Medicode Healthcare Clinic Management System

1. Project Overview

Medicode Healthcare Clinic Management System is a Windows Forms-based desktop application designed for efficient management of clinic operations. The system enables secure and user-friendly handling of patients, doctors, appointments, prescriptions, treatments, and billing.

2. Key Features

- **Role-Based Login:** Admin, Doctor, and Patient login panels with appropriate access levels.
- **Patient Management:** Add, edit, and view patient information.
- **Doctor Management:** Add, edit, and view doctor information.
- **Appointment Management:** Book, view, update, and delete appointments with double-booking prevention.
- **Prescription Management:** Doctors can issue, and review prescriptions linked to appointments.
- **Treatment Management:** Register treatments for appointments.
- **Automatic Billing:** When a treatment is added, a bill is automatically generated using an SQL trigger.
- **View Bills:** Patients can view all their bills and the total amount due.
- **Data Integrity:** Database constraints and triggers ensure consistent, reliable data.
- **Modern, Intuitive Interface:** Simple UI with DataGridView controls and clear navigation.

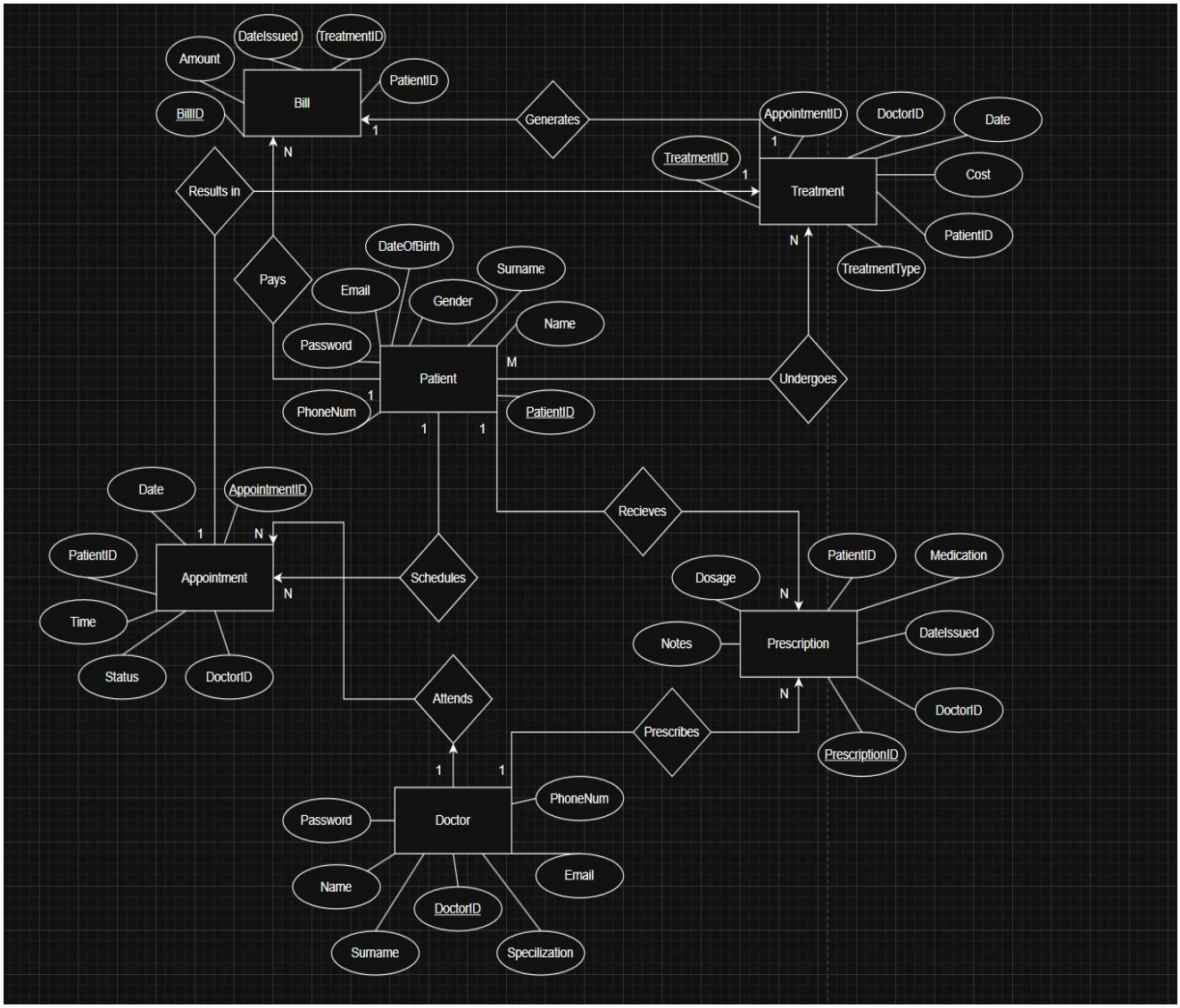
3. System Requirements

- Windows 10 or higher
- Visual Studio 2019/2022
- .NET Framework 4.7.2 or higher
- SQL Server (Express or above)

4. Database Design

- **Tables Used:**
  - Patient
  - Doctor
  - Appointment
  - Prescription
  - Treatment
  - Bill
- **Relationships:**
  - Each appointment links a patient to a doctor.
  - Prescriptions and treatments are linked to appointments.
  - Bills are generated per treatment.
- **Normalization:**
  - The design is normalized up to 3NF, eliminating data redundancy and ensuring referential integrity.

5. Entity-Relationship Diagram (ERD)



6. Application Modules and Flow

6.1. Login and Role Selection

- Users select their role (Admin, Doctor, Patient) and enter their credentials.
- The system authenticates the user and opens the relevant dashboard.

6.2. Dashboard Functionality

- Admin Panel:
    - Manage patients, doctors, appointments.
    - View system-wide reports and statistics.
- Doctor Panel:
    - View and manage only their own appointments.
    - Add and review prescriptions and treatments.
    - See bills generated for their patients.
- Patient Panel:
    - View personal appointments, prescriptions, treatments, and bills.
    - Book new appointments (with time and doctor selection).

6.3. Appointment Scheduling

- Prevents booking at the same time for the same doctor or patient.
- Checks for time conflicts and displays an error message if a slot is already taken.

6.4. Prescription and Treatment

- Doctors issue prescriptions per appointment.
- Treatments are registered and can only be linked to existing appointments.

6.5. Automatic Billing (SQL Trigger)

- When a treatment is added, the following trigger runs.

```
CREATE TRIGGER trg_AutoGenerateBill

ON Treatment

AFTER INSERT

AS

BEGIN

    INSERT INTO Bill (TreatmentID, Amount, DateIssued)

    SELECT

        TreatmentID,

        Cost,

        GETDATE()

    FROM inserted;

END;
```

This ensures every treatment automatically generates a bill, without needing to code this logic in the application.

6.6. Bill Viewing

Patients can view all their bills, including appointment date, doctor, treatment type, cost, and total amount.

7. Error Handling and Data Integrity

- **Database Constraints:**
    - Primary keys, foreign keys, and unique indexes are used to guarantee referential integrity.
- **Application-Level Checks:**
    - Double-booking is prevented by checking for conflicts before adding an appointment.
    - Deleting an appointment also removes related prescriptions and treatments.
- **Trigger Usage:**
    - Automatic bill generation as shown above.

8. How to Install and Run

1. **Clone or Download the Project Files.**
2. **Restore and Configure the Database:**
    - Use the provided SQL scripts to create all tables and triggers in SQL Server.
    - Insert sample data as needed.
3. **Update Connection String:**
    - Edit your app's configuration file to point to your own SQL Server instance.
4. **Build the Project in Visual Studio.**
5. **Run the Application.**
    - Login as Admin, Doctor, or Patient to use the system.

How the Application Works

1. **Login**
    - User selects their role (Admin, Doctor, Patient) and logs in with credentials.
2. **Dashboard**
    - Admin: Manages all users and appointments.
    - Doctor: Views their appointments, adds prescriptions, and sees related treatments and bills.
    - Patient: Views and books appointments, sees prescriptions, treatments, and bills.
3. **Appointments**
    - Appointments can be booked only if the selected time slot is available for both doctor and patient.
    - Users can update or delete existing appointments.
4. **Prescriptions & Treatments**
    - Doctors can add prescriptions and treatments per appointment.
    - Treatment addition automatically generates a bill.
5. **Billing**
    - Every treatment triggers automatic bill creation via an SQL trigger.
    - Patients can view all their bills and total amount due.
6. **Data Integrity**
    - Deleting an appointment also deletes related prescriptions and treatments (with triggers and/or cascade rules).
    - The application checks for time conflicts to prevent double booking.