University
Mohammed VI
Polytechnic

College of Computing

Conceptual Design of Moroccan National Health Services (MNHS)

Data Management Course

UM6P College of Computing

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

Team Information

Team Name	Team1/section2
Member 1	Soukaina EL KESSIRI
Member 2	Youness EL HACHIMI
Member 3	Mohamed Ziad EL ABBASSI
Member 4	Alaa Allah EL BAZZAZ
Member 5	Soukaina EL JAOUHARI
Member 6	Oussama EL HILALI
Repository Link	`https://github.com/hash-gdel`

---

1 Introduction

This project is a teamwork based project that focuses on building the Database Management System for the Moroccan National Health Services . Through the deliverable , we design the conceptual Schema to model the different entities and relationships involved in managing the Health Service . We seek to guarantee accurate data organization, efficient access to information , and ensure query handling . The project will highlight a "real-world" use case application of DBMS ,improving the healthcare system through good data handling .

2 Requirements

- Understand the MNHS project domain and define core requirements.
- Identify entities, attributes, relationships, and participation/cardinality constraints.
- Build a first-pass ER diagram.

3 Methodology

Strategy

We approached our ER-diagram by identifying the main entities , In our strategy , we aimed to minimize redundancy and ensure clear relationships, alongside making sure there were the right attributes and relations. We also tried to ensure clarity by separating entities from relationships and use of correct type of lines.

Choice Design

Patient

- **Entity:** patient

  Attributes: internal identifier, national identifier (CIN), full name, date of birth, sex, blood group, phone.

  Primary key: internal identifier (we thought about making cin a primary key but not all patient have cin)

  Relationships:

  - **has** : a one to many relationship with the entity "contact location"

- **Entity:** contact location (made as an entity because it has its own attribute and to be able to represent the one to many relationship with patient)

  Attributes: contact_id, street, city, province, postal code, and optional phone.

  Primary key: contact_id

Staff

- **Entity:** staff

  Attributes: staff id

  Primary key: staff id

  Relationships:

  1. **The ISA relationship**:
     "Staff" is a general entity of practitioners, caregiving staff and technical staff therefore we're using ISA relationship. The practitioners, caregiving staff and technical staff will inherit from "staff" its attributes but will have their own.
     Sub-entities and their attributes:
     - Practitioners: license number, specialty
     - Caregiving staff: record grade ward
     - Technical staff: modality/equipment , certifications
  2. **Interact**: is made as a trinary relationship since it links patient, staff, and clinical activity (as we considered it another entity)

- **Entity:** clinical activity

  Attributes: name (also primary key; it's the only one)

  Clinical activity is an entity and not a relationship because it will be linked later to other entities via other relationships

Hospitals and departments:

- **Entity:** hospital
  - **Attributes:** name, city, region
  - **Primary key:** name
- **Entity:** department
  - **Attributes:** dept_id
  - **Primary key:** dept_id
- **Relationships:**
  - Belongs: a relationship that links department to hospital
    - A department belongs to one hospital 1-1
    - A hospital must have a least one department 1-M
  - Works: staff work in a department O-M

Appointments:

Appointment is made as a trinary relationship to avoid redundancy with descriptive attributes linking department, patient and staff.

Descriptive attributes: date, time, reason, and status .

Scheduled, completed, cancelled are values that status can take meaning we're not representing them

Prescription

We chose to make Prescription an entity since it represents a medical order and has a strong relation with medicaments , patients and staff . Also , prescription has independent attributes from other entities .

Medicaments

We made Medicaments an entity since it has it's own attributes that defines it like **Drug_ID** which is it's primary key, it is also linked to other entities via relations.

Insurance

We made Insurance as an entity as well, since it has independent coverage types a patient may have.

Billing

We also made Billing an entity since it records financial transactions linked to a clinical activity and to other entities such us patients and insurance by relationships .

Emergencies

We made Emergency as an entity as well because it records urgent patient events.

Pharmacy Inventory

Pharmacy Inventory racks on-hand stock, reorder levels, last restock timestamp, and unit price per medication per hospital, however we made it as a relationship , linking hospitals to medication , because it creates a clear relation between these two entities : hospital goes through pharmacy inventory to access medicaments.

Assumptions

For Assumptions, We linked prescription to medicaments (one to many) with a relation to store attributes like dosage and duration , and since each prescription can include multiple medications , we also linked patient to insurance (many to many) because patients may have multiple insurances with different coverage types . In addition to that, we linked emergencies to staff (many to many) and patients (many to many) because staff handles emergencies , and each one of them is linked to exactly one patient . As for Pharmacy Inventory , it was linked with medicaments and hospital , creating a relationship between these two entities.

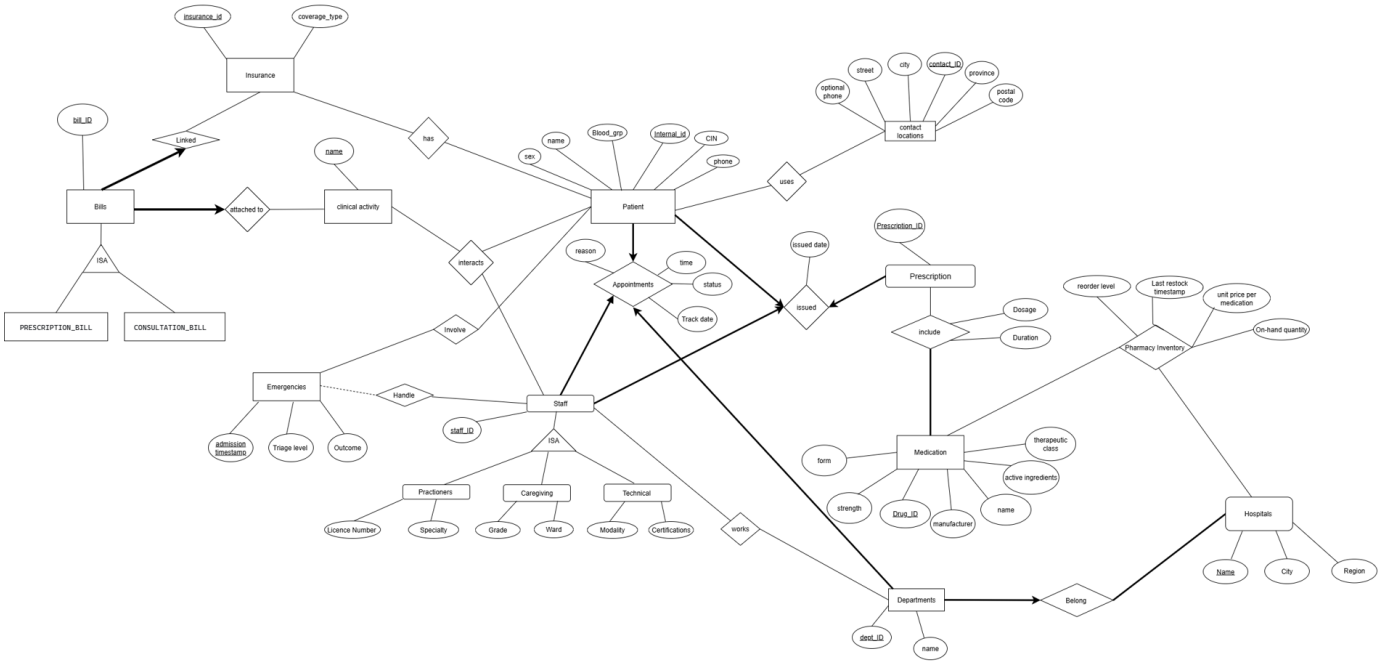

Figure 1: ERD

4 Implementation & Results

5 Discussion

5.0.1 Challenges faced

- Identifying entities and weak entities: sometimes is hard and confusing to identify from the requirements whether something is an entity, an attribute, or even a relationship.
- Choosing the right type of relationships: deciding whether it is a binary, a ternary relationship, or even an aggregation.
- Handling complex relationships: when multiple entities are related to each other, it is hard to organize the diagram and make it readable without crossing too many lines.
- Teamwork: sometimes we may interpret the same statement differently, leading to disagreements on how to represent entities, relationships, or attributes in the ER diagram.

5.0.2 Lessons learned

- Listening to others' opinions and considering their suggestions.
- Making use of specialized tools to draw the diagram.
- Importance of planning and dividing tasks among team members.

6 Conclusion

To wrap up, we successfully accomplished the conceptual design for the Moroccan National Health Services (MNHS) database by thoroughly analyzing the requirements and translating them into a detailed ER diagram. This involved making key design decisions, like that the pharmacy inventory works best as a relationship between hospitals and medications. Though it wasn't always clear-cut, as we had extensive debates about some ternary relationships, such as 'Appointment'. In the end, we can confidently say that this conceptual model gives us a strong and clear foundation to build on in the next stage of the project.