< Patient Guided Assessment System >

Analysis and Design Document

Student:Iacob Andrei Ștefan

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

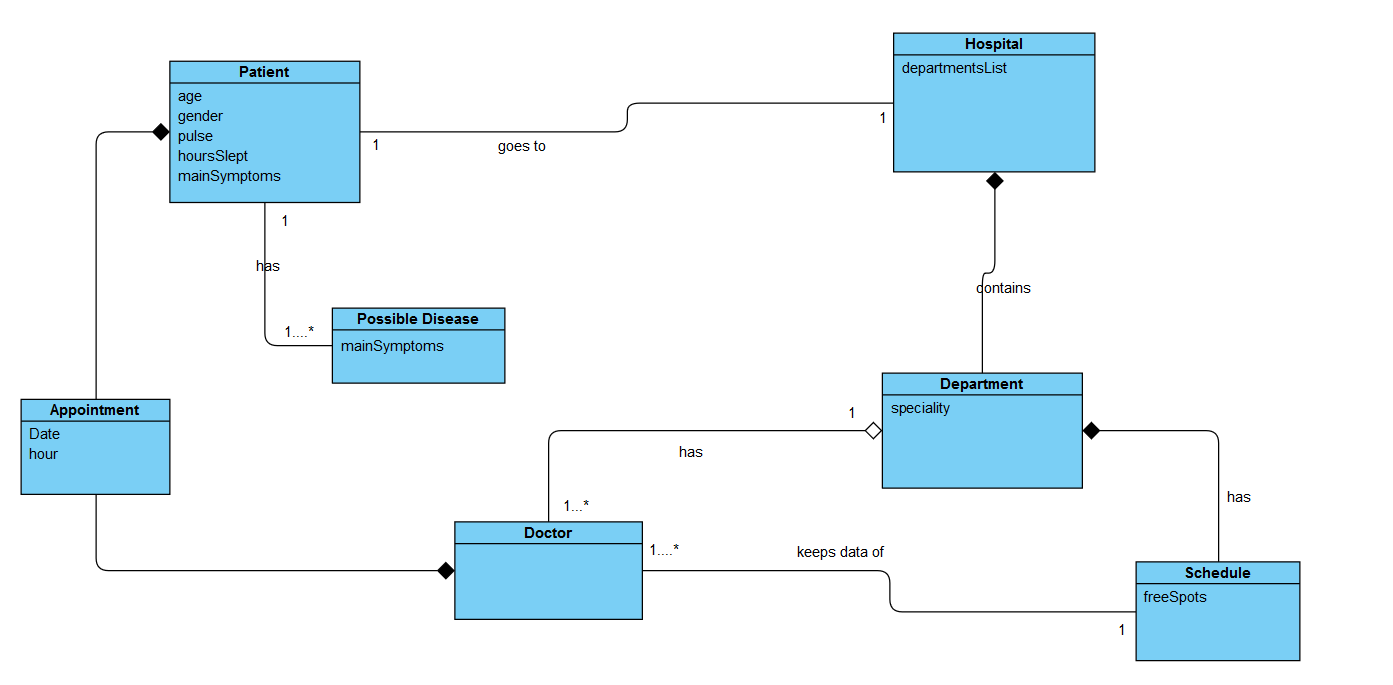
# Project Specification

# The “Patient Guided Assessment System” is a project which aims at offering the users which enter a hospital / a clinic a better view about their condition. It asks for relevant data to be input (age, gender, pulse, most important symptoms, prior conditions) to determine the possible necessities of the patient and guide him towards a department.

# It brings utility to the user( in a lot of cases , people just go to the emergency room and wait there for hours before a nurse or a physician is able to help them ) , as well as to the medical stuff ( by giving them an idea about what the patient is experiencing in terms of symptoms , his medical background etc).

# Elaboration – Iteration 1.1

# Domain Model

The domain is used by 2 main actors: the patient which enters the hospital and the medical staff . Once a patient has entered the application , he will be required to fill in a form containing some basic informations(age, gender etc.). There will also be some fields (non-compulsory) which will ask for some more specific information, like the current pulse , number of hours slept . This is why it depends a little bit on whether a physician is able to assist the user (by calculating his pulse and so on). Anyhow, after filling the form , the patient will be able to choose the most predominant(and important) symptoms that he/she is feeling at the moment. This ranges from muscular pains to headaches to simply sneezing very often. More important is the output of the system : the patient will be able to see which department is the most suitable for his/her condition and make an appointment on an available spot right away , while the physician will receive a paper with the patient’s stats (being also able to verify the results of the system and have a better idea about the patient ).

# Architectural Design

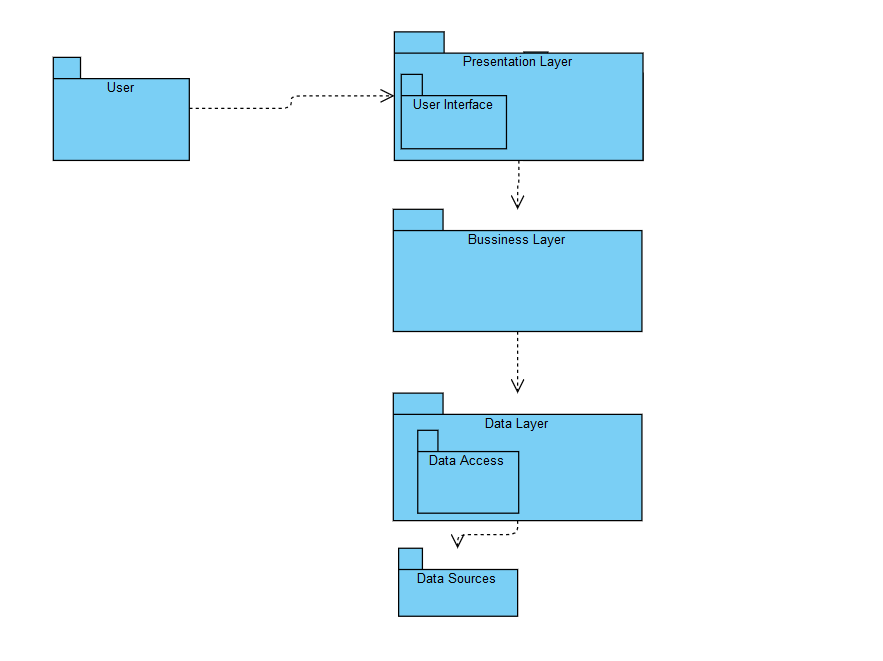
## Conceptual Architecture

*The intent of the conceptual architecture is to direct attention at an appropriate decomposition of the system without delving into the details of interface specification. It is a structural design that contains no implementation details.*

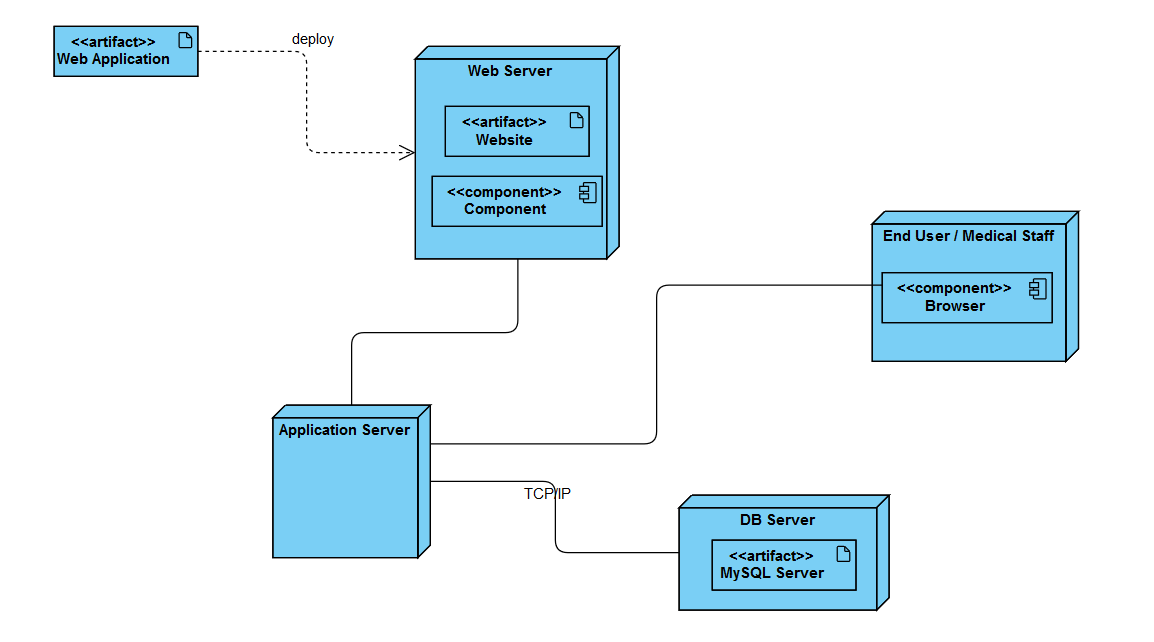
*The ”Patient Guided Assessment System”* *application will be web based. It will be hosted on a website to which the patients will have access once they enter the hospital/clinic ( maybe the website will continuously run on a device and remain ready to use for all the ‘ newcomers’). In order to have an organized structure , the project will be based on the layered architecture . Each layer will be performing a specific role within the application (presenation, logic etc.), so there will be a clear separation of concerns among components. The reasons for choosing so are , most importantly :the fact that it is easy to test, as components belong to specific layers ; it is easy to implement because naturally most applications work in layers. To sum up, it is easier to develop ,test , govern and maintain applications using this architectural pattern.*

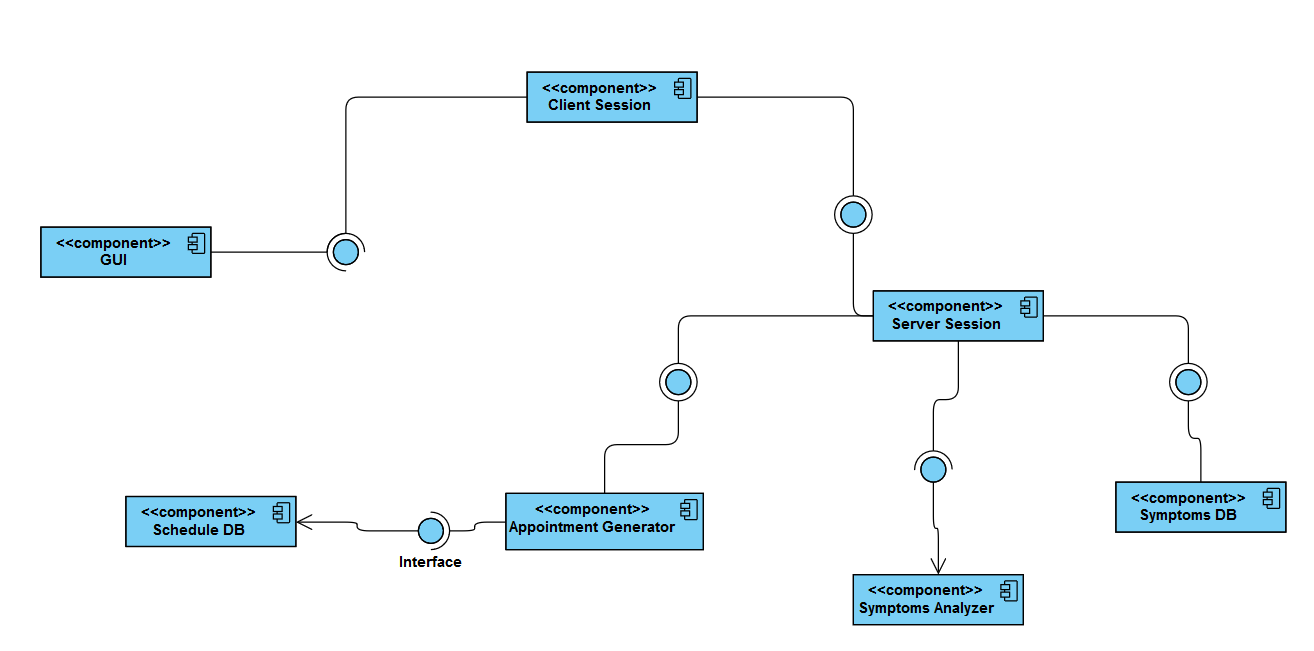
**

## Package Design



## Component and Deployment Diagrams



**

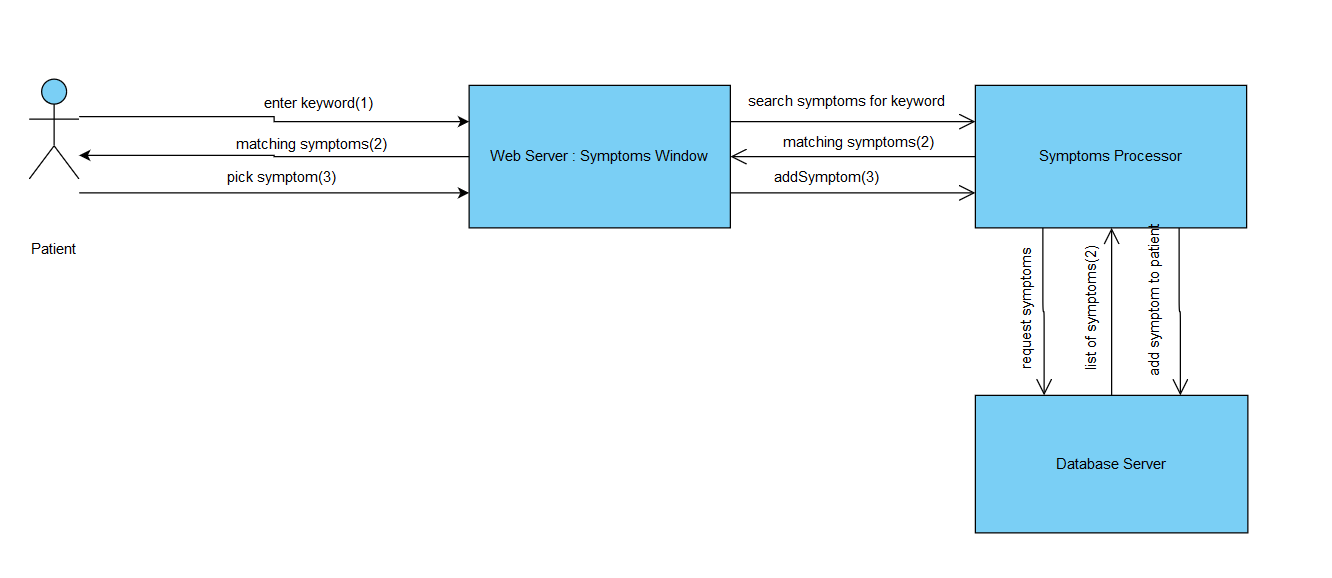
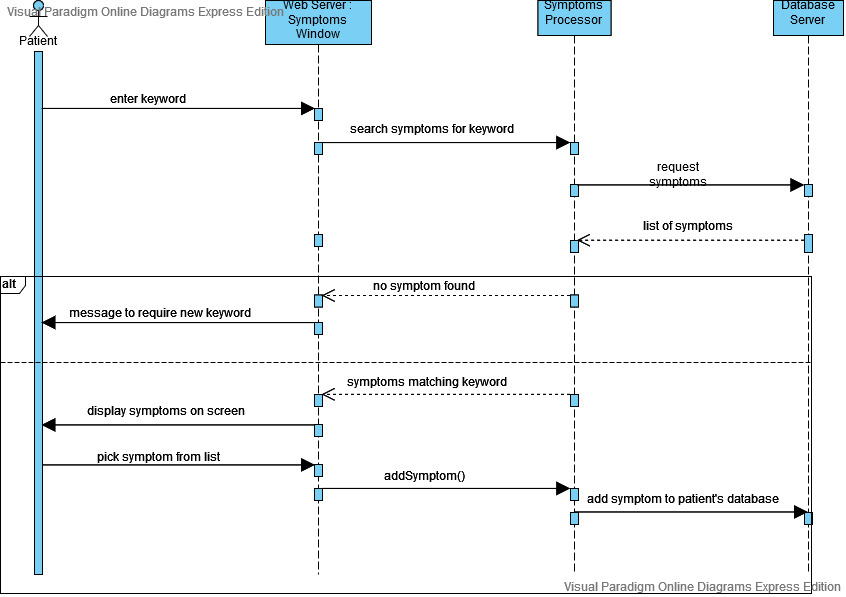
# Elaboration – Iteration 1.2

# Design Model

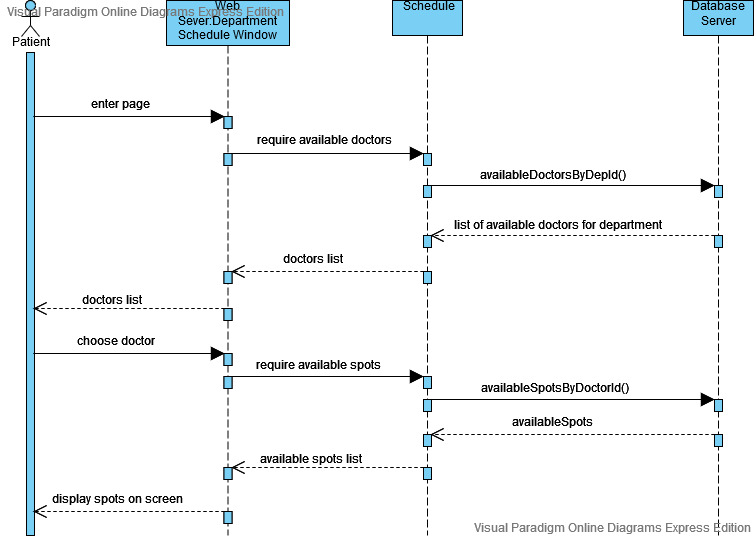
## Dynamic Behavior

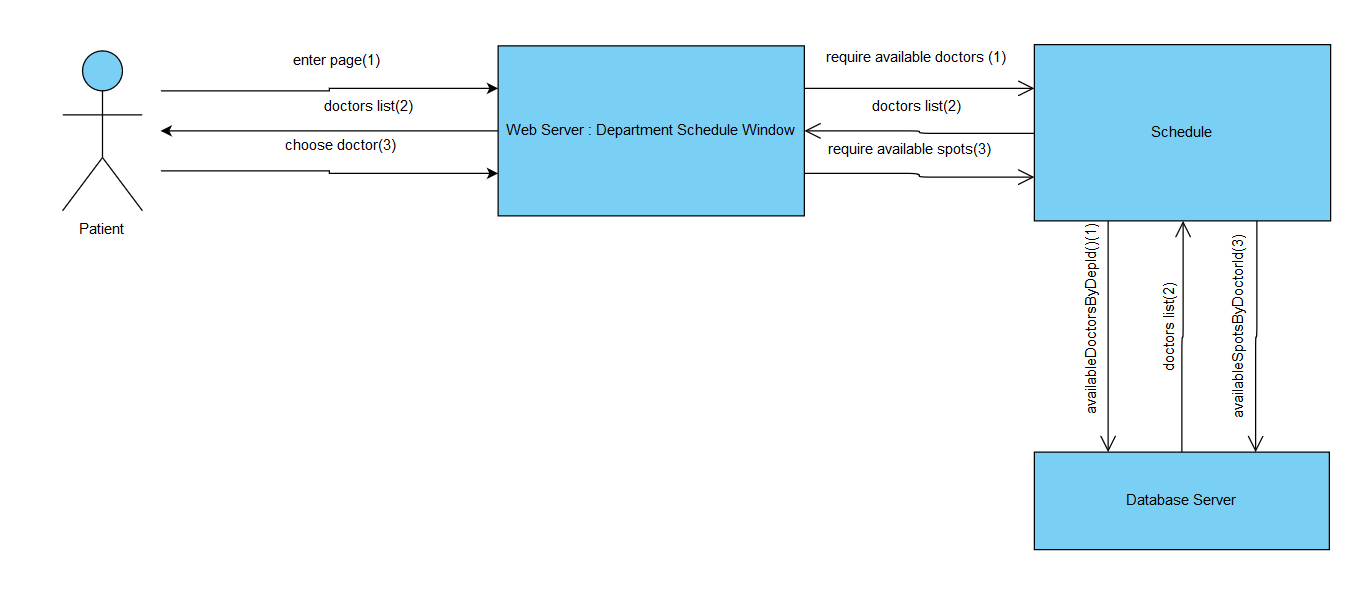
*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

*First scenario : the patient has reached the symptoms’ window. He is now asked about the most proeminent ( and important , on his opinion) symptoms. The application only works , anyway , with symptoms that are already stored in the database. So the main idea of the application , it is clear, is having a well-built database. If one or more symptoms starting with the letters entered by the patient are already in the database , they will be displayed on the screen , letting the user choose the most important. If not, a message will appear, informing that the given keyword does not correspond to any ‘known’ symptom.*



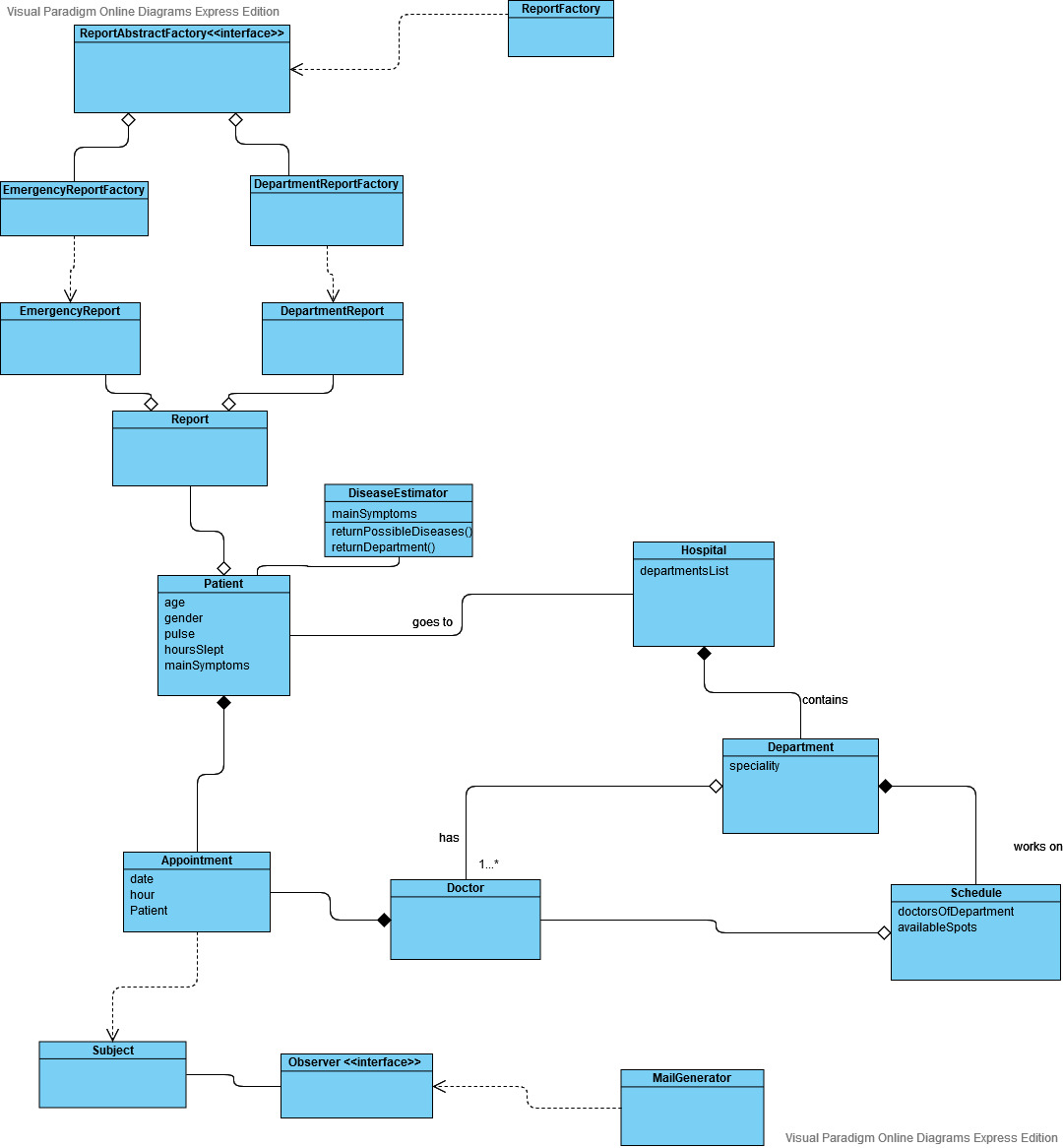
*Second scenario : one department has already been chosen for the patient and he has been redirected to a window displaying the specific schedule . First, the list of available doctors will show up. Then, upon choosing one of them , the spots that have not been already occupied for the requested doctor will become visible.*



**

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

**

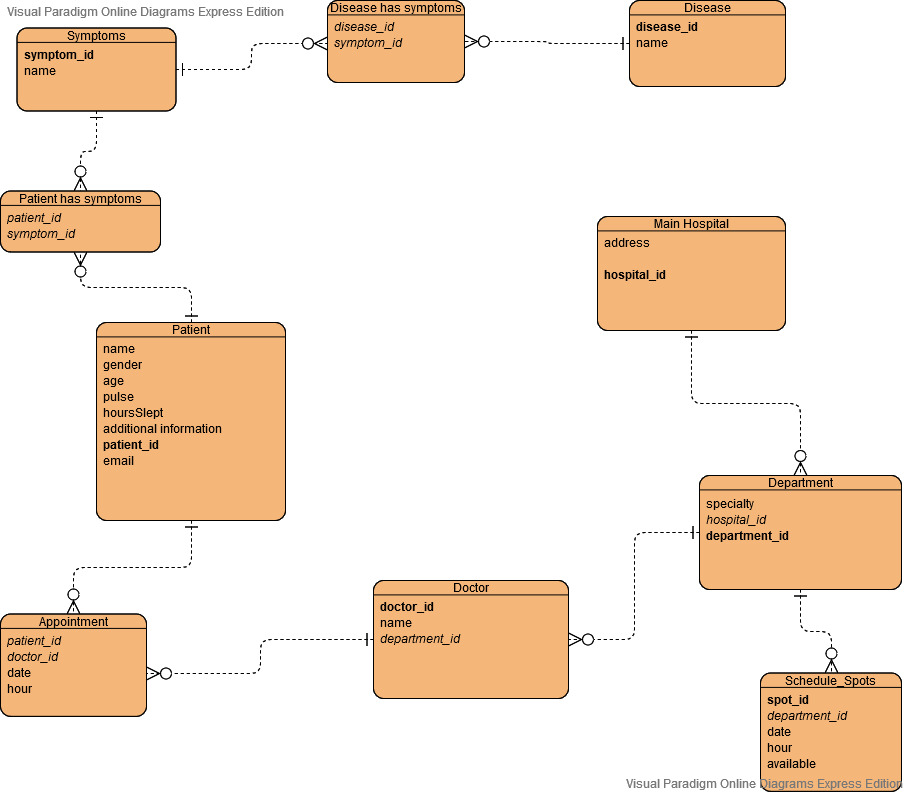
*I have two GoF patterns that I have to mention : the abstract factory and the observer .*

*The Abstract Factory is a creational pattern and it provides an interface for creating families of related or dependent objects without specifying their concrete classes.It is here used because the reports generated after the symptoms and data of the patient are of two types : a report which states the need of going to the emergency room, and another which states the department to which the patient should come at a later time , when a spot is available. The final report will have a different form on each of the two cases.*

*On the other hand, the Observer Design pattern is used for the email generation . When an appointment is set by a patient, the mail generator will be notified and given the appropriate credentials ( such that it will be able to send an email to the patient’s address with all the information necessary for the appointment) .*

# Data Model

*[Create the data model for the system.]*



# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*The system testing will be made on more than one level.*

*The first level will be the one regarding the functionality .For this to work properly, the application should fulfill the specifications intended ,as well as the functional requirements. All the links would be tested, to make sure that there are no broken links. Also , the test forms should work as expected because, especially for an application like this, there should not be any mandatory fields that are not completed with valid information .Since this is a web application which necessitates completing a number of fields in order to get an output at the end, it is clear that the business workflow should be on point.*

*The second level, that of usability testing , is just as important . We want the navigation of the site to be possible and ‘coherent’. So if the user wants to go to a specific page , he should be able to do so. The content presented on the application has to be legible.*

*The last two levels , that of Interface and Database testing , require that all the errors which may appear should be the concern of the administrator, and not the main user(he should not be able to even see them ) and all the information requested from the database should come as expected.*

# Future improvements

*Future improvements are innumerous. The first things that come to my mind , anyway , are:*

*-integrating a system for ID recognition (so that the patients would be able to give a photo of their identity card or birth certificate and then the system will identify automatically the first few details : name, birth date, CNP , gender).*

*-in times like this , when a pandemic is around the corner , the system should be able to identify quick the possibility of such dangers. If a common disease is spread within the country and the other patients may be at risk when making contact with the current user(which entered symptoms are , indeed, pointing the fact that he may suffer from the common virus) , the application should communicate directly with a more complex server and notify it.*

# Bibliography