DATA BASE FOR MONITORING BIOMEDICAL DEVICES

Data Base for monitoring biomedical devices

The aim of our project is to create a data base for the Hospital “La Paz” in order to organize all the information related with the doctors, their patients, and the biomedical devices in use.

The six entities are:

1. Appointment
2. Devices
3. Doctors
4. Manufacturer
5. Patient

Each entity has different properties, as well as an Id which identifies it. We can observe this in the ER diagram.

USER STORIES:

- As a patient, I need a device to improve my quality of life.
- As a doctor, I treat a patient to monitor their diagnosis.
- As a doctor, I accept an appointment to check on a patient.
- As a patient, I order an appointment with the doctor.
- As a manufacturer, I check what devices do the hospital need in order to fabricate them.

USE CASES:

- **Order devices:**

Actor: Doctor

~ Description: The doctor orders an amount of devices in a specific date.

~ Preconditions: An appointment must had been taken place between the doctor and the patient. The doctor must had decided that the patient needs a device according to its diagnosis.

Standard path:

1. Select doctor
2. Select the option order device
3. Select the device requested and the amount

- **Confirms an appointment :**

Actor: Doctor

~ Description: The doctor enters the application and requests an appointment on a specific date.

~ Preconditions: The doctor must be registered in the application.

Standard path:

1. Select doctor
2. Select the option request appointment

- **Request appointment :**

Actor: Patient

~ Description: The patient enters the application and requests for an appointment on a specific date.

~ Preconditions: The patient must be registered in the application.

Standard path:

1. Select patient
2. Select the option request appointment
3. Select the date of the appointment

- **Register patient:**

Actor: Patient

~ Description: The patient enters on the application and introduces his personal information to register.

~ Preconditions:

Standard path:

1. Select patient
2. Select the option register patient
3. Introduce the personal information

- **Cancel an appointment :**

Actor: Doctor

~ Description: The doctor enters the application and cancels an appointment previously made by a patient for a specific date.

~ Preconditions: The patient must have asked for an appointment.

Standard path:

1. Select doctor
2. Select the option cancel appointment

- **View information about device:**

Actor: Patient

~ Description: The patient enters the application and checks the specific information about his device.

~ Preconditions: The patient needs to be registered and must have a device.

Standard path:

1. Select patient
2. Select the option view information about device

- **View devices needed:**

Actor: Manufacturer

~ Description: The manufacturer enters the application and edits his personal information.

~ Preconditions: The Hospital must have requested devices.

Standard path:

1. Select manufacturer
2. Select the option view devices needed

- **Modify data :**

Actor: Manufacturer

~ Description: The manufacturer enters the application and modifies his own information.

~ Preconditions: The manufacturer must be registered in the application.

Standard path:

1. Select manufacturer
2. Select the option edit personal information
3. Select the information that he wants to change

- **Cancel appointment :**

Actor: Patient

~ Description: The patient enters the application and cancels the appointment.

~ Preconditions: The patient must have requested for an appointment.

Standard path:

1. Select patient
2. Select the option cancel appointment

- **View information about patient :**

Actor: Doctor

~ Description: The doctor enters the application and checks the information about the patient.

~ Preconditions: The patient needs to be registered and they must have had an appointment.

Standard path:

1. Select doctor
2. Select the option view information about patient

- **Modify data :**

Actor: Doctor

~ Description: The doctor enters the application and modifies his own information.

~ Preconditions: The doctor must be registered in the application.

Standard path:

1. Select doctor
2. Select the option edit personal information
3. Select the information that he wants to change

- **Modify data :**

Actor: Patient

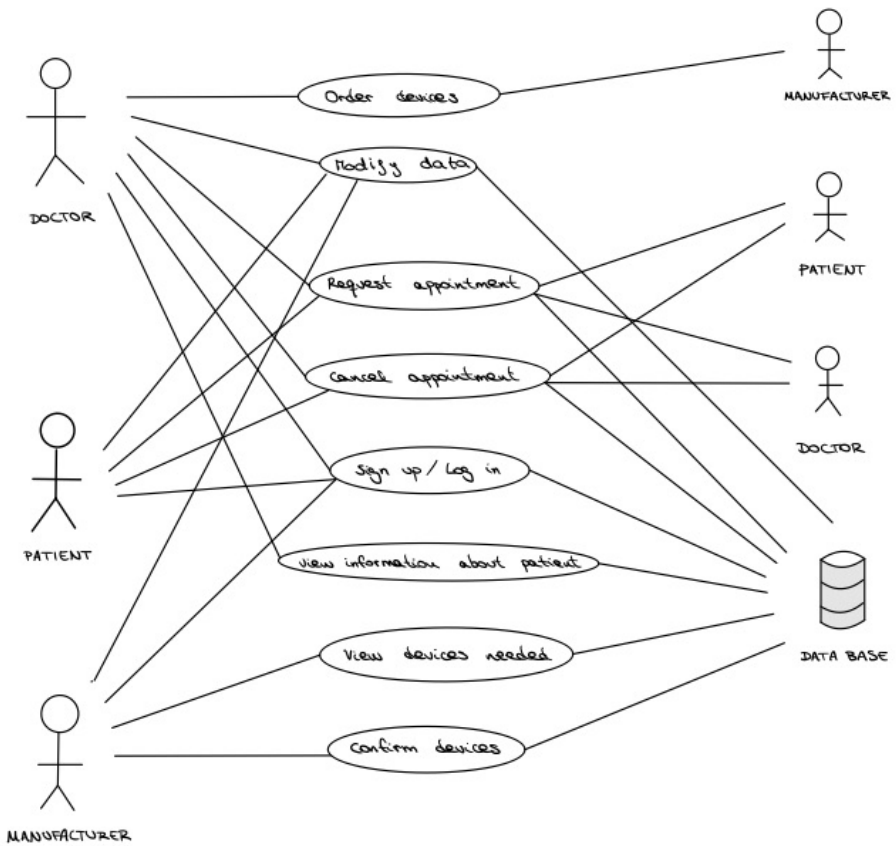
~ Description: The patient enters the application and modifies his own information.

~ Preconditions: The patient must be registered in the application.

Standard path:

1. Select patient
2. Select the option edit personal information
3. Select the information that he wants to change

UML use-case diagram:



REQUIREMENTS

FUNCTIONAL:

1 - The system stores info. about the patient in the D.B.

2 - The system stores info. about the devices in the D.B.

3 - The doctor can add a device to the system

4 - The doctor can modify the patient´s data

5 - The doctor has access to the patient´s data

6 - The doctor can add data to the system

7 - The doctor can print the list of the patients that he treats

8 - The doctor can assign a patient to him

9 - The doctor can modify his own data

10 - The patient can choose a doctor for his appointment

11 - The patient can obtain his own data

12 - The patient can obtain his own device information

13 - The patient can modify his own data

14 - The access to the app requires login

15 - The system has three roles of users

NON-FUNCTIONAL:

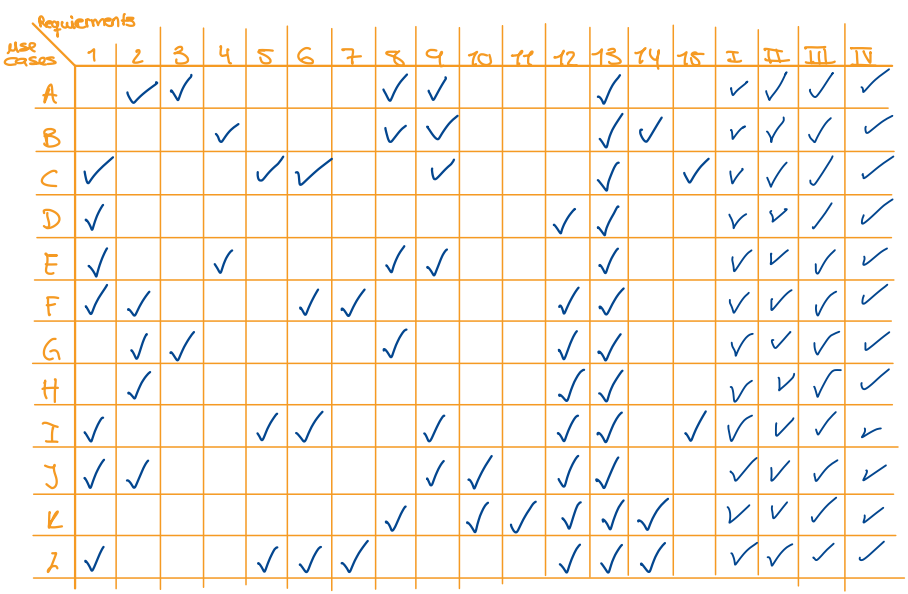
i – The system should respond to the user interaction in less than two seconds

ii - The system should be able to support multiple users at the same time without reducing performance

iii - In case of system failure, the system can be able to recover within 30 minutes

iv - The system does regular backups of the D.B.

TRACEABILITY MATRIX:



Patients

p_id	Name	Surname	Device	B date	Diagnosis
1.	Maria	Gorgojo	Pacemaker	2003-9-22	Arrhythmia
2	Sandra	Cabrera	Prosthetic Limb	2003-7-25	Amputated Leg

Manufacturer

m_id	Name	Adress	Phone number
1	Lucia	Andes Av.	689054844
2	Ana	Serrano street	629941093

Doctor

do_id	Specialty	Name	Surname
1	Cardiologist	José Luis	Luchelli
2	Traumatologist	Roberto	Rodríguez

Appointment

a_id	Date	Description	doctor_id	patient_id
1	2024-6-2	Revision	1	1
2	2024-7-5	Revision	2	2

Device

d_id	Type	manufacturer_id
1	Pacemaker	1
2	Insulin pump	2

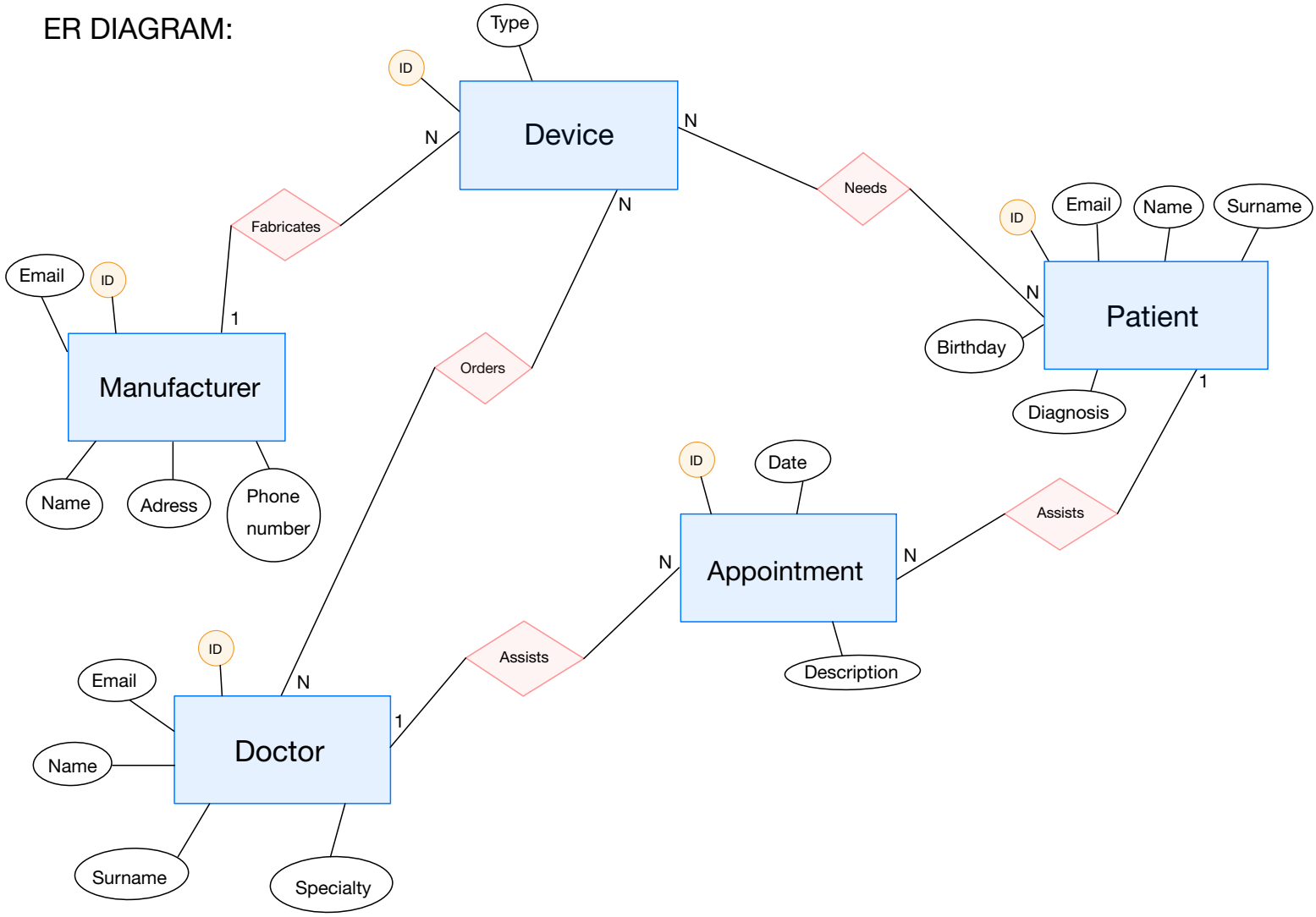
Device_doctor: Orders

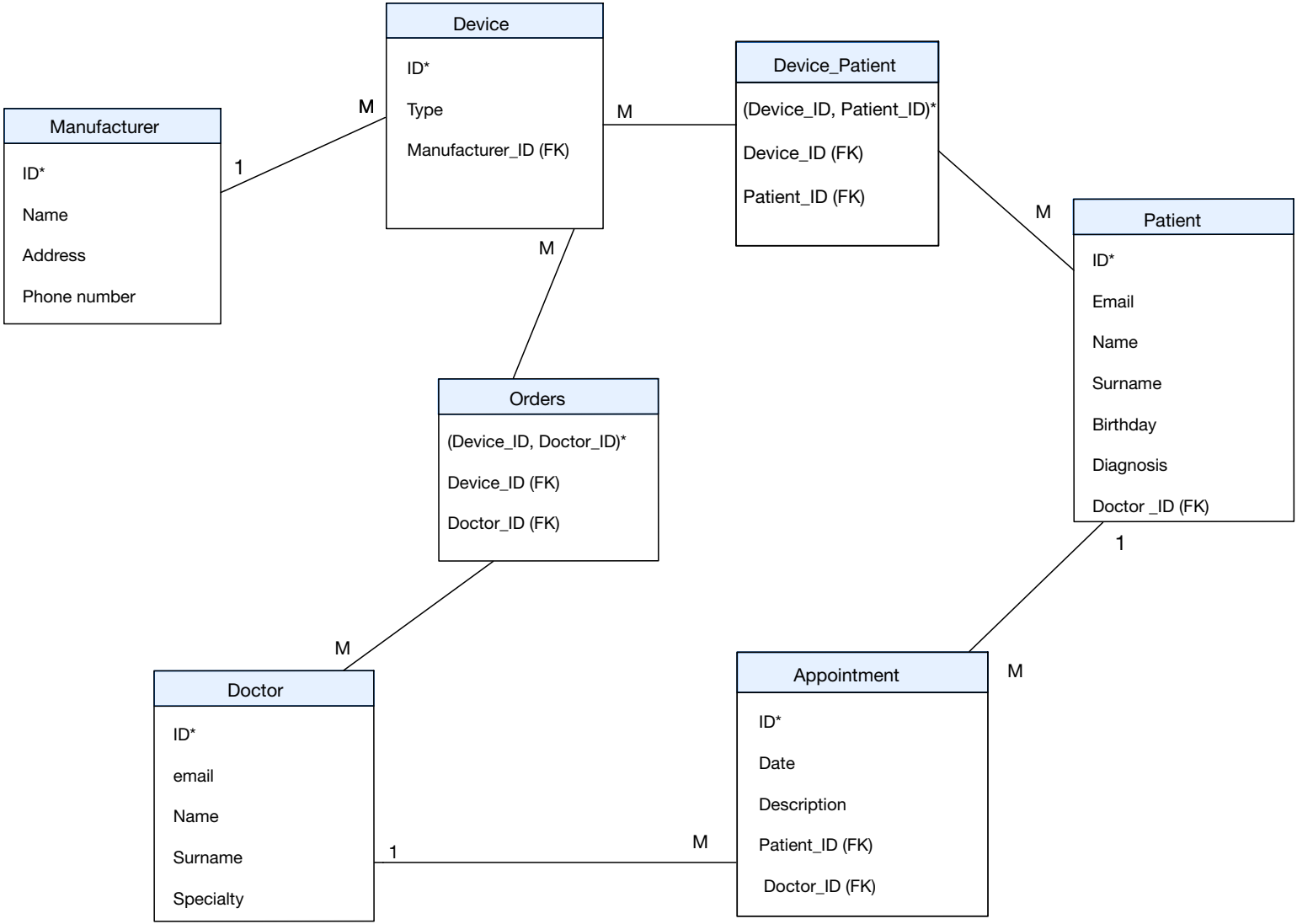
device_id	doctor_id
FK to d_id	FK to do_id
1	1
2	2

Device_patient

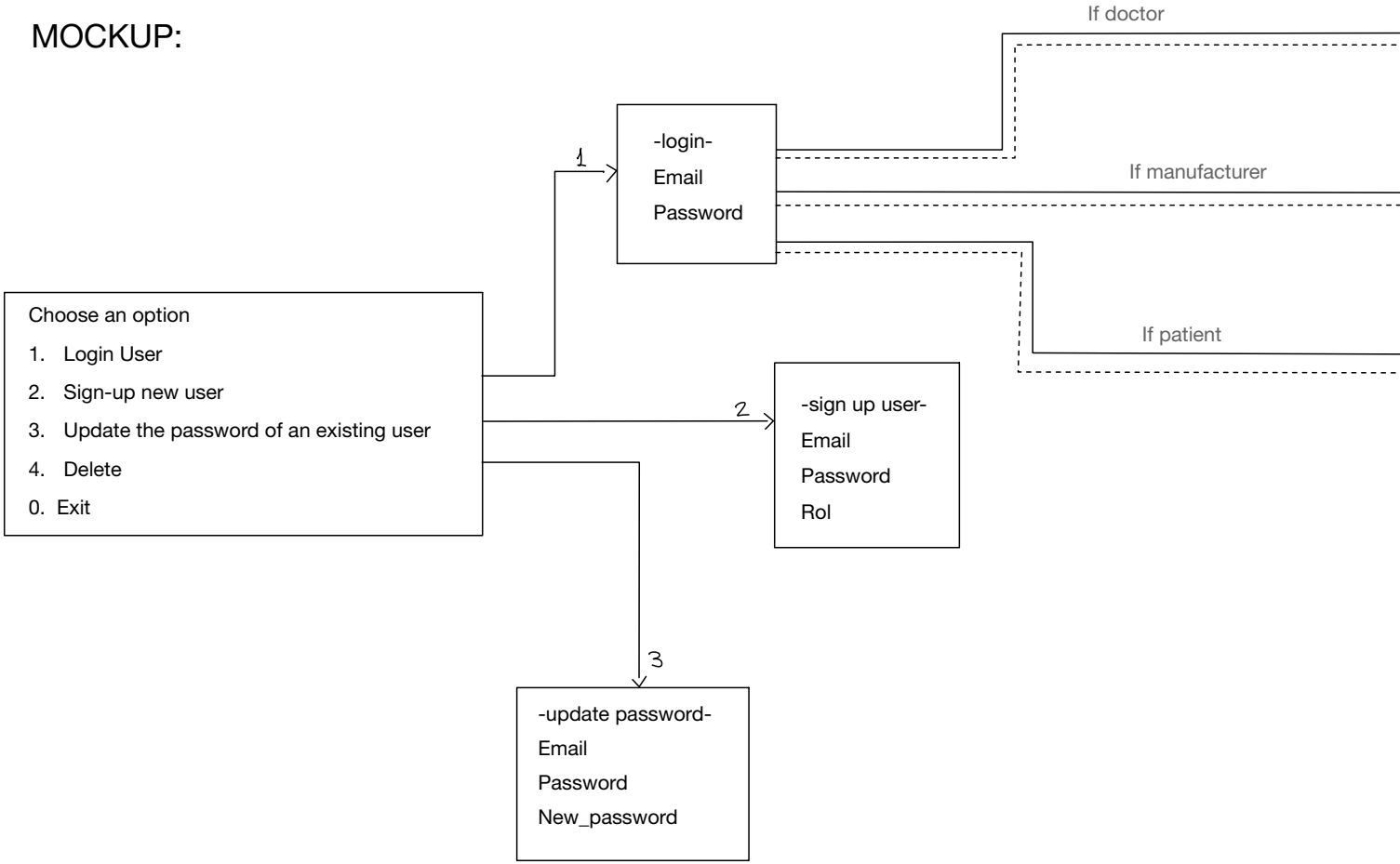
device_id	patient_id
FK to d_id	FK to p_id
1	1
2	2

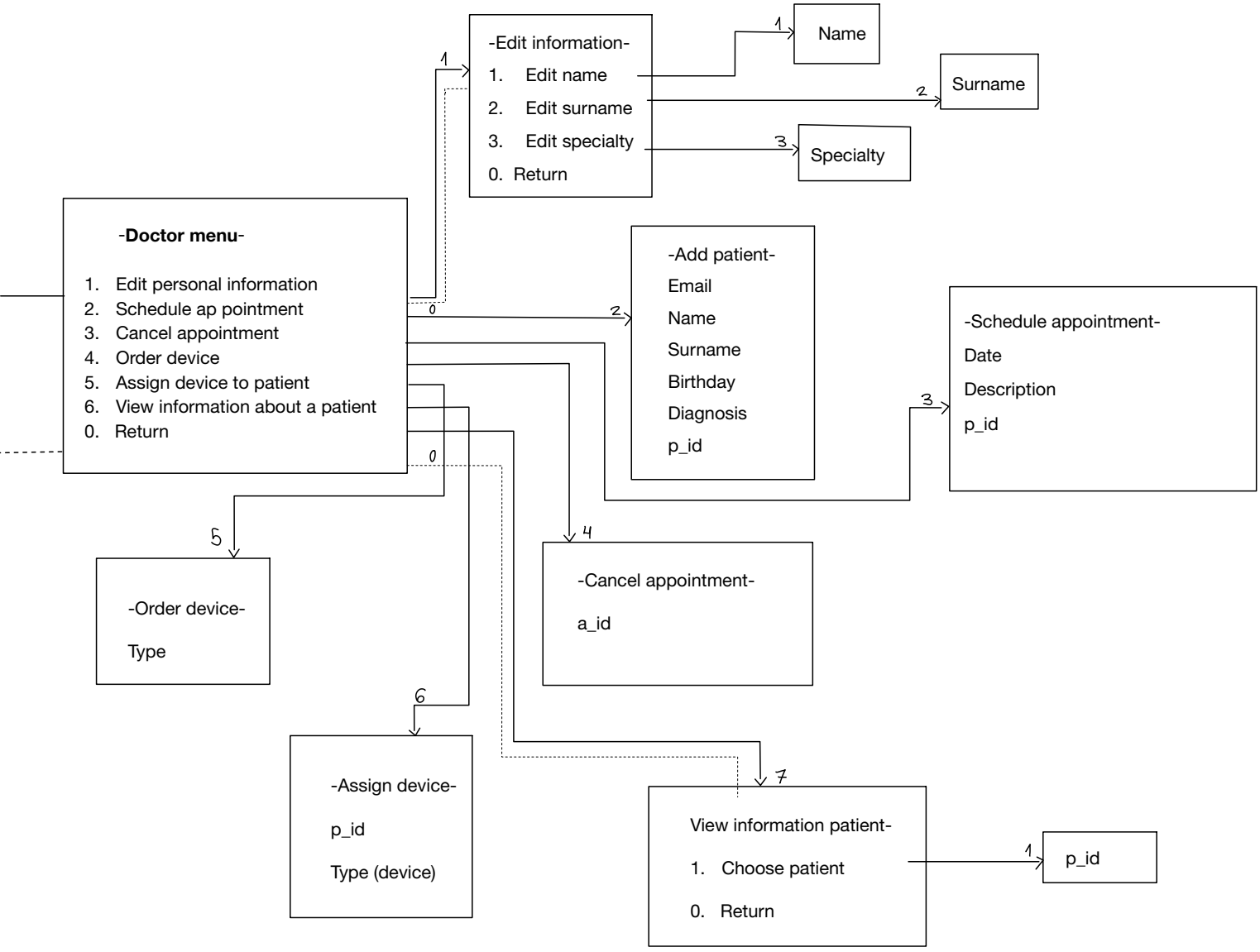
ER DIAGRAM:

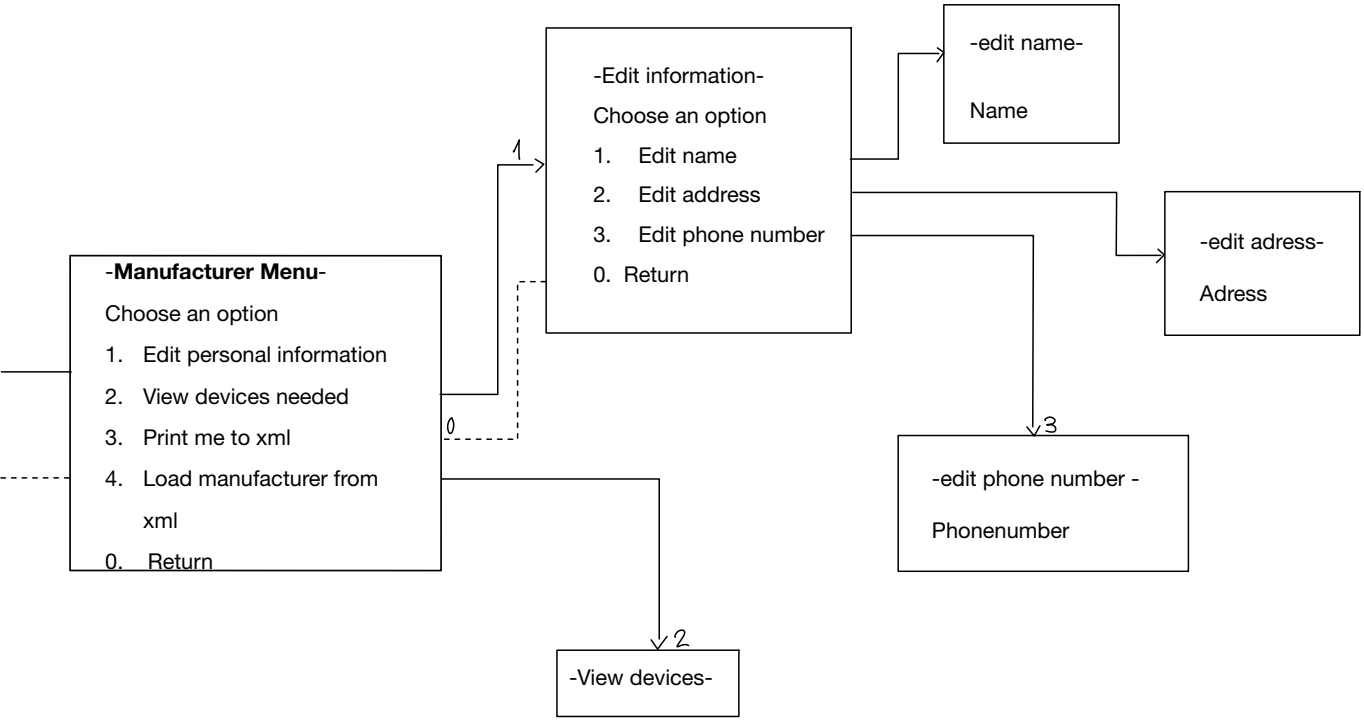

Type
ID
Device
N
N
Fabricates
Needs
N
1
Email
ID
Manufacturer
Orders
Name
Adress
Phone number
ID
Email
Name
Surname
Patient
N
Birthday
Diagnosis
1
ID
Date
Appointment
N
N
Assists
Description
ID
Email
Doctor
N
1
Assists
Name
Surname
Specialty

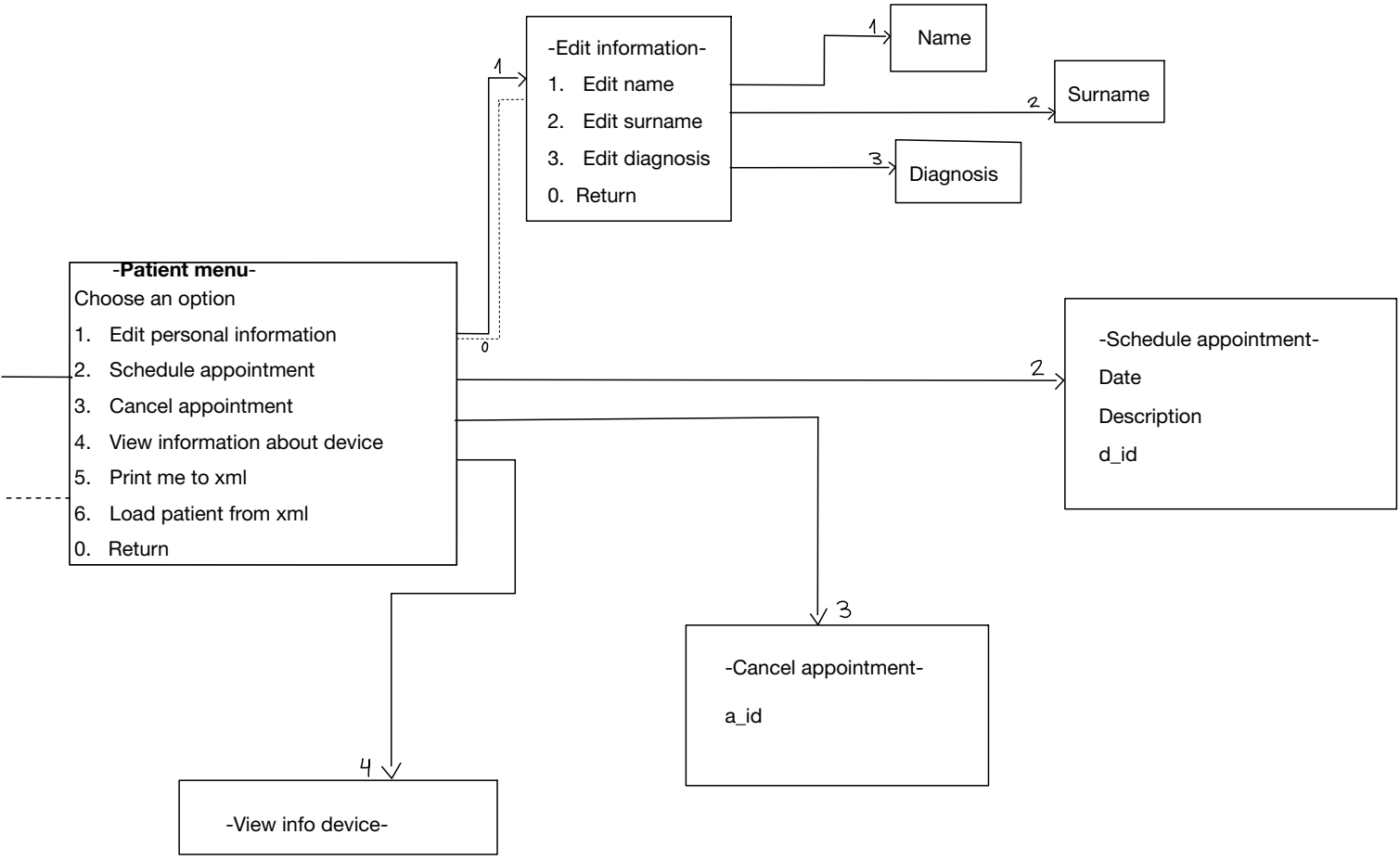

Device
ID*
Type
Manufacturer_ID (FK)
Manufacturer
ID*
Name
Address
Phone number
Device_Patient
(Device_ID, Patient_ID)*
Device_ID (FK)
Patient_ID (FK)
Patient
ID*
Email
Name
Surname
Birthday
Diagnosis
Doctor _ID (FK)
Orders
(Device_ID, Doctor_ID)*
Device_ID (FK)
Doctor_ID (FK)
Doctor
ID*
email
Name
Surname
Specialty
Appointment
ID*
Date
Description
Patient_ID (FK)
Doctor_ID (FK)
M
1
M
M
M
1
M
1
M
M
1
M

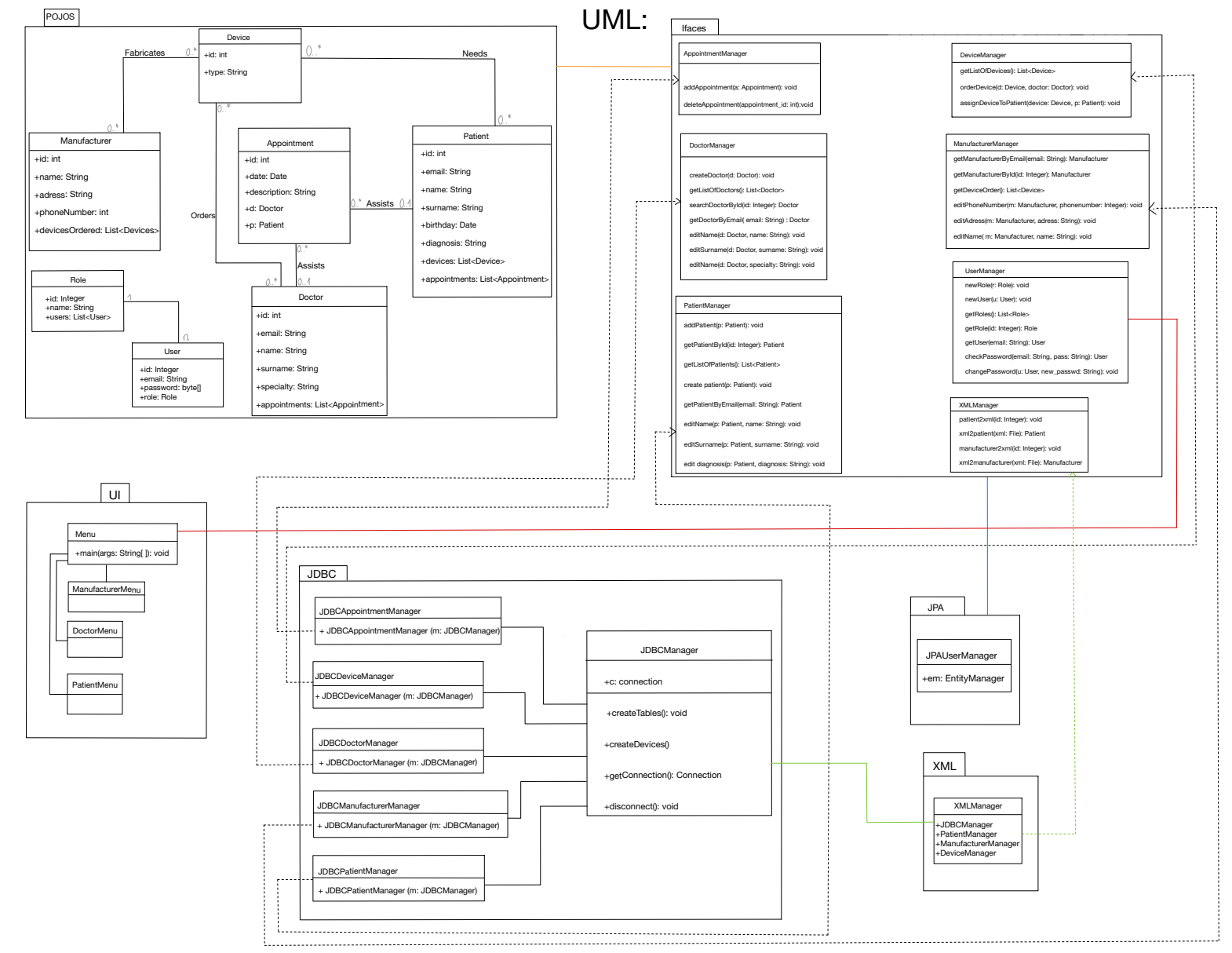
MOCKUP:


If doctor
If manufacturer
If patient
-login-
Email
Password
1
Choose an option
1. Login User
2. Sign-up new user
3. Update the password of an existing user
4. Delete
0. Exit
2
-sign up user-
Email
Password
Rol
3
-update password-
Email
Password
New_password


-Edit information-
1. Edit name
2. Edit surname
3. Edit specialty
0. Return
Name
Surname
Specialty
-Doctor menu-
1. Edit personal information
2. Schedule ap pointment
3. Cancel appointment
4. Order device
5. Assign device to patient
6. View information about a patient
0. Return
-Add patient-
Email
Name
Surname
Birthday
Diagnosis
p_id
-Schedule appointment-
Date
Description
p_id
-Order device-
Type
-Cancel appointment-
a_id
-Assign device-
p_id
Type (device)
View information patient-
1. Choose patient
0. Return
p_id


-Edit information-
Choose an option
1. Edit name
2. Edit address
3. Edit phone number
0. Return
-edit name-
Name
-edit adress-
Adress
-edit phone number -
Phonenumber
-Manufacturer Menu-
Choose an option
1. Edit personal information
2. View devices needed
3. Print me to xml
4. Load manufacturer from xml
0. Return
-View devices-
1
0
2
3


-Edit information-
1. Edit name
2. Edit surname
3. Edit diagnosis
0. Return
1
Name
2
Surname
3
Diagnosis
1
0
-Patient menu-
Choose an option
1. Edit personal information
2. Schedule appointment
3. Cancel appointment
4. View information about device
5. Print me to xml
6. Load patient from xml
0. Return
2
-Schedule appointment-
Date
Description
d_id
3
-Cancel appointment-
a_id
4
-View info device-


UML:
POJOS
Device
+id: int
+type: String
Fabricates
Needs
Manufacturer
+id: int
+name: String
+adress: String
+phoneNumber: int
+devicesOrdered: List<Devices>
Appointment
+id: int
+date: Date
+description: String
+d: Doctor
+p: Patient
Orders
Assists
Patient
+id: int
+email: String
+name: String
+surname: String
+birthday: Date
+diagnosis: String
+devices: List<Device>
+appointments: List<Appointment>
Assists
Role
+id: Integer
+name: String
+users: List<User>
User
+id: Integer
+email: String
+password: byte[]
+role: Role
Doctor
+id: int
+email: String
+name: String
+surname: String
+specialty: String
+appointments: List<Appointment>
Ifaces
AppointmentManager
addAppointment(a: Appointment): void
deleteAppointment(appointment_id: int):void
DeviceManager
getListOfDevices(): List<Device>
orderDevice(d: Device, doctor: Doctor): void
assignDeviceToPatient(device: Device, p: Patient): void
DoctorManager
createDoctor(d: Doctor): void
getListOfDoctors(): List<Doctor>
searchDoctorById(id: Integer): Doctor
getDoctorByEmail(email: String) : Doctor
editName(d: Doctor, name: String): void
editSurname(d: Doctor, surname: String): void
editName(d: Doctor, specialty: String): void
ManufacturerManager
getManufacturerByEmail(email: String): Manufacturer
getManufacturerById(id: Integer): Manufacturer
getDeviceOrder(): List<Device>
editPhoneNumber(m: Manufacturer, phonenumber: Integer): void
editAdress(m: Manufacturer, adress: String): void
editName(m: Manufacturer, name: String): void
PatientManager
addPatient(p: Patient): void
getPatientById(id: Integer): Patient
getListOfPatients(): List<Patient>
create patient(p: Patient): void
getPatientByEmail(email: String): Patient
editName(p: Patient, name: String): void
editSurname(p: Patient, surname: String): void
edit diagnosis(p: Patient, diagnosis: String): void
UserManager
newRole(r: Role): void
newUser(u: User): void
getRoles(): List<Role>
getRole(id: Integer): Role
getUser(email: String): User
checkPassword(email: String, pass: String): User
changePassword(u: User, new_passwd: String): void
XMLManager
patient2xml(id: Integer): void
xml2patient(xml: File): Patient
manufacturer2xml(id: Integer): void
xml2manufacturer(xml: File): Manufacturer
UI
Menu
+main(args: String[ ]): void
ManufacturerMenu
DoctorMenu
PatientMenu
JDBC
JDBCAppointmentManager
+ JDBCAppointmentManager (m: JDBCManager)
JDBCDeviceManager
+ JDBCDeviceManager (m: JDBCManager)
JDBCDoctorManager
+ JDBCDoctorManager (m: JDBCManager)
JDBCManufacturerManager
+ JDBCManufacturerManager (m: JDBCManager)
JDBCPatientManager
+ JDBCPatientManager (m: JDBCManager)
JDBCManager
+c: connection
+createTables(): void
+createDevices()
+getConnection(): Connection
+disconnect(): void
JPA
JPAUserManager
+em: EntityManager
XML
XMLManager
+JDBCManager
+PatientManager
+ManufacturerManager
+DeviceManager