Database Design for a Hospital Management System, Phase I&II

By:

Hussein Younes
Mikhail Gudkov
Valentin Sergeev
Daniel Atonge

Table of Contents

Use-Case motivation and assumptions 2

Use-Case Diagram of the Hospital Management System and requirements tables 3

ER diagram design decisions 20

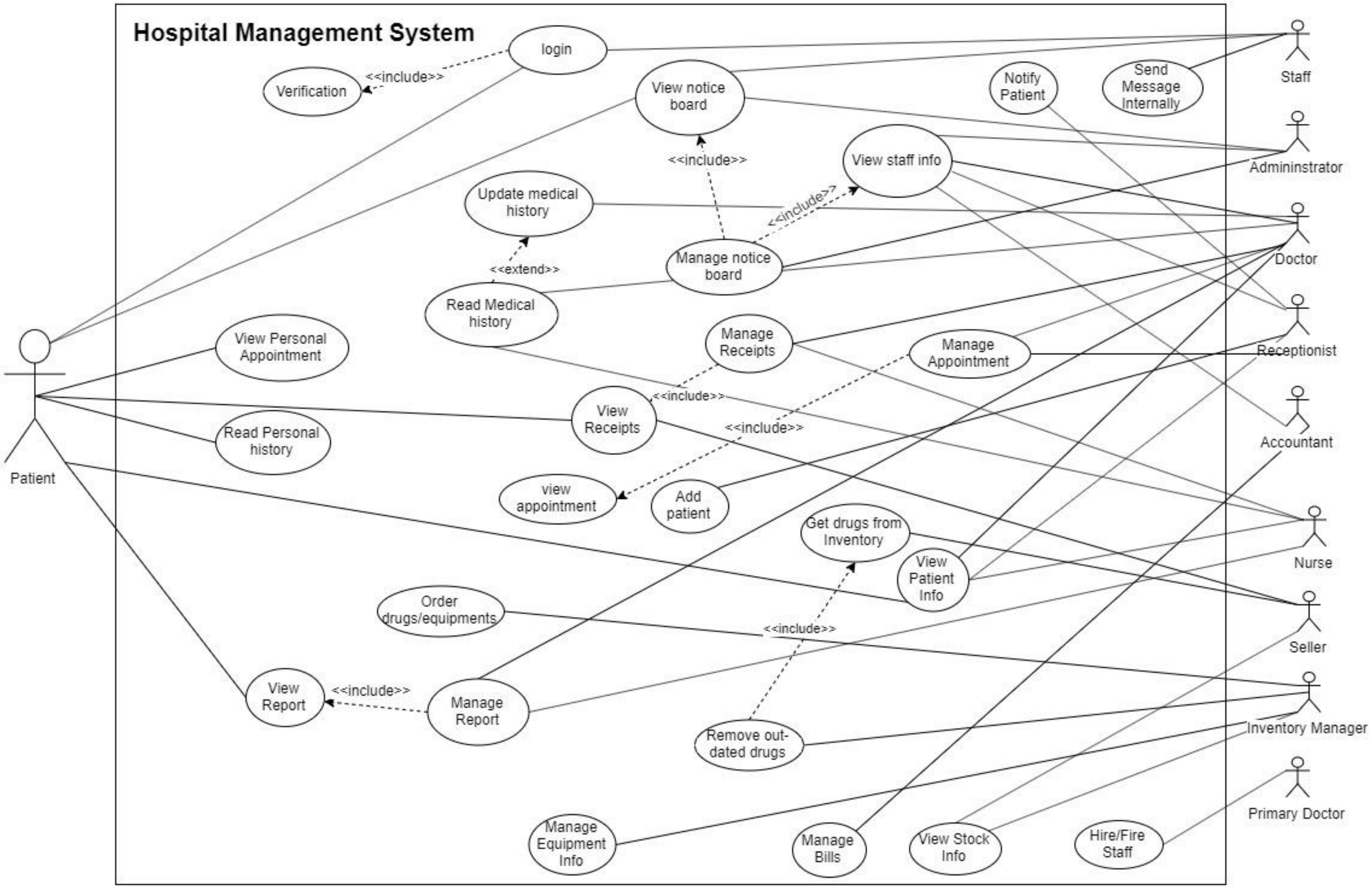
ER diagram of the hospital management system 21

Use-Case motivation and assumptions:

As per requirements, the use case diagram represented below was drawn to simulate our entire hospital system. Primary doctor, as in real life, is the main person in the hospital, he is hiring and firing staff, that's why we gave him extra use case "hire/fire staff". Most actors have UCs (here and after UC instead of Use Case) that begins with the word "view" (added in order to let the actors view things they need). For example, Inventory Manager uses "View Stock Info" to get information about the stock. Since we started to talk about him, Inventory Manager has to order drugs and equipment for the hospital. That's why we added this UC, it is done physically by Inventory Manager, but for database, this UC means adding drugs and equipment to the system. He also must look for outdated drugs and remove them if needed from inventory. As most of the actors, he has UC that begins with the word "Manage", using it we assume that the actor should be able to add, edit and remove things that he has to. For example, Inventory Manager can add/edit/remove Equipment. Seller is the other one who is able to get drugs from inventory, to sell it to the patient.

Reports are given and may be changed by either doctor or nurse. Doctors and nurses can check every report but the patient is able to check only reports connected to him. Same with appointments and medical history - patient can check only his own information. The only actor who is able to update medical history is the doctor, but nurse also can check it. We added accountant in order to give him access to bills and staff salaries, which he is able to change. Receptionist is the person who adds new patients to the database, he also notifies patients. Not all the doctors can check all staff information, only the primary doctor and the administrator can. However, receptionist should be able to view doctors' information for them to manage appointments. Doctor and Nurse can add, edit and remove receipts from the database. All the staff members are able to write internal messages to other staff members. Administrator is able to change the staff information if needed. Moreover, administrator is also able to make changes to the notice board. Logins and passwords are given by Primary Doctor (for staff) or Receptionist (for patients) during registration. It is worth nothing that the only primary actor in our Use-Case Diagram is patient, all the other actors are Secondary.

Use-Case Diagram of the Hospital Management System and requirements tables:



Requirement ID	Uniquely identifies the requirement	1	2
Title	Gives the requirement a symbolic name	Primary doctor should be able to hire staff members	Primary doctor should be able to fire staff members
Type	Functional or non-functional	Functional	Functional
Description	The definition of the requirement	Primary doctor should be able to create staff records in database, in particular, fill salary, ssn, name, age, office, role, and working days attributes. Login should be generated automatically based on a name. Id and password should be automatically generated	Primary doctor should be able to delete existing staff record from the database.
Priority	Defines the order in which requirements should be implemented. Priorities are designated (highest to lowest) from 1 to 3. Requirements of priority 1 are mandatory; 2 represents features nice to have, and 3 represents optional features.	1	1
Risk	Specifies the risk of not implementing the requirement. It shows how critical the requirement is to the system as a whole. The following risk levels are defined over the impact of not being implemented correctly. ● Critical (C) It will break the main functionality of the system. The system cannot be used if this requirement is not implemented. ● High (H) It will impact the main functionality of the system. Some function of the system could be inaccessible, but the system can be generally used. ● Medium (M) It will impact some system features, but not the main functionality. The system can still be used with some limitation. ● Low (L) The system can be used without limitation, but with some workarounds.	C: Primary doctor must have the ability to hire staff. Otherwise, no staff will exist in the database, and the main functionality of the system will be lost.	C: Primary doctor must have the ability to fire staff. Otherwise, unwanted staff will exist in the system for its whole lifetime, and they will affect the main functionality of it.

Requirement ID	3	4	5
Title	Receptionist should be able to add patients to system	Doctors and receptionists should be able to add appointments to the database	Doctors/Receptionists should be able to edit appointments in the database
Type	Functional	Functional	Functional
Description	Receptionist should be able to create patient records in the database, in particular, fill medical insurance number, name, birthday, address and gender attributes. Login should be generated automatically based on the name. Id and password should be automatically generated	Doctors and receptionists should be able to create appointment records in the database, especially, specify the date and the patient. If receptionist is creating an appointment, he should also be able to specify the doctor. If a doctor creates an appointment, he will be automatically assigned to it	Doctors/Receptionists should be able to edit appointments in the database, especially, its date attribute
Priority	1	1	2
Risk	C: Adding patients to the database affects the functionality of the whole system directly because patients are the main elements of the system, and not being able to add them will break the system's functionality.	H: because without adding appointments to the system, patients and doctor cannot have the needed information for them to meet. Thus, neither doctors will have patients assigned to them, nor patients will have doctors assigned for them. Therefore, the whole system will lose one of its main functionalities.	M: because for example, editing an appointment's date is crucial in cases where the doctor or the patient wants to change the date of the appointment. Therefore, this requirement affects a feature of the system but not the main functionality.

Requirement ID	6	7	8
Title	Doctors/Receptionists/Patients should be able to view appointments	There should be a possibility to delete appointments	Seller/Inventory manager should be able to check stock information
Type	Functional	Functional	Functional
Description	Doctors/Receptionists should be able to view all appointments. Patients should be able to view their only appointments which are attached to them	Receptionist should be able to delete appointments from the database. Old appointments (difference between appointment's date attribute and current date exceeded 1 week) should be automatically deleted	Seller/Inventory manager should have access to Equipment and Drug table
Priority	1	2	1
Risk	H: If patients or doctors do not have the access to view the appointment date and time, one the main functionalities of the system will be broken (appointments' functionality).	L: if appointments will not be deleted, they will not affect any functionality of the system. They will just remain in the database.	H: if (seller/inventory manager) are not able to check the available equipment, then they are not able to bring the equipment needed for the hospital or even provide the existing equipment when it's needed inside the hospital.

Requirement ID	9	10	11
Title	Accountant should be able to check budget information	Accountant should be able to edit budget information	Inventory manager should be able to order equipment
Type	Functional	Functional	Functional
Description	Accountant should have access to bills table, be able to view staff information (especially, salary and ssn)	Accountant should be able to edit staff salaries	Inventory manager should be able to create new equipment records in database, especially, fill name and quantity attributes. Id attribute should be generated automatically
Priority	2	1	1
Risk	M: Checking the budget information does not affect one of the main functionalities of the system. But it will affect a feature of the system which is the ability to check the budget information.	H: It will break one of the main functionalities of the system. Because budget information must be changed according to the current salaries of the staff. But the system can still work without this requirement.	H: if the inventory manager will not be able to order equipment. Then, he is not able to bring the equipment needed for the hospital (which is one of the main functionalities that keeps the hospital working with the proper equipment).

Requirement ID	12	13	14
Title	Inventory manager should be able to order drugs	Inventory manager should be able to remove broken\outdated equipment	Inventory manager should be able to add ordered drugs to stock
Type	Functional	Functional	Functional
Description	Inventory manager should be able create new drug records in database, especially, fill creation date, expire date, name and quantity attributes. Id attribute should be generated automatically	Inventory manager should be able to delete equipment records from the database	Inventory manager should be able to add equipment records to the database
Priority	1	2	1
Risk	H: if the inventory manager is not able to buy medicines, then the hospital cannot provide the needed treatments for its patients.	L: if the inventory manager is not able to remove the outdated equipment, it will just remain in the stock and will not affect any functionality of the system.	H: if the inventory manager is not able to add any ordered medicines to the stock, it means there will always be no medicines in the stock. And then the hospital will not be able to provide treatments to its patients.

Requirement ID	15	16	17
Title	Inventory manager/Seller should be able to get drugs from stock	Inventory manager should be able to remove outdated drugs from stock	Doctor/Nurse should be able to add patient's report
Type	Functional	Functional	Functional
Description	Inventory manager/Seller should be able to update drug's quantity attribute	Inventory manager should be able to delete drugs records from the database	Doctor/Nurse should be able to create report records in database, in particular, fill its content attributes. They should be able to specify corresponding medical history item and patient, for whom report is created
Priority	1	2	1
Risk	H: if the (seller/inventory manager) is not able to get any medicines from the stock, it means no one can access them and give them to the patients. Therefore, one of the main functionalities of the system will be lost.	L: if the inventory manager is not able to remove the outdated medicines, it will just remain in the stock and will not affect any functionality of the system.	H: Adding medical reports to the database allows doctors to analyze patients' situations to provide the corresponding treatments for them. Therefore, it is one of the system's main functionalities. However, system can still be used without it.

Requirement ID	18	19	20
Title	Doctor/Nurse should be able to edit patient's report	Doctor, Nurse should be able to delete patient's reports	Doctor, Nurse, and patient should be able to view reports
Type	Functional	Functional	Functional
Description	Doctor/Nurse should be able to edit contents of the any report, change the medical history item attached to it	Doctor, Nurse should be able to delete any report record from the database	Doctor and Nurse should have access to any report in the database. Patients should only have access to reports which were attached to them
Priority	1	2	1
Risk	H: Editing the information of the medical reports affects the functionality of the medical histories subsystem which performs one of the main functionalities of the entire system. However, system can still be used without it.	L: if not deleted, unnecessary reports will stay in the system. Which will not affect any functionality of the system.	H: Doctors/Nurse should have access to any medical report in the system for them to analyze patients' situations and provide the corresponding treatments. Therefore, without this requirement, one of the system's main functionalities will break. However, system can still be used.

Requirement ID	21	22	23
Title	Nurse, Doctor, Receptionist, Patient should be able to view patient’s information	Patient should be able to view medical history	Doctor, Nurse, Seller, Patient should be able to view receipts
Type	Functional	Functional	Functional
Description	Nurse, Doctor, Receptionist should have access to all patient's information (medical insurance number, name, birthday, address, and gender), which presents in the system. Patients should have access only to information, which corresponds to them	Patient should have access to all medical history items attached to him	Doctor, Nurse, Seller should have access to all Receipt records in the database. Patients should have access to only those receipts which are attached to them
Priority	1	3	2
Risk	C: If (Nurse/Doctor/Receptionist/Patient) are not allowed to access the patient's information: name, age, insurance id. Then, the main functionality of the system will break and it cannot be used anymore.	M: if the patient cannot access his medical history, then one of the system's features will break. However, the system can still be used with some limitations for the patient.	H: if (Doctor/Nurse/Seller/Patient) cannot view any receipt. Then, the receipts will not perform their roles correctly in the system. Thus, one of the main functionalities of the system will break. But the system can still be used.

Requirement ID	24	25	26
Title	Doctor/Nurse should be able to add receipts	Doctor/Nurse should be able to edit receipts	Doctor/Nurse should be able to remove receipts
Type	Functional	Functional	Functional
Description	Doctor/Nurse should be able to create new Receipt records in the system, especially, fill its validity period. He also should be able to specify the list of the drugs included into receipt and patient for whom this receipt is created for. There also should be a flag "used" which initially set to False. Id should be auto-generated	Doctor/Nurse should be able to edit all existing Receipt records, especially, modify drug list and validity period	Doctor/Nurse should be able to delete any Receipt record from the database
Priority	2	2	3
Risk	H: if (Doctor/Nurse) are not able to add any receipt. Then, no receipt actually will appear in the system. Thus, one of the main functionalities of the system will break. However, the system can still be used.	H: if (Doctor/Nurse) are not able to edit receipts. Then, the receipts will not perform their roles correctly in the system. Thus, one of the main functionalities of the system will break. But the system can still be used.	L: if not deleted, unnecessary receipts will stay in the system. Which will not affect any functionality of the system.

Requirement ID	27	28	29
Title	Nurse, Doctor should be able to check medical history	Doctor should be able to update patients' medical history	Receptionist should be able to notify patients
Type	Functional	Functional	Functional
Description	Nurse and doctor should be able to access all existing medical history items, for every patient	Nurse and Doctor should be able to create separate medical history items, especially, fill diagnosis, start and end date (may be specified later) of illness, symptoms and, possibly, some extra information (in text form). They should specify patient for whom this particular medical history item is attached	Receptionist should be able to send notifications to the patient, especially, specifying a notification reason
Priority	1	2	3
Risk	H: (Doctors, Nurse) should have access to any medical history item in the system for them to analyze patients' situations and provide the corresponding treatments. Therefore, without this requirement, one of the system's main functionalities will break. However, system can still be used.	H: Updating the information of the medical history items affects the functionality of the medical history subsystem which performs one of the main functionalities of the entire system. However, system can still be used without it.	M: Notifying patients is a feature in the system. Therefore, if it breaks, the systems can still be used without the ability to send notifications.

Requirement ID	30	31	32
Title	Primary Doctor, Admin, Accountant, Receptionist should be able to check staff information	Everyone should be able to check notice board	Admin should be able to manage notice board
Type	Functional	Functional	Functional
Description	Primary doctor, Admin, Accountant should have access to ssn, name, age, office, role and working days of any staff in the database. Receptionist should only have access to name, office and working days	Both Staff and Patients should have access to view notice board (doctor's working hours)	Admin should have access to changed data used in notice board (doctors working hours)
Priority	2	1	1
Risk	C: If (Primary Doctor, Admin, Accountant, Receptionist) are not able to check staff information. Then, staff cannot actually be hired and no staff will exist in the system and the system cannot be used properly for a long time.	M: If the notice board subsystem breaks. Only a feature of the entire system will break. But the system can still be used with.	M: If Admin cannot manage the notice board. Then, the notice board subsystem will break. Therefore, only a feature of the system will be damaged. However, the system can still be used.

Requirement ID	33	34	35
Title	Every Staff member should be able to send messages internally to each other	All Users in the system should be able to login and pass verification procedures to be able to access the system	Interface of the system should be intuitively clear. And personalized for different categories of users
Type	Functional	Functional	Non-functional
Description	Every staff member should be able to send message to any other staff member, but not to patients (internally). They should be able to specify content of a message	Users should be prohibited to do any actions until they pass the verification procedures. In particular, until they haven't entered correct pair login password	The interface of a system should be understandable by at least 90% of the users (should be tested on control group out of 100 people of different ages). Weak vision support should exist
Priority	1	1	2
Risk	C: If the staff member cannot communicate properly, the subsystems cannot be well integrated. And the functionality of the whole system will break.	C: if some users cannot access the system properly, then it cannot be used.	L: If the interface is not clear, the whole system can still work properly but with some difficulties for the users.

Requirement ID	36	37	38
Title	System should have a feature of clearing memory by deleting old information	System should have a backups configured	The system should be secure against any unauthorized users
Type	Non-functional	Non-functional	Non-functional
Description	The percentage of useless information (old receipts, old appointments, not relevant person/staff accounts) should exceed 10%. The measure of "old" was clarified in previous requirements for each entity	System should have auto-backups (stored on external storage, at least one time per week)	The system should pass all checks on pentest-tools.com
Priority	1	1	1
Risk	C: If the memory is not managed properly, the system will break after a period of time.	C: if the system does not have back-ups. After any unexpected error, the system will not be able to go back to its previous state. Therefore, the whole system will break.	C: If an unauthorized user can change critical information in the system. Then, the whole system might break at any time.

Requirement ID	39	40	41
Title	The system should be made in such a way, that it will be easy to add new features)	The system should be independent from external systems.	The system should be optimized to high-load usage
Type	Non-functional	Non-functional	Non-functional
Description	Time for adding new features of the same complexity should not change during the time	System should not have any dependencies on external services. It should remain the whole functionality while no access to external IP's is allowed	The system should be able to handle load ~100 requests/second
Priority	2	3	2
Risk	M: If the system is not extensible, then it can work properly. However, no new features can be added to the system.	C: if the system cannot work without depending on other systems. Then, its functionality will depend on other systems' functionality. Therefore, it might break at any time.	M: The system can be used without the performance requirement but with a limitation on the number of requests per second.

Requirement ID	42	43	44
Title	System should consume less energy as possible.	The system should be reliable. It should work without crashes and falls	The effort of changing some existing software should be minimum
Type	Non-functional	Non-functional	Non-functional
Description	This electricity usage caused by system work exceed limits of corresponding budget field	The percentage of crashes out of all usage (1 usage - from opening time end) should not exceed 0.1%	The system should be organized in such way, that any changes made won't break existing functionality
Priority	3	1	3
Risk	H: without the energy requirement. The whole system can be used without any limitation.	C: If the system is not reliable, then it cannot be used.	L: If the system is not modifiable, it can still be used without any limitation. However, changes on the system's features will cost a lot of effort.

Requirement ID	45	46	47
Title	The system should be easily moved to a different platform when required	The user interface of the system should be responsive	The system should be legal in all possible ways
Type	Non-functional	Non-functional	Non-functional
Description	The system should be available on such platforms as Windows, Linux, macOS, Android, IOS. The versions of each system should be chosen in such a way, that the percentage of covered users is not less than 99%	The system should have responsive, well adapted for different screen sizes UI	There may be some legal issues involving the privacy of information. i.e.: export of some restricted technology.
Priority	3	2	1
Risk	L: if the system is not portable, it can still be used properly. However, it will only work on the current platform.	L. If the user interface is not responsive, the system can still be used. However, the user experience will be unsatisfactory.	C: If the system does not meet the legal requirements. It can be stopped at any time.

ER Diagram design decisions:

First of all, we want to mention that we are using Chen’s notation. Actors that we had in UC diagram now become roles, not all of them will appear in our ER diagram. Furthermore, attributes of Staff are the same attributes of Doctor, Receptionist, and Seller. In our diagram, we have only two weak entities: APPOINTMENT and NOTIFICATION.

An appointment is identified uniquely by a doctor, connected to it (it will have a composite key made up of a partial key “datetime” and the foreign key “doctor_id”). A notification cannot exist on its own, because it should be sent to a patient (it will have a composite key made up of a partial key “time” and the foreign key “medical_insurance_id”). All the other entities are uniquely identified by their primary keys so they are strong entities. We would like to emphasize that report has only some information about patient, while medical history item contains reports, diseases and other information related to the medical history.

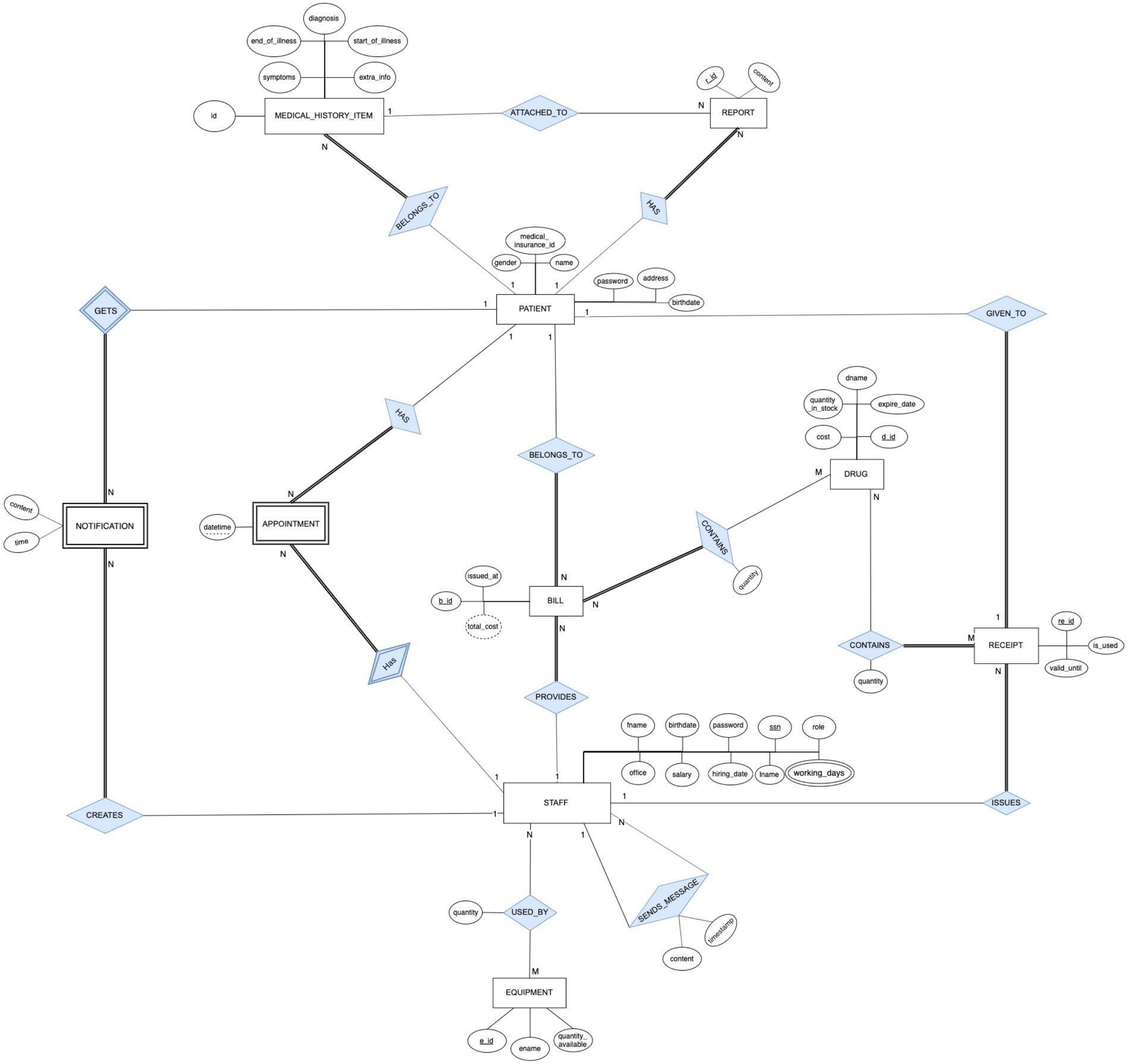
Relationship SENDS_MESSAGE has information about messages that staff send to each other (recursive).

We also have relationship USED_BY to store information of usage of equipment. We store information about notifications: time, content, receptionist, who sent it, and patient, who got it. For it we use relationships CREATES and GETS.

We also have bills, connected to drugs with relationship CONTAINS. Bills are given by sellers to patients and this is shown with relationship PROVIDES. Receipt also contains drugs given to patient (relationship GIVEN_TO) by doctor (relationship ISSUES). Doctor can also make appointments (relationship HAS).

As represented in the ER diagram, the “working_days” attribute for STAFF is multi-valued. This is because a staff may work several days per week. All other attributes apart from “working_days” are single-value attributes. To be consistent with Chen’s notation, we used single lines to denote partial participation and double lines for total participation. This means whenever a single line is used, not necessarily all entities of the associated entity set should take part in the associated relationship, and when a double line is used, all entities must take part in the associated relationship. For example, the relationship BELONGS_TO between BILL and PATIENT. All bills should necessarily belong to patient (total participation of BILL in the relationship BELONGS_TO). However, some patients might have no bills to pay (partial participation of PATIENT in the relationship BELONGS_TO).

ER Diagram of the Hospital Management System:


diagnosis
end_of_illness
start_of_illness
symptoms
extra_info
id
MEDICAL_HISTORY_ITEM
1
ATTACHED_TO
N
REPORT
r_id
content
N
N
BELONGS_TO
HAS
medical_
insurance_id
gender
name
password
address
birthdate
1
1
GETS
1
PATIENT
1
GIVEN_TO
1
1
dname
quantity
_in_stock
expire_date
cost
d_id
HAS
BELONGS_TO
M
DRUG
N
N
N
content
time
NOTIFICATION
datetime
APPOINTMENT
CONTAINS
quantity
N
N
issued_at
N
b_id
BILL
N
total_cost
N
1
re_id
CONTAINS
M
RECEIPT
is_used
valid_until
N
Has
quantity
PROVIDES
fname
birthdate
password
ssn
role
office
salary
hiring_date
lname
working_days
1
1
CREATES
1
STAFF
1
ISSUES
N
1
N
SENDS_MESSAGE
quantity
USED_BY
timestamp
content
M
EQUIPMENT
e_id
ename
quantity_
available