Deliverable #:

Moroccan National Health Services (e.g.: Conceptual Design ....)

Data Management Course

UM6P College of Computing

Professor: Karima Echihabi **Program:** Computer Engineering
Session: Fall 2025

Team Information

Team Name	Index Five
Member 1	Zakarya Aze-Dine
Member 2	Adam Ajerouassi
Member 3	Youssef Benhammouda
Member 4	Adam Biar
Member 5	Yahia Belefquih
Repository Link	`https://github.com/Youssefbenhammouda/DBMS-IndexFive`

1 Introduction

The Moroccan National Health Services (MNHS) requires a comprehensive database to manage patient care, staff operations, hospitals, appointments, prescriptions, medications, insurance, billing, and emergencies. This deliverable provides the Entity–Relationship (ER) model for the MNHS system. It captures the main entities, their attributes, and relationships to support both operational queries (e.g., patient admission, prescription tracking) and future analytics (e.g., staff workload, medication demand). The ERD ensures data consistency, supports scalability, and aligns with healthcare management requirements in Morocco.

2 Requirements

This deliverable covers:

- Patient management (personal info, insurance, address, phone, history).
- Staff management (practitioners, caregiving staff, technical staff, certifications).
- Hospital and department management (region, city, wards, specialties).
- Appointment scheduling (patient, staff, department, time, reason).
- Prescription handling (doctor, dosage, duration, medication).
- Medication inventory (pharmacy stock, suppliers, prices, restock dates).
- Insurance and billing management.
- Emergency cases and triage management.
- Linking hospitals, departments, and services under MNHS.

3 Methodology

Below we explain each entity and its relationships, with cardinalities. In our notation:

- **Bold line** = $(1 \dots N)$
- Normal line = $(0 \dots N)$
- **Bold Arrow** = (1)
- Normal Arrow = $(0, 1)$

Patient

Attributes: `internal_id`, `full_name`, `CIN`, `sex`, `birth_date`, `blood_grp`.

- Aggregation: grouped with `Location` (province, city, street, postal code) and `Phone`. A patient may have $0 \dots N$ locations and each one of them has one phone.
- Relation with `Insurance`: a patient can have $1 \dots N$ insurances.
- Relation with `Emergency`: a patient can be admitted in $0 \dots N$ emergencies.
- Relation with `Appointment`: a patient can have $0 \dots N$ appointments, each appointment must involve exactly 1 patient.
- Relation with `Prescription`: a patient can have $0 \dots N$ prescriptions.

Insurance

Attributes: `insur_id`, `coverage_type`.

- Each insurance must be linked to $0 \dots N$ patient.
- An insurance can have $0 \dots N$ bills.

Bill

Attributes: `bill_id`, `generation_time`.

- A bill can be attached to exactly 1 insurance.

Appointment

Attributes: `app_id`, `date`, `time`, `status`, `reason`.

- Each appointment must be linked to exactly 1 patient.
- Each appointment must be linked to exactly 1 staff.
- Each appointment must be linked to exactly 1 department.
- Patients, staff, and departments may have $0 \dots N$ appointments.

Staff (Hierarchy)

Attributes: `staff_id`. Specializations (hierarchy):

- `Practitioners`: license number, specialty.
- `Caregiving staff`: ward, grade.
- `Technical staff`: modality/equipment.

Relations:

- `work_in`: each staff must belong to exactly $0 \dots N$ department. A department may have $0 \dots N$ staff.
- `Handles`: staff can handle $0 \dots N$ emergencies. An emergency can be handled by $0 \dots N$ staff.
- `has_cert`: staff can have $0 \dots N$ certifications. Each certification must belong to 0 or 1 staff.
- `Appointment`: each appointment must be linked to exactly 1 staff.
- `Prescription`: each prescription must be issued by $0 \dots N$ staff.

Department

Attributes: `dept_id`, `name`.

- Each department must belong to exactly 1 hospital. A hospital can have $0 \dots N$ departments.
- Staff and appointments are linked to departments as described above.

Hospital

Attributes: `hos_id`, `name`, `city`, `region`.

- Each hospital must have $1 \dots N$ pharmacy inventory.
- A hospital may have $0 \dots N$ departments.

Pharmacy Inventory

Attributes: `phar_id`.

- Relation with `Medication`: an inventory can contain $0 \dots N$ medications. Relationship attributes: quantity, unit price, reorder level, last restock timestamp.

Medication

Attributes: `drug_id`, `name`, `strength`, `form`, `active_ingredient`, `manufacturer`, `therapeutic_class`.

- May be contained in $0 \dots N$ inventories.
- Must be included in $0 \dots N$ prescriptions. Relationship attributes: dosage, duration.

Prescription

Attributes: `prescrip_id`, `date`.

- Must be linked to $0 \dots N$ patient.
- Must be linked to exactly $0 \dots N$ staff.
- Must include $0 \dots N$ medications.

Emergency

Attributes: `emerg_id`, `triage_id`, `admission_timestamp`, `outcome`.

- Must involve by $0 \ldots N$ patient.
- Can be handled by $0 \ldots N$ staff.

Certifications

Attributes: `cert_id`.

- Each certification must belong to exactly 0 or 1 staff.

4 Implementation & Results

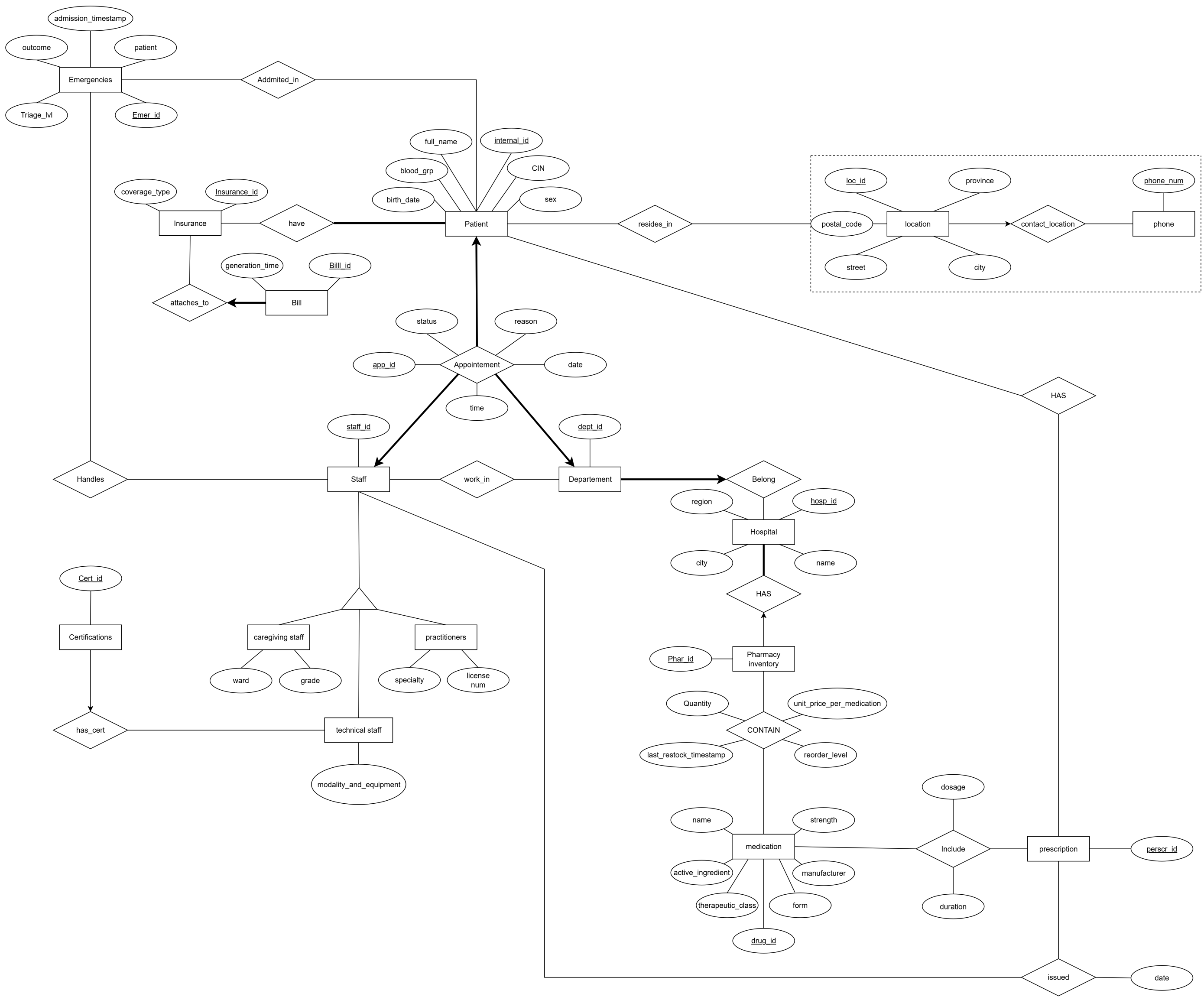

Figure 1: ER Diagram

5 Discussion

Challenges faced: The challenges we encountered are: Modeling the precription system .Since each prescription can include multiple medications and each medication needs its own dosage and duration details. To solve that,we had to create a specific relationship structure to capture this complexity. The emergency section also posed questions:should we require staff assignment for every emergency,or allow for cases where this Information might be missing ?

Observations: During the designing process, it became evident that the nature of relationships within the healthcare domain was by nature complex, particularly many-to-many relationships between entities like patients and staff with multiple appointments, and prescriptions and associated multiple medications. We added time with various components—appointment time, duration of the prescription, and timestamps for emergencies—as notable components needing accurate modeling to support report accuracy and operations. Further, we needed to interweave varying staff roles into one structure (practitioner, caregiver, technical staff), indicating the need for a design that was both flexible and exact for all entities. Through our process, we observed and confirmed clinical databases must balance detailed transactional tracking with the need to facilitate a high-level analytic questioning.

lessons learned: Through this project,we recognized that database design strikes the right balance between theoretical perfection and real-world uusability. By defining ralationship constraints(e.g.specifying that each staff member belongs to one department while a department can contain many staff). Most significantly,we understood the importance of designing database not only for present operational needs but also for future scalability by supporting analytics and expansion.

6 Conclusion

This deliverable presents a complete conceptual design for the MNHS database system by creating an Entity-Relationship(ER) diagram all the healthcare entities with their attributes and their realationships. The design follows the project requirementswhile establishing a scalable foundation for both operational efficiency and future analytical needs..