INFORMATYKA
Akademia Górniczo-Hutnicza w Krakowie

Deusto
Universidad de Deusto
University of Deusto

ERCİYES ÜNİVERSİTESİ
1978

AGH

SOFTWARE ENGINEERING

MEDICAL CLINIC

Software System to Support a Doctor

AUTHORS:
Iñaki Seco ; Sefa Kurtoğlu

PROFESSORS:
Tadeusz Dyduch ; Małgorzata Żabińska-Rakoczy

25/01/2018

Table of Contents

1. Project description

1.1 Area and subject of modeling

1.1.1 Problem domain

The company is a medical organization composed of many different employees and created to reach every person that needs medical assistance, so that is the kind of people that are going to be interested on the project.

The software is going to be used in a medical clinic, helping both employees and clients, so for example the patients will be able to ask for a meeting with the doctor when they need medical assistance and doctors will send messages and prescriptions back to the patients.

1.2 Modeling area

1.2.1 Description of organization structure

Thanks to our modelling project, the organization of any medical clinic will be extremely easy to administrate. In these systems are going to take part not only the doctors and nurses but also the patients and providers of each office. The

The employees will be able to modified things such important like medical records of theirs patients or diagnostics from the appointments, as well as others less vital stuffs like, for example, holidays and vacations. In addition to that, patients will have the choice to make online appointments in a really simple way.

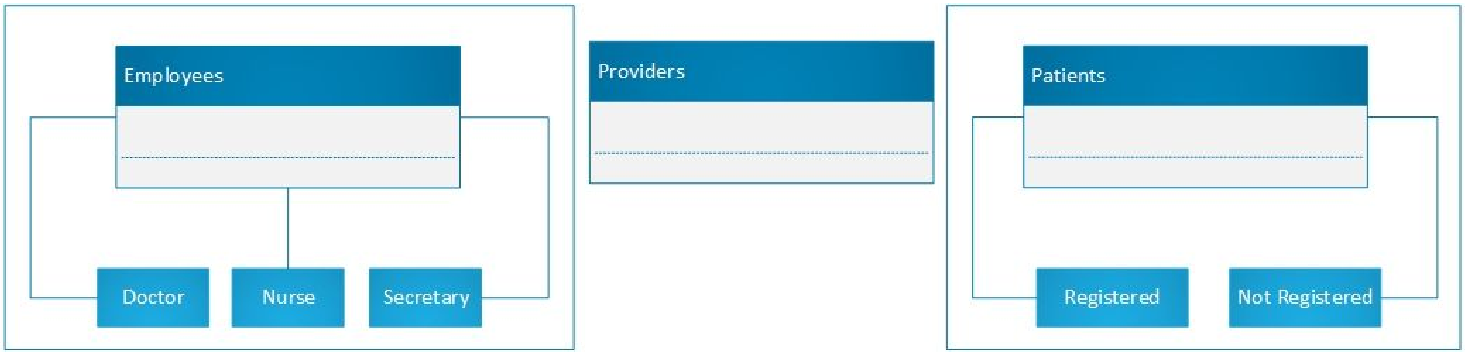


Figure 1 - Structure of Actors

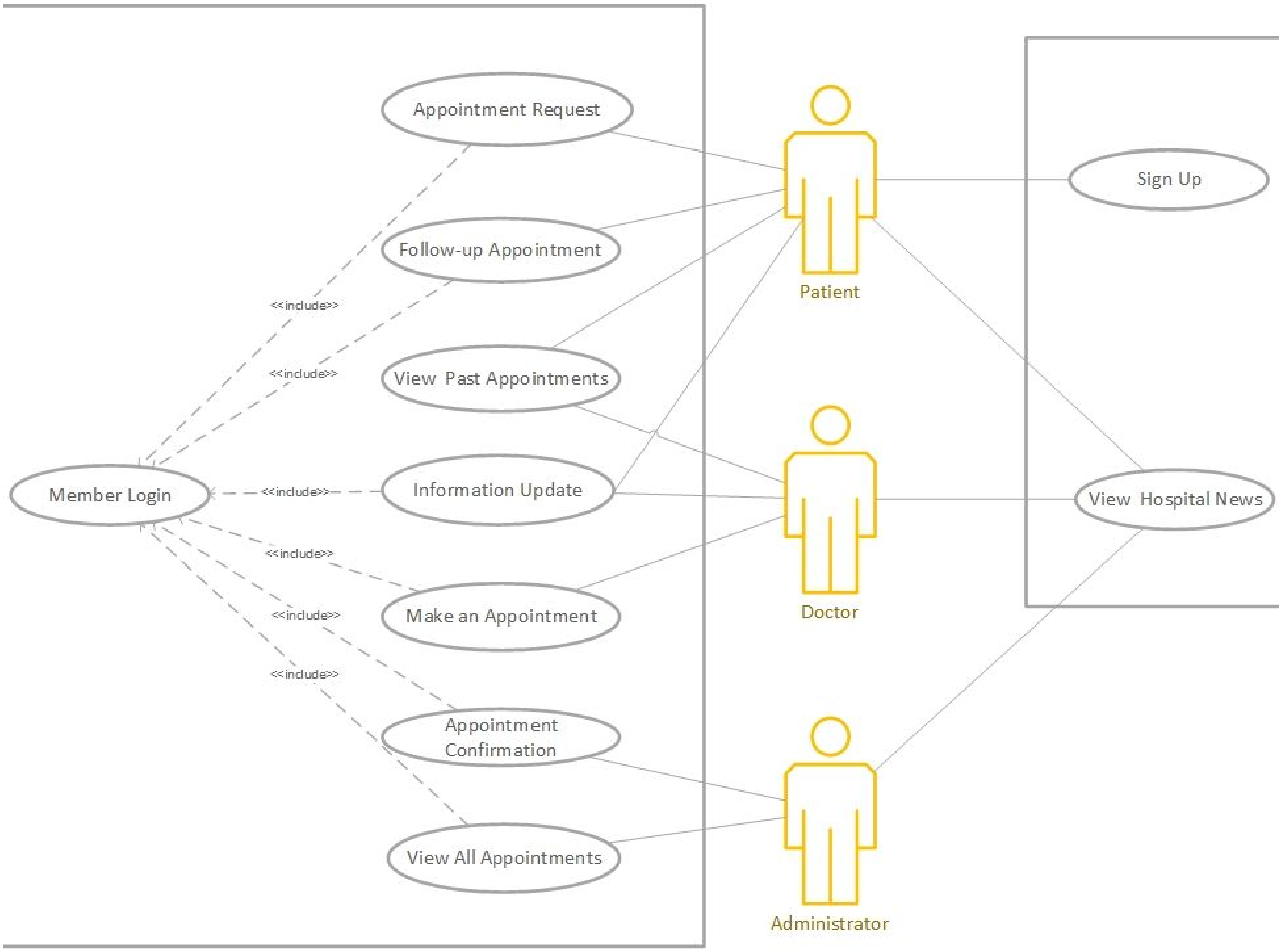


Figure 2 - Global Vision Use Case

1.2.2 Activity Areas

Each clinic will content the following areas separately:

- Administrative area
- Medical area
- Ordering area
- Human resources

1.3 Activity areas’ description

1.3.1 Posts description

- Administrative area:
  This area concerns everything what has to do with calendars, payments and identities or information about every individual interact with the clinic. Legal documentations and signed contracts from each person who has work or gain from those services.

- Medical area
  The medical records and analyzes belongs to this area, as well as treatments followed by the patients and doctors. This information is quite sensitive due to its delicacy, so it will be especially good protected.

- Ordering area
  Here we will include purchases and sales of different products which are necessary for the company. They are going to be mainly medicines and hygiene items for the maintenance of the office and services.

- Human resources
  This area will be dedicated to manage the environment between the employees, additionally, they handle all new employees and other kind of internal problems.

1.3.2 Business procedures description

- Administrative area:
  The employees in this area participate in a lot of business deals, they have to manage and store every contract done with a client, management of documents and information of the doctors.

- Medical area
  There have to be a very good and special business deal on this area with the security of data enterprises, cause here is where the medical data is managed.

- Ordering area
  This area is working directly with the providers, making deals and orders every week to have the necessary materials and medicaments for the clinic. Once they have closed a deal they must send the pertinent documents to the administrative area.

- Human resources
  Human resources negotiate with the employees their contracts and new salaries, and also are the ones in charge of the new hires and layoffs.

1.4 Verified system's responsibilities

The system must guarantee a fast and good communication between the company and the clients, because patients must need a medical response quickly in case they have a serious health problem.

Also, it must store the medical data of the employees and of the registered clients, but with a very good security system to protect the people personal and medical information and prevent legal problems.

1.5 Short problems

One of the biggest problems the system will have to handle with is the unpredictable success of itself, so we will have to work and improve the system in case we have a lot of demand. For example, if we have a lot of registered clients we must store a huge amount of data and also have a better security system.

Another problem could be the fact that older people are less used to using new technologies, so we should implement in a way that everyone find it easy. Also, we could make it more international offering a few options for the language of the system and the communication with the employees.

1.6 Goals:

1.6.1 Product's aims

This product aims are to improve communication between hospital and patients, it will help to direct referrals to other hospital. On the other hand, the product is going to be beneficial for the shipment of medicines and materials.

1.6.2 Design aims

This application is calculated specially to help older people, the design includes a small messaging application so these people will be able to communicate immediately without the needed of going to the hospital.

2. Requirements description

2.1 List of functions from the user's viewpoint

- Patients
  - Make an appointment with the doctor using the web page or calling to the reception
  - View their medical records online
- Employees
  - Manage their contract
  - Manage holidays/free days
  - Order products needed for the clinic
- Doctors
  - View medical records from their patients
  - Make a diagnosis
  - Add treatments to a patient
- Secretary
  - Add appointments
  - Manage medical records
- Nurse
  - Add treatments (limited)
  - Keep medical records (limited)
- Providers
  - Check their orders

2.2 Data to be stored in the system

In the system we will have information about all the people who is related to our clinic, as appointments, diagnosis, but also payments and salaries. To be more concrete:

- Doctors
  - Name
  - ID
  - Address
  - Phone number
  - Email address
  - Bank account
  - Status of payment
  - Department number
  - Timetable
  - Calendar

- Nurse
  - Name

- ID
- Address
- Phone number
- Email address
- Bank account
- Status of payment
- Department number
- Timetable

- Secretary
  - Name
  - ID
  - Address
  - Phone number
  - Email address
  - Bank account
  - Status of payment
  - Timetable

- Providers
  - Name
  - ID
  - Address
  - Phone number
  - Email address
  - Department number
  - Type of product
  - List of activities
  - Bank account

- Registered patients
  - Name
  - ID
  - Visits history
  - Address
  - Phone number
  - Email address
  - List of activities
  - Status of subscription payment
  - Dates of appointment

- Not registered patients
  - Name
  - Address

        - Visits history
        - Phone number
        - Email address
        - Dates of appointment

2.3 Input and output documents

- Registration file

  Registration will be an input information. When a new patient wants to subscribe to our medical clinic he has to fill a profile document about himself and his personal information such as full name, home address, phone number, bank account, born date and medical record. in order to the secretary can add all this information to the system.

- Doctor’s diagnostics

  Doctor’s diagnostics are output documents, these documents include some informations about the patient’s situation. All this diagnostics will be input information for patient’s treatments

- Patient’s treatments

  Treatments will be output information as well as input because it would be the doctor's conclusion about the treatment to be followed by a patient, but also would be the document which the respective patient has to check to be able to follow it.

- Calendar

  Here we have both type of documents as well. First of all the patient has to be able to see when the doctor is free to get an appointment, this would be output information. But then the patient has to fill a form about the appointment date, reasons and so on, this would be an input document.

2.4 Required special requirements and limitations

The most important thing to consider in our system is to have knowledge about how sensitive are the documents we deal with, in fact, we are talking about completely secret information about people.

That means we need a really secure system which doesn’t let anybody go in without permission. The connections will be especially secure when they talk about diagnostics or reports from the doctors, and the login will be totally controlled, modifying passwords every three months or adding second steps as security improvement.

2.5 Functional requirements' analysis

The system consists of 3 modules: Patients Module, Doctors Module, Administrative Module. Each module has its own use case diagram and scenarios

2.5.1 Patients Module

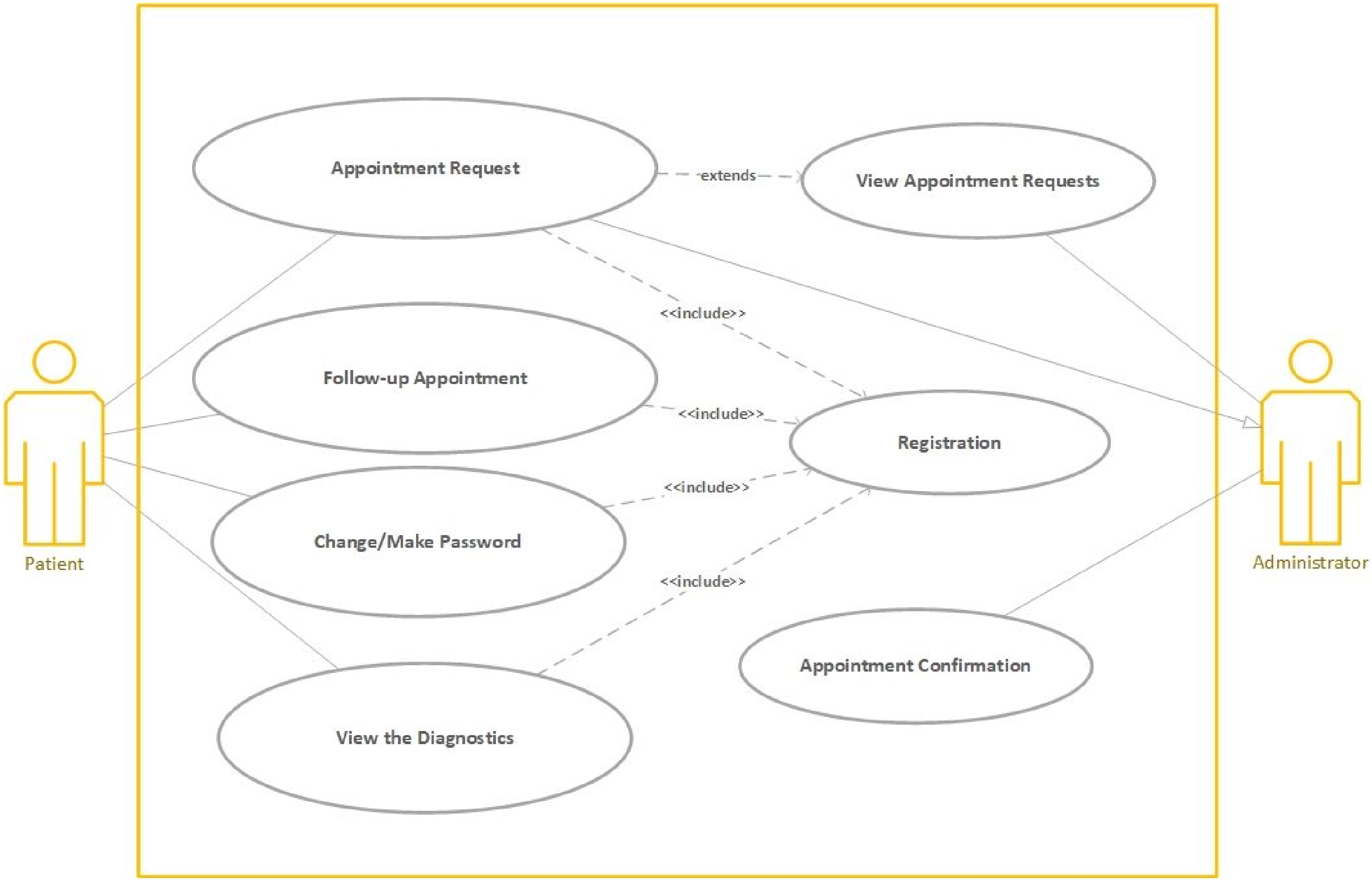


Figure 3 - Patients Movements Diagram

2.5.1.1 Patients Scenario with Cockburn’s Pattern

Number (id) and the name of the Use Case: UC-01.- Patient Use Case

Author: Iñaki and Sefa

Objective of the Use Case: Understand and have a better compression of our system functionality, in this concretely case, how patients work.

Context (business goal of the Use Case): A patient would like to know something about his treatments or reports, or want to use the medical clinic services.

Range (project range, ie. which system is a „black box”):

Level: (business level, function, subfunction with implementation details): function like request for an appointment

Main Actor: Patient

Participants (other Actors): Administrator

Trigger (Initiating event – what starts the Use Case): First step is the intention to do an appointment to go to see the doctor.

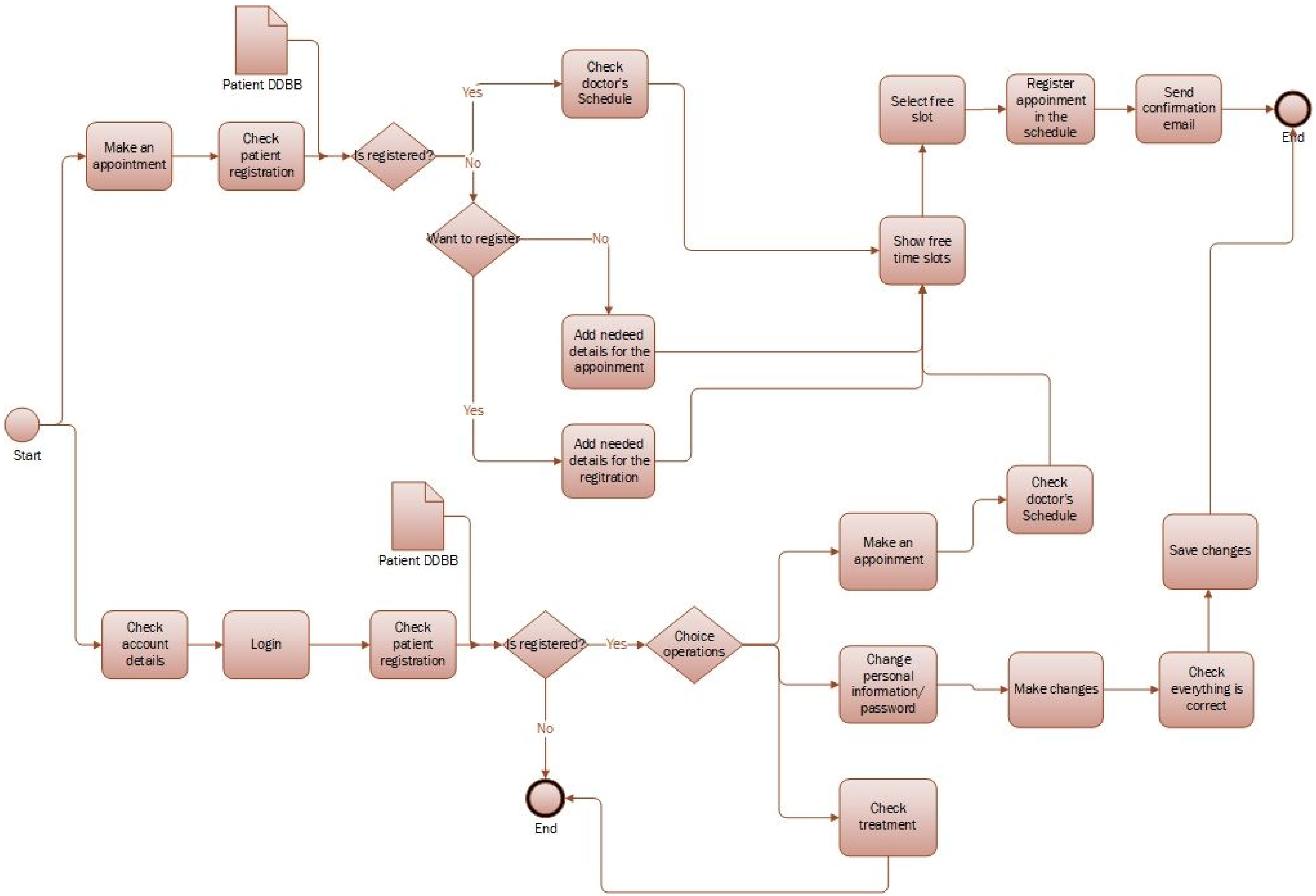
Initial condition (what state of the world we expect at the beginning): Patient is not registered, then he or she will have to register herselves or provide some specific information.

Minimal guarantee (how interests are protected at any ending, even not successful): The patients is doing an appointment and the internet conections fels down.

Main scenario of the success (3-7 points in the form: 1. Client adds the product to the basket; 2. System successfully verifies availability of the product, etc.):

- Make sure the patient is registered
- Patient make an appointment request
- System verify that the doctor is available
- Add appointment to the doctor’s calendar

Extensions - Alternative scenarios - extensions to main scenario points: <<Extend>>: Cancel the appointment, the patient has to be able to cancel an appointment whenever he need it.


Start
Make an appointment
Check patient registration
Patient DDBB
Is registered?
Yes
No
Check doctor's Schedule
Want to register
No
Yes
Add nedeed details for the appoinment
Add needed details for the regitration
Show free time slots
Select free slot
Register appoinment in the schedule
Send confirmation email
End
Check account details
Login
Check patient registration
Patient DDBB
Is registered?
Yes
No
End
Choice operations
Make an appoinment
Check doctor's Schedule
Change personal information/ password
Make changes
Check everything is correct
Save changes
Check treatment

2.5.2 Doctors Module

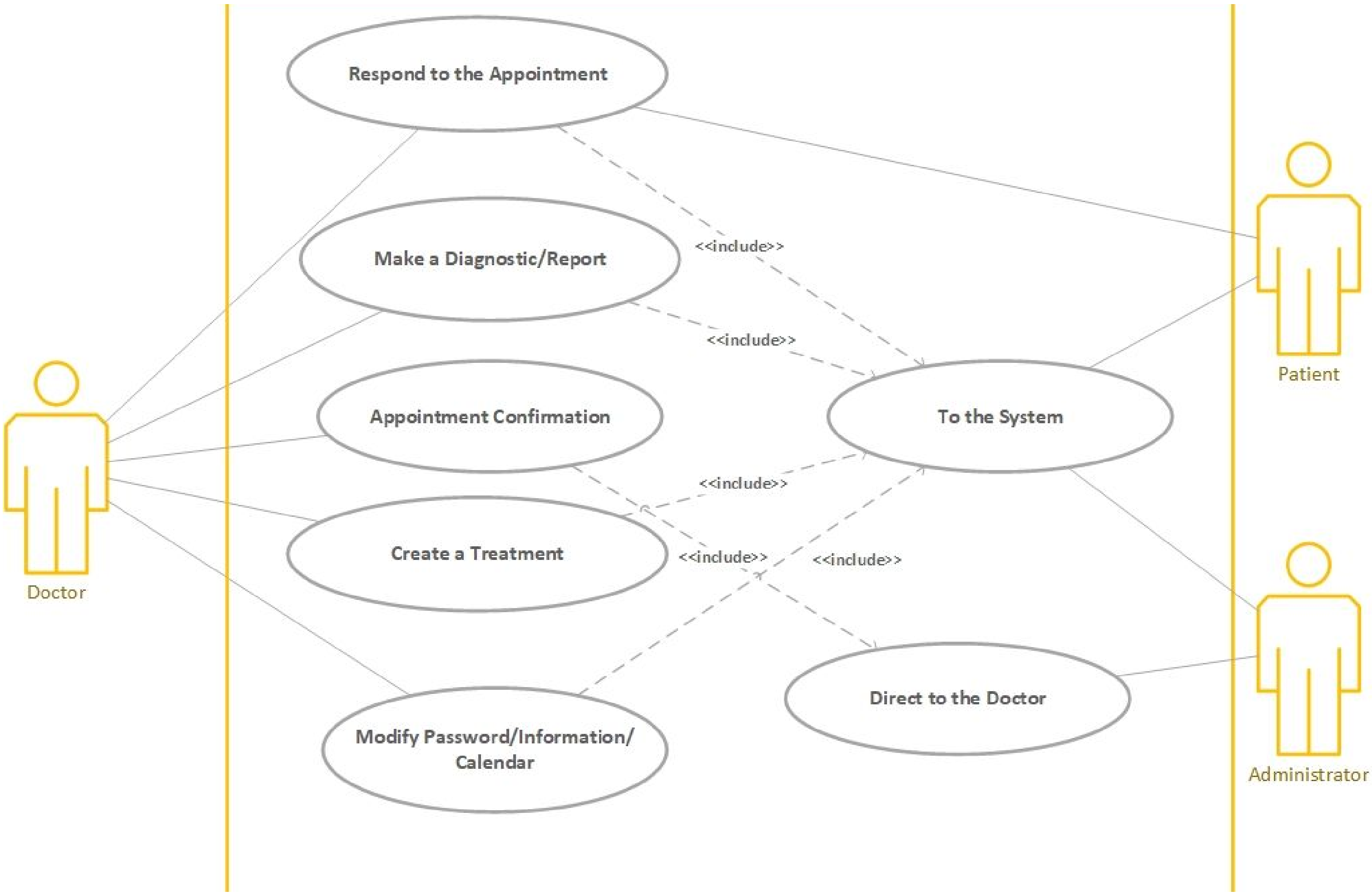


Figure 4 - Doctors Movements Diagram

2.5.2.1 Doctors Use Case Scenario with Cockburn's Pattern

Number (id) and the name of the Use Case: UC-02.- Patient Use Case

Author: Iñaki and Sefa

Objective of the Use Case: Understand and have a better compression of our system functionality, in this concretely case, how doctors work.

Context (business goal of the Use Case): During an appointment, the patient explains him all of his difficulties and problems, then the doctor has to redact an diagnosis and upload it to the medical record of the patient.

Range (project range, ie. which system is a „black box"):

Level: (business level, function, subfunction with implementation details): function, like make a report or design a treatment

Main Actor: Doctor

Participants (other Actors): Patient and Administrator

Trigger (Initiating event – what starts the Use Case): Redact conclusion of each visit and design a treatment for the patient.

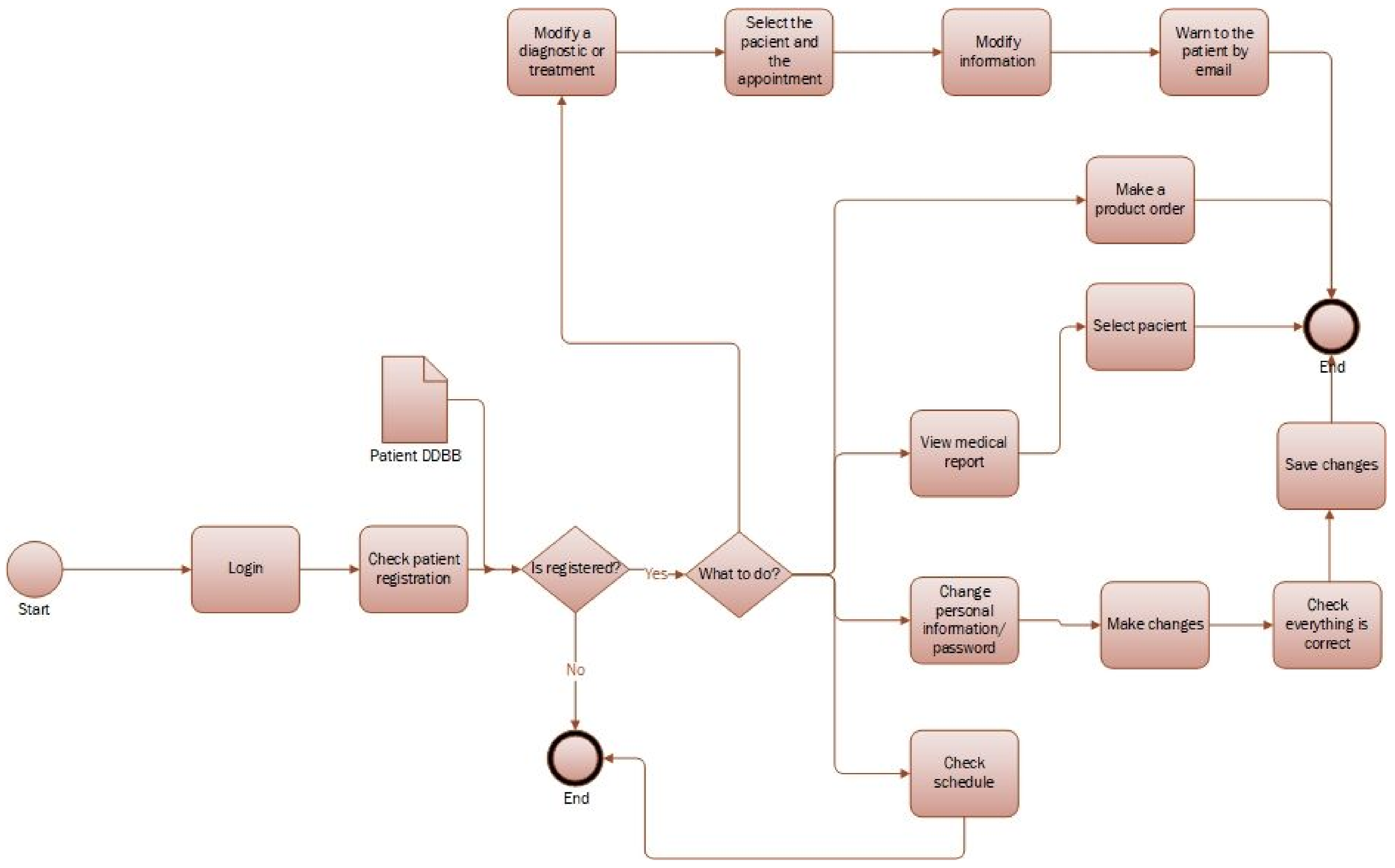
Initial condition (what state of the world we expect at the beginning): To have an appointment with any patient.

Minimal guarantee (how interests are protected at any ending, even not successful):

Main scenario of the success (3-7 points in the form: 1. Client adds the product to the basket; 2. System successfully verifies availability of the product, etc.):

- After the patient's appointment is verified and added, this has to come to the doctor’s clinic to explain which problem do he has.
- The doctor get conclusions and make a diagnosis of the appointment.
- Design a treatment and tells him.
- Add the diagnosis and the treatment to the patient’s medical record.

Extensions - Alternative scenarios - extensions to main scenario points: <<Extend>>: Be able to see the medical record of any of his patients, so that he could have as much informations as he can to make a decision.


Start
Login
Check patient registration
Patient DDBB
Is registered?
Yes
No
End
What to do?
Modify a diagnostic or treatment
Select the pacient and the appointment
Modify information
Warn to the patient by email
Make a product order
Select pacient
View medical report
Change personal information/ password
Make changes
Check everything is correct
Save changes
End
Check schedule

2.5.3 Administrative Module

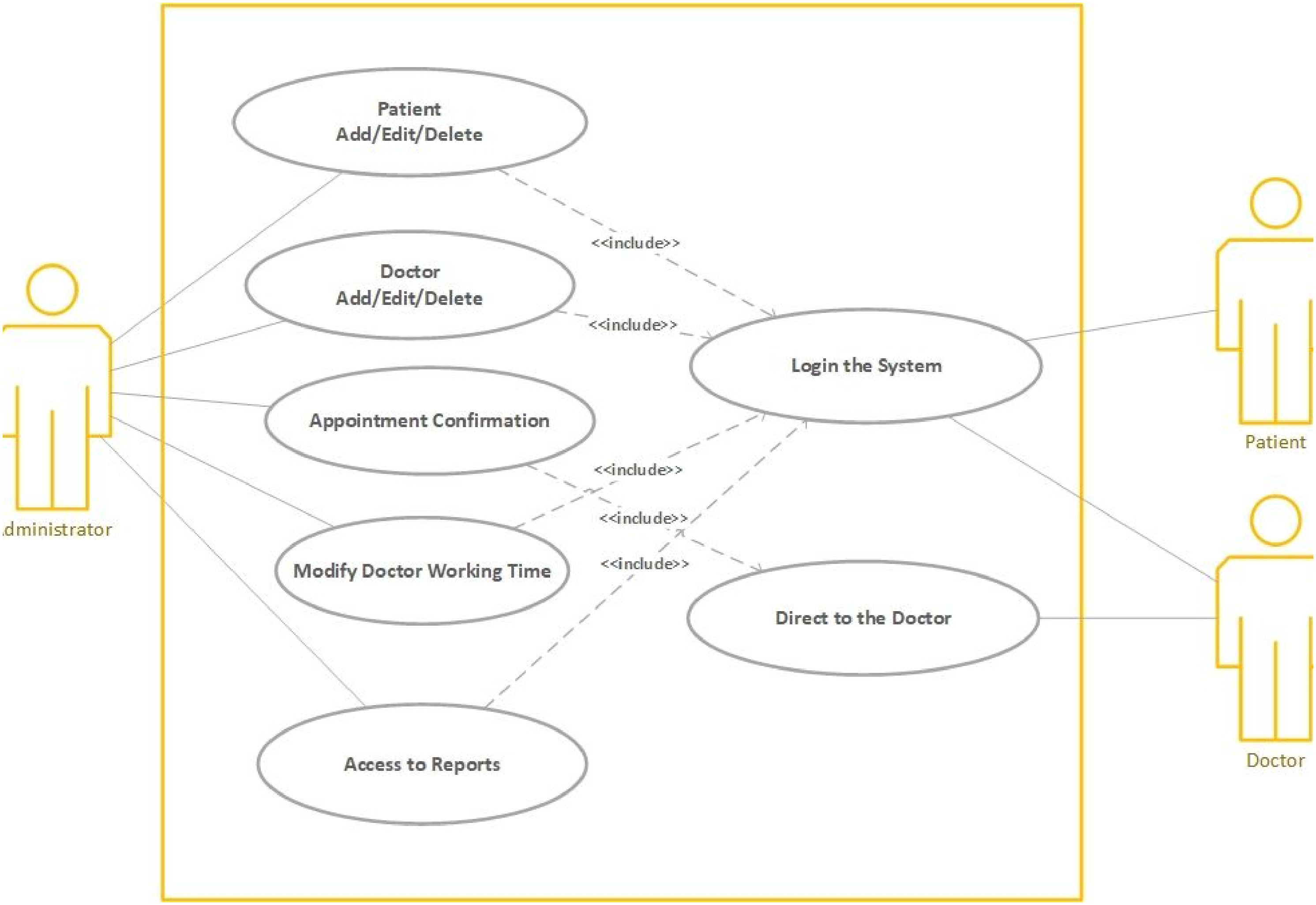


Figure 5 - Administrator Movements Diagram

2.5.3.1 Administrators Use Case Scenario with Cockburn's Pattern

Number (id) and the name of the Use Case: UC-3.- Administrator Use Case

Author: Iñaki and Sefa

Objective of the Use Case: Understand and have a better compression of our system functionality, in this concretely case, how administrator's work

Context (business goal of the Use Case): The patient's and doctor's schedule need to be modified and uploaded

Range (project range, ie. which system is a „black box"):

Level: (business level, function, subfunction with implementation details): function like add patients or access to reports.

Main Actor: Administrator

Participants (other Actors): Patient and doctor

Trigger (Initiating event – what starts the Use Case): Receive an application for an appointment from any patient, or for a modification from any doctor

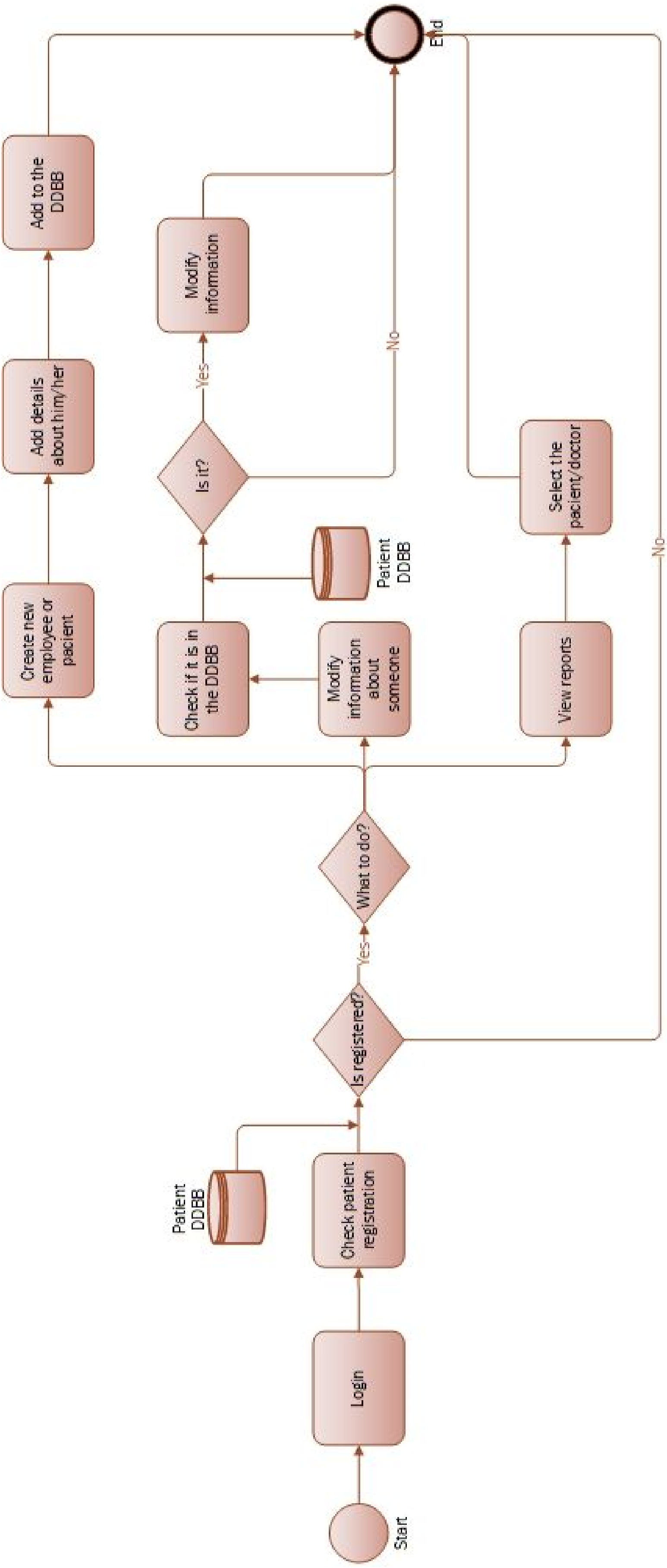
Initial condition (what state of the world we expect at the beginning): Login and has access to the information of doctors and patients

Minimal guarantee (how interests are protected at any ending, even not successful):

Main scenario of the success (3-7 points in the form: 1. Client adds the product to the basket; 2. System successfully verifies availability of the product, etc.):

- Receive a request for an appointment
- Verify if the doctor is free in the schedule
- Add the appointment
- Inform the patient about the confirmation

Extensions - Alternative scenarios - extensions to main scenario points: <<Extend>>: Be able to see and modify the schedule of the patients and doctors, in addition to the salaries.


Start
Login
Check patient registration
Patient DDBB
Is registered?
Yes
No
What to do?
Create new employee or pacient
Add details about him/her
Add to the DDBB
Modify information about someone
Check if it is in the DDBB
Patient DDBB
Is it?
Yes
Modify information
No
View reports
Select the pacient/doctor
End

2.6 Functional requirements for additional system functions

We have only three input scenarios: Patients, Doctors and Administrations inputs:

- Patients input
  - Fill the profile form
  - Get an appointment
  - Update information about profile
  - Check the reports from the doctor
- Doctors input
  - Make a diagnostic/report
  - Create a treatment
- Administrative input
  - Add/Delete a new patient to the regist
  - Modify information about the patients
  - Modify the information about employees contracts

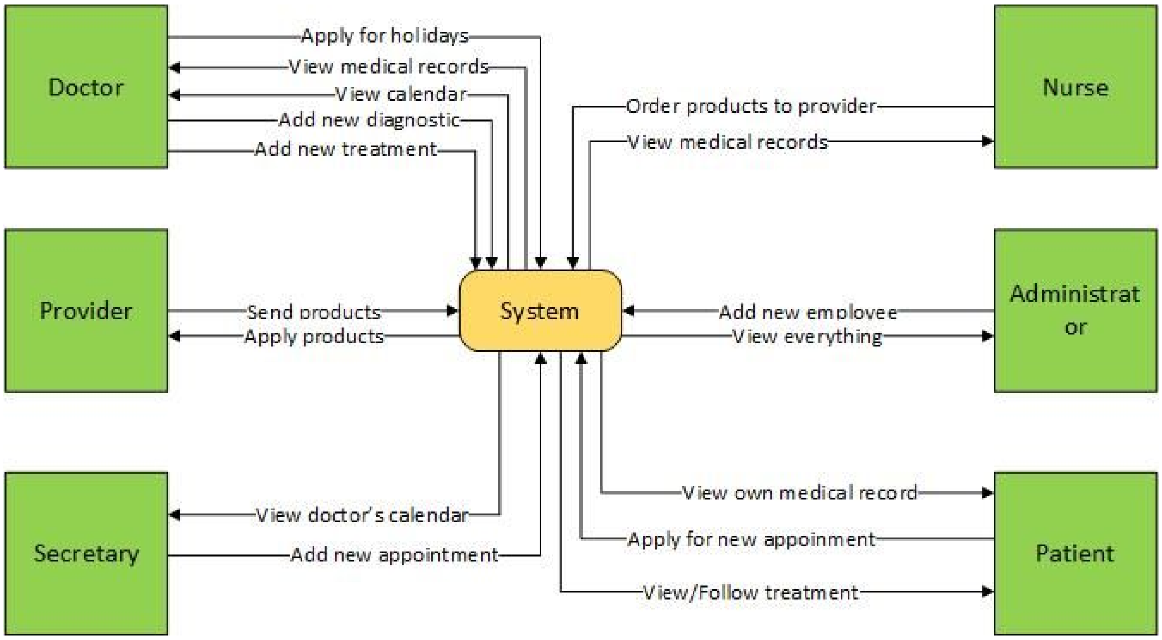
2.7 Non-functional requirements

The medical clinic will have next non-functional requirements:

- Update the information about the calendar of the doctors
- Control the amount of patients has each doctor per day
- Get statistics about the procedure and results of the clinic

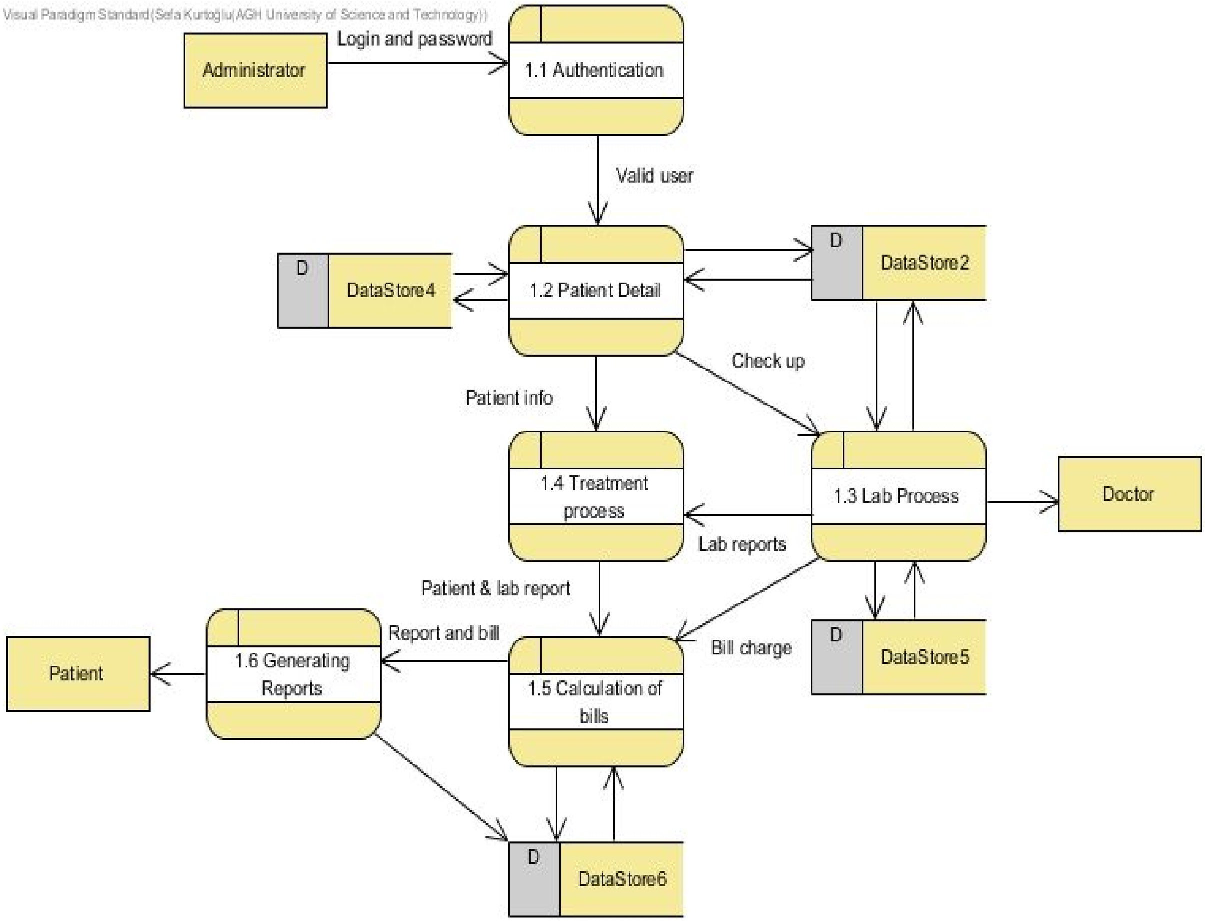
3. Functional analysis – Data Flow Diagrams

3.1 Context diagram

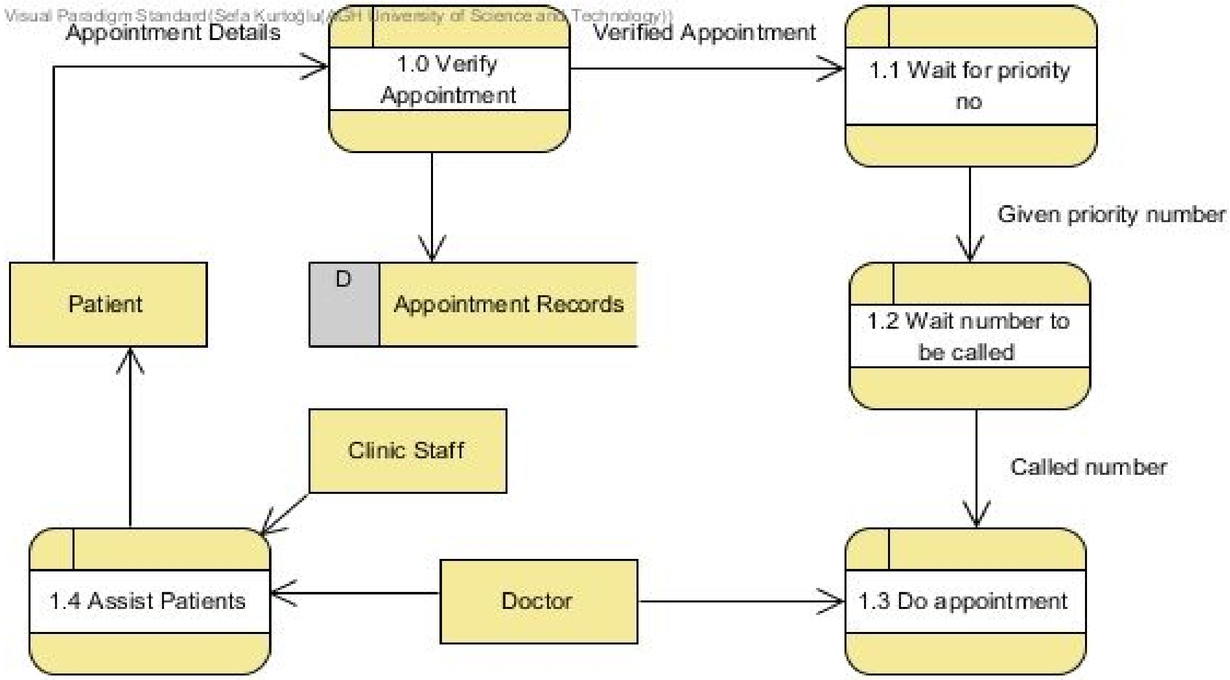


3.2 Top-down analysis(Data Flow Diagram)

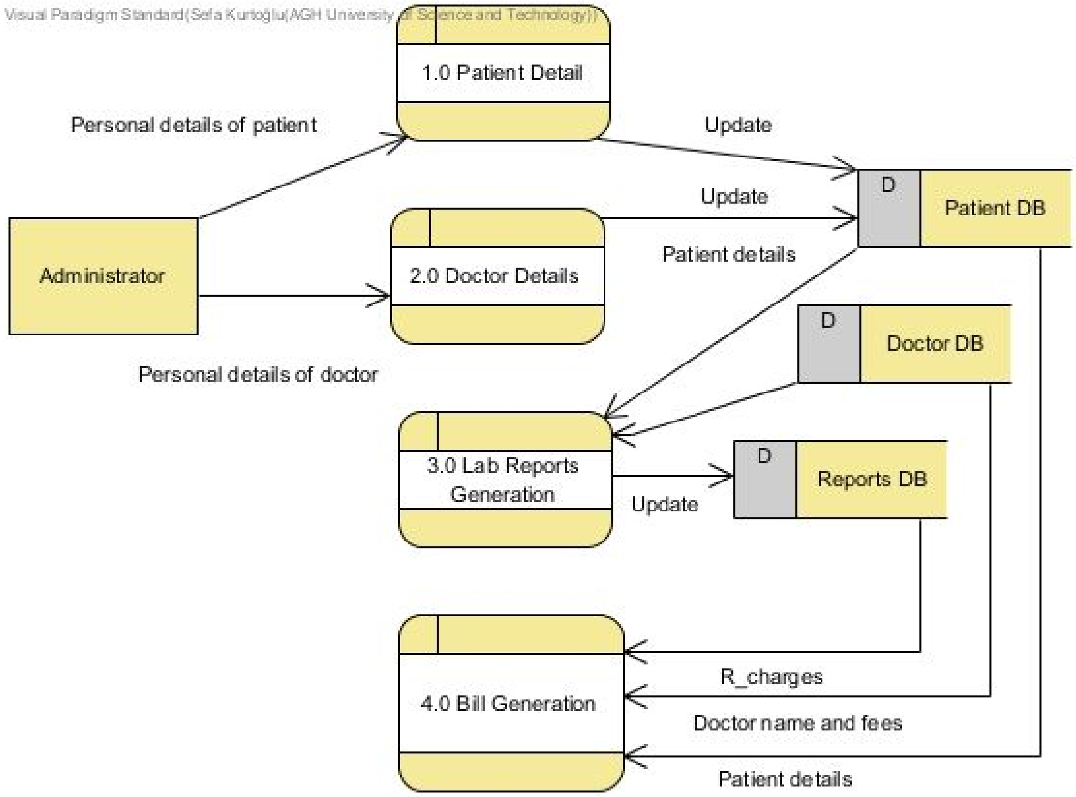
3.2.1 Level 0



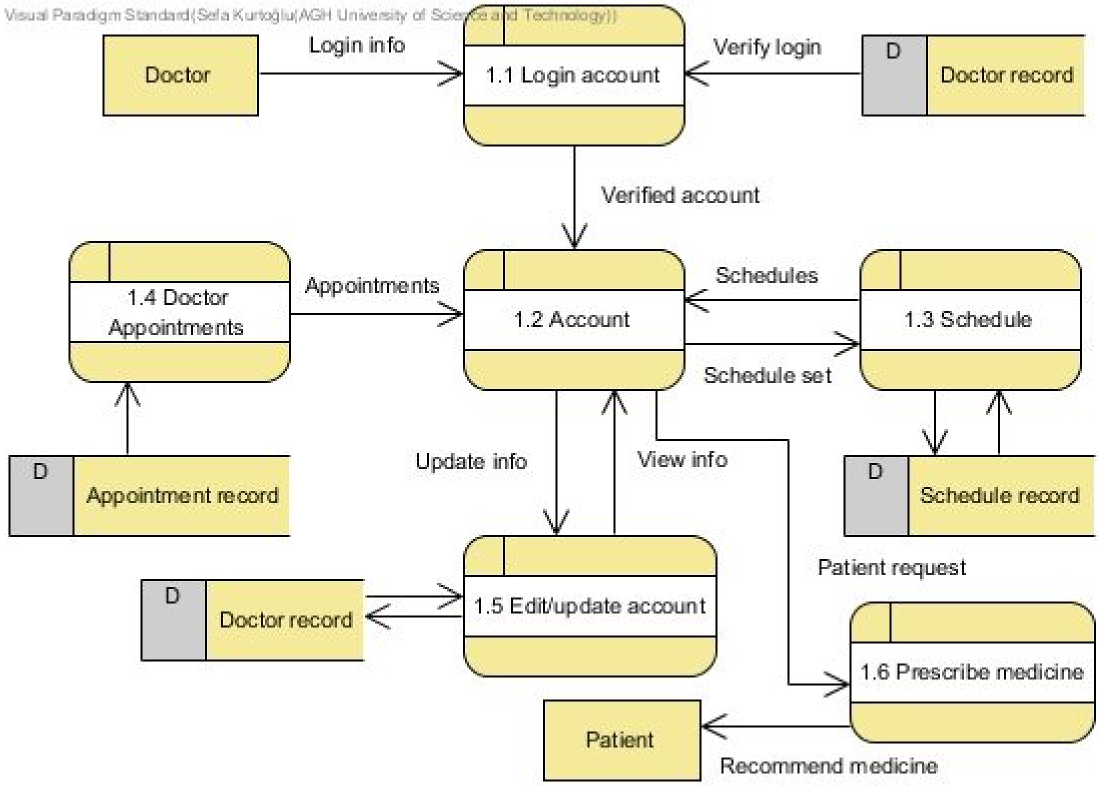
3.2.2 Level 1-Patient



3.2.3 Level 1-Administrator



3.2.4 Level 1-Doctor



3.3 Processes description

3.3.1 Patients module

Appointment request

Input: Time slot selection

Output: Confirmation message

Desc.: Makes a petition to the system to set a medical appointment into the doctor's calendar, checking that there is no any other appointment during that period of time.

Check treatments

Input: Verification identity info

Output: Treatments information

Desc.: Gets information about the treatments someone has to follow, including the prescription to get the medicine in question.

View own medical record
Input: Verification identity info

Output: Information about patient’s medical record

Desc.: Gets information about medical history and other kind of data from somebody’s medical record.

3.3.2 Doctors module

Medical record modification
Input: Information about an appointment and diagnostic

Output: Confirmation of delivery

Desc.: Sends a modification petition to the system in order to get something in the patient’s profile edit.

Add new treatment
Input: Patient and medicaments

Output: Confirmation of delivery

Desc.: Sends a petition to add a concrete type of information about a patient in his medical record, so then he could view it later.

Check calendar
Input: Verification identity info

Output: Doctor’s calendar

Desc.: Gets information about the appointments and their details that any doctor has.

Holidays application
Input: Select period of time

Output: Application confirmation

Desc.: Sends a petition to the system to book any day as a free day, checking if it is available to do that.

3.3.3 Administrative module

Patient registration

Input: Patient info (Full name, Home address, Phone number, Bank account, Born date and Medical record)

Output: Confirmation message

Desc.: Sends a petition to the system to add a patient to the database, asking for his information and checking that everything is correctly introduced.

Doctor registration

Input: Patient info (Full name, Home address, Phone number, Bank account, Born date and Medical record)

Output: Confirmation message

Desc.: Sends a petition to the system to add an employee to the database, asking for his information and checking that everything is correctly introduced.

Delete patient/doctor

Input: Doctor's or patient’s name

Output: Confirmation message

Desc.: Sends a petition to delete a patient from the database, or to put him as ”gone”, depends on the structure of the data.

View patient’s or employee's info

Input: Doctor's or patient’s name

Output: information

Desc.: Gets all the information about any person who is related with the medical clinic

Modify patient’s or doctor’s info

Input: Modified info

Output: Confirmation message

Desc.: Sends a modification petition to the system in order to get something in any profile edit.

4. Working data dictionary

4.1 Registered Patient

Entity	Attribute	Data Type	Size
R_Patient	name	String	255
R_Patient	id_Patient	Integer	10
R_Patient	address	Varchar	50
R_Patient	phoneNumber	Integer	10
R_Patient	email	Char	100
R_Patient	datesOfAppointment	Integer	15
R_Patient	soSPayment	Binary	
R_Patient	visitHistory	Date	
R_Patient	loActivities	Varchar	255

4.2 Doctor

Entity	Attribute	Data Type	Size
Doctors	name	Varchar	255
Doctors	id_Doctor	Integer	10
Doctors	address	Varchar	255
Doctors	phoneNumber	Integer	10
Doctors	email	Varchar	255
Doctors	bankAccount	Integer	15
Doctors	statusofPayment	Binary	
Doctors	depNumber	Integer	5
Doctors	timetable	Date	
Doctors	calendar	Date	

4.3 Not registered Patients

Entity	Attribute	Data Type	Size
NR_Patient	name	Varchar	255
NR_Patient	id_NRPatient	Integer	10
NR_Patient	address	Varchar	255
NR_Patient	phoneNumber	Integer	10
NR_Patient	email	Varchar	255
NR_Patient	dateoAppointment	Date	
NR_Patient	visitsHistory	Date	
NR_Patient	loActivities	Varchar	255

4.4 Nurse

Entity	Attribute	Data Type	Size
Nurse	id_Nurse	Integer	10
Nurse	name	Varchar	255
Nurse	address	Varchar	255
Nurse	phoneNumber	Integer	10
Nurse	email	Varchar	255
Nurse	bankAccount	Integer	15
Nurse	soSPayment	Binary	
Nurse	depNumber	Integer	5
Nurse	timetable	Date	

4.5 Providers

Entity	Attribute	Data Type	Size
Providers	id_Providers	Integer	10
Providers	name	Varchar	255
Providers	address	Varchar	255
Providers	phoneNumber	Integer	10
Providers	email	Varchar	255
Providers	depNumber	Integer	5
Providers	toProduct	Varchar	255
Providers	loActivities	Varchar	255
Providers	bankAccount	Integer	15

4.6 Secretary

Entity	Attribute	Data Type	Size
Secretary	id_Secretary	Integer	10
Secretary	name	String	255
Secretary	address	Varchar	50
Secretary	phoneNumber	Integer	10
Secretary	email	Char	50
Secretary	bankAccount	Integer	5
Secretary	soSPayment	Binary	
Secretary	timetable	Date	

5. Analysis of data structures in data stores

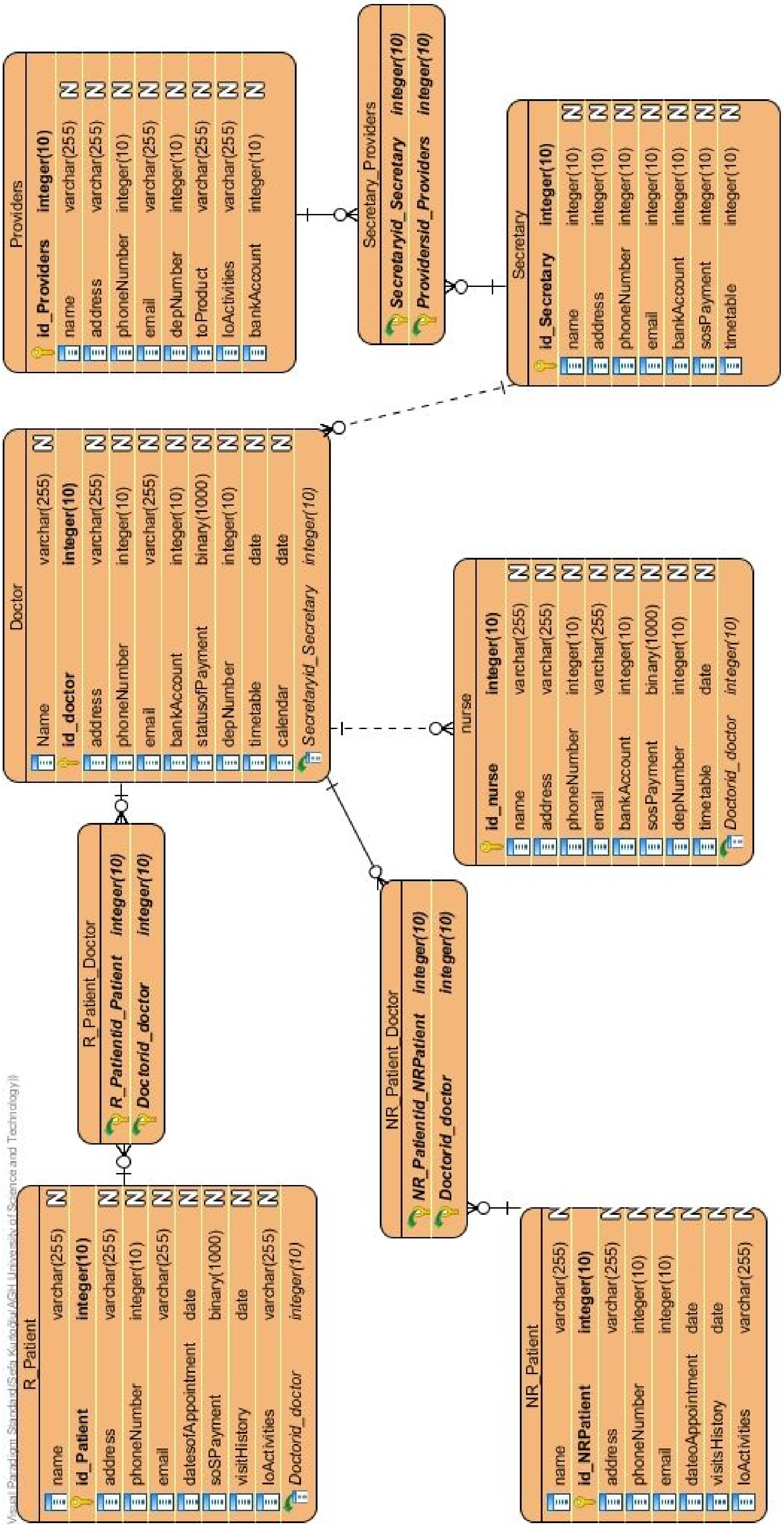
5.1 Description

First of all, between patients and doctors, no matter if they are registered or not, there is a many to many relation because we understand that each doctor works with a different part of the body or sickness, so depends on the problem with the patient we send him to one or another.

Then each nurse has a doctor assigned so they are usually going to work with them, but each doctor could have more than only one nurse, for example in case of a surgery may be it is not enough with one.

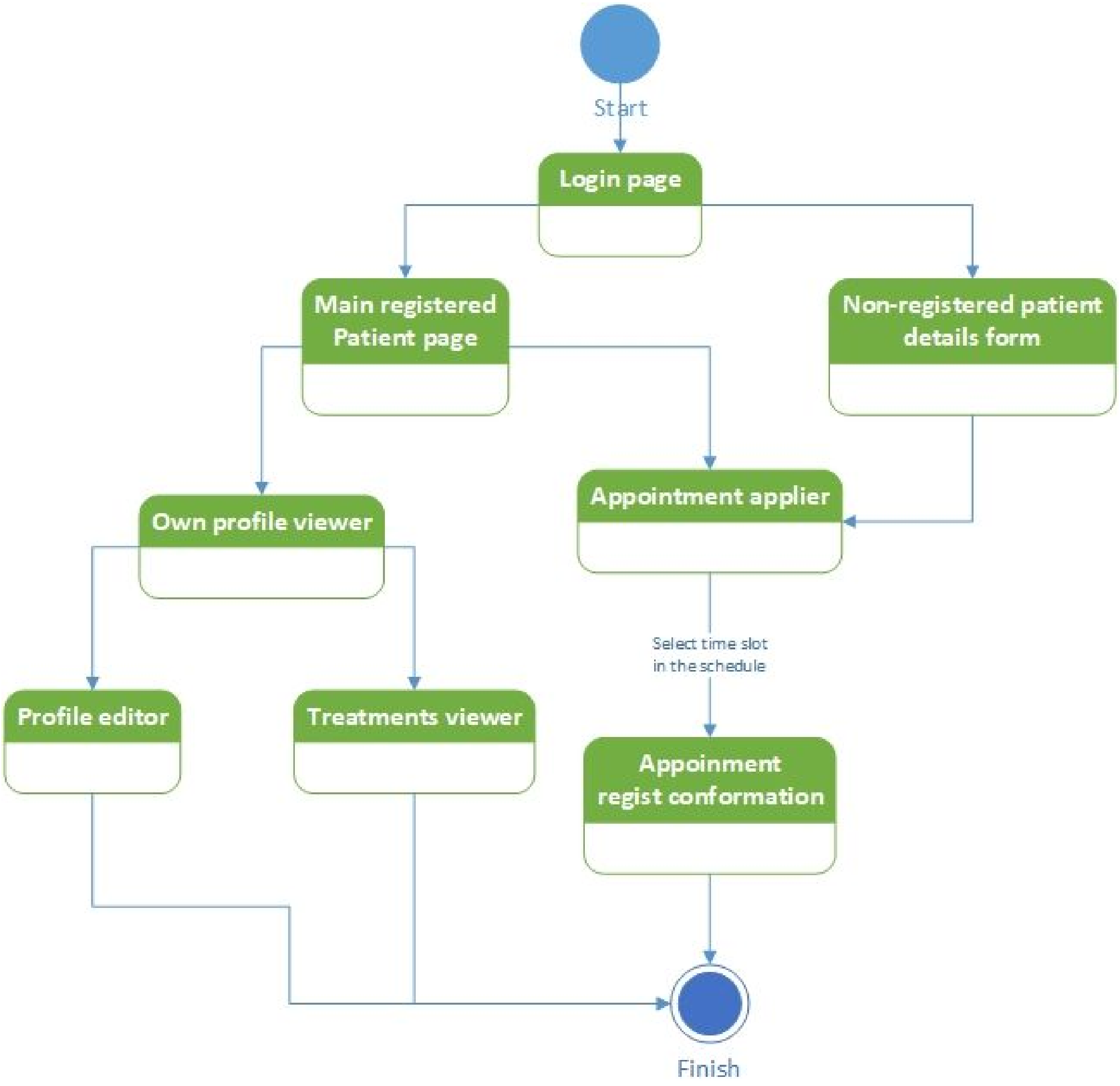
Normally there are no more than a few secretaries so each doctor will have a secretary assigned as well, so that those are not overworked. This means that everything that a doctor needs has to be preceded by their secretary. At the same time, every secretary can order whatever to any of the providers, so there we have another many to many relation.

5.2 Diagram

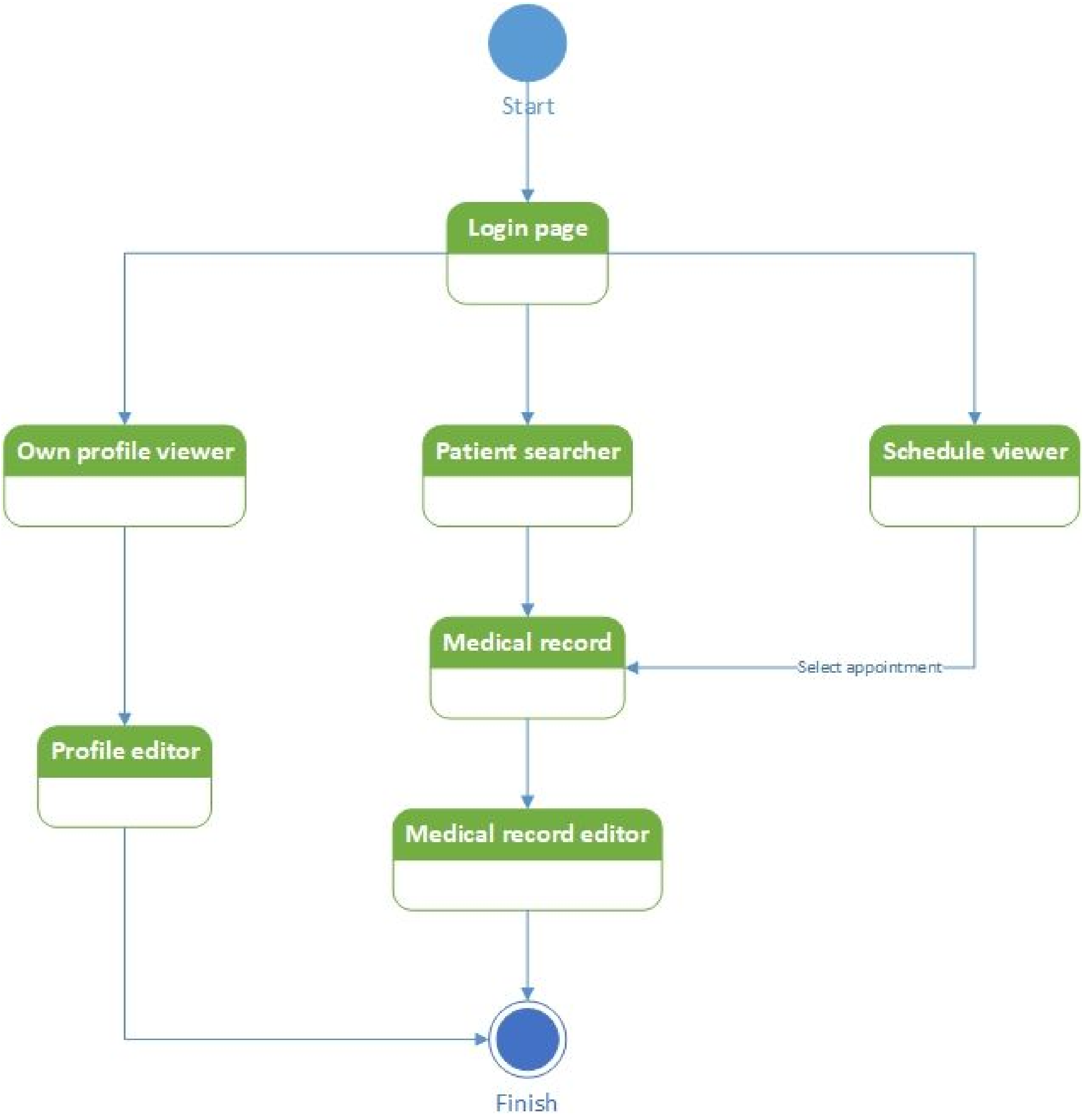

Providers
id_Providers integer(10)
name varchar(255) N
address varchar(255) N
phoneNumber integer(10) N
email varchar(255) N
depNumber integer(10) N
toProduct varchar(255) N
loActivities varchar(255) N
bankAccount integer(10) N
Secretary_Providers
Secretaryid_Secretary integer(10)
Providersid_Providers integer(10)
Secretary
id_Secretary integer(10)
name integer(10) N
address integer(10) N
phoneNumber integer(10) N
email integer(10) N
bankAccount integer(10) N
sosPayment integer(10) N
timetable integer(10) N
Doctor
Name varchar(255) N
id_doctor integer(10)
address varchar(255) N
phoneNumber integer(10) N
email varchar(255) N
bankAccount integer(10) N
statusofPayment binary(1000) N
depNumber integer(10) N
timetable date N
calendar date N
Secretaryid_Secretary integer(10)
nurse
id_nurse integer(10)
name varchar(255) N
address varchar(255) N
phoneNumber integer(10) N
email varchar(255) N
bankAccount integer(10) N
sosPayment binary(1000) N
depNumber integer(10) N
timetable date N
Doctorid_doctor integer(10)
R_Patient_Doctor
R_Patientid_Patient integer(10)
Doctorid_doctor integer(10)
NR_Patient_Doctor
NR_Patientid_NRPatient integer(10)
Doctorid_doctor integer(10)
R_Patient
name varchar(255) N
id_Patient integer(10)
address varchar(255) N
phoneNumber integer(10) N
email varchar(255) N
datesofAppointment date N
soSPayment binary(1000) N
visitHistory date N
loActivities varchar(255) N
Doctorid_doctor integer(10)
NR_Patient
name varchar(255) N
id_NRPatient integer(10)
address varchar(255) N
phoneNumber integer(10) N
email integer(10) N
dateoAppointment date N
visitsHistory date N
loActivities varchar(255) N
Visual Paradigm Standard(Sefa Kutoğlu(AGH University of Science and Technology))

6. Model of system's behaviour in time – STD-s for the main objects

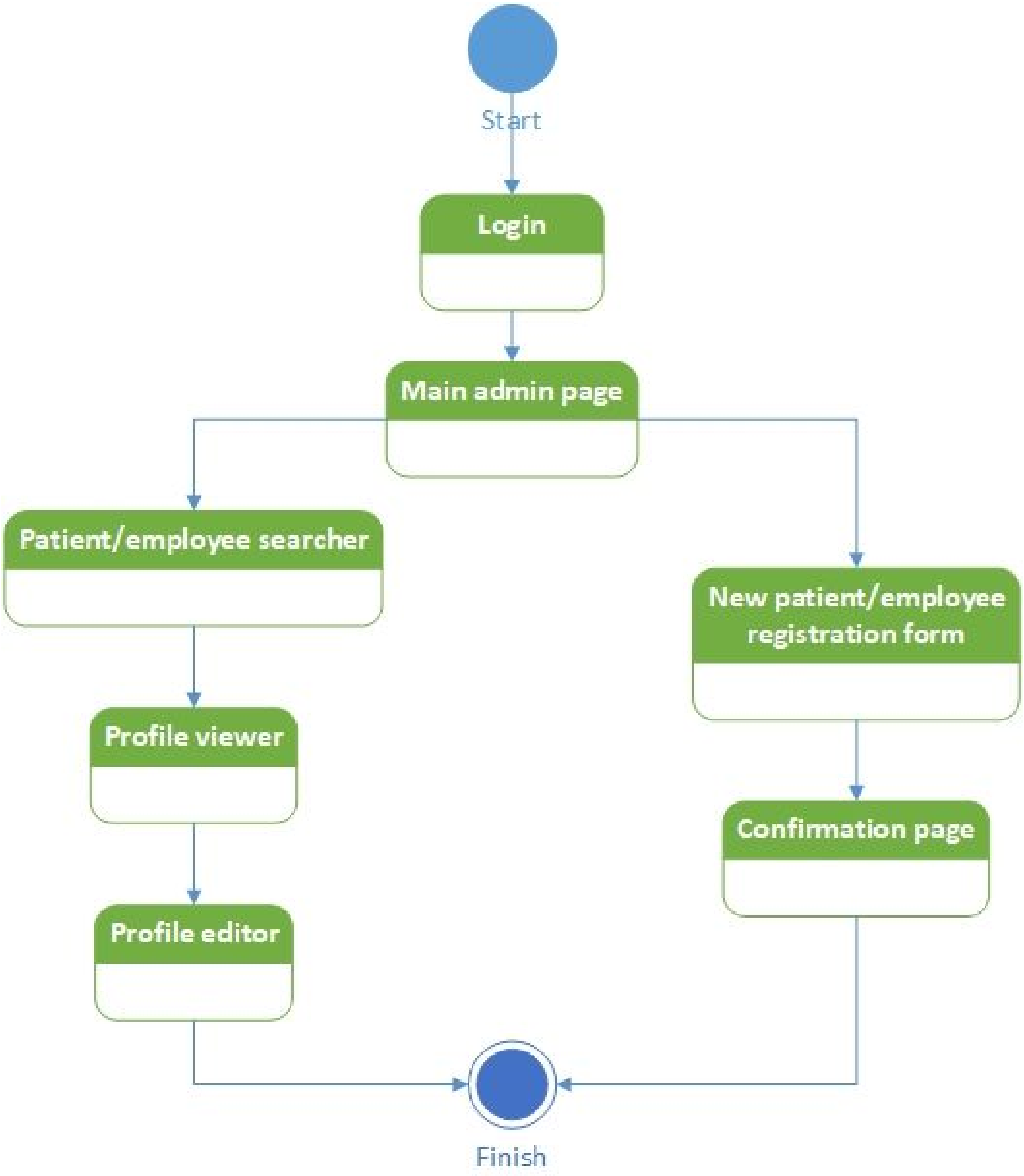
6.1 Patient's SDT


Start
Login page
Main registered Patient page
Non-registered patient details form
Own profile viewer
Appointment applier
Select time slot in the schedule
Profile editor
Treatments viewer
Appoinment regist conformation
Finish

6.2 Doctor’s SDT



6.3 Administrator's SDT



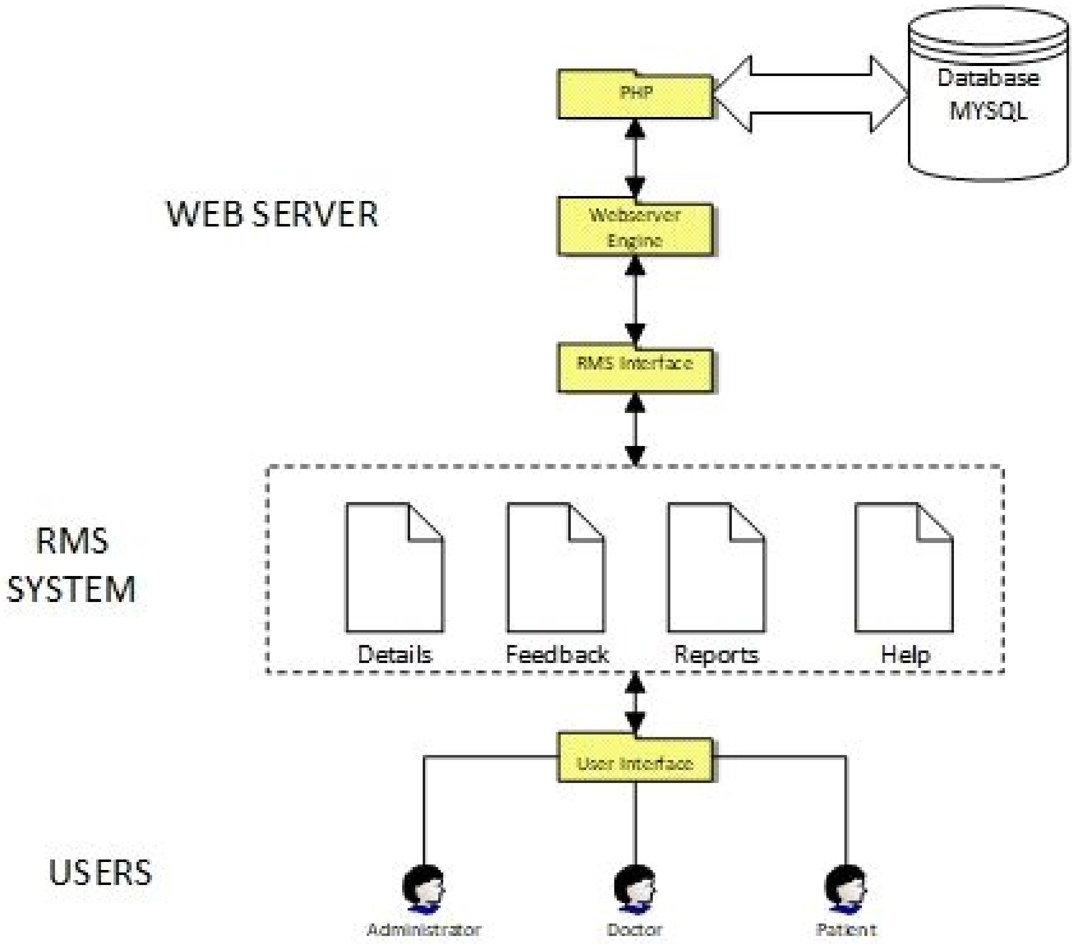
8. Comparison of models – check of mutual consistency of models and specifications

While doing the diagrams we always follow the previous one, trying to keep a continuity in our schema and drawings. Of course, this has resulted in very similar functions, with a different perspective in each drawing.

It is true some of the actions have grown changing the initial perception of that action itselves but never removing them. For example, the ordering services to medical items or medicine providers was from the beginning an option given to any employee from the clinic, however, as time goes by we saw it was easier to manage from the secretaries, not complicating it at all, both for the employees or providers and for the system configuration.

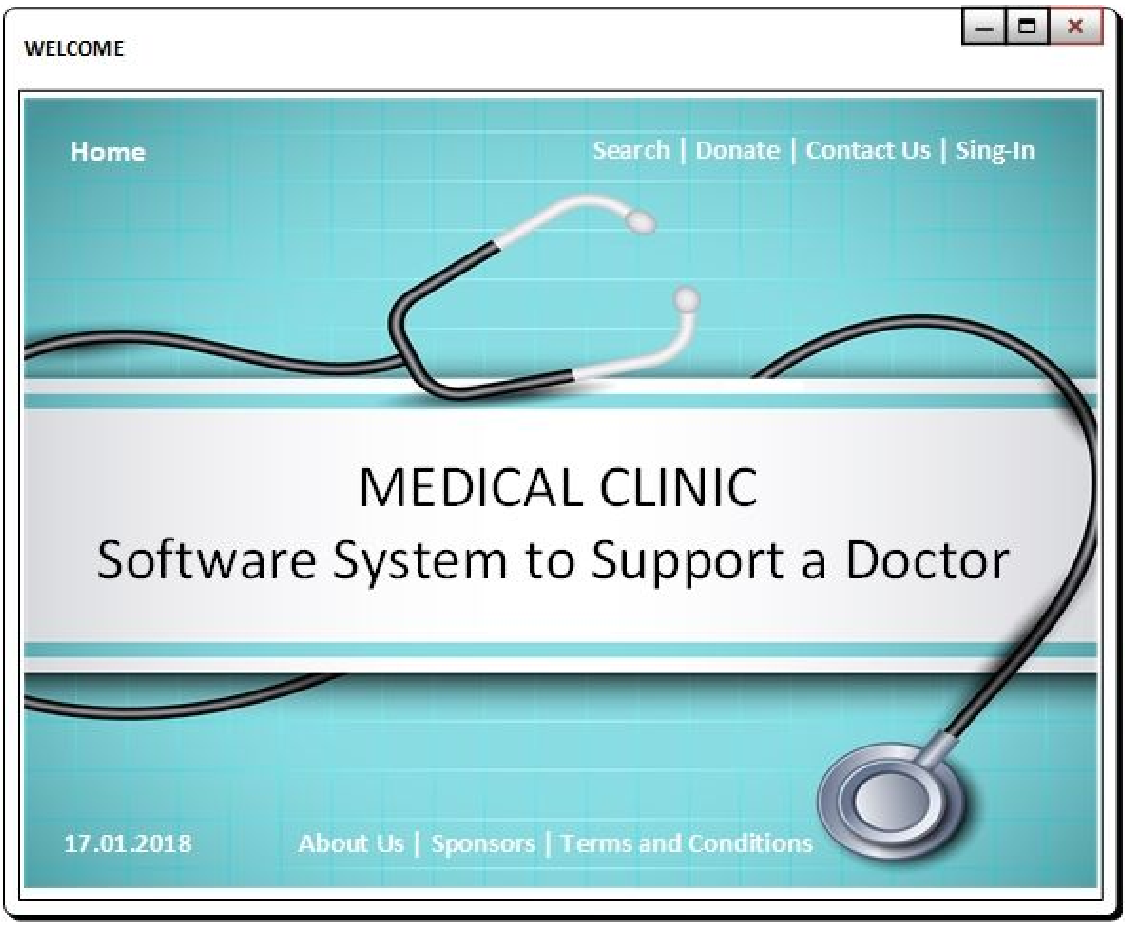
9. System's architecture (figure/s)

9.1 Architecture of the system



10. Project of the interface

10.0 Main



WELCOME

Home

Search | Donate | Contact Us | Sing-In

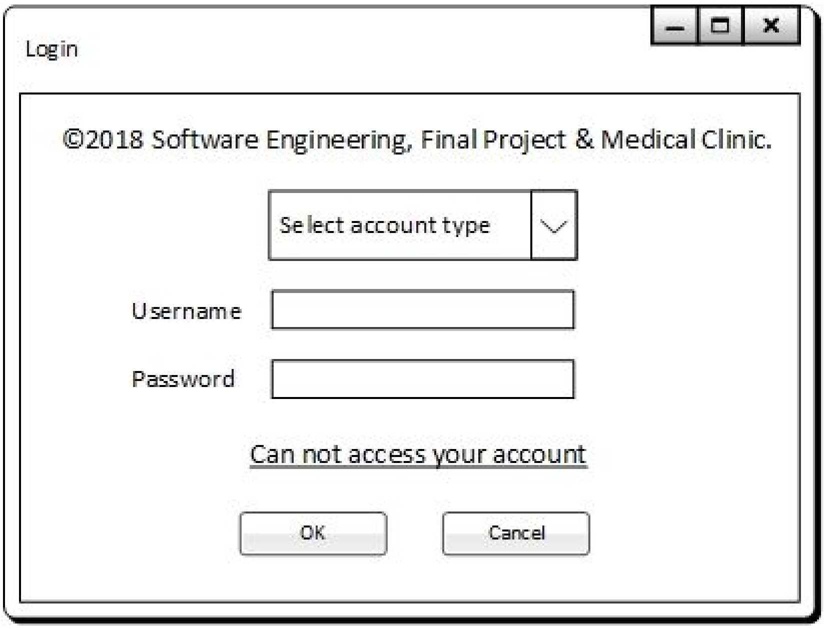
MEDICAL CLINIC

Software System to Support a Doctor

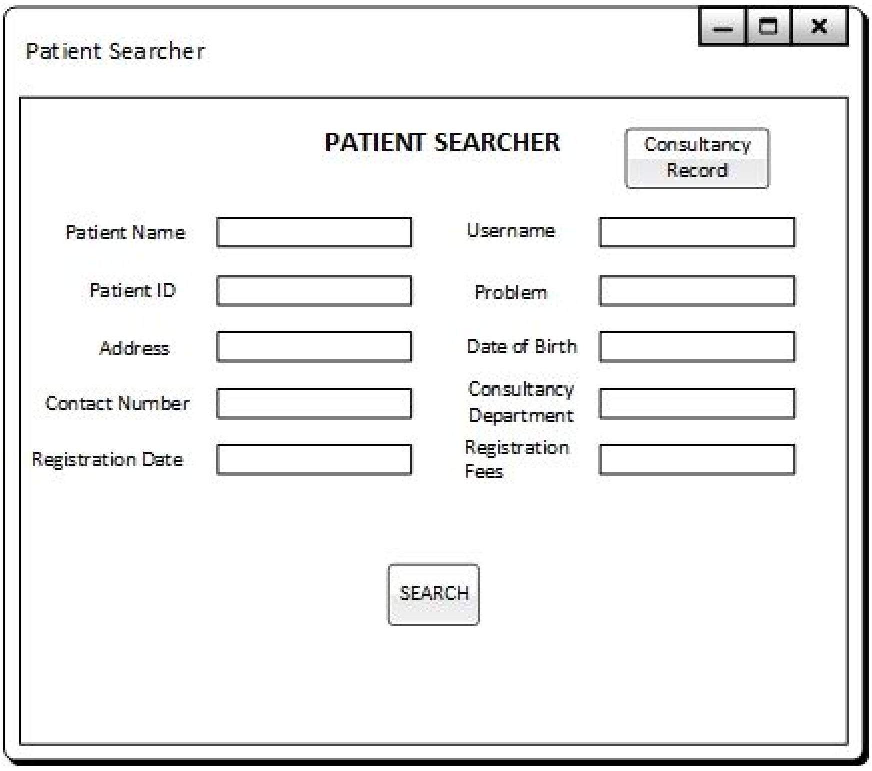
17.01.2018

About Us | Sponsors | Terms and Conditions

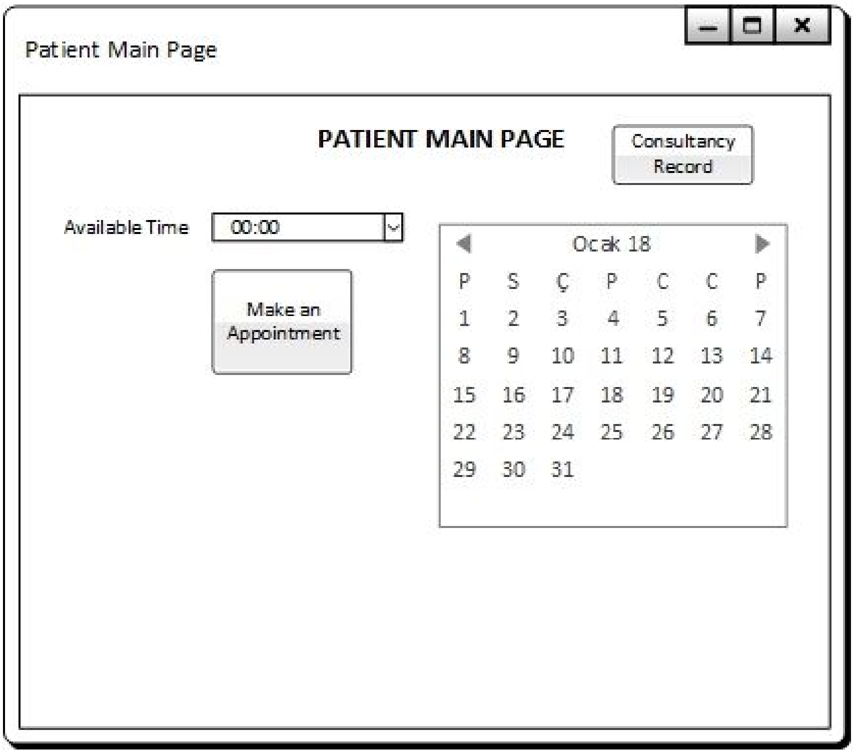
10.1 Login Interface


Login
©2018 Software Engineering, Final Project & Medical Clinic.
Select account type
Username
Password
Can not access your account
OK
Cancel

10.2 Doctor Interface



10.3 Patient Interface

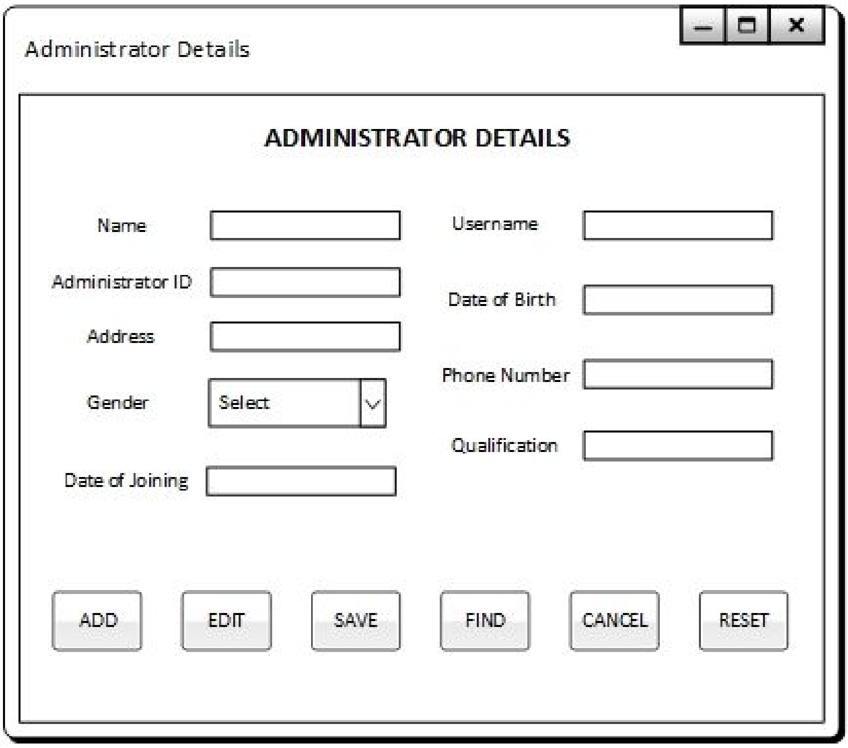


Available Time: 00:00

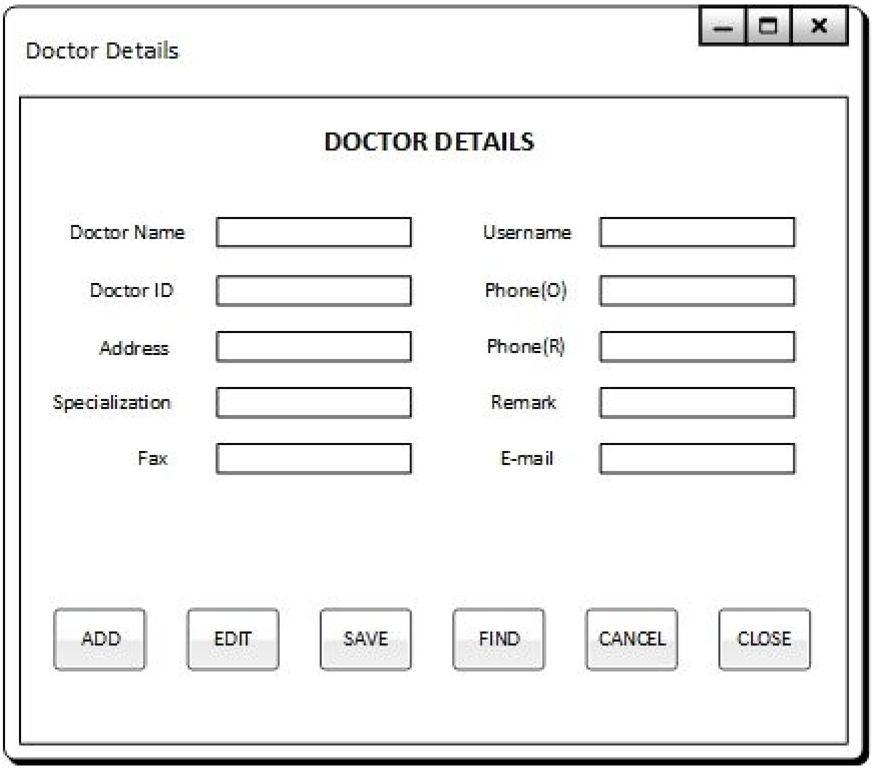
Ocak 18

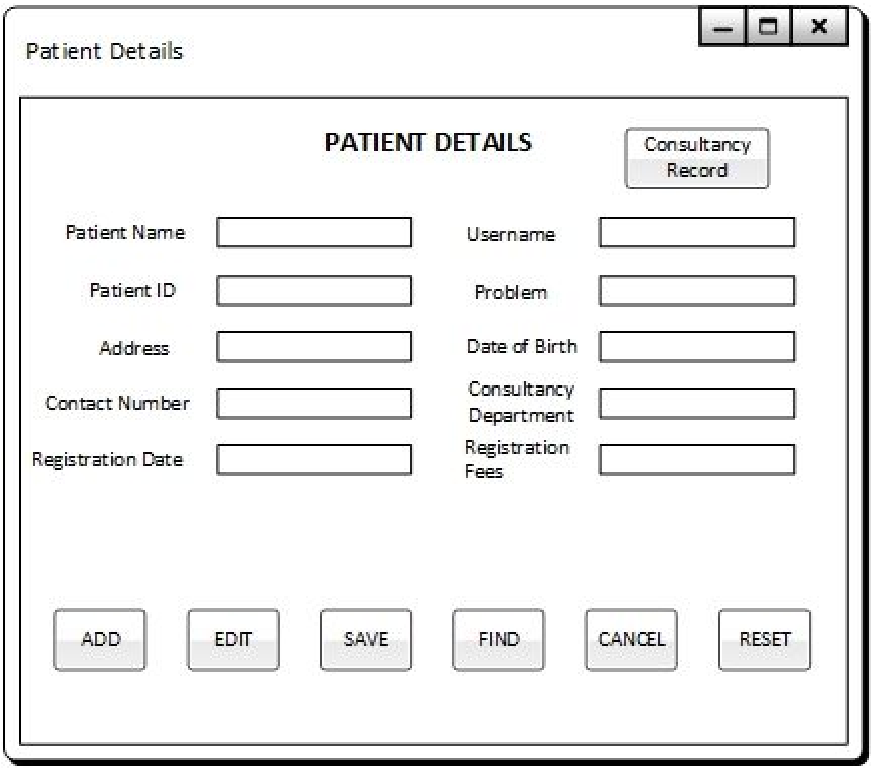
P	S	Ç	P	C	C	P
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

10.4 Administrator/Staff Interface



Gender: Select


Doctor Details
DOCTOR DETAILS
Doctor Name
Doctor ID
Address
Specialization
Fax
Username
Phone(O)
Phone(R)
Remark
E-mail
ADD
EDIT
SAVE
FIND
CANCEL
CLOSE


Patient Details
PATIENT DETAILS
Consultancy Record
Patient Name
Patient ID
Address
Contact Number
Registration Date
Username
Problem
Date of Birth
Consultancy Department
Registration Fees
ADD
EDIT
SAVE
FIND
CANCEL
RESET

11. Summary

11.1 Assumptions for implementation

When the system's development is finished we will implement it as beta version in only one medical clinic, trying to find any kind of malfunction or needed requirements, and figuring them out. After that it will be prepared to be selled and implemented in any other clinic, considering that each of them would have slightly different configurations or applications, but eventually pretty similar.

11.2 Verification of the Project

Over and above that, our goal is keep up the relation with every company who decides to get our services so that we can improve and adapt the system to any suggestion the could have.

11.3 Final remarks and conclusions

In the case of a medical clinic we must bear in mind the data storage is a critical point, after all we are working with a real high sensitive information about private people's life. It means to have the better and safer data structure, in addition to a very frequent updating.

Of course, most of the things will be adaptable to any kind, doing the system the fittest possible to the company involved, and this will help us to go on improving and developing a better and more complete software for everyone.