HOSPITAL MANAGEMENT

The aim of our project is to develop the best suited application for hospital management. This application will control practically all the logistic functions of a hospital and will be used by all the people involved in it, from doctors, nurses to patients…

The application will follow all the steps from the arriving of the patient, to his recovery. The hospital has several doctors, each one specialized in a certain disease. From the moment the patient registers, a doctor will analyze the symptoms, diagnose, and treat him with the necessary medicines.

Furthermore, depending on the disease and the status of the patient, a long stay at hospital might be needed. In that case a nurse will be needed for monitoring the patient

To sum up, this application will improve the doctor's job and their efficiency, the access of all the information regarding the logistics of the hospital. It will also improve the stay of the patients and their needs.

USE CASES:

Use case 1:

- Actor: patient/doctor/nurse
- Goal: register
- Path:

5. Input username/password
6. Input patient data (phone number and social security card)
7. Input phone number of a close family member
8. Input symptoms (in case of the patient)

Use case 2:

- Actor: doctor
- Goal: diagnose the patient
- Prerequisites: patient registered
- Path:

1. Look at symptoms
2. Consider if the patient needs to stay

Use case 3:

- Actor: doctor
- Goal: report status to the family of the patient
- Path:
    1. Call a close family member
- Precondition: use case 2

Use case 4:

- Actor: doctor
- Goal: treat the patient
- Path:
    1. Input medicines
    2. Input procedures (tests and surgeries)
- Precondition: use case 2

Use case 5:

- Actor: doctor
- Goal: discharge patient
- Path:
    1. Call family member
    2. Put the final “results” in the medical history of the patient
- Precondition: use case 2

Use case 6:

- Actor: patient
- Goal: leave hospital
- Path:
    1. Pay with credit card or cash
    2. Ask for an appointment for checking

Use case 7:

- Actor: nurse
- Goal: monitor the patient
- Path:
    1. Warn doctor if the patient is not okay

LIST OF REQUIREMENTS:

Functional requirements:

FR1: Must allow patients to register

FR2: Must allow doctors to register

FR3: Must allow patients to stay at hospital for a short or long time

FR4: Must allow the doctor to update a patient's status

FR5: Must allow the doctor to report the status of a patient to his family members

FR6: Must allow the doctor to discharge the patient

FR7: Must allow the patient to pay

FR8: Must allow the nurse to monitor a patient

Non-Functional requirements:

NFR1: Must allow the patient to pay with credit card or cash

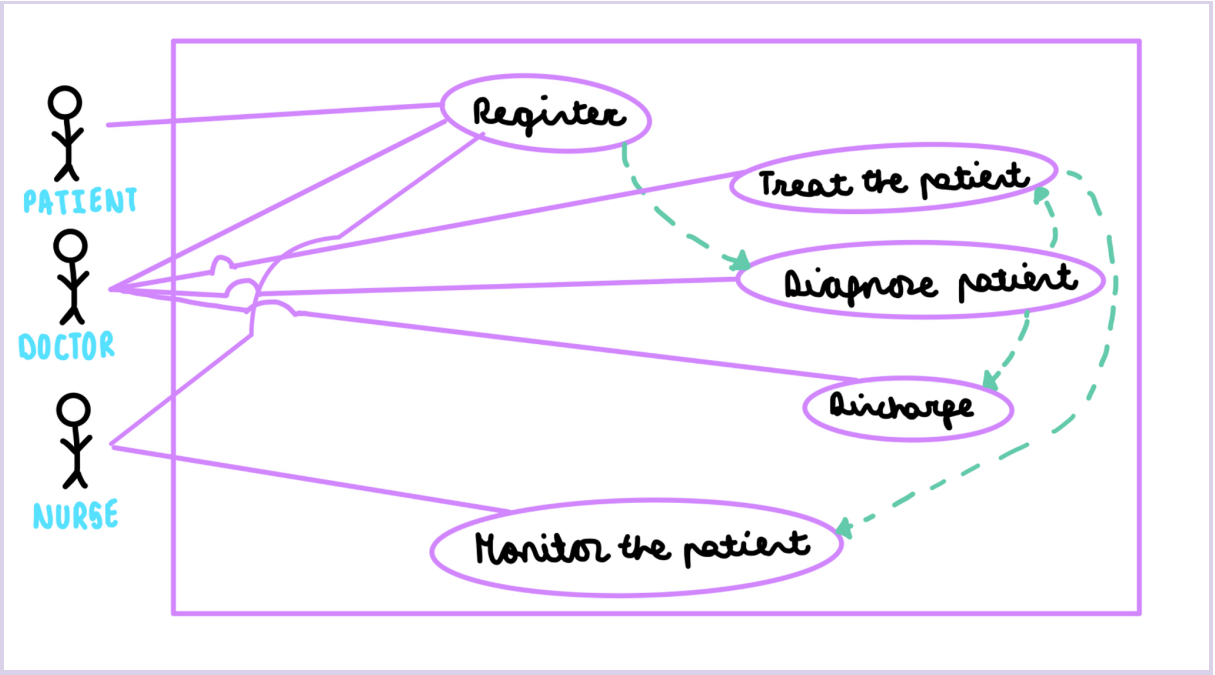
NFR2: Application must be able to run in a laptop computer

NFR3: Application must support 15 or more doctors and nurses

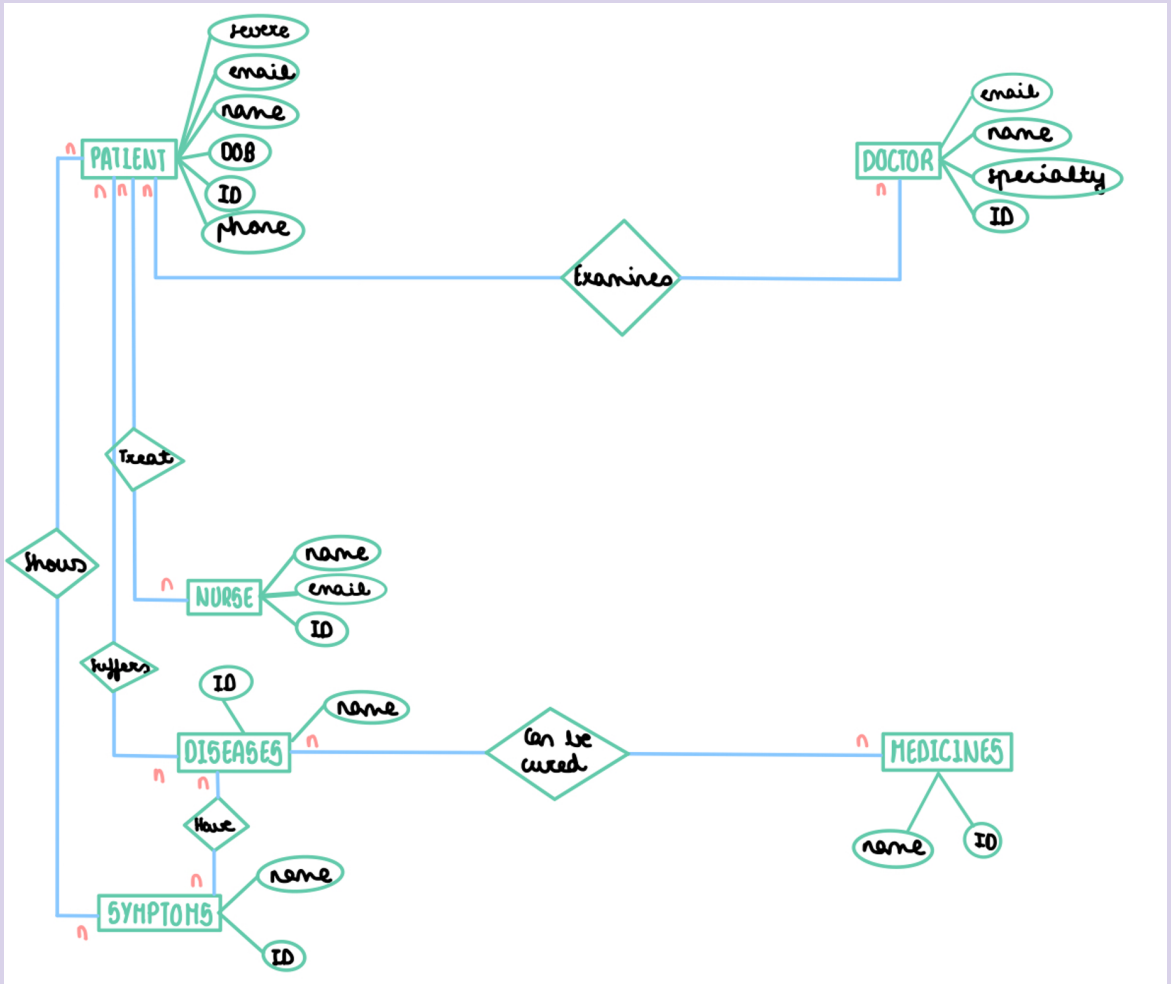
TRACEABILITY MATRIX

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	NFR 1	NFR2	NFR 3
UC1											
UC2											
UC3											
UC4											
UC5											
UC6											
UC7											

UML USE CASE DIAGRAM



E-R DIAGRAM



RELATIONAL DATA BASE'S TABLE

Patients table:

Patient ID	Name	Nurse ID	Phone number of family member	E-mail	Status	Date of birth
1	Alejandra Sánchez	1	629771031	a.sanchez@gmail.es	Severe	02/05/1998
2	Rocío Martínez	3	610328366	r.martinez@gmail.es	Severe	29/01/1970
3	Alba Martín	2	681945567	a.martin@gmail.es	Not severe	14/12/2000
4	Javier Rodríguez	2	633089221	j.rodriguez@gmail.es	Not severe	27/10/1983
5	Carlos García	3	617251903	c.garcia@gmail.es	Not severe	03/07/1990
6	Rodrigo Aguirre	1	657518224	r.aguirre@gmail.es	Severe	24/06/1993
7	María Puente	1	644972006	m.puente@gmail.es	Not severe	15/02/2005
8	Jaime Díaz	2	658820036	j.diaz@gmail.es	Severe	22/05/2002

Nurse's table:

ID	Name
1	Sofía Perez
2	Leticia Jimenez
3	Lucas Canellas

Diseases table:

ID	Name
1	Migraines
2	Stomach flu

3	Cancer
4	Covid SARS-19
5	Heart disease
6	Diabetes

Medicines table:

ID	Name
1	Gelocatil
2	Antibiotic
3	Remdesivir
4	Ibuprofeno
5	Insulin
6	Benazepril

Doctor's table:

ID	Name	Specialty
1	Julio Hernández	Cancer
2	Rosa María Posada	Head and Stomach aches
3	Carmen Lucena	COVID
4	Pablo Soler	Diabetes doctor
5	Mª Dolores Jiménez	Cardiologist

Many to many (doctor-patient) table:

Patient ID	Doctor ID
1	1
2	4
3	3
4	2
5	1
6	5
7	5
8	2

Many to many (Disease-medicine) table:

Medicine ID	Disease ID
1	1
2	2
3	4
4	1
4	2
5	6
6	6

Many to many(Patient-Disease) table:

Patient ID	Disease ID
1	1
2	4
3	5

4	2
5	3
6	6
7	1
8	2

Symptoms table:

ID	Name
1	Headache
2	Nasal congestion
3	Stomache
4	Tiredness
5	Overall pain
6	Bone pain
7	Tachycardia
8	Blackout
9	Loss of smell
10	Puke

Many to many Patient-Symptoms

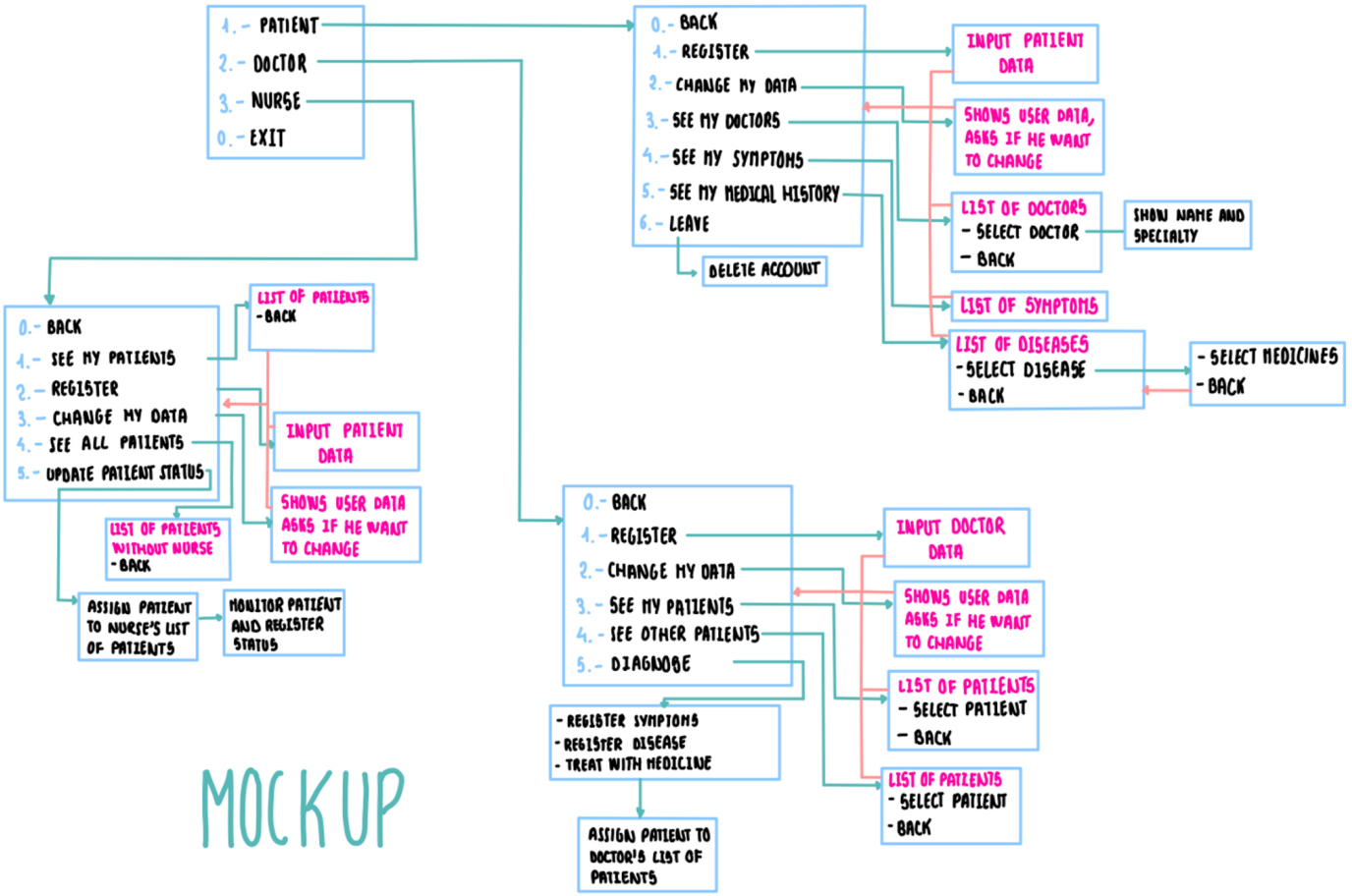
Patient ID	Symptoms ID
1	**1**
2	**2**
2	**4**
2	**9**
2	**1**
3	**7**

3	**4**
4	**3**
4	**10**
5	**5**
5	**8**
5	**4**
5	**6**
6	**8**
7	**1**
8	**10**

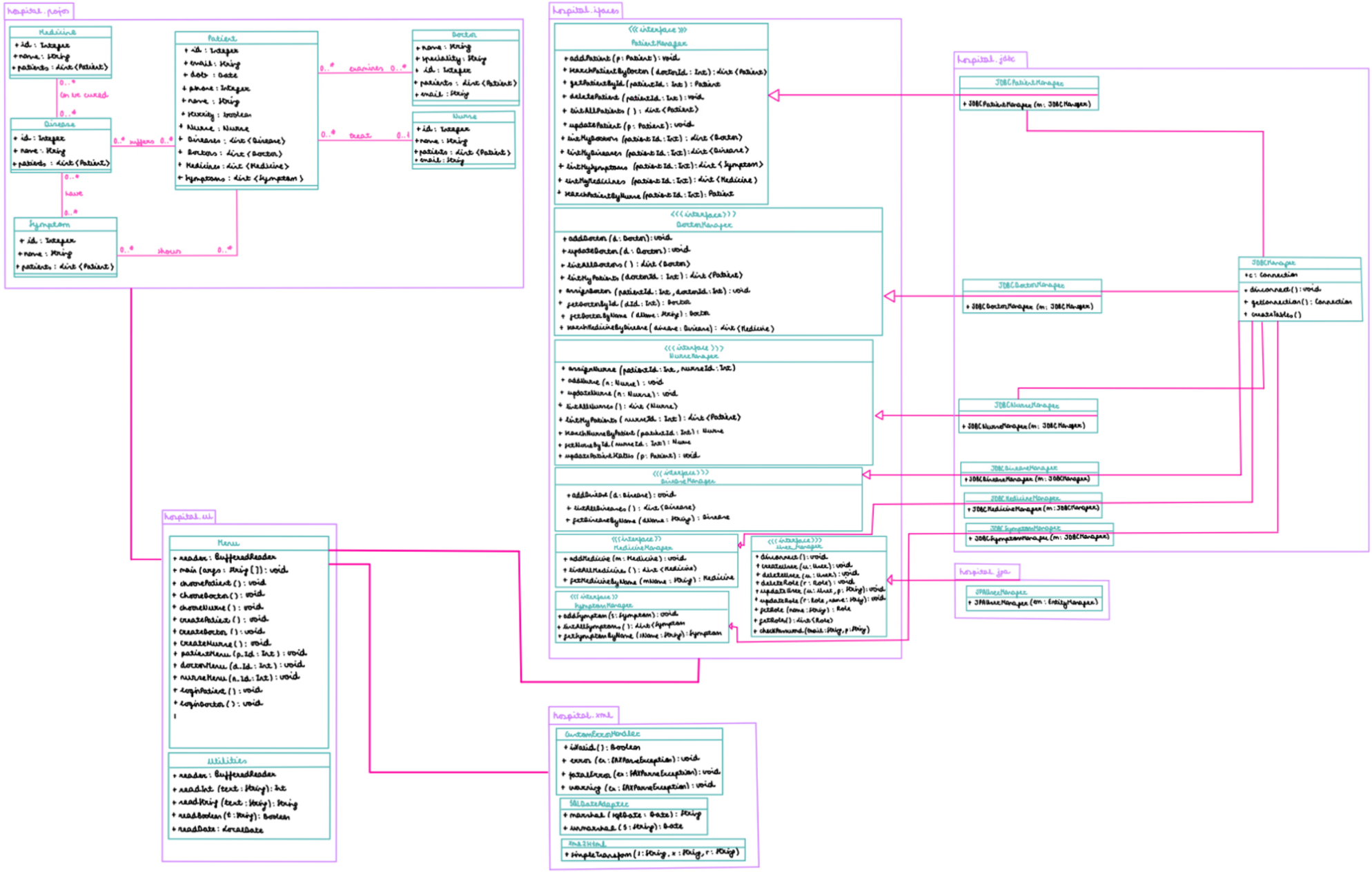
Many to many Symptoms-Disease

Symptoms ID	**Disease ID**
1	**1**
1	**4**
2	**4**
3	**2**
4	**4**
4	**3**
5	**3**
5	**4**
5	**5**
6	**3**
6	**4**
7	**5**
8	**1**
9	**4**
10	**2**

Mock up


1.- PATIENT
2.- DOCTOR
3.- NURSE
0.- EXIT
0.- BACK
1.- REGISTER
2.- CHANGE MY DATA
3.- SEE MY DOCTORS
4.- SEE MY SYMPTOMS
5.- SEE MY MEDICAL HISTORY
6.- LEAVE
DELETE ACCOUNT
INPUT PATIENT DATA
SHOWS USER DATA, ASKS IF HE WANT TO CHANGE
LIST OF DOCTORS
- SELECT DOCTOR
- BACK
SHOW NAME AND SPECIALTY
LIST OF SYMPTOMS
LIST OF DISEASES
- SELECT DISEASE
- BACK
- SELECT MEDICINES
- BACK
0.- BACK
1.- SEE MY PATIENTS
2.- REGISTER
3.- CHANGE MY DATA
4.- SEE ALL PATIENTS
5.- UPDATE PATIENT STATUS
LIST OF PATIENTS
-BACK
INPUT PATIENT DATA
SHOWS USER DATA ASKS IF HE WANT TO CHANGE
LIST OF PATIENTS WITHOUT NURSE
- BACK
ASSIGN PATIENT TO NURSE'S LIST OF PATIENTS
MONITOR PATIENT AND REGISTER STATUS
0.- BACK
1.- REGISTER
2.- CHANGE MY DATA
3.- SEE MY PATIENTS
4.- SEE OTHER PATIENTS
5.- DIAGNOSE
INPUT DOCTOR DATA
SHOWS USER DATA ASKS IF HE WANT TO CHANGE
LIST OF PATIENTS
- SELECT PATIENT
- BACK
LIST OF PATIENTS
- SELECT PATIENT
- BACK
- REGISTER SYMPTOMS
- REGISTER DISEASE
- TREAT WITH MEDICINE
ASSIGN PATIENT TO DOCTOR'S LIST OF PATIENTS
MOCKUP

UML


hospital.pojo
Medicine
+ id : Integer
+ name : String
+ patients : List<Patient>
0..*
can be cured
0..*
Patient
+ id : Integer
+ email : String
+ dob : Date
+ phone : Integer
+ name : String
+ sexMale : boolean
+ Nurse : Nurse
+ Diseases : List<Disease>
+ Doctors : List<Doctor>
+ Medicine : List<Medicine>
+ Symptoms : List<Symptom>
0..* examines 0..*
Doctor
+ name : String
+ speciality : String
+ id : Integer
+ patients : List<Patient>
+ email : String
0..* treat 0..1
Nurse
+ id : Integer
+ name : String
+ patients : List<Patient>
+ email : String
Disease
+ id : Integer
+ name : String
+ patients : List<Patient>
0..* suffers 0..*
0..*
have
0..*
Symptom
+ id : Integer
+ name : String
+ patients : List<Patient>
0..* shown 0..*
hospital.ijava
«interface»
PatientManager
+ addPatient(p : Patient) : void
+ searchPatientByDoctor(doctorId : Int) : List<Patient>
+ getPatientById(patientId : Int) : Patient
+ deletePatient(patientId : Int) : void
+ listAllPatients() : List<Patient>
+ updatePatient(p : Patient) : void
+ listMyDoctors(patientId : Int) : List<Doctor>
+ listMyDiseases(patientId : Int) : List<Disease>
+ listMySymptoms(patientId : Int) : List<Symptom>
+ listMyMedicines(patientId : Int) : List<Medicine>
+ searchPatientByName(patientId : Int) : Patient
«interface»
DoctorManager
+ addDoctor(d : Doctor) : void
+ updateDoctor(d : Doctor) : void
+ listAllDoctors() : List<Doctor>
+ listMyPatients(doctorId : Int) : List<Patient>
+ assignDoctor(patientId : Int, doctorId : Int) : void
+ getDoctorById(did : Int) : Doctor
+ getDoctorByName(dName : String) : Doctor
+ searchMedicineByDisease(disease : Disease) : List<Medicine>
«interface»
NurseManager
+ assignNurse(patientId : Int, nurseId : Int)
+ addNurse(n : Nurse) : void
+ updateNurse(n : Nurse) : void
+ listAllNurses() : List<Nurse>
+ listMyPatients(nurseId : Int) : List<Patient>
+ searchNurseByPatient(patientId : Int) : Nurse
+ getNurseById(nurseId : Int) : Nurse
+ updatePatientWithDo(p : Patient) : void
«interface»
DiseaseManager
+ addDisease(d : Disease) : void
+ listAllDiseases() : List<Disease>
+ getDiseaseByName(dName : String) : Disease
«interface»
MedicineManager
+ addMedicine(m : Medicine) : void
+ listAllMedicines() : List<Medicine>
+ getMedicineByName(mName : String) : Medicine
«interface»
SymptomManager
+ addSymptom(s : Symptom) : void
+ listAllSymptoms() : List<Symptom>
+ getSymptomByName(sName : String) : Symptom
«interface»
UserManager
+ dbInvoke() : void
+ createUser(u : User) : void
+ deleteUser(u : User) : void
+ deleteRole(r : Role) : void
+ updateUser(u : User, p : String) : void
+ updateRole(r : Role, name : String) : void
+ getRole(name : String) : Role
+ getRoles() : List<Role>
+ checkCommand(cmd : String, p : String)
hospital.jdbc
JDBCPatientManager
+ JDBCPatientManager(m : JDBCManager)
JDBCDoctorManager
+ JDBCDoctorManager(m : JDBCManager)
JDBCNurseManager
+ JDBCNurseManager(m : JDBCManager)
JDBCDiseaseManager
+ JDBCDiseaseManager(m : JDBCManager)
JDBCMedicineManager
+ JDBCMedicineManager(m : JDBCManager)
JDBCSymptomManager
+ JDBCSymptomManager(m : JDBCManager)
JDBCManager
+c : Connection
+ disconnect() : void
+ getConnection() : Connection
+ createTables()
hospital.jpa
JPAUserManager
+ JPAUserManager(em : EntityManager)
hospital.ui
Menu
+ reader : BufferedReader
+ main(args : String[]) : void
+ choosePatient() : void
+ chooseDoctor() : void
+ chooseNurse() : void
+ createPatient() : void
+ createDoctor() : void
+ createNurse() : void
+ patientMenu(p_id : Int) : void
+ doctorMenu(d_id : Int) : void
+ nurseMenu(n_id : Int) : void
+ loginPatient() : void
+ loginDoctor() : void
Utilities
+ reader : BufferedReader
+ readInt(text : String) : Int
+ readString(text : String) : String
+ readBoolean(t : String) : Boolean
+ readDate : LocalDate
hospital.xml
CustomErrorHandler
+ isValid() : Boolean
+ error(ex : SAXParseException) : void
+ fatalError(ex : SAXParseException) : void
+ warning(ex : SAXParseException) : void
XMLDateAdapter
+ marshal(xmlDate : Date) : String
+ unmarshal(s : String) : Date
XmlTest
+ simpleTransform(i : String, x : String, r : String)