Data Modeling and Databases I.
Project, Phase 2.

Lev Svalov, Shamil Khastiev, Alisa Martyanova, Bekzhan Talgat, Zhandos Kipshakbaev.
Group - BS18-05.

Table of contents

The domain of the phase 1

Changes after the feedback:

- Explanations of the risk for each requirement were added.
- Some misunderstanding for non-functional severity was removed. In particular, in the non-functional requirement **Privacy**, risk was changed from “M”(medium) to “H”(high) and was explained why this decision was made.

---

1. Study the domain.

- How your system will function:

  Patients who are assigned for this hospital have their personal account, where they can access their medical history, the results of analyzes and communicate with doctors. In case of illness, the patient can register for an appointment with the doctor on the site. He can choose the appropriate time and preferable specialist. Doctor in his account can check all the patients with whom he has an appointment for the day, also he can view the list of patients who is currently treated by him and their medical history. Nurses are able to see their schedule for the day, receive notifications about emergency situations. They should keep track of stationary patients, check their canteen menu, take them for analyzes. The system is also can be used for internal communications. The doctor can send results of medical examination to patients and certificate of recovery after the end of the treatment. Other staff of the hospital should be able to manage the rest of the hospital problems. Financial management, keeping track of the condition of the medical equipment, registration of guests, tracking of donations is the responsibility of other staff of the hospital.

- Which entities will it have and how they will be related to each other:

  In our system, we have four main entities: patient, doctor, nurse, hospital staff. Patients are divided into two types: ambulatory patient and stationary patient; as well as in staff we have three main characters: economic department, supply manager and receptionist. Treatment of patients is controlled by the assigned doctor. Every nurse is assigned to a group of stationary patients. Economic department manages salaries, spendings on patients and other financial questions. Supply manager tracks the condition of medical equipment and the whole hospital. The receptionist is responsible for managing reports for the patient, and the organization of charity events and donation receiving.

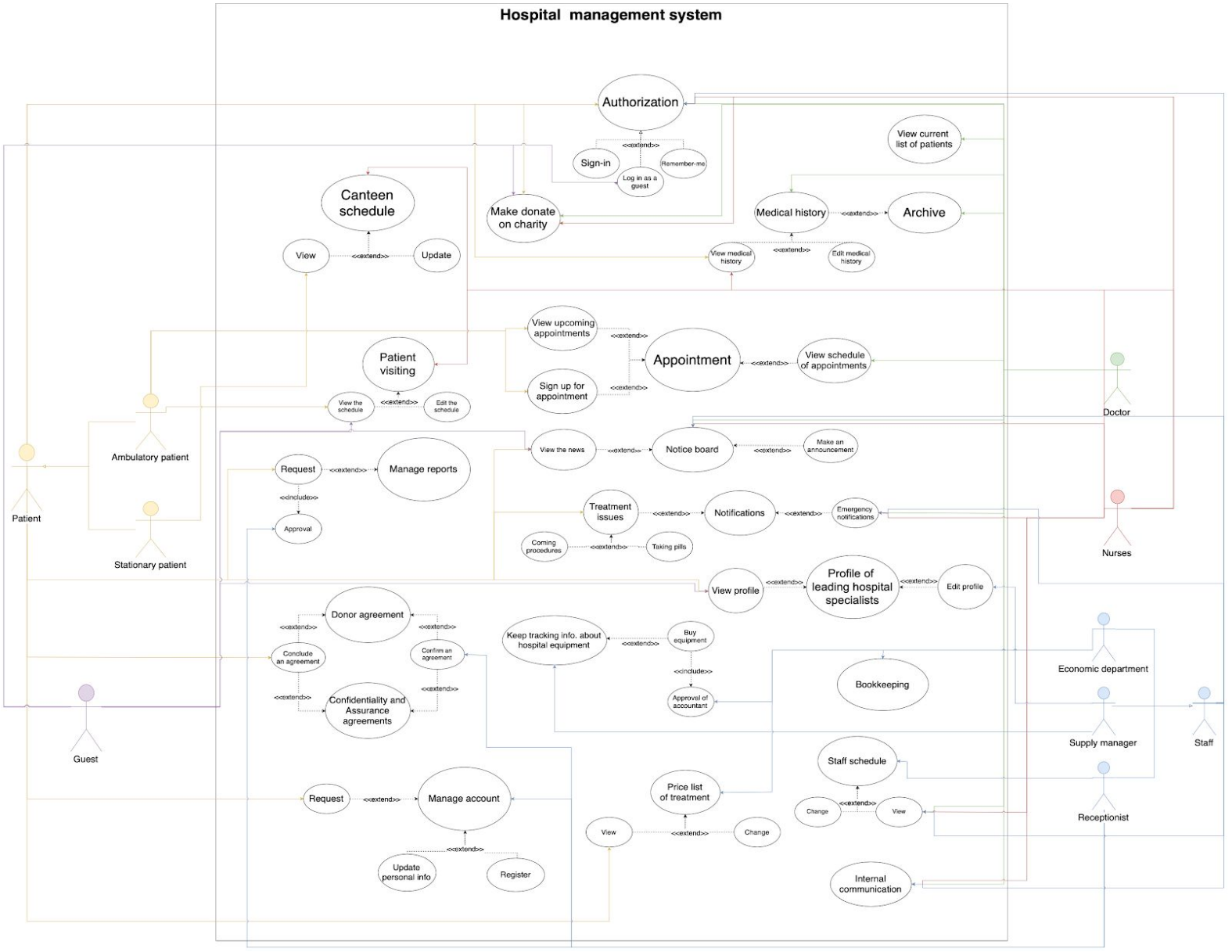
- What will be the level of detailization:

  Our system is on the optimal level of detailization. Most of the functions of the hospital can be done by using our system avoiding unnecessary paperwork. Besides the main functionality that is covered in the system, there are some narrow use-cases such as the Management of canteen menu and the Profile of leading specialist in the hospital.

2. Describe the domain.

1. Use-Case diagram.

   p.s. : for better readability **link** to the our saved diagram in the cloud.



2. Functional and non-functional requirements.

There is the list with all requirements and templates below.

p.s. : In the list below we will provide the name of the requirement in the **Use-Case** diagram in the brackets.

Functional requirements:

1. Multi user account system. (Authorization)
2. Medical History.
3. Monitoring the whole hospital system. (Keep tracking info. about hospital equipment)
4. Management of all types of users' account.
5. Notice Board.
6. View Appointments.
7. Appointment Management.
8. Notifications.
9. Invoice Management.
10. Medical Report Management. (Manage Report)
11. Internal Communication.
12. Management of the list with current pricing for particular treatment. (Price list of treatment)
13. Profile of leading hospital specialists.
14. Management of an assurance and confidentiality agreements between the patient and the hospital. (Confidentiality and Assurance agreements)
15. Current list of patients.
16. Donor agreement.
17. Patient visiting.
18. Canteen menu.
19. Staff schedule.
20. Charity.

Non-functional requirements:

21. Security.
22. User friendly interface.
23. Scalability and Reliability.
24. Portability.
25. Privacy.
26. Accessibility.
27. Development environment.
28. Response time.
29. Backup.

30. Testability.
31. Integrability.
32. Internalization.
33. Robustness.
34. Payment service.
35. Partnership

Functional requirements

Requirement ID	1
Title	**Multi user account system**
Type	**Functional**
Description	Any user should be able to create/manage his/her personal account.
Priority	1
Risk	C **Explanation:** Information about patients/doctors/nurses are stored in different accounts. Without this part we would not be able to create the system.

Requirement ID	2
Title	**Medical history**
Type	**Functional**
Description	Doctors can see the list of all patients and edit their medical history.
Priority	2
Risk	H **Explanation:** Without this part doctors/nurses would not be able to correctly treat patients.

Requirement ID	3
Title	**Monitoring the whole hospital system**
Type	**Functional**
Description	Staff of hospital should be able to check the condition of the all equipment of the hospital (technical, medical etc.).
Priority	2
Risk	H **Explanation:** Lack of equipment or medications would lead to the problems in the work of the hospital.

Requirement ID	4
Title	**Management of all types of users' account**
Type	**Functional**
Description	The system supports several types of accounts with different level of access and functionality.
Priority	1
Risk	C **Explanation:** Accounts of Patients/Doctors/Nurses have different access level and functionality and without this part we could not implement the system.

Requirement ID	5
Title	**Notice Board**
Type	**Functional**
Description	Users can see general information about the hospital, upcoming events.
Priority	3
Risk	L **Explanation:** This will make process of announcements easier. But it is not crucial

	for the system.

Requirement ID	6
Title	**View Appointments**
Type	**Functional**
Description	Doctors should have an opportunity to see their working schedule and list of patients for consultation. Patients should have an opportunity to view their upcoming appointments.
Priority	2
Risk	M **Explanation:** This is one of the features of the doctors' accounts. But it does not affect work of other users accounts.

Requirement ID	7
Title	**Appointment Management**
Type	**Functional**
Description	All patients should have an opportunity to choose convenient time-slot for the medical examination with particular doctor and sign up for it.
Priority	2
Risk	M **Explanation:** This is a way of making an appointment with doctors, but this would not affect the work of other features of the system.

Requirement ID	8
Title	**Notifications**
Type	**Functional**
Description	Authorized users should be notificated about issues, according to their account type. For instance, for doctors - information about operations and unplanned meetings and for patients - notification about procedures and taking pills/drugs/medicaments.
Priority	3
Risk	L **Explanation:** This will help users to stay informed about latest news. It is not an important feature.

Requirement ID	9
Title	**Invoice Management**
Type	**Functional**
Description	Financial department of the hospital have to be able to track the balance of the financial resources.
Priority	2
Risk	M **Explanation:** Keep track of finances is an important part of the hospital work process, but without this feature our system will still function.

Requirement ID	10
Title	**Medical Report Management**
Type	**Functional**
Description	Patients can receive(from the hospital) the medical report about their illness using the system. It can be either formal document with the period of treatment or it can be detail review of medical history.
Priority	3
Risk	L **Explanation:** Patients can receive information from doctors much easier, but there is no crucial effects of not implementing this feature.

Requirement ID	11
Title	**Internal Communication**
Type	**Functional**
Description	Medical staff can communicate and have some conversations about diagnosis of patients and problems of the hospital.
Priority	3
Risk	L **Explanation:** Communication between hospital staff could be done without this feature, but with our system this process is simplified.

Requirement ID	12
Title	**Management of the list with current pricing for particular treatment.**
Type	**Functional**
Description	Patients can view the list with current pricing of the procedures and treatments. The financial department can make changes to the list basing on the hospital balance.
Priority	2
Risk	L **Explanation:** Useful feature for patients, but not crucial for the system.

Requirement ID	13
Title	**Profile of leading hospital specialists**
Type	**Functional**
Description	Users can view the list of the top specialist and their medical experience. Hospital staff can edit this list.
Priority	3
Risk	L **Explanation:** This is an additional information which in no way affects work of the hospital and our system.

Requirement ID	14
Title	**Management of an assurance and confidentiality agreements between the patient and the hospital**.
Type	**Functional**
Description	Patient can conclude the confidentiality and assurance agreements using the system.
Priority	2
Risk	M **Explanation:** Some features of the system are not allowed without this part, but the rest are still available.

Requirement ID	15
Title	**Current list of patients**
Type	**Functional**
Description	Doctors have their own list of patients that are getting treatment currently.
Priority	2
Risk	L **Explanation:** This information helps doctors in their work, but does not affect the work of the system.

Requirement ID	16
Title	**Donor agreement**
Type	**Functional**
Description	Patient will have an option to conclude an agreement with the hospital. The agreement confirms that if the patient passes away, his/her body will transfer either to the Organ Procurement and Transplantation Network or to the Research laboratory of the hospital.
Priority	3
Risk	L **Explanation:** This is not the main purpose of the system, and does not affect the whole system.

Requirement ID	17
Title	**Patient visiting**
Type	**Functional**
Description	Friends and relatives of a patient may visit the hospital only during the particular period of time. They can view a schedule of visiting hours in the system. Nurses can edit the schedule for internal reasons of the hospital.
Priority	3
Risk	L **Explanation:** Patients can register guests for visit much easier. Not the reasonable function of the system.

Requirement ID	18
Title	**Canteen menu**
Type	**Functional**
Description	For every group of patients there are different meal plan stored in the system, nurses should keep track on this schedule to provide appropriate meal for the patient.
Priority	3
Risk	L **Explanation:** Makes work for nurses easier. Not the reasonable function of the system.

Requirement ID	19
Title	**Staff schedule**
Type	**Functional**
Description	Hospital staff can see their tasks for the day and schedule of their duties.
Priority	2
Risk	M **Explanation:** Schedule organizes the whole work in the hospital and makes control over the staff easier. Additional feature.

Requirement ID	20
Title	**Charity**
Type	**Functional**
Description	Any person should be able to donate money/organs/medicaments in need of hospital.
Priority	3
Risk	L **Explanation:** Does not affect the system. Is an additional feature.

Non-Functional requirements

Requirement ID	21
Title	**Security**
Type	**Non-functional**
Description	System should have two step authentication, restriction on complexity of password, confirmation via phone number / ID medical number.
Priority	1
Risk	H **Explanation:** Personal information of the patients should be protected. Without this part the system should not be used.

Requirement ID	22
Title	**User friendly interface**
Type	**Non-functional**
Description	User should be able to understand all the functional of the system. Interface should be readable and understandable, must be in light colors and pleasant to use.
Priority	2
Risk	H **Explanation:** System is designed for arbitrary user which should be able to understand how to use it. Otherwise there will be no meaning for such a system.

Requirement ID	23
Title	**Reliability and Scalability**
Type	**Non-functional**
Description	System must handle increase in number of users and prevent any possible exceptions and errors.
Priority	1
Risk	H **Explanation:** Since system controls the work of the hospital this system should be able to handle errors.

Requirement ID	24
Title	**Portability**
Type	**Non-functional**
Description	System should be supported on any device and operating system. Also, it can support several types of files/extensions/formats.
Priority	2
Risk	M **Explanation:** Users with any device should be able to access the system, but this is not crucial for the work of the system.

Requirement ID	25
Title	**Privacy**
Type	**Non-functional**
Description	All the information stored in the system should not be accessible for the people outside of the system.
Priority	1
Risk	H **Explanation:** Personal information of the patients should be protected. Without this part the system should not be used.

Requirement ID	26
Title	**Accessibility**
Type	**Non-functional**
Description	System design should have several modes for comfortable use by people with disabilities.
Priority	2
Risk	L **Explanation:** This is an additional feature. Would not affect the main functionality of the system.

Requirement ID	27
Title	**Development environment**
Type	**Non-functional**
Description	System should be implemented in a way such that it can be updated and set up for adding new information or functionality.
Priority	1
Risk	H **Explanation:** New features should not affect other features, otherwise it will hurt the work of the system.

Requirement ID	28
Title	**Response time**
Type	**Non-functional**
Description	System response time should be less than 5 seconds.
Priority	2
Risk	L **Explanation:** Makes use of the system easier, but does not affect other features of the system.

Requirement ID	29
Title	**Backup**
Type	**Non-functional**
Description	In case of crush of the system, all the data should be easily restored.
Priority	2
Risk	L **Explanation:** Used rarely, but helps to recover from crushes.

Requirement ID	30
Title	**Testability**
Type	**Non-functional**
Description	Developer can test every function of the system, to be sure that system works properly.
Priority	3
Risk	L **Explanation:** Helps during development of the system and adding new features, but does not affect work of the system.

Requirement ID	31
Title	**Integrability**
Type	**Non-functional**
Description	System can be used in collaboration with others systems/applications in the field.
Priority	3
Risk	L **Explanation:** System is independent and this feature is just for comfort of the user.

Requirement ID	32
Title	**Internationalization**
Type	**Non-functional**
Description	System can be adapted to various languages and regions without engineering changes.
Priority	3
Risk	L **Explanation:** Makes system easier for use. Is not a main functionality.

Requirement ID	33
Title	**Robustness**
Type	**Non-functional**
Description	System should be able to handle the invalid input and errors during the execution.
Priority	1
Risk	H **Explanation:** Mistakes and errors during the use of the system could lead to crush of the system, this feature is important for stable work.

Requirement ID	34
Title	**Payment service**
Type	**Non-functional**
Description	System should support inside payment for additional medical services.
Priority	2
Risk	M **Explanation:** Some features are not available without payment, but most of them are free.

Requirement ID	35
Title	**Partnership**
Type	**Non-functional**
Description	System should provide the list of operations/procedures that can be done in partner hospitals.
Priority	3
Risk	L **Explanation:** Patients could easily get information useful for them. Not the main feature of the system.

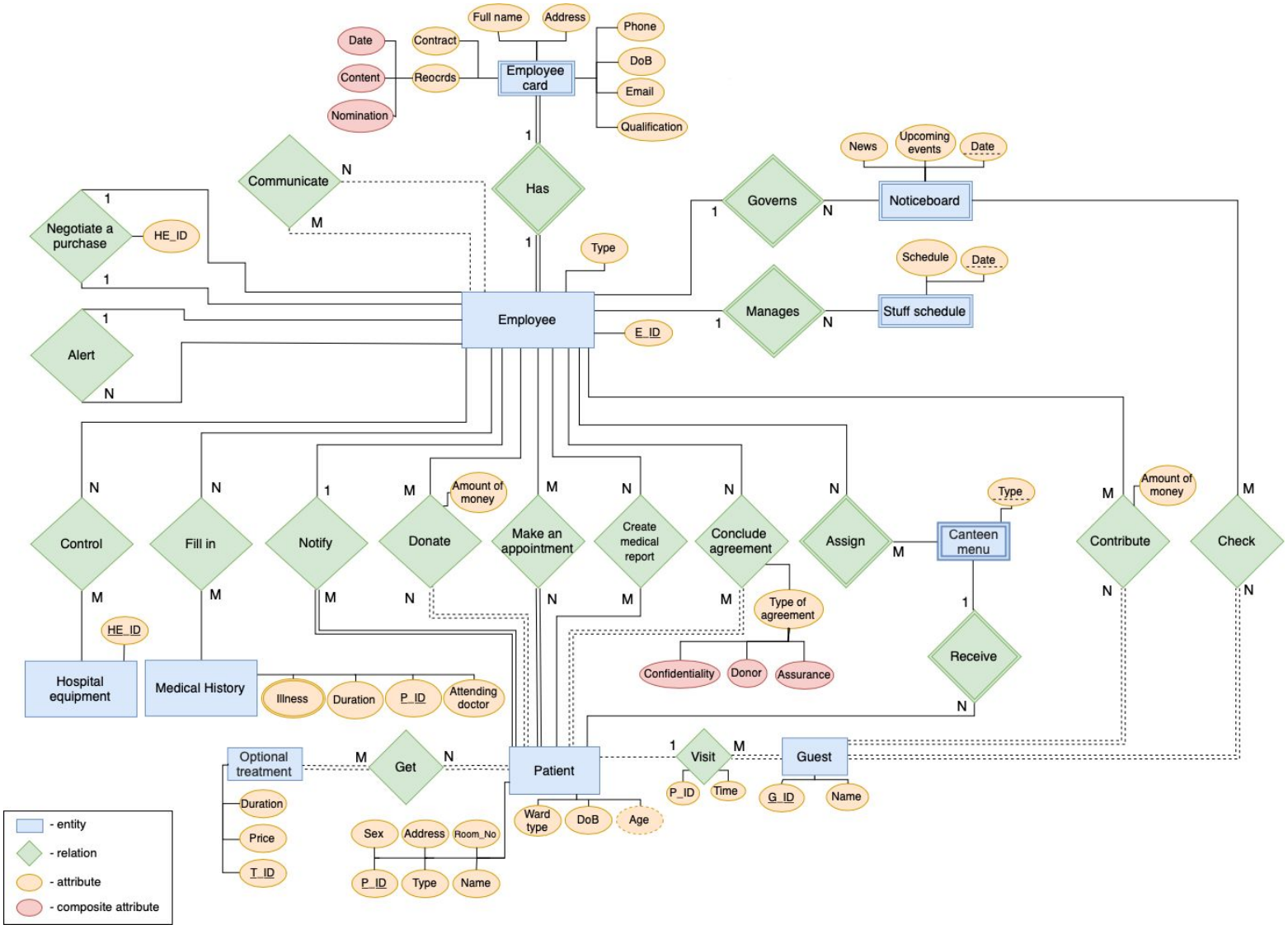
The domain of the phase 2

1. The ER-Diagram.

PS: for better readability and understanding:

[link to draw_io]

[link to cloud]



2. The explanation of our design decisions.

Now we are going to describe our diagram in more details. Here are descriptions of every entity, their attributes and relation they participate in.

Employee card:

- It is a weak entity which is created for detail description of the employee. It is weak because it can not be declared without employee.

Noticeboard:

- It is a weak entity as it can not be defined uniquely without the employee who governs it.
- It has "**Date**" attribute as a partial key. Also, it has "**Upcoming events**" and "News" attributes.
- Partial relation "**Governs**" with "**Employee**" means that one receptionist governs many news, upcoming events.

Stuff schedule:

- It is a weak entity which keeps a schedule of all employees for each day. It is weak because it can not be declared without employee id(e_id). Partial key is a "**Date**" attribute.
- Partial relation "**Manages**" with "**Stuff schedule**" means that one receptionist governs many schedules.

Hospital equipment:

- It has a Hospital equipment ID as a key attribute
- Partial relation "**Control**" with "**Employee**" means that many supply managers keep track of many hospital equipments. For example, one supply manager keeps track of the number of equipments and another supply manager fills documents to make a purchase of equipments.

Employee:

- It has attributes: Type and Employee ID, which is a key attribute (E_ID).
- It has total participation with one-to-one relation "**Has**" with weak entity. "**Employment Card**", it means that there can not exist an employment card without employee himself/herself.
- Partial relation "**Manages**" with "**Stuff schedule**" means that one receptionist governs many schedules. Avoiding misunderstanding, between several receptionists, while scheduling.
- Partial relation "**Governs**" with "**Noticeboard**" means that one receptionist governs many news and upcoming events. Avoiding misunderstanding between several receptionists, while they are editing noticeboard.
- Recursive relation "**Communicate**" means that in our system many employees can communicate with many employees.
- Recursive relation "**Negotiate a purchase**" is between one supply manager and one economic manager. Supply manager gives document with hospital

equipments to buy, assigned with HE_ID, then economic manager allocates money.

- Recursive relation "**Alert**" means that one receptionist alerts many doctors about coming emergencies situations(unplanned meetings, operations). To not bother doctor in vain.
- Partial relation "**Control**" with "**Hospital equipment**", *(as we mentioned before)* means that many supply managers keep track of many hospital equipment. For example, one supply manager keeps track of the amount of equipments and another supply manager fills documents to make a purchase of equipments.
- Partial relation "**Assign**" with "**Canteen menu**" means that many nurses should keep track on many menus to provide appropriate meal for the patients. It is better menu to be provided by several nurses to make it properly.
- Relation "**Notify**" with "**Patient**", is one to many, because it is better for patient to be notified by one receptionist and one receptionist can notify many patients.
- Relation "**Donate**" with "**Patient**", means several receptionists can obtain and certificates donations from the many patients for the hospital. This relation has attribute "**Amount of money**" indicates amount of donation one makes.
- Relation "**Create medical report**" with "**Patient**" means that every time, different receptionists may give medical report for a different patients.
- Relation "**Conclude agreement**" with "**Employee**" means different receptionist may conclude agreement with several patients. Relation has attribute type of the agreement which may be of Donor/Confidentiality/Assurance type.
- Relation **"Fill in"** with entity **"Medical history"** is partial with many-to-many cardinality. Only doctors out of all employees can view and edit medical history of patients. Many doctors can view and edit medical history of many patients.
- Relation "**Make an appointment**" with entity "**Patient**" means several doctors may be assigned for an appointment with several patients.
- Relation "**Contribute**" with entity "**Guest**" means every receptionist can certificate money from several guests(Contribute relation saves the amount of money).

Guest:

- It has Guest ID as a key attribute.
- Relation "**Visit**" with entity "**Patient**" means several guests may visit several stationary patients, for that they have to register time slot for every patient.
- Relation "**Contribute**" with entity "**Employee**" means every guest can donate money(Contribute relation saves the amount of money).
- Relation "**Check**" with entity "**Noticeboard**" means every guest can view upcoming events and news from noticeboard.

Patient:

- It has a “**Patient ID**” as a key attribute. It also has “**Age**” as a derived attribute, it’s derived from another its attribute “**DoB**” (day of birthday).
- Relation "**Get**" with entity "**Optional treatment**" means several patients may receive several treatments.
- Relation "**Visit**" with entity "**Guest**" means only stationary patients may be visited by guests. Patients may be visited by several guests.
- Relation "**Notify**" with entity "**Employee**" suggests that several patients may be notified by one employee.
- Relation "**Donate**" with entity "**Employee**" means several receptionists can certificate donations that are given from several patients. This relation has attribute "**Amount of money**" which indicates amount of donation one makes.
- Relation "**Make an appointment**" with entity "**Employee**" means several patients may assign for an appointment with several doctors.
- Relation "**Create medical report**" with entity "**Employee**" means that any time when patients needs medical report from hospital they can request it from different receptionists.
- Relation "**Conclude agreement**" with entity "**Employee**" means every patients may conclude agreement with different receptionist. Relation has attribute type of agreement which may be of Donor/Confidentiality/Assurance type.
- Relation "**Receive**" with entity "**Canteen menu**" means several patients may receive specific canteen menu assigned for them.

Canteen menu:

- Weak entity with partial key attribute **"Type"** (many patients can have the same menu).
- Partial relation **"Assign"** with entity **"Employee"** means that many nurses should keep track on many menus to provide an appropriate meal for the patients.
- Relation **"Receive"** with entity **“Patient”** means several patients may receive a specific canteen menu assigned for them.

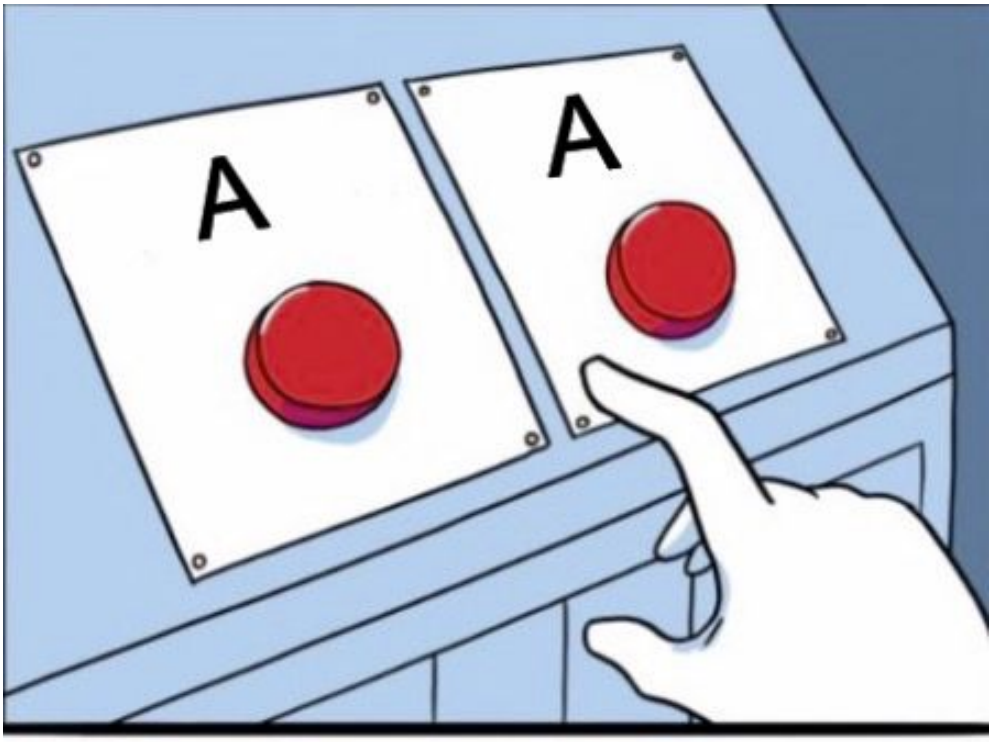
Medical History:

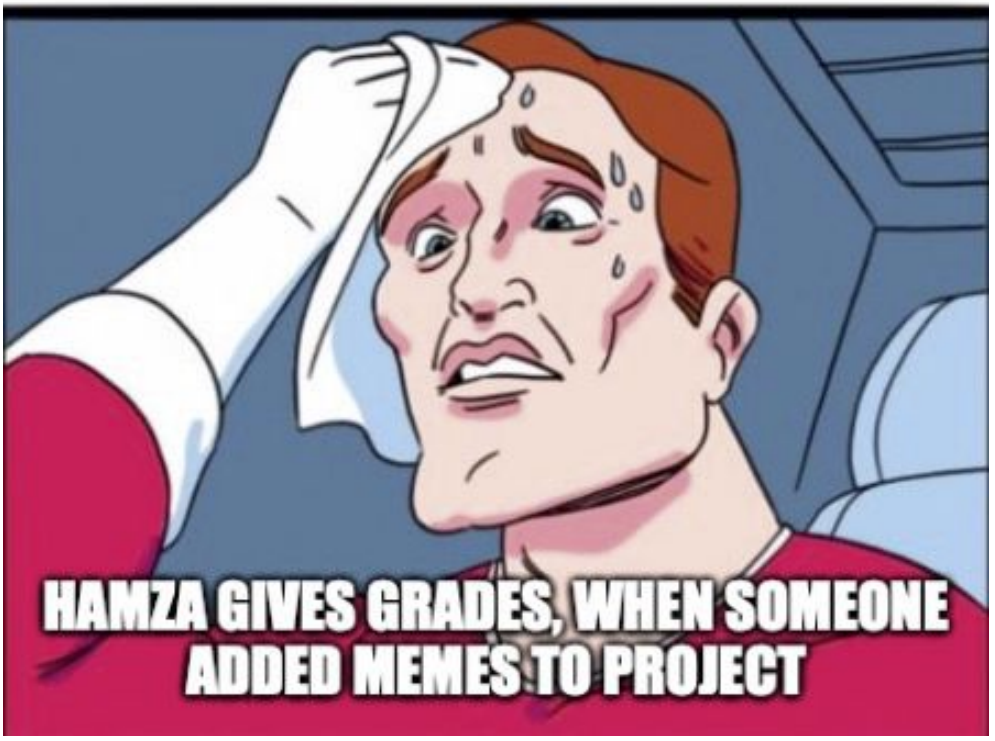
- Entity, containing information about illnesses, their duration and attending doctors of each patent.
- Key attribute is Patient ID
- Relation **"Fill in"** with entity **"Employee"** is partial with many-to-many cardinality. Only doctors out of all employees can view and edit medical history of patients. Many doctors can view and edit medical history of many patients.

- Attribute **"Illness"** is multivalued, because one patient could have several illnesses in his/her life.

Optional treatment:
- Entity, containing information about price and duration of optional treatments.
- Key attribute is Treatment ID.
- Relation **"Get"** with entity **"Patient"** is total, optional, with many-to-many cardinality. All patients are able to get optional treatment, if they want. One patient can have many treatments and one treatment can be done to many patients.

Thanks for your attention!


A
A


HAMZA GIVES GRADES, WHEN SOMEONE
ADDED MEMES TO PROJECT