

DEFINITION:

Management of organ donation in Spain.

FEATURES:

Our app is going to connect organ donors and organ receptors. Medical staff will be able to upload patients (donors and receptors) to the data base, it's job will be to match the receptors and donors by specific parameters. Donors will also be able to register before they are in the situation of donating.

THINGS TAKEN INTO ACCOUNT

In our application, we are assuming the following points:

- A receptor can **only receive one organ** (one request)
- A person **cannot be a donor and a receptor at the same time** (can be changed later)
- We **only delete donors** (in the case they are alive, and they do not want to donate their organs anymore)
- Donors can register in the database whenever they want, they **must** insert his dni, dob and organs (type of organ). However, their **organs** can only be donated **when** they have **passed away**. More than one organ can be donated from the same person.
- The **doctor** can only register a **donor** it alive is **FALSE**
- The people that can enter in the database and be considered as a **user** are **donor and doctor**.
- When a donor registers as a user, the **alive** attribute will be by default true. Also the attribute available of the organ is by default false.
- A **donor** can only see his data and insert his dni, dob and organs (type of organ).
- **Only** the **doctor** will be able to introduce the rest of the **info** in the donor.
- The application provides a match between the organs from a donor and receptor.
- The **tables that are constant** are type_organ
- **For matching**, we are creating a list with two conditions to be on that list:
 - Needing that organ
 - Being the same blood type (not true in real life)
- After producing the list, we calculate a mean of several factors that influence organ donation. Those factors are:
 - Compatibility (antigens and antibodies).
 - Urgency of the operation.
- The **match function is triggered** each time a donor (not alive) or a receptor is uploaded to the data base.
- Once the **match is done**, the organ is directly assigned to the receptor. To do that, in the table of organs, available = FALSE and in the table of request, received = TRUE
- Things that **cannot be changed**:
 - Receptor: blood type
- Our JAXB classes are organ and request
- Match function:

- Donor and receptor need to have the same bloodtype, antigens (a,b, dq), antibodies, id del type organ, available true

ENTITIES:

RECEPTOR:

- DNI
- DOB
- Status (accepted, rejected, waiting, operating)
- Blood type
- Alive
- Urgency (1: least urgent; 5: most urgent)
- Antigens (A, B, C, DP, DQ, DR)
- Antibodies (anti I, anti II)
- Location (coordinates)
- Request (organ needed)

DONOR:

- DNI
- DOB
- Alive
- Blood type
- Antigens
- Antibodies
- Location (coordinates)
- Doctor in charge
- Organs (donated)

TYPE OF ORGAN: lung, kidney, heart, liver, pancreas, bowel, bone marrow.

- ID
- Lifespan (hours):
 - Kidney: 24-50
 - Liver: 17-30
 - Pancreas: 17-30
 - Lung: 3-8
 - Heart: 3-8
 - Bowel: 6-12

- Bone marrow:

ORGAN:

- ID
- Size
- Available

REQUEST

- ID
- Size
- Received
- Organ_ID

DOCTOR:

- Medical ID
- Phone number
- Name

LOCATION:

- Latitude.
- Longitude.

ANTIGEN:

- A.
- B.
- C.
- DP.
- DQ.
- DR.

ANTIBODY:

- Class I.
- Class II.

USE CASES:

USE CASE 0:

Actor: doctor.

Goal: register doctor.

Path: Medical ID, phone number, Name.

USECASE 1

Actor: doctor

Goal: register receptor

Path: input: id, blood type, antigen, antibodies, organ needed, DOB, location, doctor in charge (can be NULL), status (after operation), state and organ received.

USECASE 2

Actor: doctor/donor

Goal: register donor.

Path: input: id, blood type, antigen, antibodies, organs donated, DOB, location, doctor in charge (can be NULL), state.

USECASE 3

Actor: doctor.

Goal: match.

Path: Mathematical operations.

USECASE 4

Actor: doctor.

Goal: retrieve the organ.

Path: transportation.

Triggers: UC5.

USECASE 5

Actor: doctor.

Goal: transplant the organ.

Path: operation (medical).

USECASE 6

Precondition: operation (medical) completed.

Actor: doctor.

Goal: report status and state.

Path: update status and state.

USECASE 7

Precondition: alive and status = accepted.

Actor: receptor.

Goal: go home.

Path: discharge the receptor.

REQUIREMENTS LIST

Functional requirements:

- 1.-Must allow to register as a receptor.
- 2.-Must allow to register and remove as a donor.
- 3.-Must allow to register as a doctor.
- 5.-Must allow to update the status of the receptor (after operation).
- 6.-Calculate the compatibility between receptors and donors.
- 7.-Take into consideration the capabilities of the receptor (do the list) (position in the list, location and distance between them...).
- 8.-Must allow the doctor to change the state of the receptor.
- 9.- Must allow the doctor to choose his way of transportation.

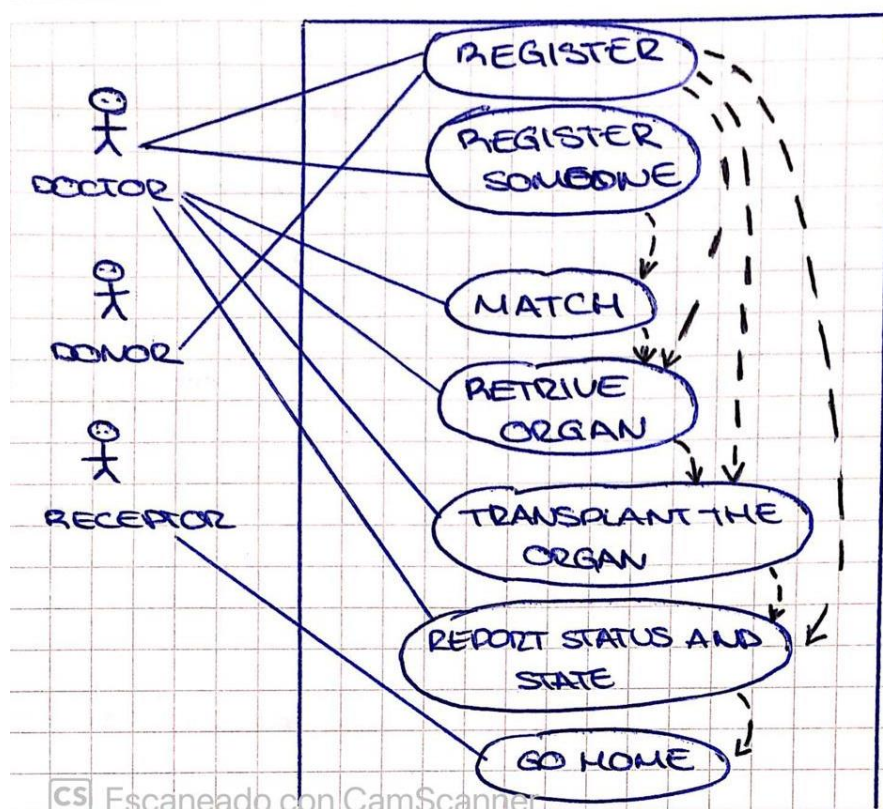
Nonfunctional requirements:

- 1.-Application must support all the doctors from the hospitals in Spain
- 2.-Must be able to run in a laptop computer for the use of doctors and donors that want to register.
- 3.-Doctor must be able to travel by boat, train, helicopter or car.

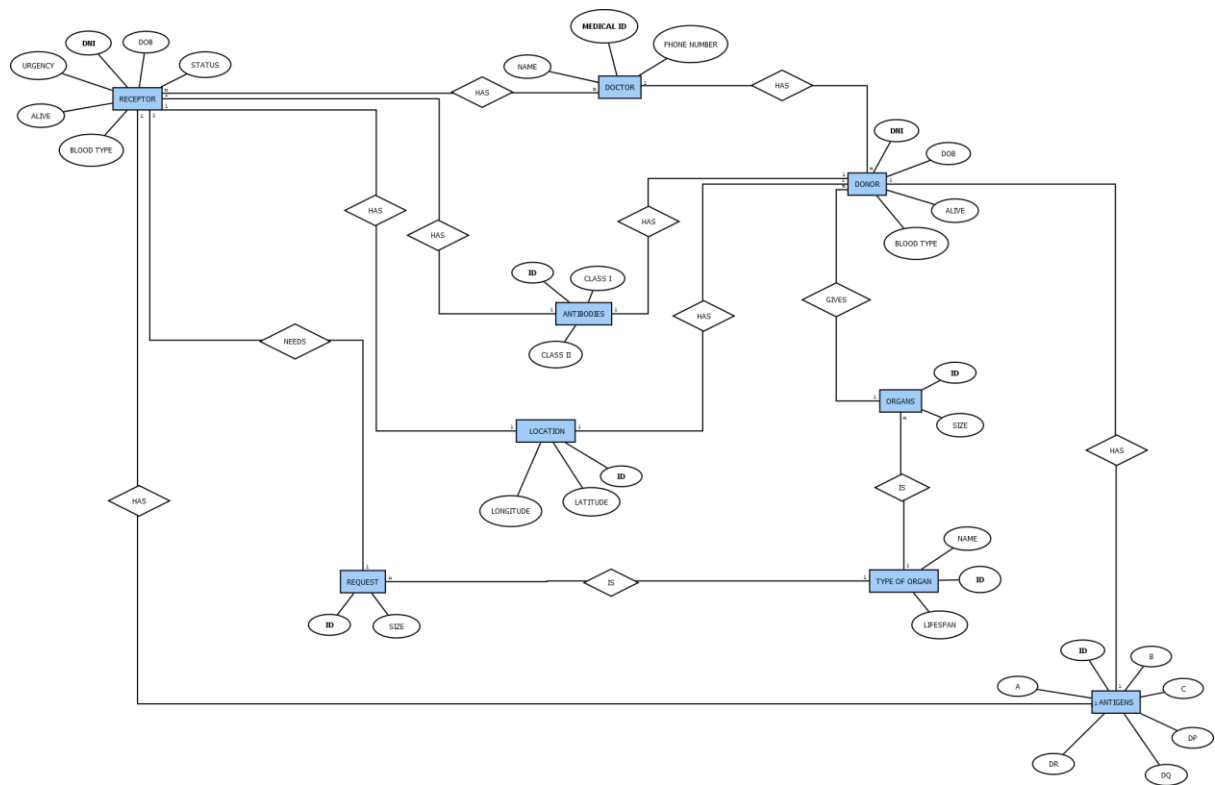
REQUIREMENTS TABLE

	FR1	FR2	FR3	FR5	FR6	FR7	FR8	FR9	NF1	NF2	NF3
UC0			x				x		x	x	
UC1	x								x	x	
UC2		x							x	x	
UC3					x	x	x		x	x	
UC4							x	x	x	x	x
UC5							x		x	x	
UC6				x			x		x	x	
UC7							x		x	x	

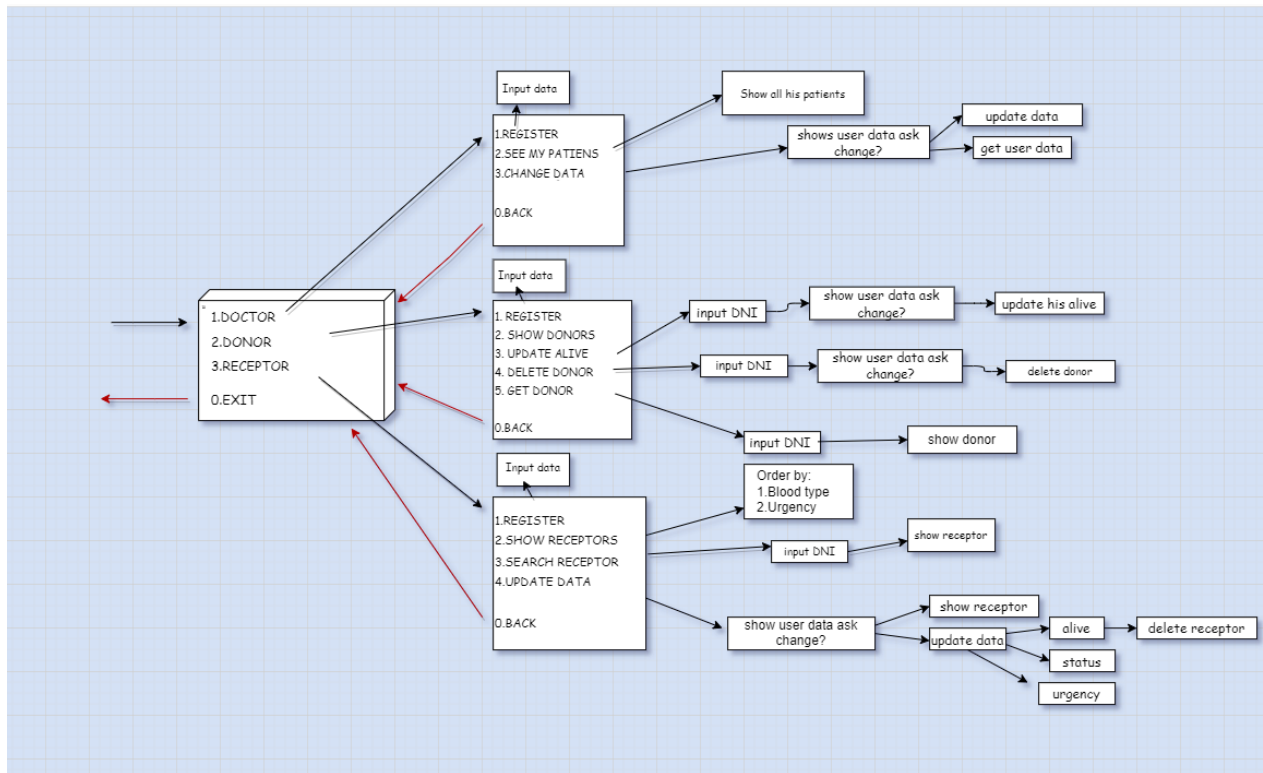
UML USE CASE DIAGRAM



E-R DIAGRAM



MOCK-UP



SQL QUERIES

Remember to change the concrete values to ? (using PreparedStatement)

OPTION 1 : DOCTOR

Register Doctor: INSERT INTO doctor (medical_id, phone_number, name) VALUES (1234, 678678678, "Hulio Firulais")

See my patients: SELECT * FROM receptor WHERE id_doctor_charge = "1234"

Change data of doctor:

- **Show doctor:** SELECT * FROM doctor WHERE medical_id = "1234"
- 1. **Update doctor:** UPDATE doctor SET phone_number = "654432432", name = "Dr. Hulio Firulais" WHERE medical_id = "1234"

OPTION 2: DONOR

Register donor: INSERT INTO donor (DNI, DOB, blood_type, alive, id_antigens, id_antibodies, id_doctor_charge, id_location, id_organs)
VALUES ("375", "2002-02-02", "B+", TRUE, "1", "1", "1234", "1", "1")

Show donors: SELECT * FROM donor

Update alive: UPDATE donor SET alive=FALSE WHERE DNI = "375"

Search donor:

- **By blood type:** SELECT * FROM donor WHERE blood_type = "A+"

Get donor by DNI: SELECT * FROM donor WHERE DNI = "3"

Delete donors: DELETE FROM donor WHERE DNI = "4"

OPTION 3: RECEPTOR

Register receptor: INSERT INTO receptor (DNI, DOB, blood_type, status, alive, urgency, id_antigens, id_antibodies, id_doctor_charge, id_location, id_request)
VALUES ("1", "2002-02-02", "A+", "Waiting", TRUE, "1", "1", "1", "1234", "1", "1");

Show receptors: SELECT * FROM receptor

Change data: (alive, status, urgency)

- **Show receptor:** SELECT * FROM receptor WHERE DNI = "2";
- **Update receptor:** UPDATE receptor SET alive = FALSE, status = "Operating", urgency = "2" WHERE DNI = "3";

Search receptor:

- **By blood type:** SELECT * FROM receptor WHERE blood_type = "A+"
- **By urgency:** SELECT * FROM receptor ORDER BY urgency DESC

Get receptor by DNI: SELECT * FROM receptor WHERE DNI = "3"

OTHER QUERIES NEEDED:

Insert antigens: INSERT INTO antigen (A, B, C, DP, DQ, DR) VALUES (FALSE, TRUE, TRUE, FALSE, TRUE, FALSE)

Insert location: INSERT INTO location (latitude, longitude) VALUES (12, 2)

Delete location: DELETE FROM location WHERE id= "4"

Insert antibodies: INSERT INTO antibody (class_I, class_II) VALUES (TRUE, TRUE)

Insert organs: INSERT INTO organ (size_organ, id_type_organ) VALUES (12, 2)

Insert request: INSERT INTO request (size_organ, id_type_organ) VALUES (12, 2)

Insert type of organ:

INSERT INTO type_of_organ (name, lifespan) VALUES ("kidney", 50)

INSERT INTO type_of_organ (name, lifespan) VALUES ("liver", 30)

INSERT INTO type_of_organ (name, lifespan) VALUES ("pancreas", 30)

INSERT INTO type_of_organ (name, lifespan) VALUES ("lung", 8)

INSERT INTO type_of_organ (name, lifespan) VALUES ("heart", 8)

INSERT INTO type_of_organ (name, lifespan) VALUES ("bowel", 12)

INSERT INTO type_of_organ (name, lifespan) VALUES ("bone marrow", 8)

Search donor by alive (dead): SELECT * FROM donor WHERE alive = FALSE

Update Received (in request): UPDATE request SET received = TRUE WHERE DNI = "3";

Update donor ID: UPDATE request SET DNI_donor = "5";