University
Mohammed VI
Polytechnic

College of Computing

Deliverable #1: Lab2; Conceptual Design.)

Data Management Course

UM6P College of Computing

Professor: Karima Echihabi **Program:** Computer Engineering

Session: Fall 2025

Team Information

Team Name	.....
Member 1	Adam Chikri
Member 2	Amine Abdellaoui
Member 3	Aya Bouayadi
Member 4	Aya Abid
Member 5	Nissrine Benallal
Member 6	Yasmin Bouaboud
Repository Link	`https://github.com/Adam-chi/Lab2-DBMS`

---

1 Introduction

This document outlines the conceptual design for the Moroccan National Health Services (MNHS) data management system. The project aims to model healthcare data to manage: patients, staff, hospitals, appointments, prescriptions, emergencies and billing operations.

2 Requirements

The conceptual design satisfies these following requirements:

2.1 Patients

- Entities:
  - Patients
  - Contact locations
- Attributes:
  - Patients:
    - Internal ID (primary Key), CIN, Full Name, Date of birth, Sex, Blood group, Phone.
  - Contact locations:
    - Contact ID (primary Key), street, postal code, city, province, optional phone.
- Relations:
  - Patient may use a contact location

2.2 Staff

- Entities:
  - Staff, could be: Practitioner, Caregiving, Technical staff.
- Attributes:
  - Staff:
    - StaffID
  - Practitioner:
    - License number, Specialty
  - Caregiving:
    - Grade, ward
  - Technical:
    - Modality/Equipment, Certifications

- Relations:
  - Staff interacts with patients.

2.3 Hospitals and Departments

- Entities:
  - Hospitals
  - Departments
- Attributes:
  - Hospitals
    - HospitalID (primary Key), name, city, region.
  - Departments
    - DepartmentID (primary Key)
- Relations:
  - Departments belong to one hospital.
  - Staff are assigned to departments.

2.4 Appointments

- Entities:
  - Appointments
- Attributes:
  - Appointments
    - AppointmentsID (primary Key), date, time, reason, status.
- Relations:
  - An appointment links exactly one patient and one staff member.
  - An appointment occurs in one department.

2.5 Prescriptions and Medications

- Entities:
  - Prescriptions
  - Medications
- Attributes:
  - Prescriptions:
    - PrescriptionID (primary Key)

  - Medication:
    * DrugID (primary Key), name, form, strength, manufacturer, therapeutic class, active ingredient.

- Relations
  - A Prescription is issued for a patient by a staff member on a given date.
  - A Prescription may include several medication, for each, record dosage and duration.

2.6 Insurance and Billing

- Entities:
  - Insurance
  - Bill
  - Attributes:
    * Insurance:
      · InsuranceID (primary Key), Insurance Type
    * Bill:
      · BillID (primary Key)
  - Relations:
    * Bill must be generated after a consultation or a prescription.
    * A Patient can have more than one insurance.
    * Bill is only linked to one insurance.

2.7 Emergencies

- Entities:
  - Emergencies
- Attributes:
  - EmergencyID (primary Key)
- Relations:
  - Records Patients, admission timestamp, triage level, outcome.
  - Handled by a staff.

2.8 Pharmacy Inventory

- Relations:
  - Track for each hospital the on-hand quantity, reorder level, last restock timestamp, and unit price per medication.

3 Methodology

- Patient is an entity and has its attributes: name, sex... and is related to exactly one contact location (use), therefore we used a thick arrow.

- Staff is an entity related to patient and both are using a 0-to-many relationship (thin line). They work in at most one department (thick line).

- Practitioners, caregiving staff, and technical staff are subclasses of staff; each of them has its attribute, so we use ISA.

- Departments belong to exactly one hospital (thick arrow).

- The fact that staff are assigned to departments and work in departments is actually not the same thing: a staff can be assigned to an X department but be called to work in a Y department.

- Appointments are involved in a ternary relationship with staff and patients because an appointment links one staff and one patient (thick arrow). Appointments occur in exactly one department (thick arrow).

- Prescription is an entity and has date as an attribute. It's issued for exactly one patient by exactly one staff member, so it is a ternary relationship (thick arrow on the prescription).

- Prescription has a 0-to-many relationship with medications (thin line) through the relationship "include", we added the attributes dosage and duration to the relationship to track them based on different prescriptions.

- Insurance is an entity that has the attribute type. It takes the values CNOPS, CNSS, RAMED, private, or none.

- Bills is an entity attached to clinical activities, but we considered this last one as an abstract notion that could be represented as appointments or prescriptions that generate the bill; therefore, we relate them directly to bills using two binary "generate" relationships.

- The relation "own" is a 0-to-many relationship because a patient can or cannot have insurance. A bill is linked to exactly one insurance (thick arrow).

- Emergencies is an entity that records (links) exactly one patient and exactly one staff member that handles it (thick arrow), and the relationship record has the attributes Admission time, triage level, and Outcome.

- Emergencies record the patient and the staff, so it is linked to them (thick arrow). We decided that admission time, triage level, and outcome are attributes of the relation record because it is not that complex of a notion to make it an entity; it only holds a single value about the patient's emergency.

- Hospital is related to medication by the track relationship that has the attributes reorder level, on-hand quantity, unit price, and restock timestamp. Thus, keeping the inventory of the hospital tracked through this relationship.

4 Implementation & Results

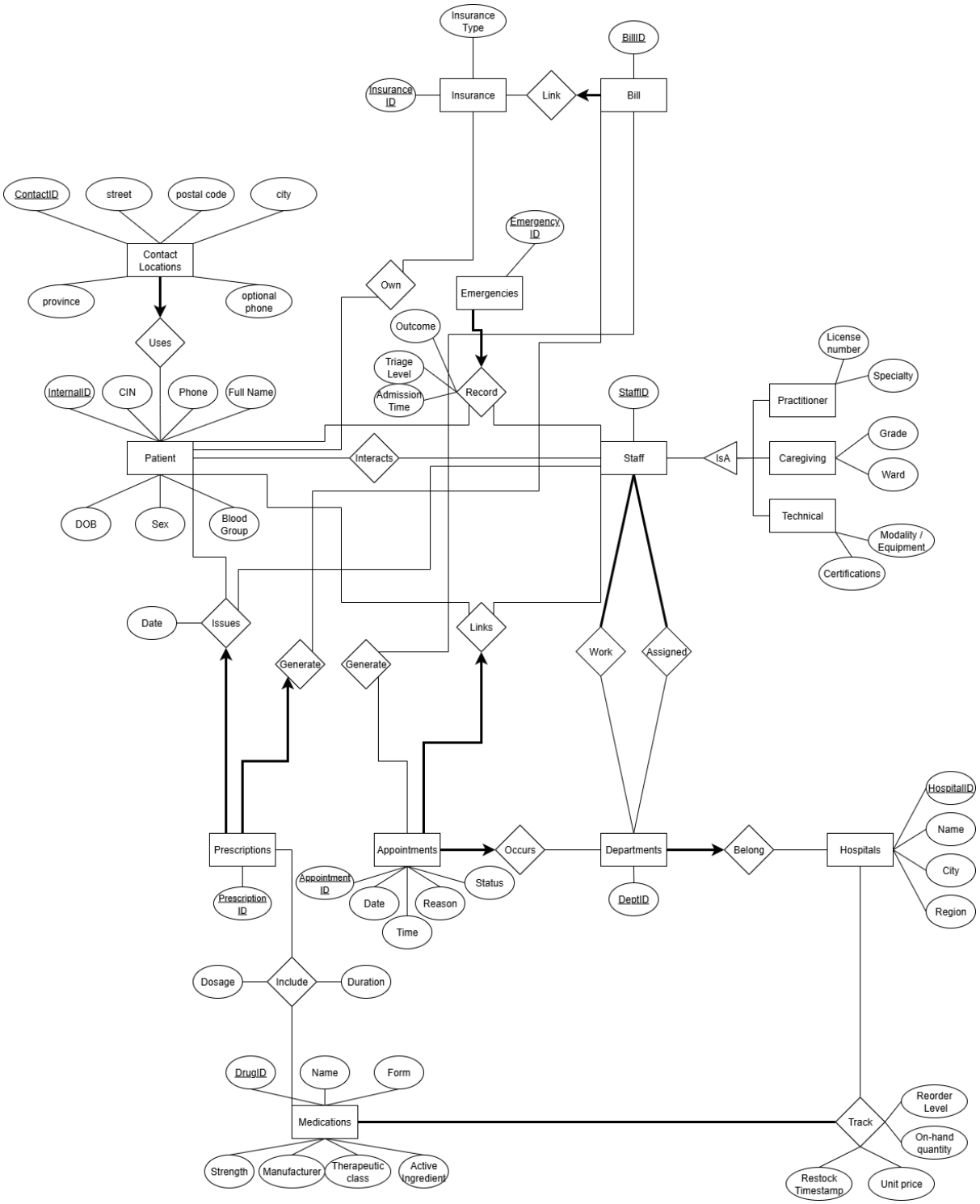

Figure 1: ER Diagram of the system

5 Discussion

- Challenges face:
  - During our project, we faced a lot of hardships, among the main challenges we can cite: distinguishing between some entities and attributes (For example: representing insurance types as attributes and not "IsA" relations) as well as entities and relationshipss, When to and not to use "IsA", Determining abstract notions that don't need entities, and finally, keeping the ER Diagram clean, Sometimes each different team member has a different perspective on the same notion which makes it to an agreement.
- Observations:
  - While working on the project, we noticed that pagestyle patients and staff interact across multiple contexts (appointments, prescriptions, emergencies), indicating the need for centralized identifiers and clear foreign key constraints. Inventory tracking per hospital shows that some data is location-dependent, requiring composite relationships between hospitals and medications with inventory-specific attributes.
- Lessons learned:
  - This project was full of valuable lessons learnt, We discovered that understanding requirements is crucial to designing a diagram that accurately reflects entities and relationships. We also learnt that there's no single right conceptual model, different design approaches can effictively capture the same requirements. This shows the importance of evaluating multiple models and justifying the design choices rather than searching for a perfect one.

6 Conclusion

The design gives a solid and flexible base for the MNHS database. It can handle key healthcare tasks like appointments, prescriptions, billing, and inventory, while keeping data ready for future analysis.