

Report Presented by:

Karim Hamawi

Charbel Azar

Vladimir Abdel Nour

To: Professor Dr. Joe Tekli

IDs: 201602739, 201905620, 201800303 (respectively)

Course Title: Database Systems

December 15, Fall 2021

Contents

[1 Abstract 3](#_Toc90446094)

[2 Introduction 3](#_Toc90446095)

[3 Background 3](#_Toc90446096)

[3.1 Context 3](#_Toc90446097)

[3.2 Problem Analysis 3](#_Toc90446098)

[3.3 Information needs 4](#_Toc90446099)

[4 Proposal 4](#_Toc90446100)

[4.1 Conceptual Data Model (E-R) 4](#_Toc90446101)

[4.2 Logical Data Model (Relational Data Model) 4](#_Toc90446102)

[4.3 Software Application Design and Implementation 6](#_Toc90446103)

[5 Experimental Evaluation 6](#_Toc90446104)

[5.1 Testing Queries for Expert User Interface 6](#_Toc90446105)

[5.2 Testing Queries for Non-Expert Users 6](#_Toc90446106)

[5.3 Highlighting and Testing All System Functionalities 7](#_Toc90446107)

[6 Limitations 7](#_Toc90446108)

[7 Conclusion 7](#_Toc90446109)

[References 7](#_Toc90446110)

# 1 Abstract

Due to the immense pressure hospitals have been facing with the pandemic, our project was targeted at creating a software using a database through web development to manage the data of a hospital. First, we will be discussing the purposes of our project, context, and data needs of the organization. Following that, we will be displaying or CDM and LDM models which were used in the implementation and creation of our database. Finally, we will mention the software that we used and will be discussing our product’s functionalities, testing phases, and limitations while drawing conclusions from our learning experience.

# 2 Introduction

With the pandemic going on, and the situation in hospitals getting worse every minute with every new variant coming along, we thought there will be a huge need for someone to help manage the huge amount of data and patients incoming to the hospitals all over Lebanon and the world. Therefore, we decided to make our project about hospitals, where we will create a database to manage a hospital’s data. To capitalize on our skills, we decided to opt for web development where the software will run on a website. This database is intended to manage the information flow in the hospital be smooth while keep track of all the records and information of all users, materials, medicine, and equipment needed and used in the hospital.

# 3 Background

## 3.1 Context

Our database project is intended to organize and manage the flow of information throughout a hospital smoothly. It will demonstrate the relationships and roles of each member of the staff (doctors, managers, nurses, cashiers) clearly and present each of them with the information they are authorized to access, and allow each of them certain functions that come with their duties.

## 3.2 Problem Analysis

After a long thought process, and a lot of discussion as to how the website will be ran, we came to the conclusion that these are the functionalities that will come with the website, and how it would work. However, this was only the first step before any implementation:

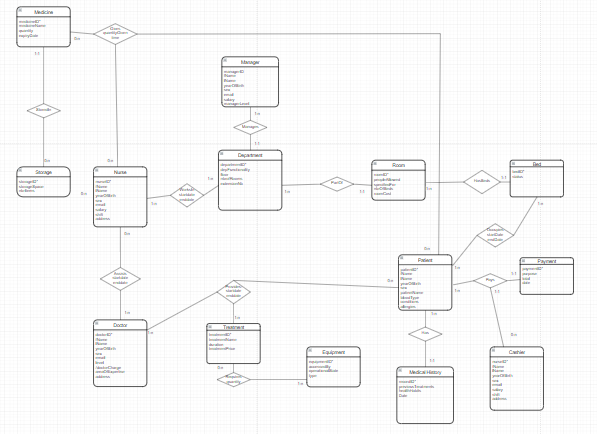
* A portal will exist where there will be accounts for doctors, nurses, managers, and cashiers, each of which have a specific role in the database.
* If the login information of the username and password are entered correctly, the portal will take each user to their corresponding pages.
* The manager will be allowed to see details of all other users and people working at the hospital. He/she is able to see the all the doctors’, nurses’, cashiers’, patients’ information. In addition to that, he/she is able to fire any doctor, nurse, or cashier with a click of a button with it showing in the database and the website. The manager should also see all the patients’ medical histories, beds available in the hospital, available medicine, and a list of paid and unpaid dues on the patients.
* The cashier should simply be able to see the due payment that had not been paid yet by patients, and remove with a click of a button when they are met.
* The doctor should be able to see all the treatments they are supposed to give, in addition to any patients’ medical information and history. They should also be able to see the available equipment for any treatment with the quantities. Finally, the doctor is supposed to add any treatments they intend to give where they are added to the data after they are filled in.
* Lastly, the nurse is able to take any quantity of medicine from the database, where the new quantity is shown, and he/she has access to patients’ records and the quantity of available medicine at all times.

## 3.3 Information needs

All we needed was a little bit of information of how a hospital was ran. In addition to that, we got a few possibilities of medicines, treatments, information on medical records, names of equipment usually used in different types of treatments. All this information was put in the database and used for testing and in all our queries. Our source of information was Charbel’s mother who happened to be a nurse.

# 4 Proposal

## 4.1 Conceptual Data Model (E-R)

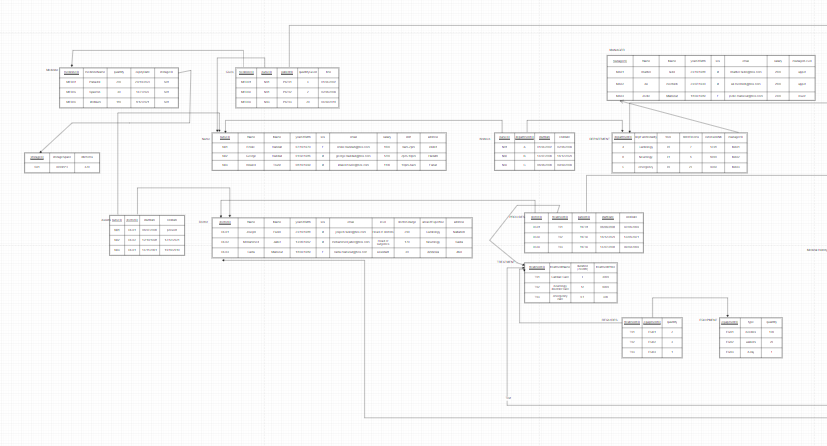


Notes regarding CDM:

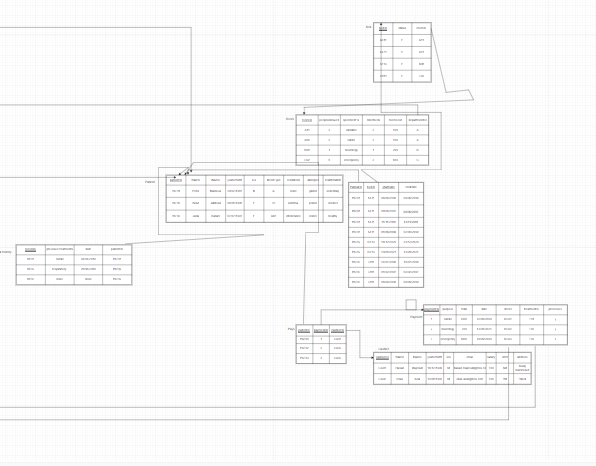
We have 14 entities. Medicine is stored in the storage. Nurses assist doctors. Nurses also give medicine to patients while working at his/her department. The manager manages the department. Rooms are a part of departments and a room has beds. Patients occupy beds, have medical history, and pay payments to cashiers. Finally, doctors give treatments to patients and treatments require equipment.

## 4.2 Logical Data Model (Relational Data Model)

Left part:



Right part:



Notes about LDM:

Unfortunately, due to the size of the LDM and all our entities, we could not fit it in one picture even after getting all the relations together. However, it is just a direct implementation of transformation from the E-R followed by normalization.

## 4.3 Software Application Design and Implementation

The database was implemented using MySQL and PHP was used to query the statements and implement the backend functionalities of our website.

The design was implemented using HTML and CSS.

Doctor account: D01 Password: asd

Manager account: MA01 Password: asd

Cashier account: CA01 Password: asd

Nurse account: N01 Password: asd

For a perfect trial of the project: Open home page- login manager account- hire a doctor, fill in information with the restrictions put in the code- hire a nurse- hire a cashier- fire the doctor- fire the nurse- fire the cashier- check a patient’s medical record- login with cashier- remove a payment- check if it is gone and also gone in the manager’s page- login as nurse- give a medicine and fill in information- check the new table after that- login as a doctor- check a patient’s medical history- add a treatment- check the treatment in the table- check it in the patient’s medical history- check if the payment shows in the manager’s page and the cashier’s page. It is also worth noting that any person that the manager hires will be able to login normally as well.

Done!

# 5 Experimental Evaluation

Testing queries

## 5.1 Testing Queries for Expert User Interface

Since our data is fetched from the database every time a person logs in, any queries or changes in the database that are relevant will appear in the respective tables that they are affecting. For instance, adding a row to the medicine table in a query with “INSERT INTO MEDICINE VALUES (“ME004”, “Advil”, 40,20-10-2023, “S01”);” will add this row to the table and it will show in the nurse’s account. Any expert user can also do any modifications to the database and they will work and show as long as they do not violate any domain, referential, integrity, or general constraints of the database.

Moreover, up until now, some functionalities can only be done by the DBA, such as adding a medicine type, or a new treatment kind all together, and a few others, and these are part of the expert user testing we went through as well.

## 5.2 Testing Queries for Non-Expert Users

Just the same as we mentioned before, we ran the following test trial to check everything in the website and try everything that could possibly go wrong:

For a perfect trial of the project: Open home page- login manager account- hire a doctor, fill in information with the restrictions put in the code- hire a nurse- hire a cashier- fire the doctor- fire the nurse- fire the cashier- check a patient’s medical record- login with cashier- remove a payment- check if it is gone and also gone in the manager’s page- login as nurse- give a medicine and fill in information- check the new table after that- login as a doctor- check a patient’s medical history- add a treatment- check the treatment in the table- check it in the patient’s medical history- check if the payment shows in the manager’s page and the cashier’s page. It is also worth noting that any person that the manager hires will be able to login normally as well.

We also checked all the tables that can be affected after a certain functionality. For instance, a payment must be issued with the total amount of the doctor charge and treatment charge after a doctor enters a payment. The payment should disappear from both the nurse’s, and the manager’s pages when it is removed by the cashier. Any firing or hiring will create that new account or remove the person from the database and tables. A medicine given has the amount reduced from the storage. Also, in addition to many more, addition of a treatment makes it show in a patient’s medical history.

We can add to before that trying to enter illogical data or inacceptable data will always result in an error message. For instance, entering an incorrect ID-password combination, entering a start date which is after the end date, invalid email format that does not end with “@hos.com” and many others.

## 5.3 Highlighting and Testing All System Functionalities

We mostly attempted to access the website through four different users and tested what they were permitted to perform. As a consequence, we can state that our website performed flawlessly and produced accurate, precise, and desirable results.

# 6 Limitations

-Divide payment into multiple deposits-Add restrictions to nurse giving medicine outside her/his shift-Assigning patients to beds and restrictions to time-The ability of the hospital management software to insert new patients to the hospital database without the DBA's help-Change in the design where the user can use a search bar for the desired data instead of scrolling in the table in case there was too much data-Not enough dummy variable due to too many tables-Input new kinds of treatments and other kinds of data without the need for the DBA-Our biggest limitation is that we can expand the data as much as we want adding new entities, functionalities, and relations. For example, we can have a technician, equipment, janitor, and many other things

# 7 Conclusion

This whole project was a very fruitful experience. We learned how to create the database from scratch. We followed that with creating the website while using all the different queries we need to be put into the website in addition to implementing our knowledge of design in creating it. All the website’s functionalities turned out exactly as planned at the end as expected, although there were a lot of struggles and problems along the way. Overall, however, the site turned out to be fully functional, dependable, and applicable in a real-world setting. Thank you!

# ­­­­­­­References

-Charbel’s Mother

-W3schools

-stack overflow