

Design and Implementation of a Database for a Medication Management System

1.0 Project Description

Chronic disease should be heart failure
(mainly focused on)

Medication manager is a system designed to assist doctors with managing medications prescriptions to patients. The system aims to consolidate the various activities associated with the prescription of medication for the treatment of chronic diseases.

2.0 Objective

The objective is to design and develop a database system for medication management.

The database will support:

PostgreSQL will be implemented

- User data management across various clinical institutions (Administrators, Doctors, and Patients) must do about multi organisations
- Prescription management, allow doctors to flexibly adjust current prescriptions
- The system should keep track of historical changes to prescriptions (doctors can view past prescriptions)
- Customizable scheduling of medications (daily, every other day, before or after meals, specific time of day)
- Reminders to assist patients with medication compliance
- Tracking patient compliance to medication schedule, symptoms and side-effects

Hint:
Create
Prescription
Table

The system shall ensure data consistency, integrity, and security while meeting the requirements of the project.

soft delete function

Hint:
Create
Prescription_Log
Table

3.0 Scope

Database Design

Schema Design – The database should capture the following data:

- Doctors
- Patients
- Medications
- Prescriptions
- Symptoms
- Side-effects

Look Like the system got the announcement feature, will this system got the notification system. If so, will we care about the operation for messaging between the users and doctors, such as the fcm_key to track the device to send the email.

Hint:
Create
Medication
Table

- Reminders
- Other relevant entities and attributes based on discussions with the client, your personas and user stories

Data Relationships: Establish relationships between tables to support complex queries and enforce data integrity. For example:

- Doctors may have many patients
- Patients may be prescribed multiple medications by their doctor
- Patients may experience symptoms or side-effects after taking a medication

Functional Components

- User Management: Store user profiles for medical staff and patients, including personal information (e.g., name, contact details) and authentication credentials.
- Search and Filter: Allow for efficient querying and filtering of database records such as medication and symptoms based on various criteria (patient, type, time, access level). using Gin Index
- Medication prescriptions: Enable adding, amending, and terminating prescriptions for patients. Amendments to prescriptions should preserve and track the prior details so that doctors can review the historical prescriptions of a specific record. Support the storage of unstructured data such as doctor's notes or documents associated with a prescription. jsonB format
- Medication reminders and compliance: Allow patients to set reminders based on their medication schedule and record when medication was taken.

Technological Requirements

- Database Management System (DBMS): Use a relational DBMS such as MySQL, PostgreSQL, or SQL Server to implement the database.
- Data Security: Implement measures to ensure the privacy and security of patient data, including:
 - Role-based access control for medical staff.
 - Encryption of sensitive data (e.g., login credentials).

Project Deliverables

- Project Proposal: A detailed plan outlining the timeline, milestones, and team responsibilities.
- Database Design Document: A comprehensive document detailing the database schema, entity-relationship (ER) diagram, relationships, and design decisions.

- Database Schema: SQL scripts or database files for creating and populating the database.
- Final Report: Summarize the design process, challenges encountered, and the final database implementation.
- **These are just example deliverables. Please refer to Canvas assignments and marking rubrics for actual deliverables.**

Evaluation Criteria

- Design Completeness: The database schema should include all necessary entities, attributes, and relationships to support the system's functionalities.
- Data Integrity: Implementation of constraints (e.g., primary keys, foreign keys) to maintain accuracy and consistency of the data.
- Performance: The database must support efficient data retrieval and updating, particularly for querying historical vitals and managing medical records.
- Security: Proper role-based access control and encryption mechanisms should be in place (where applicable) to protect sensitive data.
- Usability: The database design should facilitate easy integration with an application interface and support user requirements.
- **These are just example evaluation criteria. Please refer to Canvas assignments and marking rubrics for actual evaluation criteria.**