Software Requirements Specification

for

<Patient Management System>

Version 1.0 approved

Prepared by <Ahmet Berat Akdoğan, Başar Aslan, Renas Barış Özkal, Gülden Ünal>

<organization>

<date created>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This project is to provide a website that allows patients to make appointment from internet. In the patient management system, purpose of the project is to allow patients to get their appointments in a shorter and more suitable time with developing a website. In addition, another aim of the project is to create a software project that facilitates the appointment process of patients in the age of rapid development of technology. The software requirement document was prepared for the first version of Patient Management System, which was designed from scratch.

1.2 Document Conventions

Main headings and subheadings in the document are written in bold and numbered order. This enabled the document to be a remarkable article, easier to understand and to express the contents more clearly.

1.3 Intended Audience and Reading Suggestions

This document mainly intended for hospital counselor, users like patients and doctors, developers such as programmer, researchers, project managers, designers, testers, documentation writers. In the 1st introductory part, after briefly talking what the project is mainly about, there are short information about the structure and technical features of the product in the second part. In the third part, there is technical information about the interface requirements. In the 4th part, the features of the system are mentioned. In the 5th part, there are sections related to the non-functional requirements of the system. In the last part, which is 6th part, there are visuals that explain the project with diagrams and schemas in terms of design. Readers (users) (patient, doctor, etc.) who are not interested in the technical parts of the program can skip sections 2, 3, 4 and 6. So sections 1 and 5 are sufficient. However, readers who are directly involved with the program (developers, testers, project manager, etc.) are recommended to read the entire document, including technical information.

1.4 Product Scope

The purpose of the project is to help patients to get an appointment from a website instead of going to the hospital physically. This brings a benefit especially for elders to make their appointment easily. Additionally, it allows patients to save time by taking their appointments in certain time periods, days, and the desired doctor at the beginning of the appointment process. It is important because especially in these days that we are in the pandemic period, we are expecting to prevent long and crowded waiting queues. Therefore, the scope of the project consists of all people who need an appointment and have access to the internet.

1.5 References

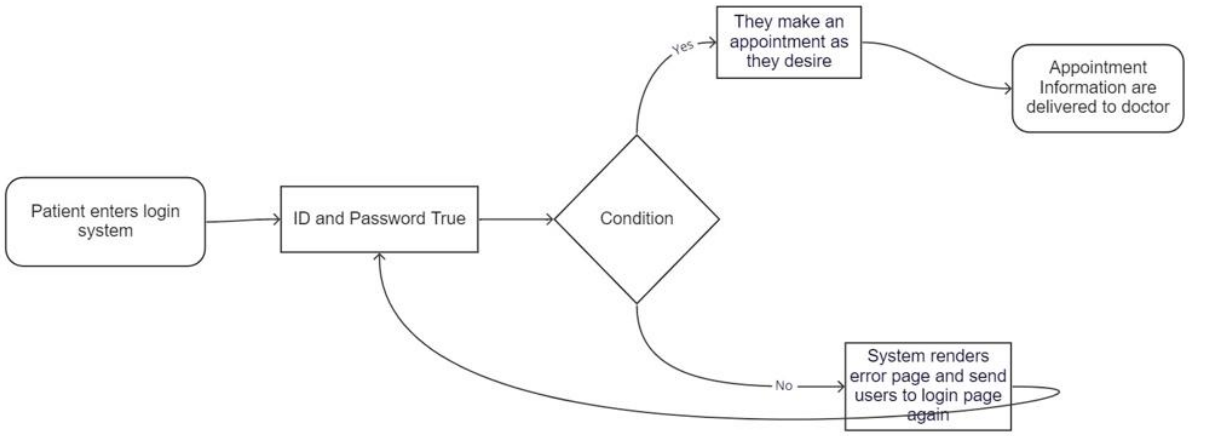
This document is referenced from “KarlE.Wiegers-1999:SoftwareRequirementsSpecification for Project”. IEEE-Software-Requirements-Specification

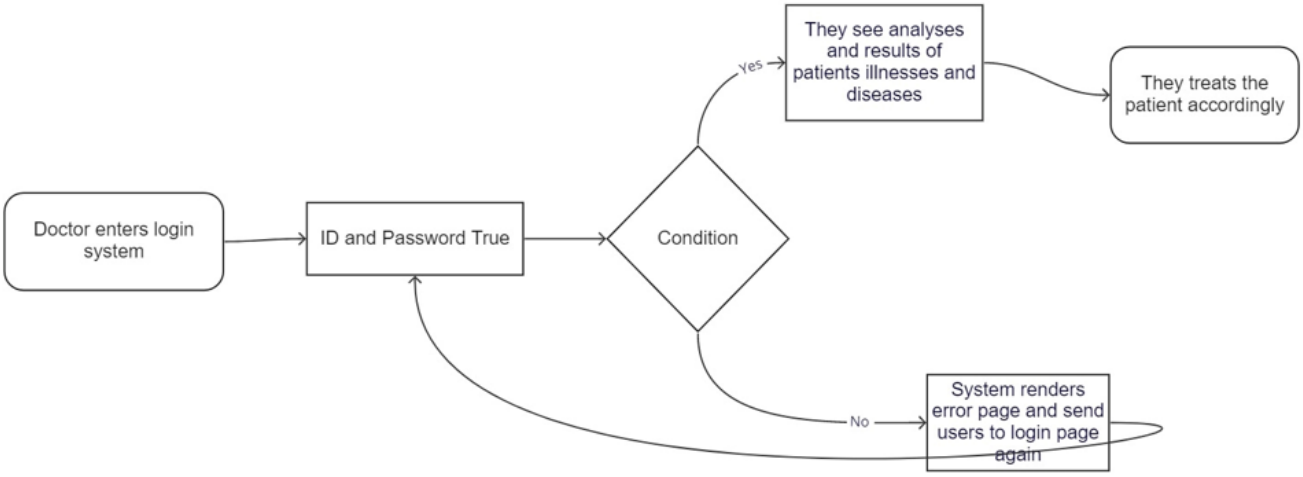
2. Overall Description

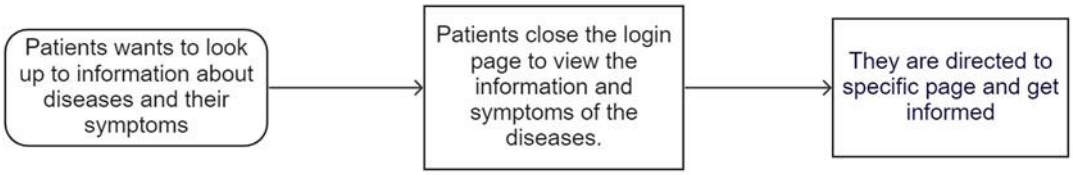
2.1 Product Perspective

Patient management system is kind of a replacement for certain existing systems that is a such a program which meets the needs of businesses. It was designed as an alternative to other appointment programs used in the market. It aims to meet the needs of small and medium-sized hospitals and polyclinics and to alleviate the burden of the unit where the appointment is made. In this way, patients can make their appointments very easily.

2.2 Product Functions







Patient management system is a multi-user, easy-to-use software program. Notable features: setting appointments that can be taken quickly, being accessible from a variety of different devices (hardware), such as tablets, phones, and computers. Detailed specifications of the project can be found in the third section. It is aimed to be easily adaptable in a modular structure to meet the needs of all patients, doctors, and hospital consultants.

As non-functional requirements, we want our system to render very quickly. Thus, users will be able to do all their work in a short time. At the same time, we want our security to be at the highest level so that users cannot see each other's user features and steal this information. At the same time, we expect our interface not to be too complex because this time it will take both patients and doctors to figure out where the necessary systems are located.

2.3 User Classes and Characteristics

Since the patient management system is a public program, it should appeal to everyone who makes an appointment. We can classify user groups according to frequency of use, departments used, technical expertise, security level, training group and experience.

2.3.1 Patients

It is the main target audience of the project. It must remain connected to the system at all times and will use the program almost every day. They do not have a high degree of technical expertise as they are kind of sick in the elderly. Ease of use is medium for this group. Patients will be able to get their appointments according to their location, the time they want to make an appointment and the doctors they want.

2.3.2 Doctors

Although it is not the main target actor, it is an important actor for the sustainability of the appointment system. The reason for this is that the patient must choose the doctor during the appointment process and the doctor logs into the system just like the patient. Another important point is that the doctor is in control of the patient's treatment process. Such information on the patient's treatment process is linked to the database system.

2.3.3 Hospital Counselor

They are actors who have the maximum level of technical knowledge. He arranges appointments at the computer. The hospital counselor must set the working hours and inform the doctor. The harmonious operation of the entire system is important for this user group.

2.3.4 IT Department

They are the actors that responsible for speed of the database. They do performance testing such as load testing and capacity testing once a week so that the system does not crash due to some overwhelming operations. They mostly deal with the speeding database algorithms and managing the database.

2.3.5 Database System

Database system will include both patients and doctors' ids, passwords, patients' analyses. So that any doctor can see the results of patients analyses and patients can make an appointment through login system.

2.4 Operating Environment

The Patient Management System software can be used on laptops and desktop computers, as well as on phones and tablets with Android or MacOS operating systems that can access the Internet. The program can be accessed using the desired web browser after the internet connection is provided with the device. It does not require high system features. Regardless of the operating system, it can be easily accessed with Linux, Windows, and similar operating systems. It is expected to work on machines with minimum processor and ram such as Pentium and 512 mb. It should work seamlessly with at least one database server.

2.5 Design and Implementation Constraints

Since the constraints that follows will also be used on low-capacity computers, the system should not use much memory and processor, the hardware should be made at the cheapest cost, and there should be no data redundancy in the database system. While the patient is making an appointment, the system is expected to respond quickly to the appointment process within the network. The information in the database must be secure and kept confidential. Web development languages such as HTML, CSS and JavaScript and PostgreSQL database server have been decided for development.

2.6 User Documentation

Before logging into the system, the patient encounters the information page where he/she can have information about the units he/she will make an appointment with. In this way, he/she gets an idea about the disease and symptoms. The site also has a FAQ section. In this way, users or those who want to access the site, what does the site do? What is the aim of this website? As well as getting answers to questions such as general information about the system and how to log in, they can benefit from the recorded video tutorials.

2.7 Assumptions and Dependencies

The programming language used in the database should work synchronously and simultaneously with each other. In other words, since any changes, updates, queries made in the program (RDBMS) will directly affect the database, these systems are interconnected. Since the database will be used, PostgreSQL, and since a website will be designed, it is assumed that software integrated development environments such as Visual Studio Code are ready to be installed on the computer.

3. External Interface Requirements

3.1 User Interfaces

The Patient Management System program will appear in a single window. When the user (patient) logs in, he/she first encounters a screen where he/she can have information about the diseases. Then, by pressing the Log-in button, he/she will be able to start the appointment process. This program appeals to anyone with Turkish citizenship, as it is entered with a Turkish citizenship ID number. In addition, the doctor will be able to log into the system as a doctor for various procedures. There will also be a button called 'FAQ' related to frequently asked questions on the interface. In this way, the user will be able to access some information that he or she needs or is curious about. The user will act by pressing the buttons and/or filling in the required parts for the appointment. The output of the interface is the screen, and the inputs are the mouse and keyboard.

3.2 Hardware Interfaces

Mouse and keyboard are required for use. There are no specific requirements regarding device type when logging into the website. The user can make an appointment from a tablet, phone, or computer. The hardware in the PCs is sufficient for the program to work. If the person is going to connect to the server over the network, a there must be TCP/IP supported network and operating system.

3.3 Software Interfaces

PostgreSQL, the database application supported by the Patient Management System program, must be installed and open to communication. Since the program can work in harmony with operating systems such as Windows, MacOS, and Linux in terms of operating system, connection will be easily provided in terms of the program and operating system. It will come ready in the libraries that the program will use. Platforms such as Visual Studio for web development languages and DBeaver for PostgreSQL are required in the process of the project.

3.4 Communications Interfaces

Patient Management System needs communication functions such as web browser, network server communication protocol. TCP/IP network protocol will be used in the communication of the program. In communication with the database, the data will go back and forth without any encryption or compression.

4. System Features

4.1 Getting an Appointment

4.1.1 Description and Priority

This is the feature of our system, which has the highest priority for our appointment making project. Because we impose our project entirely on users to make appointments from anywhere, they want in a fast, easy, and dynamic way. We can scale the priority of this feature as 9 out of 9.

4.1.2 Stimulus/Response Sequences

When users make an appointment, after entering their IDs and passwords respectively, our system will render them according to the location of the user, the hospitals, the polyclinic where the user wants to be examined, and the doctors who are available at certain hours. If the user enters the wrong ID or password into the system, this time the interface of our system will render an error page. Let's say the user has made his appointment successfully, this time our system will render the appointment information in the system and say that the appointment has been taken successfully, and the user will understand that this process has been completed successfully.

4.1.3 Functional Requirements

The appointment system is a functional requirement of our system, as all users need to keep their id, password, body features, past appointment information and also the information of the diseases they have. Since our system is responsible for keeping this information, if inaccurate information is entered in the appointment system on the login screen, it will give an error page and redirect the user to the login screen and show the error made.

REQ-1: User must enter valid inputs to appointment login system.

REQ-2: The user should be able to make an appointment that belongs to the specific polyclinic at any time, from any place and from any doctor.

4.2 Checking Diseases and Its Symptoms

4.3 4.2.1 Description and Priority

Our site, which is a service in the feature of this system, should show the information about this disease in any branch, how it should be protected, what the symptoms are and what to do before coming to the hospital if the patient thinks that he has this disease. This is the second functional requirement of our system. Because the patient should see all the diseases and know what to do practically. We can scale the priority of this feature as 8 out of 9. Because this feature has the second highest priority.

4.2.2 Stimulus/Response Sequences

When our user enters the interface to see what the disease may be, a screen for making an appointment will appear for him, but later, if he only wants to look at the diseases and their symptoms, we will send this screen to the diseases page interface from here. In this service, our system works in this way.

4.2.3 Functional Requirements

This functional requirement is a necessary feature in our system. If the patient makes an appointment without knowing what his illness is, it may even be too late for everything. If our patient is not sure about his disease and wants to control their symptoms, he will first be directed to the interface of diseases and their symptoms.

REQ-1: User must exit login page and must be directed to diseases and symptoms page in order to check their suspicions

4.3 Checking assay data and giving treatment for patient by doctor

4.3.1 Description and Priority

Another main feature of our system is that doctors will log in to the system and write the treatment of diseases according to their fields of expertise, and at the same time, they will look at the analyzes of the diseases and determine the treatment method in the future according to the patients. This service of ours is also very important because in this way, doctors will receive patient information and act on it, and it is an absolutely very important service for priority, and it has 9 importance out of 9 again.

4.2.2 Stimulus/Response Sequences

In our system's interface of doctors, of course, there will be an entry belonging to doctors first, then if the correct entry is made, we will direct to the section of patients with a different interface and their analyzes. If the doctor enters with any non-valid id and password on the login screen, the error screen will be rendered, and we will ask them to enter it again.

4.2.3 Functional Requirements

In our functional necessity with this system, we expect the doctor to examine the analysis and prescribe treatment accordingly. It is obvious that this should be one of the indispensables of our system. Because if the doctors cannot see the analyzes in the patient database, they can neither arrange the treatment according to the patient, nor can they know the previous diseases and deal with him in the most appropriate way if the patient comes with an appointment.

REQ-1: User (Doctors this time) will open a login page and they will be directed to analyses page and give patients appropriate treatment

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response time: The system should give response in 1 second after patient did any request. It is important to be fast since our main purpose is to facilitate patient's appointments. Also, people don't like anything that runs slow. If the system does not respond to the user in a convenient time, there might be long waiting durations, even system may crash. Therefore, these leads customers to be unsatisfied. The software results with an unsuccessful conclusion.

Capacity: The system should support ... people at a time. Since it is a national site, it is important to have a big capacity in order to prevent any system problems like crashing.

User-interface: The user-interface screen should respond quickly in a 0 to 1 seconds. Also, it should be aesthetic.

Conformity: The system must conform to all kinds of OS.

5.2 Safety Requirements

Data protection and security have a major priority in our program.
Data loss may lead great damage since business records and accounts are held.
First of all, Transaction properties will be used in the database for the integrity of the data.
For system integrity such as database collapse, went off the electricity, logging feature will be used. Data backup and transfer feature in the program will be found. As the information stays on the server, the damages that may occur on the clients will not remove the data.
Some important security requirements are as follows:

People are error prone, but we can limit the negative effects of common errors by backing up our data and by provide a good security for our customers.

Security: Doctors can see their patients' information and it is important to keep patient's information as private. So, it should be satisfied by our site by making doctors' accounts protected. Doctors will have their passwords to log in which is our first security checkpoint.

Back-up: To prevent possible loss, our site has an advanced backup. It prevents losing patients' information that are important.

5.3 Security Requirements

Altering the settings is significantly important. Anyone who takes over the server or steals the files can access the recordings. Security of the server and operating system's security is similar. For patients, ID number and password will be asked when opening the program. The system will check permissions using this information

Patient Identification: The system requires the patient to identify themselves using their ID

Doctor Identification: The system requires doctors to identify themselves using their ID and passwords.

Log on ID: the log on ID for both patients and doctors is their own ID.

Modification: Any modification (insert, delete, update) for the Database shall be synchronized and only by the administrator

5.4 Software Quality Attributes

Bug free software which contains all necessary requirements
Customer satisfaction.

Availability: The system should be available all the time.

Reliability: the system should be reliable. Speeds up form development but does not limit functionality.

Reusability: part of the code that is going to be used again. Produces simple and independent code modules that can be reused.

Usability: The system should be easy to use since it will be used by all ages. The user interface screen must be easy to use and understandable.

5.5 Business Rules

- All patients need a valid ID number and password to make an appointment.
- When patients are making appointment, they must specify a doctor, living place, and valid place.
- It Department must do performance, load, and capacity testing in order to not to lose any service performance.
- Hospital counselor must inform doctor about the schedule working hours of the doctor.
- IT Department must do backup data once in a week so that the system does not slow down.

6. Other Requirements

Appendix A: Glossary

HTML	Hypertext markup language that has elements which tell the browser to how to show content
CSS	Cascading Style Sheet. It styles the elements of HTML (Border, Background Color, Text Align etc.)
JavaScript	Is a scripting programming language. displays vital data for you to look at — when it shows regular posts, visualizations, dynamic 2D/3D visuals, and so on — you may know that JavaScript is included.
TCP/IP	Transmission Control Protocol / Internet Protocol is a set of internet communication protocols that allow network communication between devices with one another.
RDBMS	Stands for Relational Database Management System. RDBMS is the basis for SQL and all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL and Microsoft Access.
DBeaver	It is a SQL client and database management tool.
Windows	It is a suite of operating systems produced by Microsoft.
MacOS	Apple Inc. It is an operating system designed by Macintosh computers.
Linux	I t is one of the core software that is the most basic part of computer operating systems.
Visual Studio Code	Code Editor that allows users to write codes in any language (IDE).
FAQ	Frequently Asked Questions.

Appendix B: Analysis Models

- **Sequence Diagrams of Patient Management System**

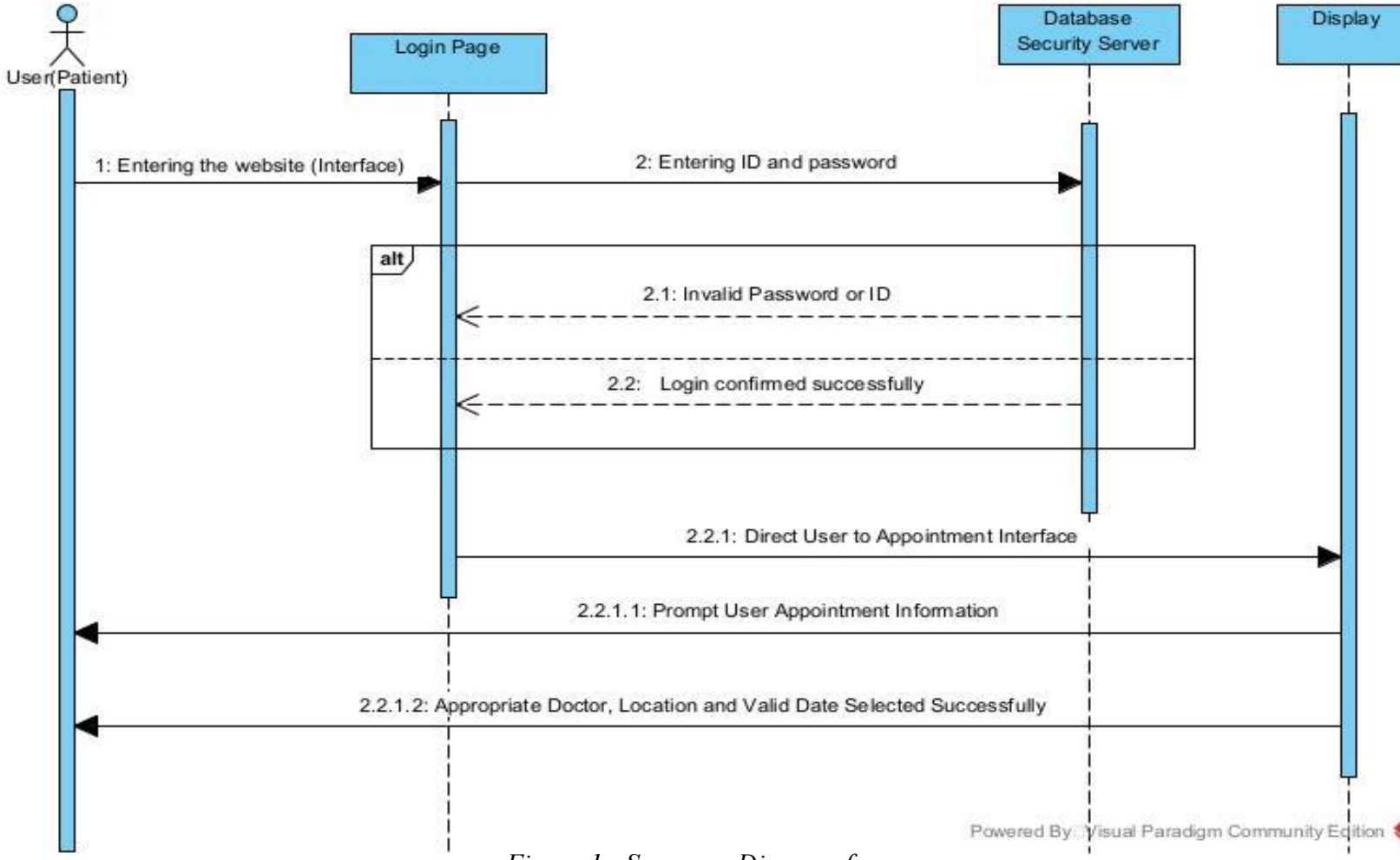


Figure 1 : Sequence Diagram for Making Appointment

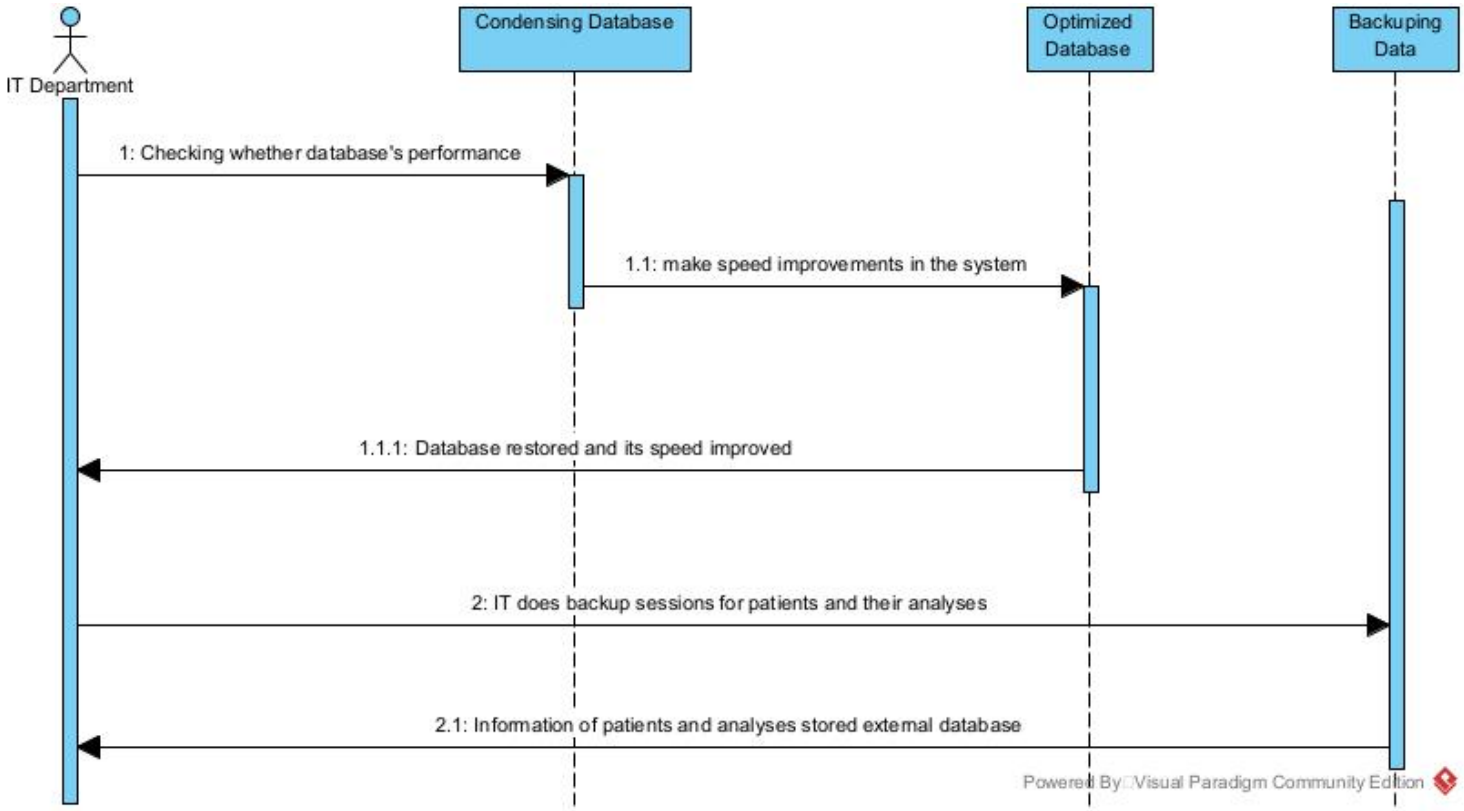


Figure 2 : Sequence Diagram for Performance Testing and Backuping Data

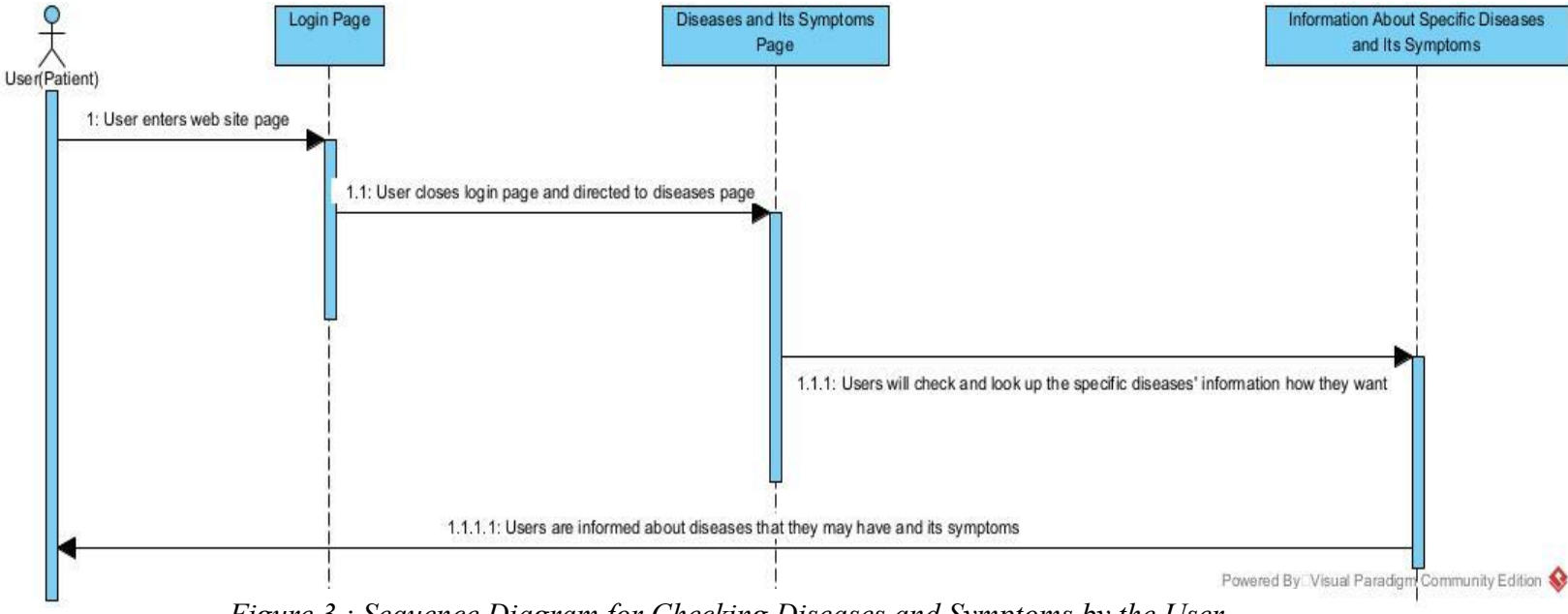


Figure 3 : Sequence Diagram for Checking Diseases and Symptoms by the User

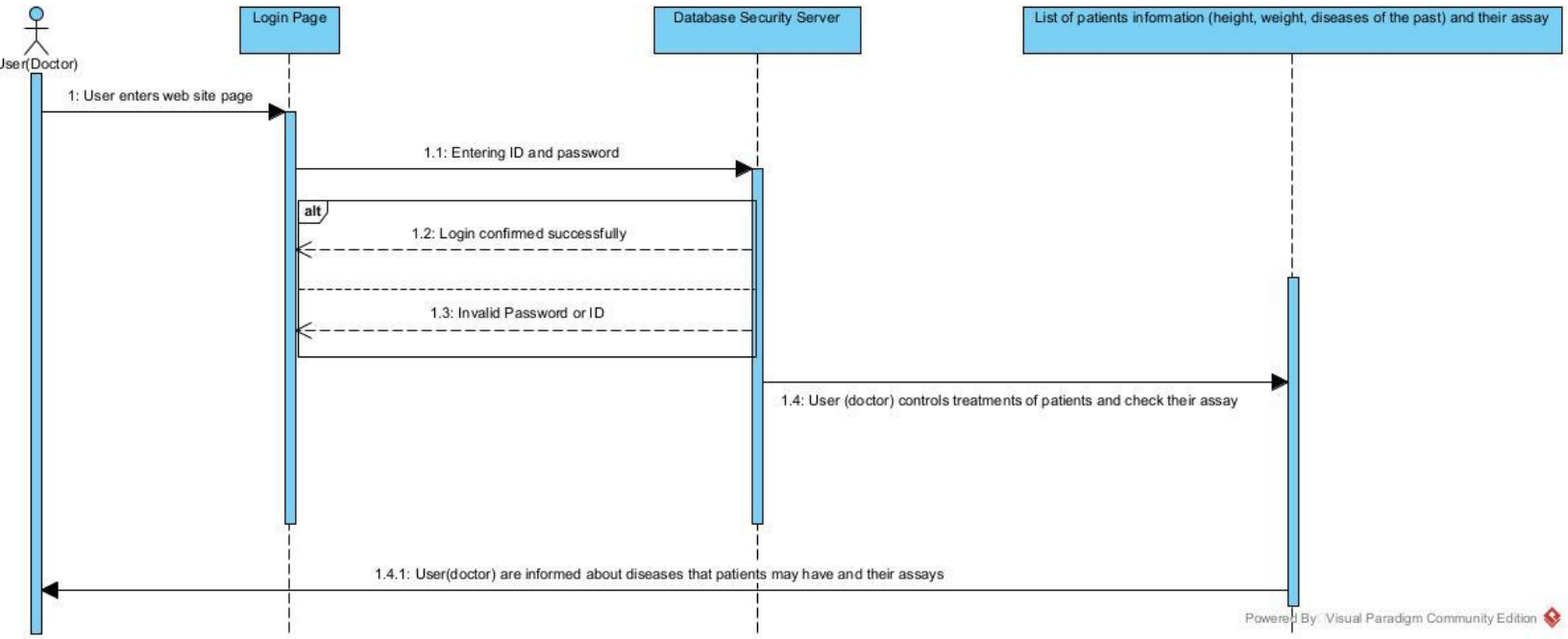


Figure 4 : Sequence Diagram for Checking Assays and Patients' Information by the User(Doctor)

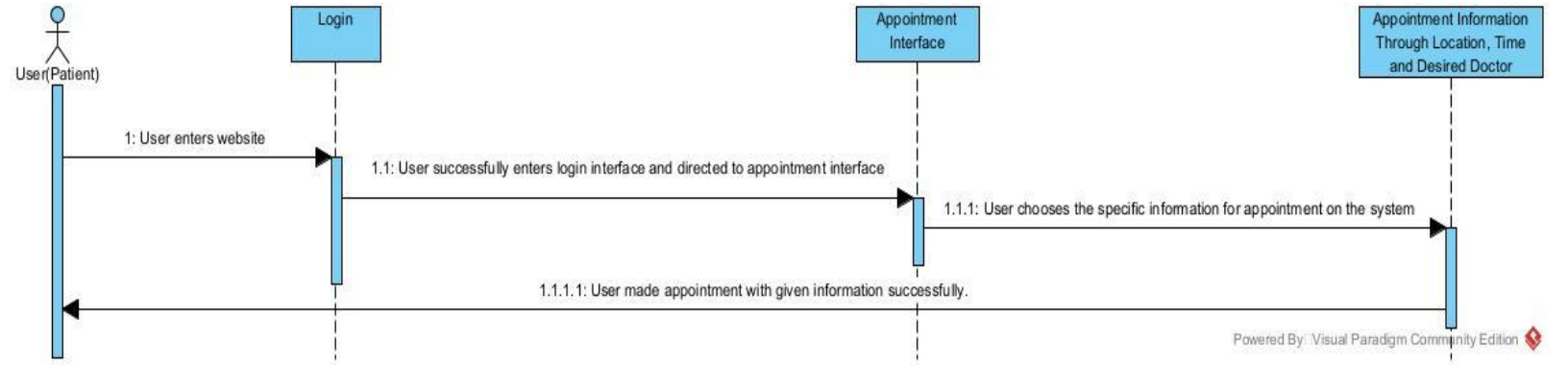


Figure 5 : Sequence Diagram for Making Appointment Through Desired Choices

- **Activity Diagrams of Patient Management System**

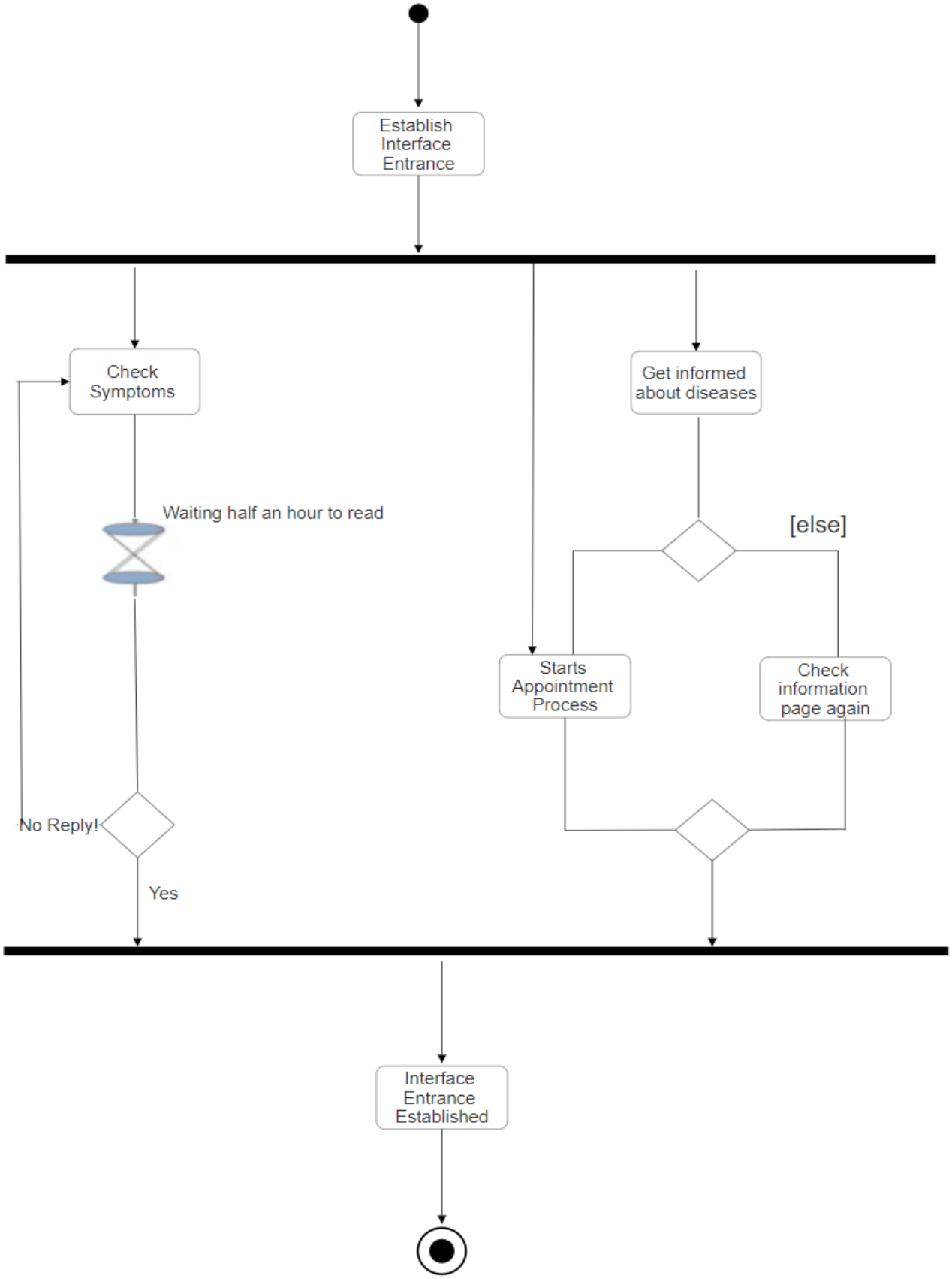


Figure 6 : UML Activity Diagram: Entering the Interface

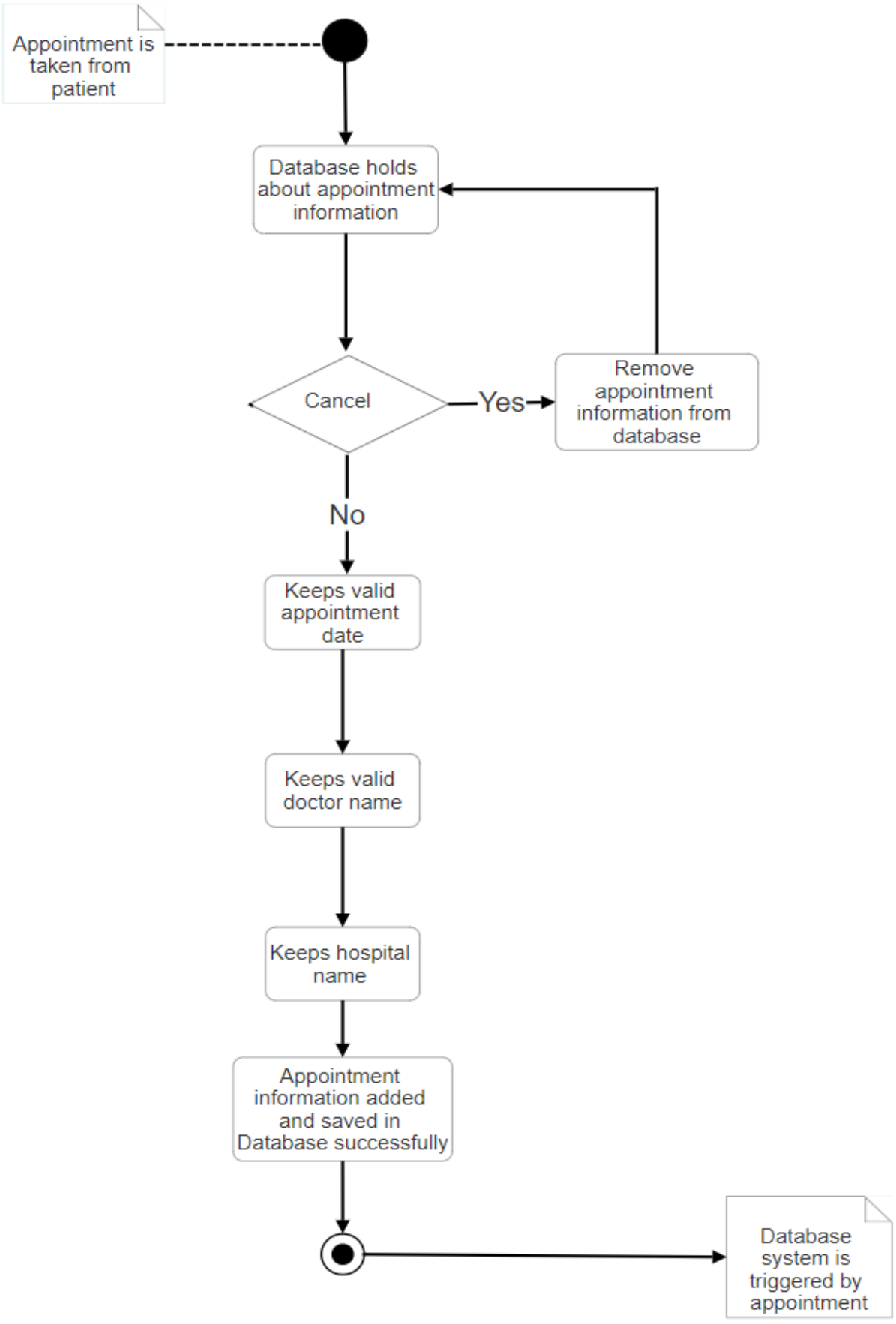


Figure 7 : UML Activity Diagram for Appointment Information of Patient in Database System

Appendix C: To Be Determined List

Not applicable.