

# XpertBot Academy Internship

## **Xpertenurse**

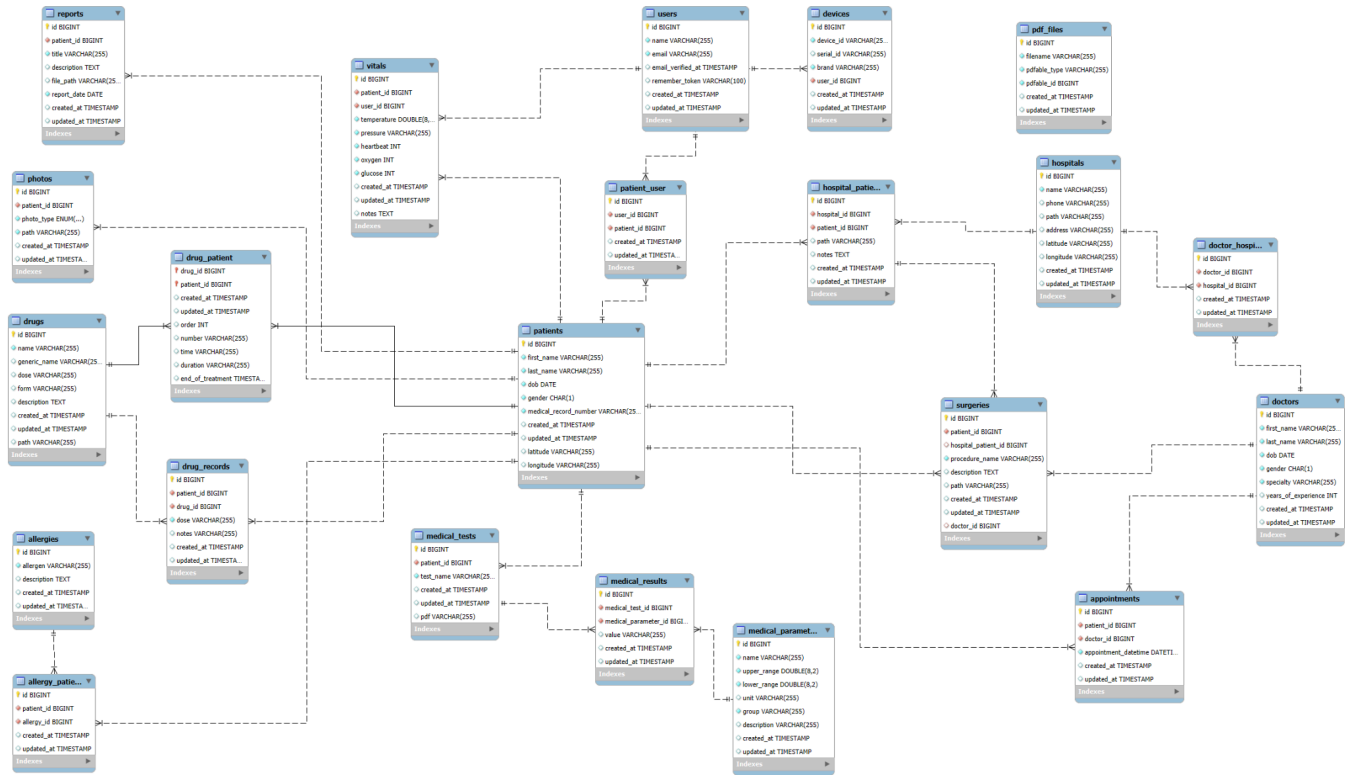
Objectives: Extract insights, Build models,  
and Create data-driven solutions

Intern: Israa Hmadi

Email: [israa.al.hmd@gmail.com](mailto:israa.al.hmd@gmail.com)

# Phase 1:

## 1. ERD:



## 2. Data profiling - Analyzing patients, medications, vitals, visits data

Analysis is available on github in phase 1 codes.

## 3. Schema documentation - Map relationships in healthcare data model

Tables and their relations:

### 1. Users

- One user → many devices (`devices.user_id`)
- One user → many vitals (`vitals.user_id`)
- Many-to-many with patients via `patient_user`

### 2. Patients (Central Entity)

- **Vitals** – One patient → many vitals (`vitals.patient_id`)
- **Drug Records** – One patient → many drug administration records (`drug_records.patient_id`)
- **Drug–Patient relationship** – Many-to-many with drugs via `drug_patient`
- **Allergy–Patient relationship** – Many-to-many with allergies via `allergy_patient`
- **Photos** – One patient → many photos (`photos.patient_id`)
- **Reports** – One patient → many reports (`reports.patient_id`)
- **Surgeries** – One patient → many surgeries (`surgeries.patient_id`)
- **Medical Tests** – One patient → many medical tests (`medical_tests.patient_id`)
- **Appointments** – One patient → many appointments (`appointments.patient_id`)
- **Hospitals** – Many-to-many with hospitals via `hospital_patient`
- **Users**- Many-to-many with users via `patient_user`

### 3. Drugs

- One drug → many drug records (`drug_records.drug_id`)
- Many-to-many with patients via `drug_patient`

#### 4. Drug–Patient (`drug_patient`)

- Join table for many-to-many between patients and drugs.

#### 5. Drug Records (`drug_records`)

- One patient → many drug records
- One drug → many drug records

#### 6. Hospitals

- Many-to-many with patients via `hospital_patient`
- Many-to-many with doctors via `doctor_hospital`

#### 7. Hospital–Patient (`hospital_patient`)

- Many-to-many join table between hospitals and patients.
- One-to-many with surgeries: hospital–patient record → many surgeries (`surgeries.hospital_patient_id`)

#### 8. Doctors

- One doctor → many appointments (`appointments.doctor_id`)
- Many-to-many with hospitals via `doctor_hospital`

#### 9. Doctor–Hospital(`doctor_hospital`)

- Join table for many-to-many between doctors and hospitals.

## 10. Appointments:

- One patient → many appointments (`appointments.patient_id`)
- One doctor → many appointments (`appointments.doctor_id`)

## 11. Medical Tests

- One patient → many medical tests (`medical_tests.patient_id`)
- One medical test → many medical results (`medical_results.medical_test_id`)

## 12. Medical Parameters

- One medical parameter → many medical results  
(`medical_results.medical_parameter_id`)

## 13. Medical Results

- **Many-to-One with Medical Tests** – One medical test can have many medical results  
(`medical_results.medical_test_id`)
- **Many-to-One with Medical Parameters** – One medical parameter can be associated with many medical results (`medical_results.medical_parameter_id`)

## 14. Reports

- One patient → many reports (`reports.patient_id`)

## 15. Allergies

- Many-to-many with patients via `allergy_patient`

## 16. Allergy–Patient (`allergy_patient`)

- Join table for many-to-many between patients and allergies.

## 17. Surgeries

- One patient → many surgeries
- One hospital–patient record can have many surgeries
- One doctor → many surgeries (`surgeries.doctor_id`)

## 18. Photos

- One patient → many photos (`photos.patient_id`)

## 19. Devices

- Belongs to a user (`devices.user_id`)

## 20. PDF Files

- Currently not linked to other entities in the healthcare schema.

## Note: Tables Excluded from Medical Schema

These are system/Laravel-related tables and not part of the healthcare domain model:

- `migrations`
- `password_resets`
- `personal_access_tokens`
- `failed_jobs`
- `telescope_entries`

- `telescope_entries_tags`
- `telescope_monitoring`
- `action_events`

## **Other Phases:**

The code of phase 1 and other phases is found in the repository