

CSEN B401 Database I, Winter 2023  
Milestone 2

Submission: 8.12.2023 (11:59 PM)

In this milestone, you are required to correct your EERD and schema using the evaluation's notes, you also need to check the user stories. You should also take into consideration the original project description that was posted for Milestone 1.

## 1 Requirements

You are required to submit a **zip** file, named after your team number, having the following documents:

- a) SQL files containing the script of both the data and the schema of the database. However no wizards will be used to perform any of the tasks.  
The file will include the following:
  1. SQL file containing the script for tables creation and the relations between them including all needed keys, constraints and formulas.
  2. SQL file containing the script for inserting **consistent and realistic data** inside the tables that covers all the possible scenarios of the user stories, with the mandatory data mentioned in Section 3.
  3. SQL file containing the script for creating all the stored procedures and functions that will be used to implement all functionalities mentioned in the User Stories below (Section 4). You must include a comment above each procedure or function containing the user story number and description corresponding to it. In other words, you should document your SQL file.
  4. SQL file containing the script for executing all stored procedures and functions in your database.
- b) The name of the procedure should be written exactly as it is mentioned in the description or it will NOT be graded.
- c) A PDF file containing:
  1. Team members' names.
  2. Team members' tutorial groups.
  3. Team members' GIU ID.

### Submission guidelines:

- a) Only one copy of the documents will be submitted via ONE team member.
- b) A submission link will be posted on CMS.

Please Note that:

- Stored procedures and functions have to be ordered according to the order of the user stories listed below.

- The tables, columns, and any other objects should have descriptive names.
- No late submissions will be accepted.
- It is your responsibility to make sure that the files were uploaded successfully.
- Not following any of the conventions, or having any problems with the order of tables creation would lead to losing marks.

## 2 Needed Modifications to Milestone 1

This section lists the most common mistakes and modifications needed.

- The most common modification is to ENSURE ALL TERNARY RELATIONSHIPS ARE MODELED CORRECTLY.
- Each Doctor works in only one hospital, However surely multiple doctors can work in the same hospital.
- You should have Lab tests as an entity (either strong or weak)

## 3 Data Insertion

Your file should contain insertions for **at least** the following:

- 16 users of different types. (covering all 4 types)
- 5 Hospitals
- 10 Prescriptions with different combinations of doctors and patient.
- 5 Emergency Medical Services (Providers).
- 5 Insurance Companies.
- 10 Treatments with different types.
- 5 Lab Tests

## 4 User Stories

### 4.1 As an unregistered user I should be able to:

- Register to the website by using my information(attributes), email, and password

**Signature:**

Name: PatientRegister

Input: [Patient's Attributes]

Output: Nothing

**Signature:**

Name: DoctorRegister

Input: [Doctor's Attributes]

Output: Nothing

**Signature:**

Name: PharmacistRegister

Input: [Pharmacist's Attributes]

Output: Nothing

**Signature:**

Name: NurseRegister

Input: [Nurse's Attributes]

Output: Nothing

## 4.2 As any registered User I should be able to:

- a) login using my username and password.

**Signature:**

Name: userLogin

Input: ID int, password varchar(20)

Output: Success bit

- b) add my mobile number(s).

**Signature:**

Name: addMobile

Input: @ID int, @mobile\_number varchar(20)

Output: Nothing

## 4.3 As an admin (Nurse) I should be able to:

- a) List all Patients in the system.

**Signature:**

Name: PatientList

Input: Nothing

Output: table

- b) List all Doctors in the system.

**Signature:**

Name: DoctorsList

Input: Nothing

Output: table

- c) List all Pharmacists in the system.

**Signature:**

Name: PharmacistList

Input: Nothing

Output: table

- d) View the profile of any Patient that contains all his/her information.

**Signature:**

Name: AdminViewPatientProfile

Input: PatientId int

Output: table

- e) Add information to the profile of any Patient

**Signature:**

Name: AdminAddPatientInfo

Input: PatientID (INT), [Patient's info]

Output: Success bit

- f) Update information in the profile of any Patient

**Signature:**

Name: AdminUpdatePatientInfo

Input: PatientID (INT), [Patient's info]

Output: Success bit

- g) List all Doctors' names, their Patients, and Hospitals  
**Signature:**  
Name: AdminViewPatientAndHospitalPerDoc  
Input: Nothing  
Output: table
- h) List all patients and the medications they take  
**Signature:**  
Name: AdminViewAllPatientsAndMeds  
Input: Nothing  
Output: table
- i) Find a certain patient and the medications he/she takes  
**Signature:**  
Name: AdminViewPatientsAndMeds  
Input: PatientID (INT)  
Output: table
- j) List the treatment decided by each doctor to each patient  
**Signature:**  
Name: AdminViewTreatmentPerDocAndPatient  
Input: Nothing  
Output: table
- k) List the Lab tests requested by each doctor to each patient  
**Signature:**  
Name: AdminViewLabPerDocAndPatient  
Input: Nothing  
Output: table
- l) List the Prescription written by each doctor to each patient  
**Signature:**  
Name: AdminViewPresPerDocAndPatient  
Input: Nothing  
Output: table
- m) List all Pharmacists, where they work and the prescriptions they checked  
**Signature:**  
Name: AdminViewPharmacists  
Input: Nothing  
Output: table
- n) View the Profile of any Pharmacist that contains all his/her information  
Name: AdminViewPharmacistProfile  
Input: PharmacistID  
Output: table
- o) List all Nurses, their info, and their assigned patient  
**Signature:**  
Name: AdminViewNurse  
Input: Nothing  
Output: table
- p) List all Hospitals, their info, and the patients' they offer care for  
**Signature:**  
Name: AdminViewHospital  
Input: Nothing  
Output: table

- q) List which emergency medical service providers offer ambulances to which hospitals  
**Signature:**  
Name: AdminViewAmb  
Input: Nothing  
Output: Table
- r) List which emergency medical service providers Manage which call centers  
**Signature:**  
Name: AdminViewCall  
Input: Nothing  
Output: Table
- s) Check which Insurance companies have contracts with which hospitals  
**Signature:**  
Name: AdminViewCon  
Input: Nothing  
Output: Table
- t) Check whether a certain Treatment is covered by Insurance or by Patient  
**Signature:**  
Name: AdminViewCover  
Input: TreatmentID (INT)  
Output: BIT

#### 4.4 As a Patient I should be able to:

- a) Check hospitals and their doctors  
**Signature:**  
Input: Nothing  
Output: Table
- b) Check doctors of each hospital  
**Signature:**  
Input: Hospital Name (VarChar(10))  
Output: Table
- c) Patients make an appointment with a doctor.  
**Signature:**  
Input: DoctorID (INT)  
Output: Success bit
- d) View my profile that contains all my information.  
**Signature:**  
Name: viewMyProfile  
Input: PatientId int  
Output: table
- e) Edit my profile (change any of my personal information).  
**Signature:**  
Name: editMyProfile  
Input: [Any of Patient's Info]  
Output: Nothing

#### 4.5 As a Doctor I should be able to:

- a) View the patients' medical history information  
**Signature:**  
Name: DocViewHis

Input: PatientID (INT)

Output: Table

- b) Request Lab tests for a patient

**Signature:**

Name: DoctorReqLab

Input: DoctorID, (INT), PatientID (INT), LabtestID (INT)

Output: Success bit

- c) Write a Prescription for a patient

**Signature:**

Name: DocWritePres

Input: DoctorID, (INT), PatientID (INT), Date (Date)

Output: Success bit

#### 4.6 As a Pharmacist I should be able to:

- a) Check a Prescription

**Signature:**

Name: CheckPres

Input: PharmacistID (INT), DoctorID, (INT), PatientID (INT), Date (Date)

Output: Table

- b) View my profile and update my personal information.

**Signature:**

Name: PharViewProfile

Input: PharmacistID

Output: table

**Signature:**

Name: UpdatePharProfile

Input: PharmacistID

Output: Nothing