Phase 3: Design Document

For

MediCare Pro

Complete Clinic Management Solution

by

Hossam Elsabbagh 202202228

Ammar Ahmed 20230087

Mohamed Ahmed 202201266

22-Nov-2025

Table of Contents

1. Introduction

1.1 Purpose of the Document

The purpose of this Design Document is to describe how the system will be structured and implemented. It provides the architectural design, module design and UI specifications required for developers and stakeholders.
This document ensures that all team members share a common understanding before coding begins.

1.2 Scope of the Design Phase

The design phase covers the following aspects of MediCare Pro:

- System architecture (monolithic)

- Module breakdown and responsibilities

- MVC pattern mapping

- Class design and interactions

- Detailed sequence descriptions

- UI/UX wireframe references

- Data design

1.3 Intended Audience

- Developers: To guide implementation.

- Testers: To understand system behavior and interaction flow.

- Project Managers: To track design completeness.

- Stakeholders & Clinic Staff: To understand system structure.

1.4 Overview of Contents

This document includes:

- System overview and design goals

- Architecture and technology stack

- Detailed MVC-based design

- UML-based system modeling

- UI/UX outline

- Data schema and dictionary

- Final summary of decisions made

2. System Overview

2.1 Brief Description of the System

The System is a web-based clinic management platform supporting doctors and administrators. It digitizes patient records, manages appointments, processes financial transactions, stores medical images, and provides analytics.

The system centralizes clinic operations under a secure, role-based interface, allowing multi-clinic management and real-time reporting.
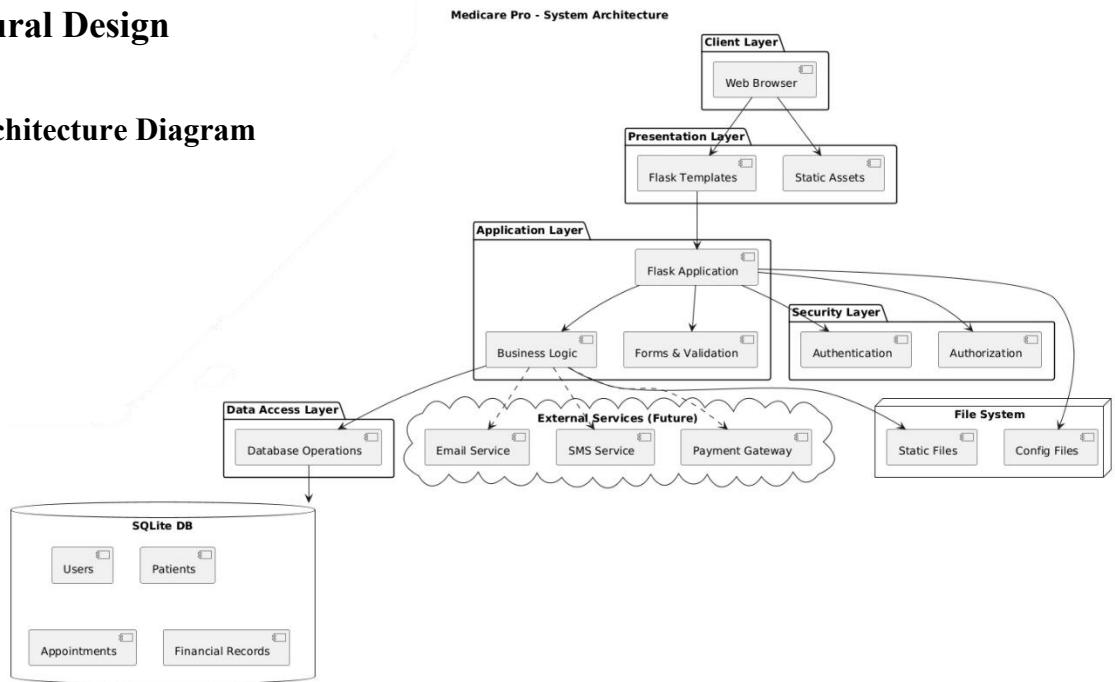
2.2 Key Design Goals and Constraints

Design Goals

- **Scalability:** Support multiple clinics and increasing patient loads.

- **Modularity:** Clear separation of features and components.

- **Security:** Encrypted medical data and role-based access.

- **Usability:** Intuitive dashboards for non-technical users.

- **Reliability:** Consistent uptime and robust data storage.

3. Architectural Design

3.1 System Architecture Diagram

3.2 Discussion of Architectural Style and Components

Architectural Style:

All components (authentication, patient management, appointments, finance, images, reporting) run as part of a single backend application.

Main Components

- **Presentation Layer (View):** HTML/CSS, Flask templates.

- **Application Logic Layer (Controller):** Flask routes, controllers for each module.

- **Data Layer (Model):** SQLite database, ORM models.

3.3 Technology Stack and Tools

Category	Technology
Backend	Python + Flask
Frontend	HTML, CSS, JS
Database	SQLite
API	REST (JSON-based)
Authentication	JWT
Hosting	Linux (Ubuntu Server)
Tools	Draw.io (diagrams), GitHub, VS Code

4. Detailed Design

4.1 Model–View–Controller (MVC) Design Pattern

4.1.1 Description of MVC Pattern

MVC divides the system into:

- **Model:** Data structures, database tables, business logic

- **View:** User interface pages

- **Controller:** Handles inputs, routes, and interactions between Model and View

Reasons for choosing MVC:

- Clean separation of logic

- Easy maintainability

- Supports multi-role interfaces

- Prevents mixing of UI and business logic

4.1.2 Responsibilities of Model, View, and Controller

Model Responsibilities

- Represent system data using ORM classes

- Enforce data constraints

- Provide logic (search, update, validate data)

- Handle relational mappings (patient → records, clinic → doctors, etc.)

View Responsibilities

- Display dashboards, forms, tables

- Provide user interaction elements (inputs, buttons)

- Show dynamic data returned by controllers

Controller Responsibilities

- Route requests

- Validate input

- Call model functions

- Prepare data and send to views

- Manage session state and permissions

4.1.3 Interaction Between Components

Example flow (Manage Patient Record):
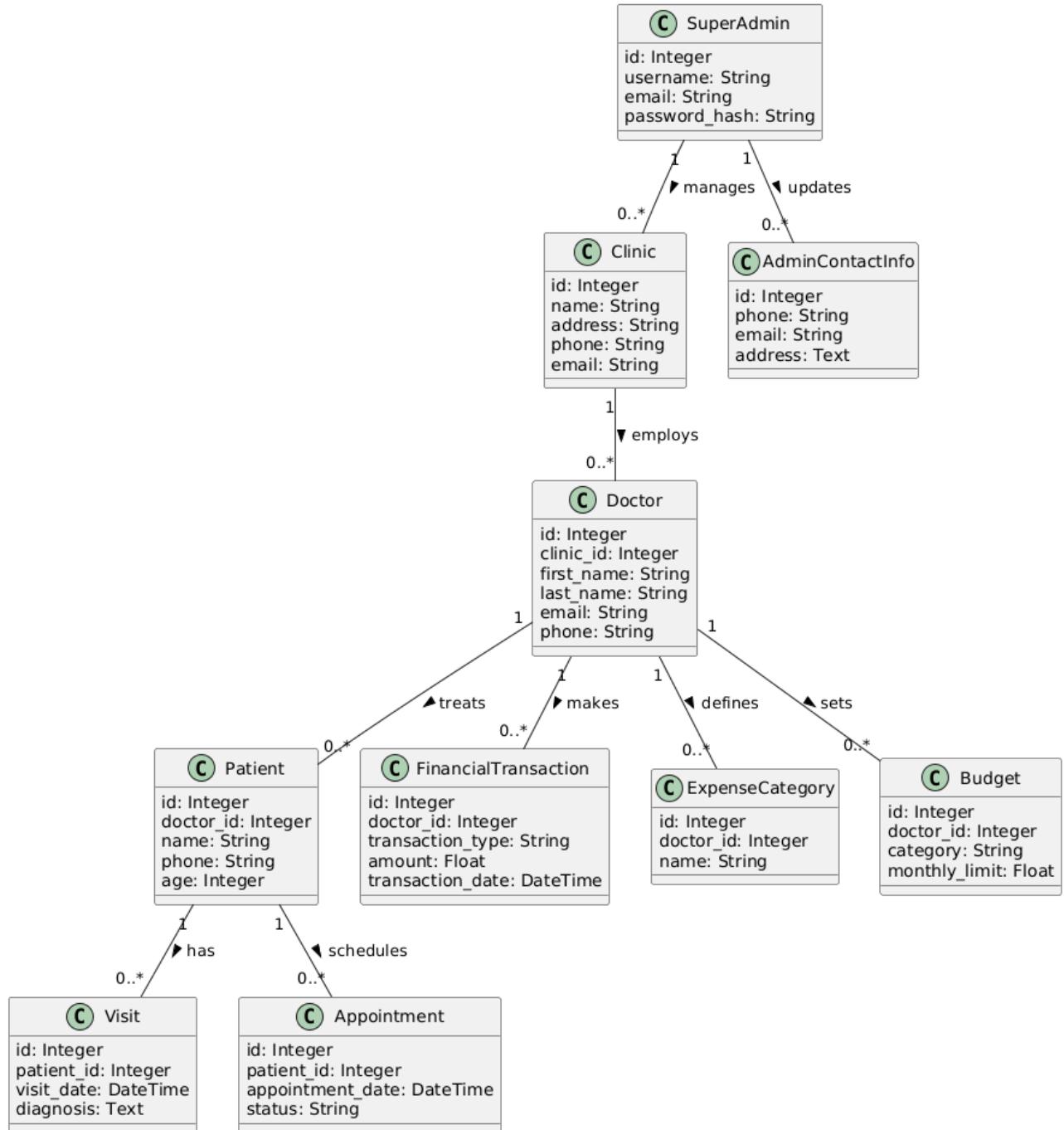
1. User clicks "View Patient"

2. Controller receives request

3. Controller fetches patient data from Model

4. Controller sends data to View

5. View displays patient record

6. User submits changes

7. Controller validates and updates Model
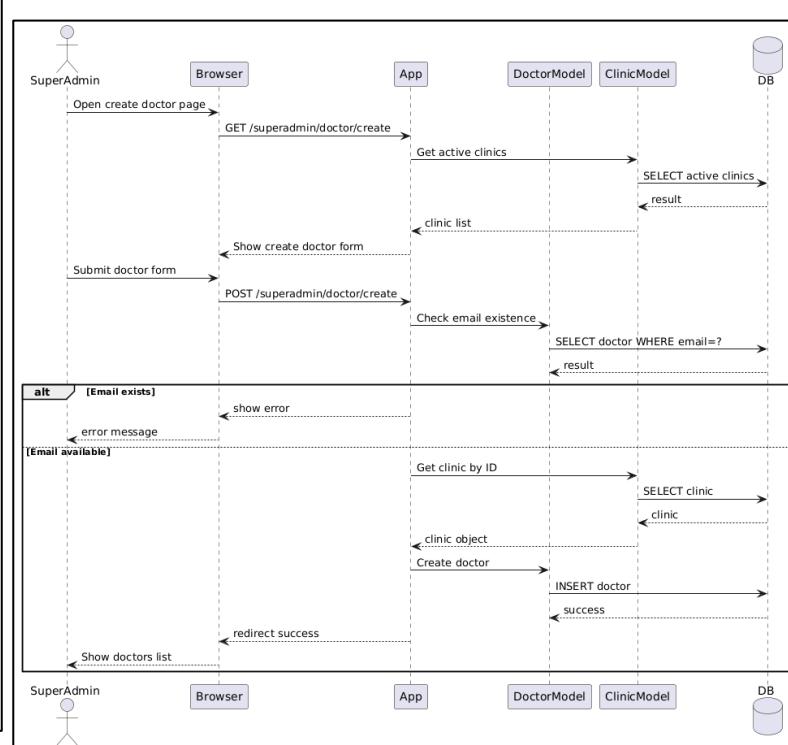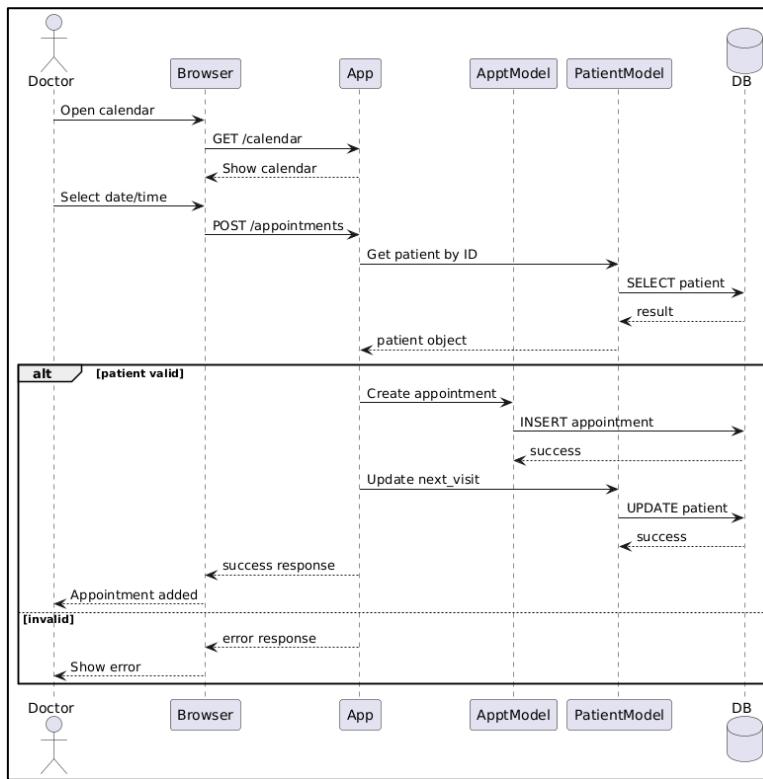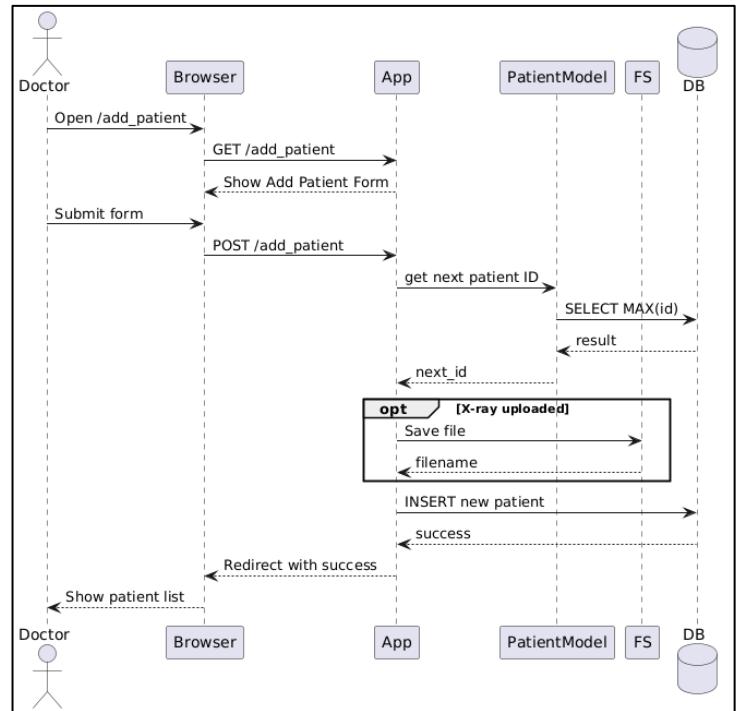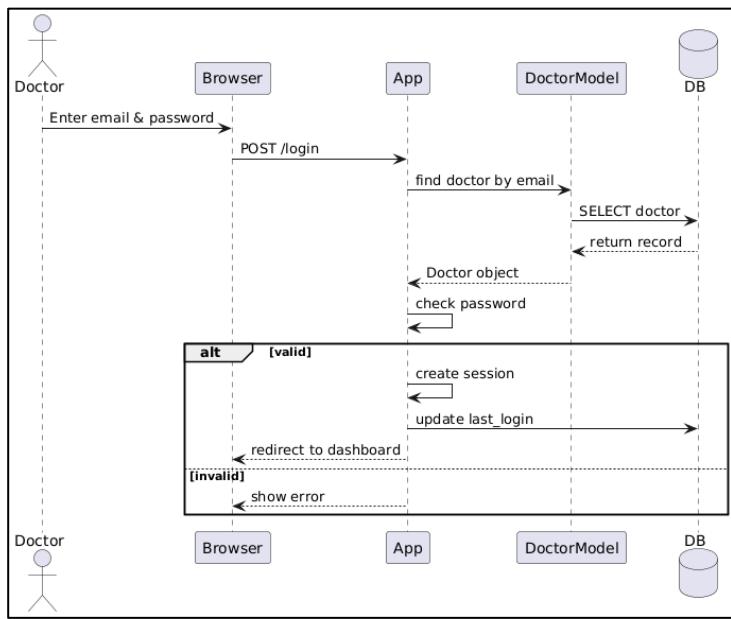
4.2 UML Diagrams

4.2.2 Class Diagram

Medicare Pro - Class Diagram
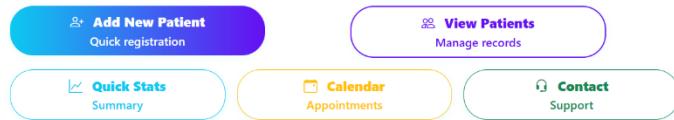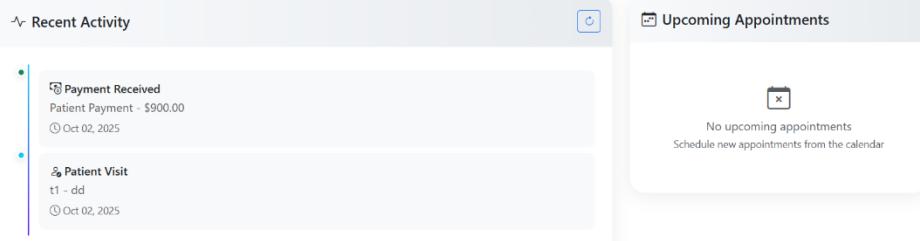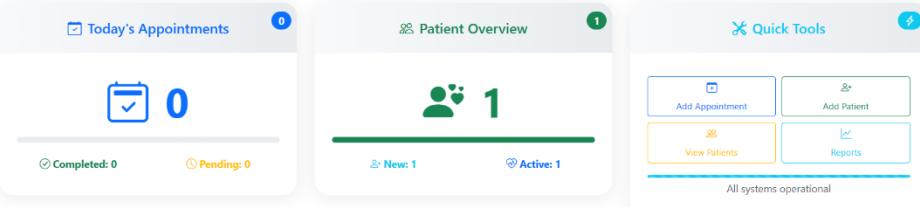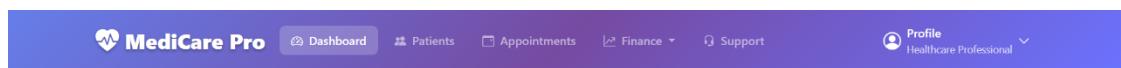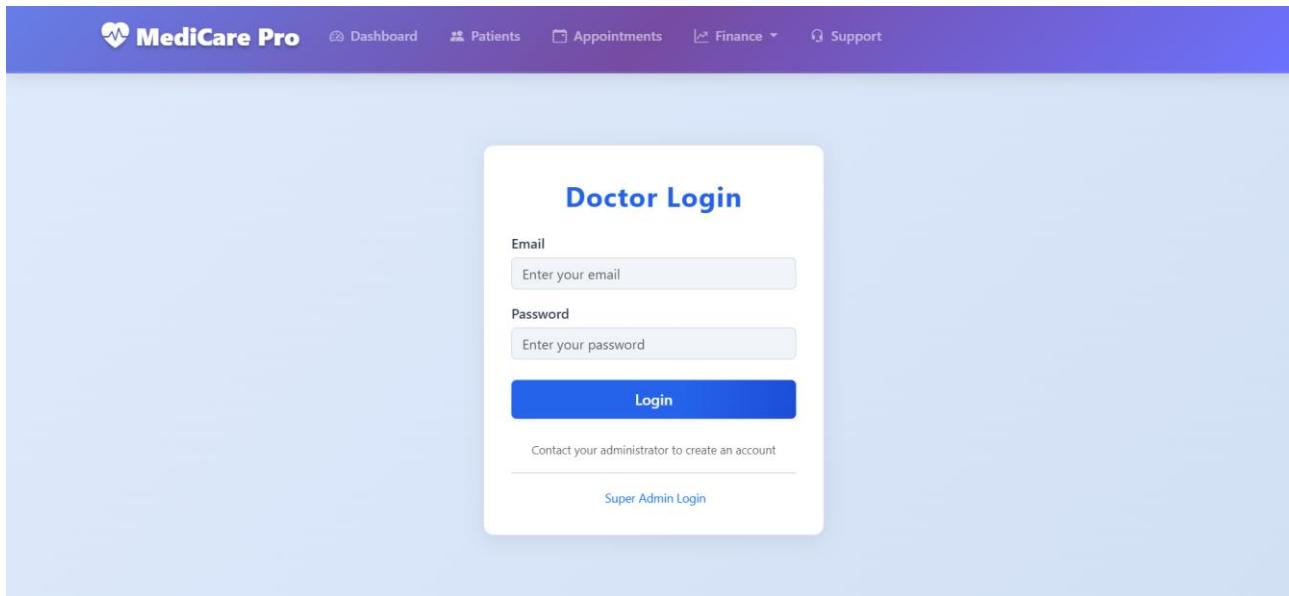
### 4.2.3 Sequence Diagrams

4.3 UI/UX Design

4.3.1 Wireframes / Mockups

♥ **MediCare Pro** ⌂ Dashboard 👥 Patients ▢ Appointments 📈 Finance ▾ 🎧 Support 👤 **Dr. Ahmed** Healthcare Professional ▾

👥 All Patients

| Search by name or phone or ID | 🔍 Search |

# ID	👤 Name	📞 Phone	▢ Age	⚙ Actions
1	t1	123	11	⊙ View 🗑 Delete

♥ **MediCare Pro**

Your Complete Healthcare Management Solution
Empowering healthcare professionals with cutting-edge technology to deliver exceptional patient care. Streamline appointments, manage patient records, and optimize your practice efficiency with our comprehensive, secure, and user-friendly platform designed specifically for modern medical practices.

🎧 **Contact & Support**

📍 **Address:**
Egypt

📞 **Phone:** +20 123 456 7890

✉ **Email:** admin@clinic.com

⚡ **Quick Access**

⊙ Dashboard
👥 Patient Records
▢ Appointments
📈 Financial Reports

🎧 Get Support
🛡 Privacy Policy
▤ Terms of Service
? Help Center

● All Systems Operational

© 2025 **MediCare Pro** - Healthcare Management System. All rights reserved.   🛡 HIPAA Compliant | Secure | Encrypted

---

♥ **MediCare Pro** ⌂ Dashboard 👥 Patients ▢ Appointments 📈 Finance ▾ 🎧 Support 👤 **Dr. Ahmed** Healthcare Professional ▾

Contact Information
Get in touch with our administrative team

Email Address
admin@clinic.com

Phone Number
+20 123 456 7890

WhatsApp
+20 123 456 7890

Office Address
Egypt

♥ **MediCare Pro**

Your Complete Healthcare Management Solution
Empowering healthcare professionals with cutting-edge technology to deliver exceptional patient care. Streamline appointments, manage patient records, and optimize your practice efficiency with our comprehensive, secure, and user-friendly platform designed specifically for modern medical practices.

🎧 **Contact & Support**

📍 **Address:**
Egypt

📞 **Phone:** +20 123 456 7890

✉ **Email:** admin@clinic.com

⚡ **Quick Access**

⊙ Dashboard
👥 Patient Records
▢ Appointments
📈 Financial Reports

🎧 Get Support
🛡 Privacy Policy
▤ Terms of Service
? Help Center

● All Systems Operational

© 2025 **MediCare Pro** - Healthcare Management System. All rights reserved.   🛡 HIPAA Compliant | Secure | Encrypted

4.4 Data Design

4.4.1 Database Schema / ER Diagram

5. Conclusion

5.1 Summary of Design Phase

This Design Document defines how the MediCare Pro system will be implemented using Flask. It provides:

- Architectural breakdown

- Technology stack

- Class and sequence descriptions

- UI/UX outline

- database structure