Technical Documentation – STUDI-mobile-app

Table of Contents

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 doctor data is paramount. We implemented secure authentication mechanisms and data encryption.

c. **User Experience**: The application had to be intuitive and user-friendly. We used Flutter to ensure a responsive design between Android and iOS platforms.

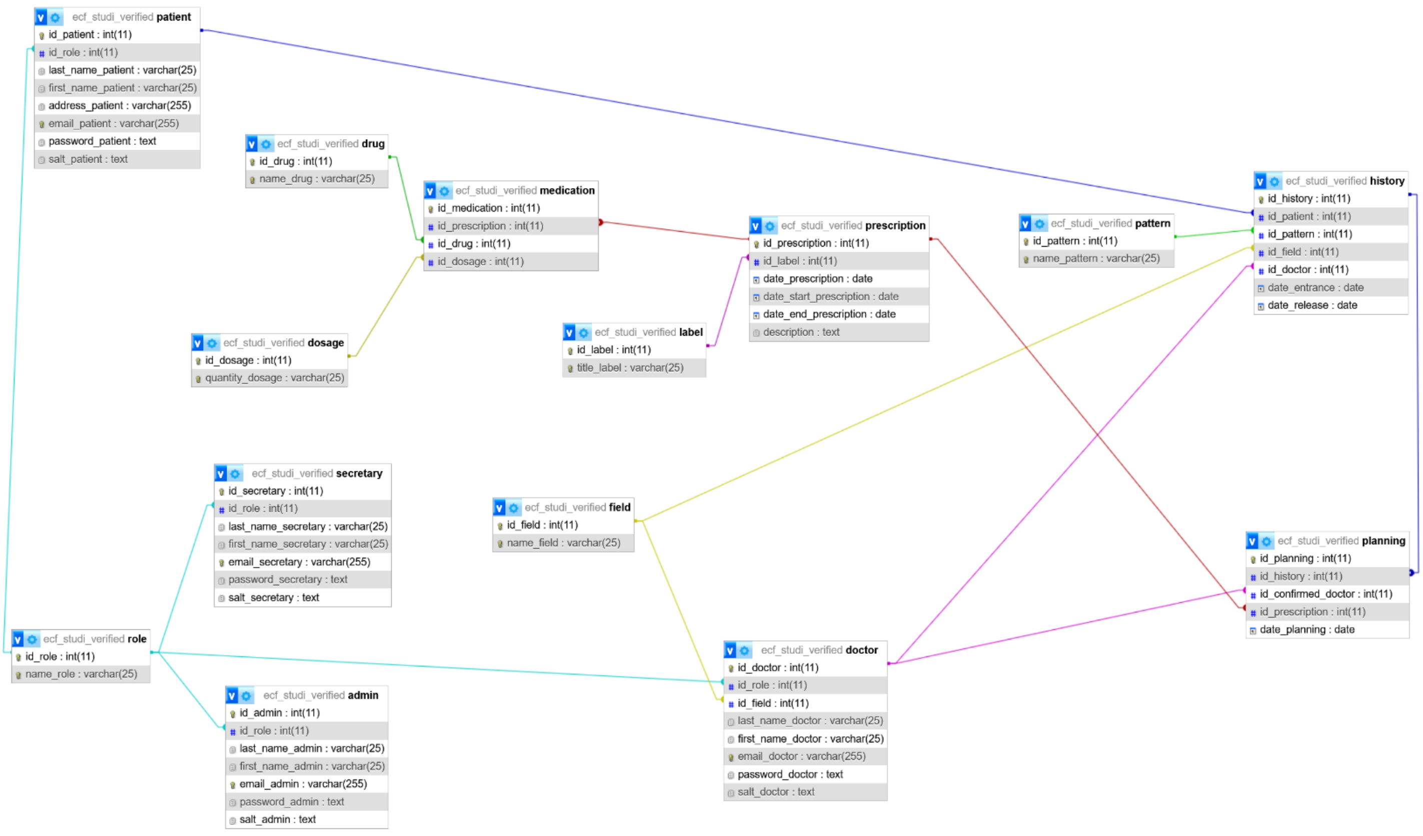
2. Work Environment Configuration

a. **Development Environment**:
   - **Operating System**: MacOS
   - **Mobile Server**: Apache 2.4.58
   - **Framework**: Flutter 3.22.2
   - **Programming Language**: Dart 3.4.3
   - **Database**: MySQL 15.1
   - **Version Control**: Git
   - **Development Tools**: Android Studio, phpMyAdmin

b. **Deployment Environment**:
   - **Platform**: Local smartphone.

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 doctor:**

- View its information
- View today patient list and refresh it
- Display patient prescription
- Add patient prescription

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** retrieves the doctor information and today patient list in parallel.

b. **View doctor information:**

1. **Doctor** clicks on the specific button from log in home.
2. **System** displays the data.

c. **View today patient list:**

During doctor log in:
1. **System** checks the patient(s) for the day date.
2. **System** retrieves the list (empty or not).
3. **System** displays the list.

d. **Refresh today patient list:**

1. **Doctor** clicks on the specific button from log in home.
2. **System** checks the patient(s) for the day date.
3. **System** retrieves the list (empty or not).
4. **System** displays the list.

Note: System also retrieves the list when Doctor comes back to the log in home.

e. **Display patient prescription:**

1. **Doctor** clicks on the specific button from log in home.
2. **System** redirects to the prescription view.
3. **System** retrieves the current patient prescription if exists.
4. **System** displays the information if exists.

f. **Add patient prescription:**

1. **Doctor** enters the prescription then confirms.
2. **System** checks entries.
3. **System** validates the entries or not.
4. **System** saves data or not.
5. **System** confirms the success or not.

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.
   - **Approach**: Manual tests from emulator.
2. **Integration Tests**:
   - **Objective**: Ensure modules and components work together.
   - **Approach**: Manual tests from emulator.
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 from emulator with SQL injection.
5. **Performance Tests**:
   - **Objective**: Ensure the application performs well under expected load.
   - **Approach**: Manual tests from emulator.