**Clinic Management System**

**“Project 3”**

Group members: id:

|  |  |
| --- | --- |
| Menna khaled | 221000165 |
| Noha alsayed | 221002299 |
| Sara ahmed | 22100 |
| Habiba essam magdy | 231001460 |

**1.Introduction:**

* Our Clinic Management System project aims to develop well structured and accurate database for managing the most important operations of a medical clinic. This includes scheduling patient appointments, maintaining medical histories, organizing the clinic data, and tracking doctor availability. In the process, we might face little challenges such as, scheduling conflicts, data inconsistency, and manual record keeping inefficiencies. The database makes the access to patient information easier and accurate and most importantly helps in delivering faster and more reliable healthcare services.

**2.Requirements and Scope:**

* As we mentioned earlier, our Clinic Management System is designed to manage the important operations of a multi-department medical clinic by computerizing patient management and scheduling appointments. The system must store detailed patient information, track appointment data and record patient diagnoses. It should also allow querying historical data, such as diagnoses, appointments and also generate reports when needed.

**2.1 Entities Involved:**

1) Department: department ID and name.  
2) Clinic: clinic ID, name, address, and associated department  
3) Doctor : includes doctor ID, name, phone number, address, and assigned department  
4) Patient : holds patient ID, name, contact details, birth date, and job  
5) Appointment: appointment ID, date, patient ID, doctor ID, start time, end time, cost, status, and diagnosis  
  
**2.2 Key Features:**

* Storing patient medical history and personal information
* Scheduling appointments
* Linking doctors to specific departments
* Generating useful queries, such as:  
  1) Diagnoses by date  
  2) Appointments by doctor or department  
  3) Total payments made by a specific patient

**3.Conceptual Design:**

* **3.1 Entities and Their Attributes:**

Entity: **Patient**   
shows: people receiving care at the clinic.  
Attributes:  
1) Patient\_ID (Primary Key)  
2) patient name (Composite)  
3) Phone\_Number (Multivalued)  
4) Birth\_Date  
5) Job  
6) address (Composite)  
7) Age (Derived from Birth\_Date)

Entity: **Doctor**  
shows: Medical professionals who provide care

Attributes:  
1) Doctor\_ID (Primary Key)  
2) Doctor\_Name   
3) Phone\_Number (Multivalued)   
4) Address (Composite)5) Department\_ID (Foreign Key)

Entity: **Appointment**  
 shows: The scheduled medical sessions between patients and doctors.  
 Attributes:  
 1) Appointment\_ID (Primary Key)  
 2) Appointment\_Date

3) Start\_Time

4) End\_Time  
 5) Status   
 6) Diagnosis  
 7) Cost  
 8) Patient\_ID (Foreign Key)   
 9) Doctor\_ID (Foreign Key)

Entity: **Clinic**  
 shows: The physical units in the departments

Attributes:  
 1) Clinic\_ID (Primary Key)  
 2) Clinic\_Name  
 3) Address (Composite)  
 4) Department\_ID (Foreign Key)

Entity: **Department**  
 shows: divisions within the clinic system.  
 Attributes:  
 1) Department\_ID (Primary Key)  
 2) Department\_Name

**3.2 Relationships Between Entities:**

1) Patient — Books —> Appointment  
- A patient can book many appointments, but each appointment can only be booked by one patient. (One-to-Many)

2) Doctor — Schedules —> Appointment  
  
- One doctor can schedule many appointments, but each appointment is assigned to one doctor. (One-to-Many)

3) Doctor — Employs —> Clinic  
- A clinic can employ many doctors, and a doctor can work at more than one clinic. (Many-to-Many)

4) Clinic — Has —> Department  
- Each clinic belongs to one department. (Many-to-One)

5) Doctor — Belongs to —> Department  
- Many doctors are associated with one department. (Many-to-One)

**3.3 Assumptions Made:**

1) Age is a derived attribute and is calculated from Birth\_Date. That’s why it is not stored but calculated when needed.

2) A clinic belongs to only one department, but a department can have multiple clinics.

3) one doctor can work in multiple clinics (even though this isn’t shown in the relational schema image but, it is shown in the ER diagram.)

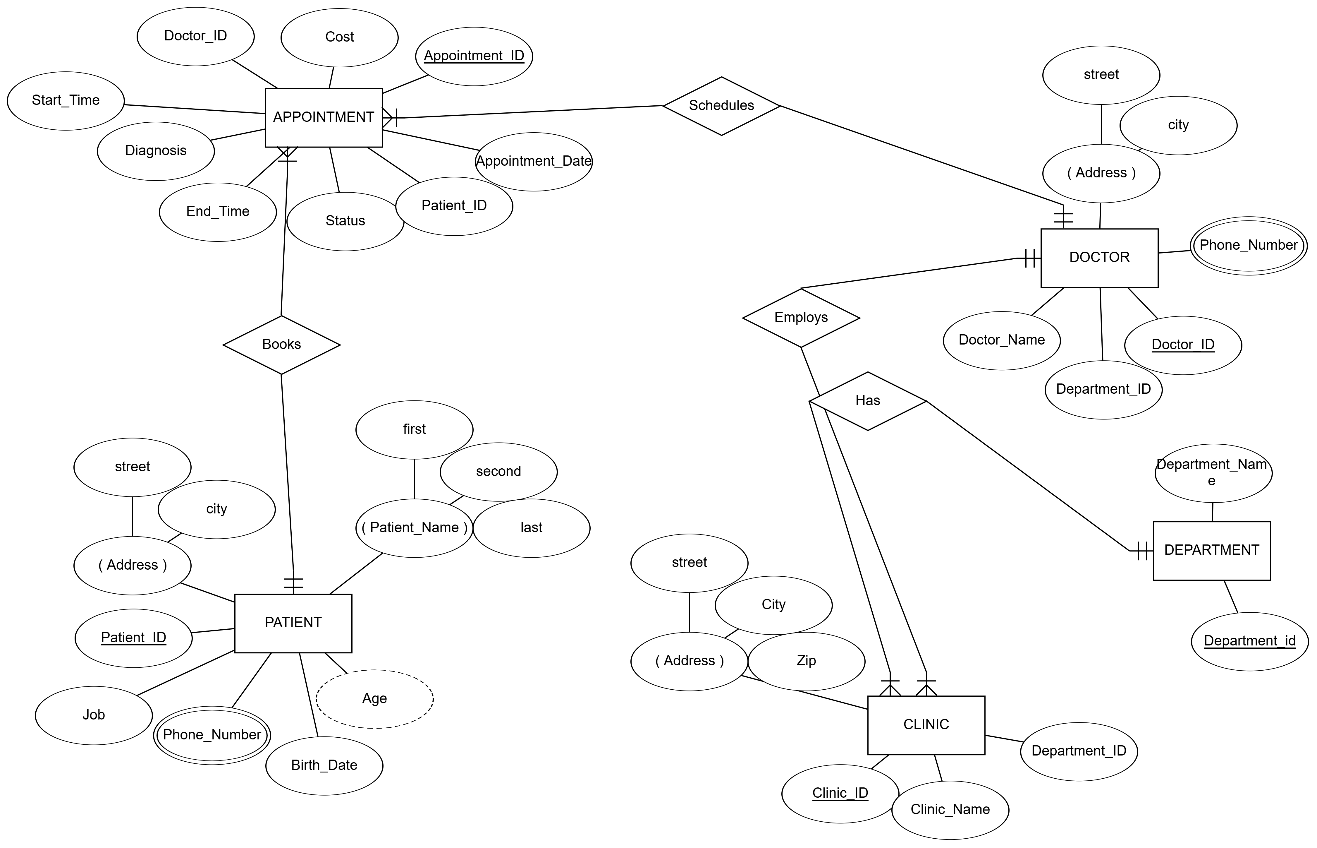
4) The phone numbers are made as separate entities in the relational schema but, they are shown as attributes in the ER diagram.

5) Appointment status is limited to its states (scheduled, in progress, or postponed.)

6) Each appointment is with exactly one patient and one doctor at a time.

7) Clinic address is detailed using Street, City, and Zip rather than a single attribute for easier querying.

**4.4 Entity-Relationship (ER) Diagram:**

****

**4.5 ER-to-Relational Mapping:**

**A diagram of a phone number

AI-generated content may be incorrect.**

1. **SQL code:**

These are the main SQL codes that we used, or in other words the most important codes in our project.

1)A screenshot of a computer

AI-generated content may be incorrect.

* Trigger code

What is it responsible for: It adds the day date automatically when a new medical record is saved.

2)

A screenshot of a computer

AI-generated content may be incorrect.

* Create view code

What is it responsible for: It shows the patient, doctor, clinic, and appointment details, all in the same place.

3) A screenshot of a computer

AI-generated content may be incorrect.

* Insert code

What is it responsible for: It adds new patient information to the database, its made for every entity.

4) A screenshot of a computer

AI-generated content may be incorrect.

* Creating table code

What is it responsible for: It creates two tables to store the department and patient information in the clinic database.

1. **Test data:**

To make sure that our Clinic Management System works well, we have added 10 records to each table. Each one was made to reflect actual healthcare scenarios and to show accurate query results. The following part is a summary of the test data we used, as well as screenshots taken from our application.

1. Patient table:

-We added patients with different information such as age, job, address and date of birth. And that shows real world clinical variety.

A screenshot of a computer

AI-generated content may be incorrect.

1. Doctor Table:

- Doctors were entered with different departments, specializations, and contact info to show proper connection with the clinics and appointments.

A screenshot of a computer

AI-generated content may be incorrect.

1. Clinics Table:

-Each clinic was entered with a unique ID, name, location, and associated department.

A screenshot of a computer

AI-generated content may be incorrect.

1. Appointments table:

-We added appointment scheduling as more than one different status (scheduled, in progress and postponed).

A screenshot of a computer

AI-generated content may be incorrect.

1. **Medical records:**

- Each appointment includes the patient diagnosis, and that allows us to track and the medical histories by patient or by time.

A screenshot of a computer

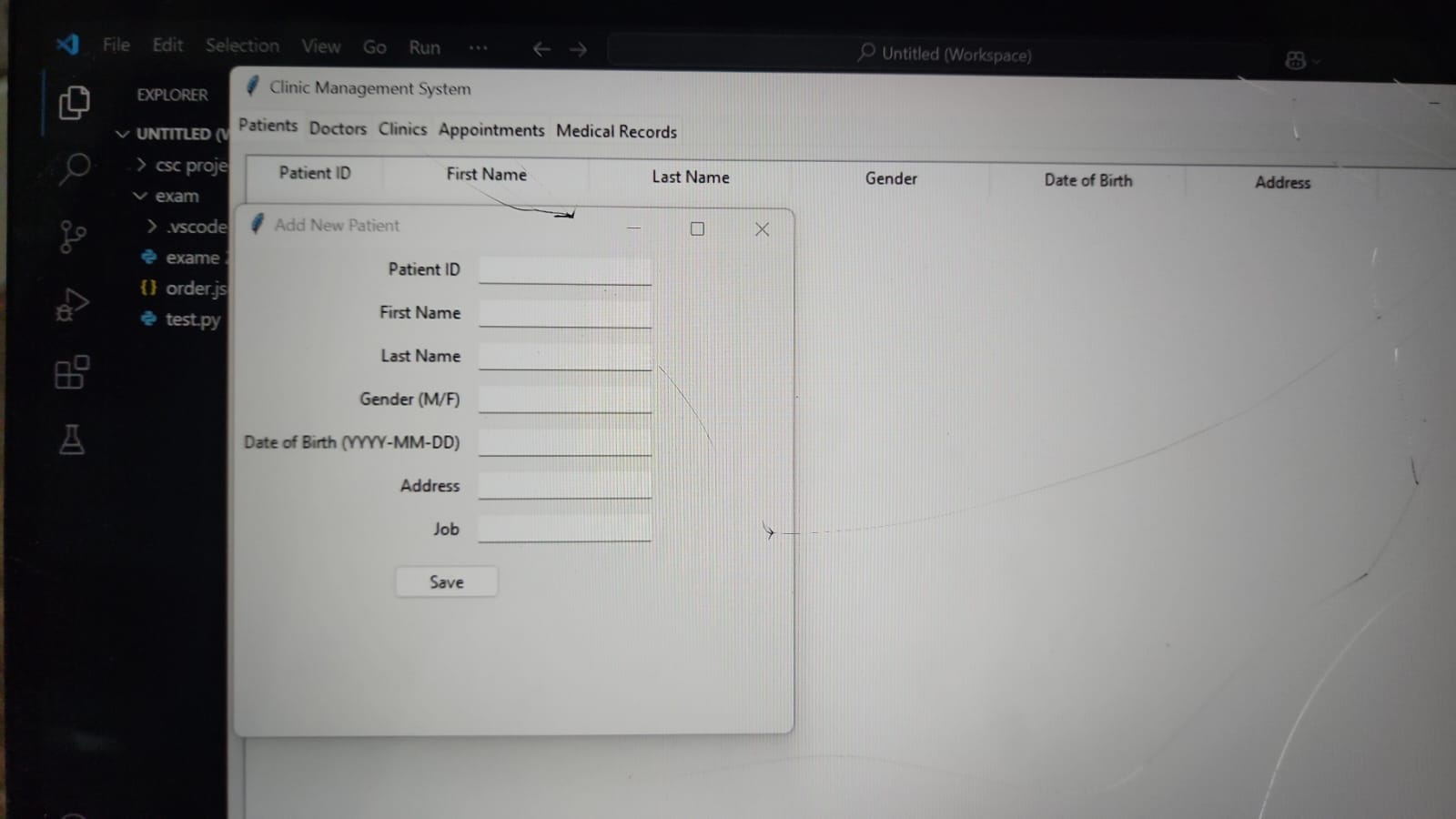
AI-generated content may be incorrect.

1. **The bonus (GUI):**

For the bonus part, we developed a graphical user interface (GUI) for our Clinic Management System using Python with the Tkinter library. Our application was made and shown in Visual Studio and that helps with providing a more user friendly interaction with the database.

The GUI also allows users to:

* See and control patient records
* Add, edit, or delete data
* Schedule and update appointments
* Keep track with diagnoses and medical records



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a computer program

AI-generated content may be incorrect.