Project Phase I: Domain Description
Data Modelling and Databases I, Fall 2019

Group[BS18-02] members:
Amina Miftahova
Dmitriy Podpryatov
Anna Startseva
Temurberk Khujaev
Ozioma Okonicha

25.09.2019

FUNCTIONAL REQUIREMENTS

Requirement ID	001
Title	Mandatory authorization
Type	Functional
Description	The access to the system has to be given only to registered users and only after their authorization to their particular account.
Priority	1
Risk	C

Requirement ID	002
Title	Should have a records management
Type	Functional
Description	This requires the system to allow a user (doctor or nurse) to add more information to an existing patient medical history, to create new records for first-time patients, to make changes to previous records and for a user (patient) to obtain their medical history or reference.
Priority	1
Risk	C

Requirement ID	003
Title	User hierarchy
Type	Functional
Description	User’s access to the data should depend on their status (doctor, patient, etc.), the actions that a user can perform should be defined by its type of the account.
Priority	1
Risk	C

Requirement ID	004
Title	Should have an accounts management by the IT-staff
Type	Functional
Description	The IT-staff should have the ability to change the kind of account, add a new user and delete a user from the system. They should be able to provide technical support for the users (change the password, personal info, etc.)
Priority	1
Risk	C

Requirement ID	005
Title	Should have an internal communication
Type	Functional
Description	This requires the system to have some panel to have discussions internally and for one user (staff) to communicate with another user (staff). Through this chat, a doctor may contact a nurse and give information or ask questions, medical staff can inform the receptionist about changes or the guards inform about suspicious movements. The chat should support text messages, tagging, file attachments. Also, there can be separate personal and group chats.
Priority	1
Risk	H

Requirement ID	006
Title	Should have an appointment management
Type	Functional
Description	This requires the system to have a means for patients to schedule appointments, see existing appointments, create new appointments, cancel previous appointments and reschedule. It also allows for doctors to set up regular appointments with patients.
Priority	1
Risk	H

Requirement ID	007
Title	Should have an invoice management
Type	Functional
Description	The payment system including the ability to send receipts to the clients and store the reports about each transaction. Each transaction is recorded and sent to the accountant.
Priority	1
Risk	H

Requirement ID	008
Title	Internal calendar with the doctor's schedule
Type	Functional
Description	Calendar with the doctor's schedule could be accessed by the receptionist; patients might see the time they can get an appointment, the doctor himself can make changes to his/her schedule.
Priority	1
Risk	M

Requirement ID	009
Title	Should have an equipment (including medicine) management
Type	Functional
Description	Allows to monitor the availability of medicine in the warehouse and check the condition of the present equipment. (Sends the notification if equipment got broken or if the hospital ran out of some medicine). Moreover, it has a warehouse number of equipment, number of remaining, the opportunity to mark equipment as taken/returned and an opportunity to order the new equipment.
Priority	1
Risk	M

Requirement ID	010
Title	Should have access to the building based on hierarchy
Type	Functional
Description	This functionality gives the security clearance to all parts of the building.
Priority	2
Risk	H

Requirement ID	011
Title	Should have a notification system
Type	Functional
Description	This requires the system to send prompts to a user (patient) to remind them of upcoming appointments or tasks to perform regarding what the user (doctor) prescribed. The notification system has to be fully customizable in terms of setting notification at a certain time before the upcoming appointment, changing the ringtone, enabling and disabling them. Also, the notification can be set on changes to the notice board.
Priority	2
Risk	M

Requirement ID	012
Title	Storing the statistics
Type	Functional
Description	The system should save the daily statistics and log files. The system should have the ability to show statistics and log files to the administrator account.
Priority	2
Risk	M

Requirement ID	013
Title	Should have an emergency handling system
Type	Functional
Description	The system may have an optional controlling system to handle emergency calls and availability of resources (emergency care, doctors, nurses, etc.)
Priority	2
Risk	L

Requirement ID	014
Title	Video surveillance system
Type	Functional
Description	The video control system includes a number of cameras and control center, and provides better security.
Priority	2
Risk	L

Requirement ID	015
Title	Should have a reward system
Type	Functional
Description	The system may have an option to handle with a reward system. In this, the system should use some algorithm to calculate the reward amount for every user, according to pre-given user constraints.
Priority	3
Risk	L

Requirement ID	016
Title	Should have an online donor queue
Type	Functional
Description	The system should support the online queue for the patients who are waiting for donors (queue for blood and organs) The mechanism of working of the queue should be specified by Government Standards. The patient which is given the opportunity by the doctor to enter the queue can do it and can leave the queue in case of getting the needed help.
Priority	3
Risk	L

Requirement ID	017
Title	Should have a feedback option
Type	Functional
Description	This is an optional functionality which leaves users with the possibility of rating the services provided by the system. They are able to recommend and share their suggestions and ideas and the IT-staff will take into account to improve the system.
Priority	3
Risk	L

NON-FUNCTIONAL REQUIREMENTS

Requirement ID	018
Title	Safety of stored data
Type	Non-functional
Description	This establishes that whatever data is present in our system cannot be accessible by just anyone. The patient information will be secure and confidential. The system should prevent leaks and unauthorized access to the data stored, it should be non-vulnerable to all of the kind of hacker attacks.
Priority	1
Risk	C

Requirement ID	019
Title	Reliable
Type	Non-functional
Description	Users should be almost sure that the system is not going to break at some unexpected moment, the system has to work correctly all of the time.
Priority	1
Risk	C

Requirement ID	020
Title	Should support multiple users
Type	Non-functional
Description	Requires the system to have different types of users with a hierarchy of access level: medical staff, patients, receptionists, guard, etc. The system simultaneously can be used by multiple authorised users with the different types of accounts without affecting its performance.
Priority	1
Risk	C

Requirement ID	021
Title	Portable
Type	Non-functional
Description	The system could be accessed from different devices and operating systems.
Priority	1
Risk	H

Requirement ID	022
Title	Fast response time
Type	Non-functional
Description	The user gets a response quickly without a big delay.
Priority	2
Risk	M

Requirement ID	023
Title	User-friendly interface
Type	Non-functional
Description	This establishes the interface that is easy to use from the user's point of view. The interface should be intuitively understandable, aesthetically pleasant and designed according the UI design principles.
Priority	2
Risk	L

Requirement ID	024
Title	Making an auto-backup
Type	Non-functional
Description	The system by itself will make a backup of all the data it has (medical records, user data, videos) with the specified frequency. Also, as an option, the backup time can be specified explicitly for each type of data.
Priority	3
Risk	C

Requirement ID	025
Title	Easy to maintain
Type	Non-functional
Description	In case if the system breaks down (which is in the ideal case is very unlikely), then it is easy to fix in terms of human resources and financial resources.
Priority	2
Risk	L

Requirement ID	026
Title	Extensible
Type	Non-functional
Description	The system should be flexible, the functional changes can be done without much effort and will not affect the correct functioning of the whole system.
Priority	2
Risk	L

Questions from the study of domain:

How your system will function?
Our Hospital Management System will function according to the requirements of both functional and non-functional types listed in the tables above. The system provides both employees of the hospital and its clients with a user-friendly interface helping to establish reliable communication between them by the use of the chat and interactive schedule with the ability to manage appointments and view medical reports and prescriptions. Also, it provides useful tools for equipment management, invoice management, and surveillance system.

Which entities will it have? And how they will be related to each other?
The entities we have to include:

- Patient
    - New
    - Existing
- Staff
    - Medical
        - Doctor
        - Emergency surgeon
        - Nurse
    - Non-medical
        - Pharmacist
        - IT specialist
        - Head of the hospital
        - Security
        - Supply manager
        - Cleaning person
        - Receptionist
        - Accountant

What will be the level of detailization?
The level of detailization is marked as medium. As our system supposed to work with the majority of hospitals, we decided that medium detailization level is considered to be optimal, as it can be easily adjusted to the specific requirements. Another reason is that by increasing detailisation, the system definitely becomes more complicated and tangled. That makes deleting, replacing or adding entities more complex, as it can lead to more problems because of the dependencies. In addition, the higher level of detailization makes the system less understandable both for the developer and for the customer.

Project Phase II: Domain Description
Data Modelling and Databases I, Fall 2019

Group[BS18-02] members:
Amina Miftahova
Dmitriy Podpryatov
Anna Startseva
Temurbek Khujaev
Ozioma Okonicha

20.10.2019

Corrections from TA after Phase II	**Changes made**
Chat is not a UC	“Chat use” case was replaced by “Send message” and “Receive message” use cases.
Requirements not sorted in any particular order	Requirements have been sorted by Priority and Risk.
Novelty	-
There are less functional requirements than use cases	The following requirements were added: accounts management by the IT-staff (id 004); access to the building based on hierarchy (id 010); feedback option (id 017).

Design decisions

Brief description of the system:
This masterpiece ER diagram below is a representation of our Hospital Management system transformed from a use-case diagram. Using Chen's notation, it shows the relationships between hospitals' entities specified below.

The whole system works as follows: every "user" has an account that stores their information in the system (e.g. login, password, etc.) Users are divided into two categories: patients and staff, where staff can be either medical or non-medical staff, and there are doctors who are a specification of medical staff.

Patients can make payments, receive notifications, they have their own medical record, and they can hold appointments. Doctors participate in appointments as well. All the staff has a communication system, so they can chat with each other. Also, there is inventory management which includes medical and non-medical equipment.

Finally, there is a table for video surveillance system which stores all the files from cameras (records are stored on the hard disk, and can be accessed via the path).

Entities and Attributes:

------------------------------------Strong Entities------------------------------------

- **PATIENT**
  The entity patient is a *strong entity* because it does not depend on any other entity in our system
  - ➢ Patient_id - **primary key**
  - ➢ Full_name
  - ➢ Address
  - ➢ Date_of_birth
  - ➢ Passport_number
  - ➢ Insurance_policy_number
  - ➢ Credit_card_number
  - ➢ Age - **derived attribute**(from the Date_of_birth)
  - ➢ Gender

- **STAFF**
  The entity staff is a *strong entity* because it is independent of any other entity in our system
  - ➢ Passport_number - **primary key**
  - ➢ Full_name
  - ➢ Position
  - ➢ Email - **multivalued** attribute

- **DOCTOR**
  The entity doctor is a *strong entity* because it is not dependent on any other entity in our system
  - ➢ Doctor_id - **primary key**

    - ➢ Specialization
    - ➢ Time

- **MEDICAL_STAFF**
  The entity medical staff is a *strong entity* because it does not depend on any other entity in our system
    - ➢ MS_id - **primary key**

- **NONMEDICAL_STAFF**
  The entity non-medical staff is a *strong entity* because it is independent of any other entity in our system
    - ➢ NMS_id - **primary key**

- **EQUIPMENT**
  The entity equipment is a *strong entity* because it is not dependent on any other entity in our system
    - ➢ Name
    - ➢ Quantity
    - ➢ Equipment_id - **primary key**

- **NONMEDICAL_EQUIPMENT**
  The entity nonmedical_equipment is a *strong entity* because it does not depend on any other entity in our system
    - ➢ NME_id - **primary key**

- **MEDICAL_EQUIPMENT**
  The entity medical_equipment is a *strong entity* because it is independent of any other entity in our system
    - ➢ ME_id - **primary key**

- **VIDEO_RECORDS**
  The entity video records is a *strong entity* because it is not dependent on any other entity in our system
    - ➢ Video_id - **primary key**
    - ➢ Camera_number
    - ➢ Date
    - ➢ Path

-----------------------------------Weak Entities-----------------------------------

- **APPOINTMENT**
  The entity appointment is a *weak entity* because the entity appointment depends on the patient and the doctor
    - ➢ Appointment_id - **primary key**
    - ➢ Date
    - ➢ Time
    - ➢ Price

- ➢ Room

- **MEDICAL_RECORD**
  The entity medical record is a *weak entity* because the entity medical record is dependent on the patient entity
  - ➢ Record_id - **primary key**
  - ➢ Date_created
  - ➢ Prescription - **multivalued attribute**
  - ➢ Diagnosis - **multivalued attribute**

- **NOTIFICATION**
  The entity notification is a *weak entity* because it depends on the entity patient
  - ➢ Notification_id - **primary key**
  - ➢ Date
  - ➢ Event
  - ➢ Sound

- **ACCOUNT**
  The entity account is a *weak entity* because it is dependent on the entities patient and staff
  - ➢ Account_id - **primary key**
  - ➢ Login
  - ➢ Password
  - ➢ Last_time_online
  - ➢ Date_of_creation
  - ➢ Permission_level

- **PAYMENT**
  The entity payment is a *weak entity* because it depends on the patient entity
  - ➢ Invoice_number - **primary key**
  - ➢ Amount
  - ➢ Date
  - ➢ Description

- **MESSAGE**
  The entity message is a *weak entity* because the entity depends on the STAFF entity
  - ➢ Message_id - **primary key**
  - ➢ Text

Relationships:

- PATIENT -- **MAKES** -- PAYMENT
  Relation type : weak
  Cardinality : one-to-many
  The relationship is weak because the entity PAYMENT depends on the PATIENT entity.
  The cardinality can be explained as one patient can make many payments and each payment must be made by exactly one patient

- PATIENT -- **RECEIVES** -- NOTIFICATION
  Relation type : weak
  Cardinality : one-to-many
  The relationship is weak because the NOTIFICATION entity is dependent on the PATIENT entity.
  The cardinality can be explained as one patient can receive many notifications and each notification must be received by exactly one patient

- PATIENT -- **OWNS** -- MEDICAL_RECORD
  Relation type : weak
  Cardinality : one-to-one
  The relationship is weak because MEDICAL RECORDS depends on PATIENT.
  The cardinality can be explained as each patient owns exactly one medical record and each medical record is owned by exactly one patient

- DOCTOR and PATIENT -- **HOLDS** -- APPOINTMENT
  Relation type : weak
  Cardinality : one-to-many
  The relationship is weak because one of the entities involved(APPOINTMENT) depends on the other entities involved (DOCTOR and PATIENT).
  The cardinality can be explained as one doctor/patient can hold many appointments while each appointment should have only one patient/doctor

- DOCTOR -- **IS_AN_INSTANCE_OF** -- MEDICAL_STAFF
  Relation type : strong
  Cardinality : one-to-one
  The relationship is strong because neither DOCTOR nor MEDICAL_STAFF depends on the other.
  The cardinality can be explained as one doctor represents exactly one instance of MEDICAL_STAFF, and for each MEDICAL_STAFF there cannot be more than one DOCTOR

- MEDICAL_STAFF and NONMEDICAL_STAFF -- **IS_A** -- STAFF

  Relation type : strong
  Cardinality : one-to-one
  The relationship is strong because none of its participating entities is dependent.
  The participation is total for MEDICAL_STAFF and NONMEDICAL_STAFF, as each instance of them is also an instance of staff.
  The cardinality means that MEDICAL_STAFF and NONMEDICAL_STAFF are disjoint sets and no instance of STAFF can be simultaneously medical and nonmedical

- STAFF -- **SENDS** -- MESSAGE

  Relation type : weak
  Cardinality : one-to-many
  The relationship is weak because MESSAGE is dependent on STAFF
  The cardinality can be explained as one STAFF can send many messages

- STAFF -- **GETS** -- MESSAGE

  Relation type : weak
  Cardinality : many-to-many
  The relationship is weak as MESSAGE is a weak entity and depends on the message sender and message receiver.
  The cardinality, firstly, means that one instance of STAFF can get multiple messages, and, secondly, that there may be cases when one message can be received by multiple users (for example, in a group chat).

- STAFF and PATIENT -- **HAS** -- ACCOUNT

  Relation type : weak
  Cardinality : one-to-one
  The relationship is weak because the entity ACCOUNT is dependent on other entities (STAFF and PATIENT)
  The cardinality is one-to-one because each staff/patient must have exactly one account and each account must be owned by exactly one staff/patient

- MEDICAL_EQUIPMENT and NONMEDICAL_EQUIPMENT -- **IS_AN** -- EQUIPMENT

  Relation type : strong
  Cardinality : one-to-one
  The relation is strong as all three entities participating in a relation are strong. For the MEDICAL_EQUIPMENT and NONMEDICAL_EQUIPMENT, it is total participation as each instance is also an instance of equipment. The cardinality means that MEDICAL_EQUIPMENT and NONMEDICAL_EQUIPMENT are disjoint sets and no of the equipment can belong to both simultaneously

Address
Date_of_birth
Passport_number
Patient_id
Insurance_policy_number
Full_name
PATIENT
Credit_card_number
Age
Gender
Time
Doctor_id
Specialization
DOCTOR
IS_AN_INSTANCE_OF
MS_id
MEDICAL_STAFF
NMS_id
NONMEDICAL_STAFF
IS_A
Date_sent
Time_sent
SENDS
STAFF
MESSAGE
Text
Message_id
GETS
Email
Position
Full_name
Passport_number
MAKES
RECEIVES
OWNS
HOLDS
PAYMENT
Date
Description
Invoice_number
Amount
NOTIFICATION
Notification_id
Date
Event
Sound
MEDICAL_RECORD
Diagnosis
Prescription
Creation_date
Record_id
APPOINTMENT
Date
Time
Appointment_id
Room
Price
HAS
ACCOUNT
Account_id
Login
Password
Permission_level
Date_of_creation
Last_time_online
NME_id
NONMEDICAL_EQUIPMENT
MEDICAL_EQUIPMENT
ME_id
IS_AN
EQUIPMENT
Name
Equipment_id
Quantity
VIDEO RECORD
Camera_number
Path
Date
Video_id
1
1
1
1
1
1
1
1
1
1
1
N
N
N
M
N
N
N
1
1
1
1
1

F19 DMD assignment phase 3

How to run python apps

PostgreSQL

1. Needed libraries: PyQt5, psycopg2
2. Create database
3. Run *hospital_postgresql.sql* - dump file
4. Execute `python ui.py`
5. Enjoy beautiful moustached men!

MySQL

1. Needed libraries: PyQt5, mysql-connector-python
2. Create database
3. Run *hospital_mysql.sql* - dump file
4. Execute `python ui.py`
5. Enjoy beautiful moustached men!

Files and folders

1. MySQL creation&population
2. MySQL queries
3. PostgreSQL creation&population
4. PostgreSQL queries
5. Population source scripts
6. Updated pdf
7. INSERT statements which will have 3rd query result nonempty