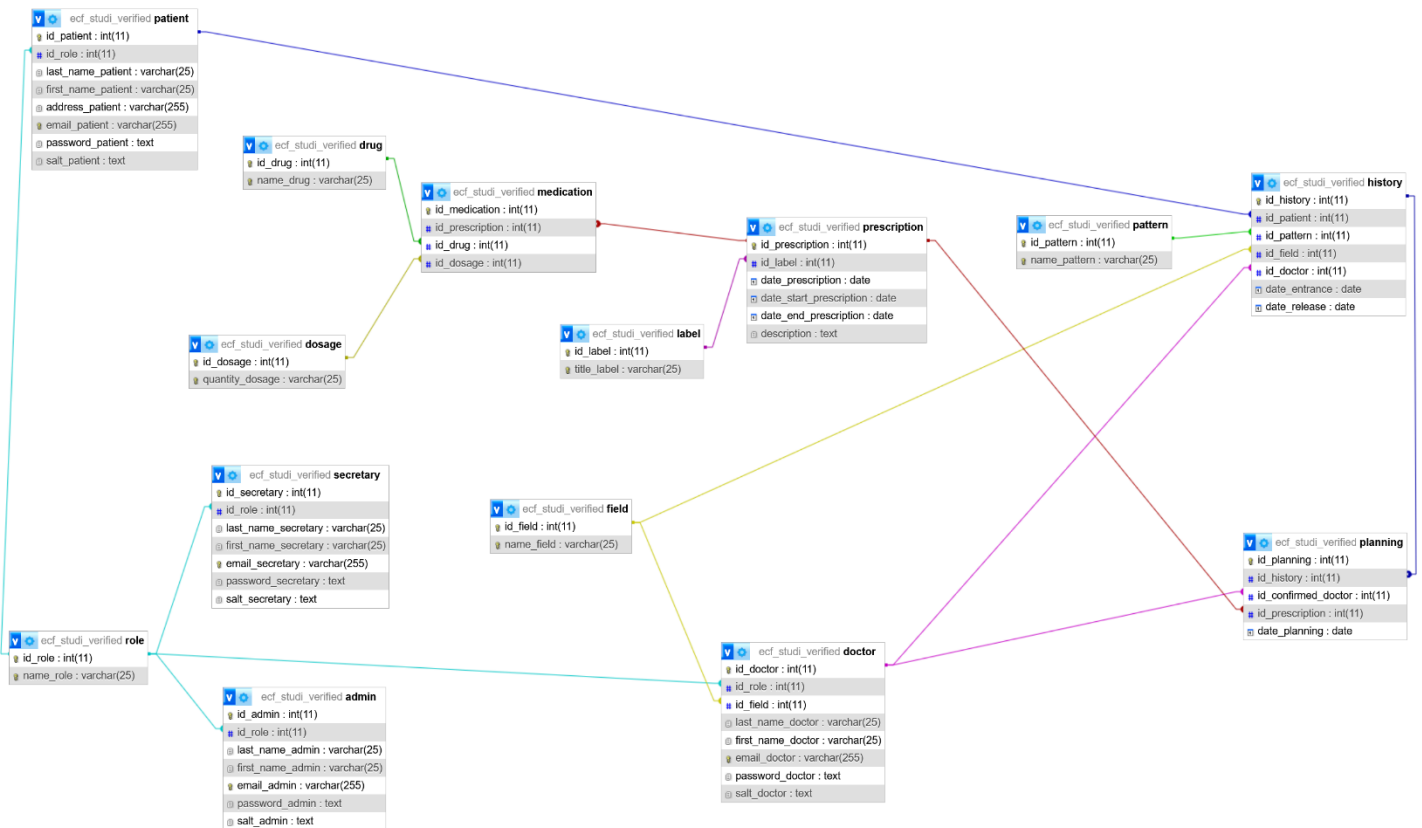
- a. **Scalability:** The application needed to handle an increasing number of users and data without performance degradation. We chose to re-use web back-end.
- b. **Security:** Protecting secretary and patient data are paramount. We implemented secure authentication mechanisms and data encryption.
- c. **User Experience:** The application had to be intuitive and user-friendly. We used Tkinter to ensure a responsive design and Graphical User Interface (GUI).

2. Work Environment Configuration

- a. **Development Environment:**
 - **Operating System:** MacOS
 - **Desktop Server:** Apache 2.4.58
 - **Programming Language:** Python 3.7.9
 - **Graphical User Interface:** Tkinter 8.6
 - **Database:** MySQL 15.1
 - **Version Control:** Git
 - **Development Tools:** Visual Studio Code, phpMyAdmin
- b. **Deployment Environment:**
 - **Deployment:** Local Computer.

3. Conceptual Data Model (MCD)

The conceptual data model defines the main entities and their relationships:



4. Use Case Diagram

The use case diagram represents the interactions between different user types and the system.

Key use cases:

a. **For visitor:**

- Log in

b. **For secretary:**

- View patient list for selected date
- Display patient information

5. Sequence Diagram

The sequence diagram illustrates the flow of operations for:

a. **Log in:**

1. **Visitor** enters email and password then confirm.
2. **System** checks entries.
3. **System** validates the entries or not.

4. **System** confirms the success connection to the visitor or not.
5. **System** opens the next GUI.
- b. **View patient list for selected date**
 1. **Secretary** selects date from calendar.
 2. **System** checks the patient(s) for the day date.
 3. **System** retrieves the list (empty or not).
 4. **System** displays the list.
- c. **Display patient information:**
 1. **Secretary** clicks twice on the patient from table.
 2. **System** redirects to the patient information view.
 3. **System** retrieves the current patient information if exist.
 4. **System** displays the information.

6. Plan Explanation

The test plan was designed to ensure comprehensive testing of the application, covering unit tests, integration tests, and user acceptance tests (UAT).

1. **Unit Tests:**
 - **Objective:** Verify individual components and functions. The main objective is to test all functions in [4. Use Case Diagram](#).
 - **Tools:** Python tests.
2. **Integration Tests:**
 - **Objective:** Ensure modules and components work together.
 - **Approach:** Manual tests.
3. **User Acceptance Tests (UAT):**
 - **Objective:** Validate the application against user requirements. The main objective is to test all functions in [4. Use Case Diagram](#).
 - **Approach:** Conducted with actual users to simulate real-world scenarios.
4. **Security Tests:**
 - **Objective:** Identify vulnerabilities and ensure data protection.
 - **Approach:** Manual tests with SQL injection.
5. **Performance Tests:**
 - **Objective:** Ensure the application performs well under expected load.
 - **Approach:** Manual tests.