Table of content

Figures list

Tables list

General Introduction

General Introduction

Nowadays, Information Technology is the most innovative revolution in modern life and it is undergoing a spectacular technological advance in all sectors. Certainly, the invention of the computer has made it possible to computerize the data systems of companies by saving and processing information.

Thus, in the face of the many problems arising from the traditional management of enterprises, this science plays an important role in the development of enterprises in general, health institutions, and medical analysis laboratories in particular.

On the other side, the current management methods used and the medical sectors are local that means depends only to a local server, of course, at least in our region. In this project, I have tried to use the 3-tiers architecture to resolve this problem.

Therefore, like everyone else, I was a temporary patient and I needed to analyse sometimes. So I asked; why can I not access my results online, why should I wait that long time in the laboratory to get my results. My knowledge of this field at the start of the project was casual. I knew generally about analyses and about taking samples and testing.

As this is a subject I am interested in starting my career with, building a 3-tiers application that manage a medical laboratory and offer to the patient an easy way to consult the results and also to pay seemed a natural choice of study which I knew would be challenging and interesting.

The learning curve embarked on has been considerably steeper than previous work I have undertaken. This project constitutes my first true web application culminating in a final product. My previous knowledge of the JavaScript language did not cater for the scale of this work, and my skills in Php, as used in Symfony Framework, were only of a basic level.

Through the development, I have learnt everything necessary about these languages and how they can be applied to creating a web management application.

From the theory aspect, I have done much research into the principals of management of a medical laboratory and all its details.

Chapter One: General Framework

Chapter One: General Framework

Introduction

Performing an internship during the last semester of the final year of studies presents an important step to complete our curriculum in the Higher Institute of Technological Studies. It helps students to get ready to the future professional life. The mine is titled “Medical Laboratory Web Application” and it is realized in order to obtain the degree of Information Technologies Diploma for the year 2019/2020.In this first chapter, we present the host company where our internship took place and gives a general idea about our project.

1. “Think Smart” Presentation

It is a digital agency for IT development and innovation. It proposes to provide the following main services:

- Development of digital solutions (web, mobile,..)
- Deploy efficient and useful strategies within E^{sa}.
- Study and develop innovative solutions using both VR and AR technologies ... it is a set of services designed to go out to make a market composed of several E^{sa} of different actors.

Figure 1 Think Smart Logo

2. Project Overview

2.1. Brief Description

Our project will engage in develop and design of an application that will enable the management of a medical laboratory. It will authorize to the director or the biologist as known, and it will allow the staff to manage their work and to share the results with the patient.

After intensive discussion and brainstorming, we named our application as “E-Medical Laboratory”. The stakeholders for this system are the biologist, the senior technician, and the medical assistant and also the patient as a client. Based on their roles, hence the stakeholders will be used as actors in the use case diagram.

2.2. Aims and objectives

The objective of this project is to develop a web application that can manage a medical laboratory. During the process, the related knowledge and skills supposed to be acquired in order to learn and understand the information system development process appropriately.

The core objectives that was been designed as fundamental to the project are:

- Identify, understand and describe the workflow of medical laboratories.
- Research, understand and describe the current used ways (traditional).
- Suggest methods to simplify the management and to enable the patient to a limited access.
- Learn and master the Symfony Framework.
- Storing information on a remote server.
- Automate processed tasks.
- Allows the patient to consult his medical analysis results online through his device.
- The biologist manages and controls the application in and out of his laboratory.
- It facilitates to the admins to handle their tasks due to its simplicity.
- Ease, simplicity and speed of access to data through a search, and through multi-criteria consultation.
- Confidential and secure access to information.

- Saving time for both laboratory staff and the patient, by speeding up and facilitating tasks for the first ones and by offering to the second to see his results on his device instead of waiting hours in the laboratory.

2.3. Application Architecture

- Three-Tier Architecture

'Three-tier' is a client-server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

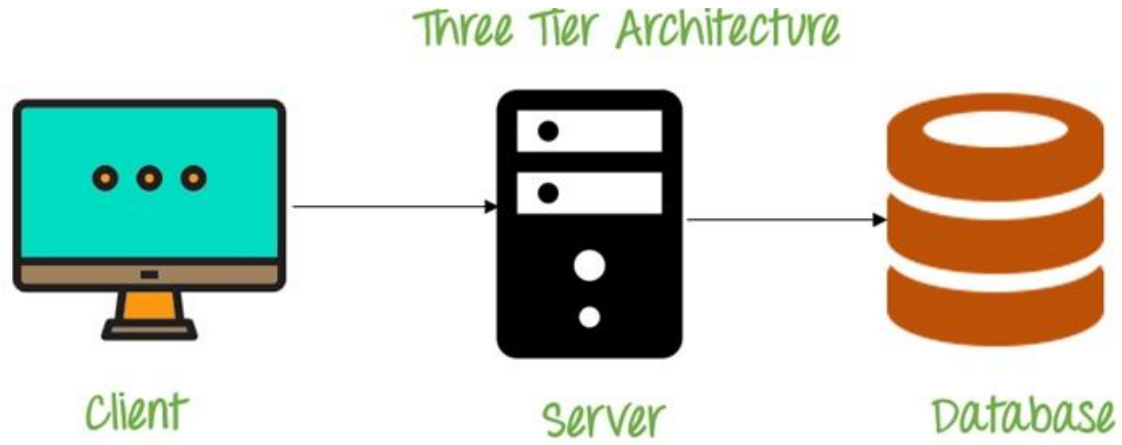

Figure 2 Three-Tier Architecture

2.4. Application life cycle

The methodology is a collection of procedures, techniques, principles, and tools that help developers building information system.

This very small and simple project uses the Waterfall method where every stage started in linear and sequence manner. However, during the development process the feedbacks and comments that received have been used to update and refine the previous steps of the previous stages. This assembles the Iterative method. As a conclusion, this project uses the combination of the Waterfall and the Iterative methodologies. In the real situation, the gathering information process starts from the problem statements and the process may involve the following activities:

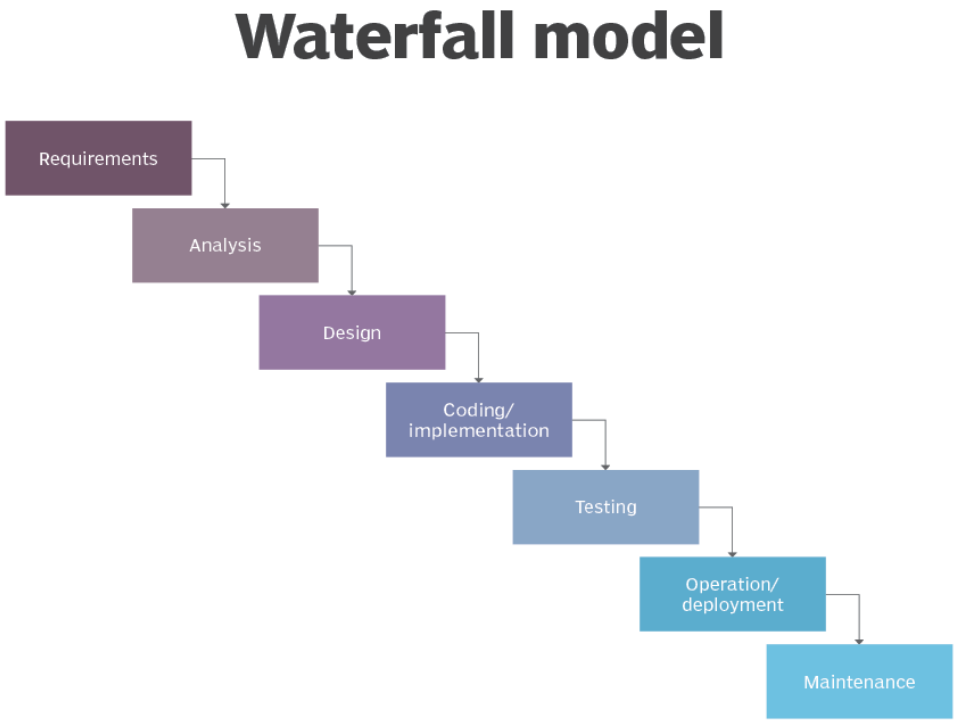

Figure 3 Waterfall Model

Conclusion

After presenting the general framework of our project, stating the problem and its solution. We will study in the following chapter the existing system in order to better understand the project environment.

Chapter Two:
Analysis and Specification of Requirements

Chapter Two: Analysis and Specification of Requirements

Introduction

Preliminary system study is the first stage of system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what actually the physical system is?

In practice, the initial system study involves the preparation of a System proposal, which lists the Problem's Definition, Objectives of the Study, Terms of reference for Study, Constraints, Expected benefits of the new system, etc.

1. Study of the existing system and critics

The communication has become the very important and necessary component in our daily life to accomplish our tasks. People have to work together cooperatively so that they come out with the solutions for certain problems. To share the information resource among the people who look for solutions must have to be connected to each other to facilitate the communication process. While the majority of medical applications use a local database to store and share, on a limited way, their huge data.

In addition, the current applications and ways used for managing the laboratories raise many problems such as:

- Large volume of information processed manually, which sometimes leads to errors in the preparation of documents (the situation of laboratories using traditional methods).
- Difficulty of storage due to large number of archives (when using local servers with limited storage).
- Lack of simplicity in the use of other applications.
- Difficult search of registers.
- Insecurity of information.
- Possibility of error in the filling of the different documents.
- Possibility of error in the calculation of exam amounts.

2. Solution proposed

For these previous reasons and critics, we have created this application to organize their work. We have proposed to create a web application that helps to facilitate the management of medical laboratories as well as improve their services.

3. List of Requirements

In this stage, we need to list the functional and non-functional requirements in the provided formatted documents based on the information gathered from the previous completed tasks. The functional requirements define the basic functions or services of the system. The non-functional requirements are the constraints on the services and the functionality of the system. The recommendations of these things can be based on the IEEE standard, Software Requirement Specification (SRS): std-830-1993 and the revised version, std-830-1998.

3.1. Functional Requirements:

The Functional Requirements Specification documents the operations and activities that a system must be able to perform are:

- This project will handle the whole activities of a medical laboratory.
- It provides to the biologist to manage the staff and all other required operation.
- It facilities to the medical assistant to manage and handle his tasks (patients, payments and reports management).
- It authorize the senior technician to manage and handle his tasks (results and reports management).
- It provides to the patient to consult his results online.

3.2. Non-Functional Requirements

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behaviour. One could also think of non-functional requirements as quality attributes for of a system.

- Security requirements are important factors in this system as classified data will be stored in the database.
- The system should be easy to use
- The system should response at the time
- System should provide specific information for specific user
- Supportability; the application can be easy updated and upgraded.

Based on the main functional requirements, on the next stage we start preparing the flow-of-event documents together with the use case diagrams respectively.

4. General Use Case diagram:

From the functional and non-functional requirement list, we prepare the use case diagrams followed by the description in the flow-of-event documents (in the next chapter). Having identified use cases and actors from the requirement list, a use case diagram can be constructed. A use case diagram is meant to show relationships between use cases and actors. The relationships may include extend, include and association.

4.1. Identification of Actors:

An Actor in the Unified Modeling Language (UML) specifies a role played by a user or any other system that interacts with the subject.

We will show the actor that intervenes in our application:

- The biologist: admin: he is the responsible of the management
- The medical assistant and the senior technician: users: they can do all the functionalities of the admin except the management of user
- The patient: client: he can only consult his results

4.2. Identification of Uses Cases:

The different tasks provided by our application will be shown in the general use case diagram.

- Use case manage The Staff
- Use case manage The Patients
- Use case manage The Results
- Use case manage The Reports
- Use case manage The Payments
- Use case consult results for the patients

The textual description of these cases will be shown in the next chapter.

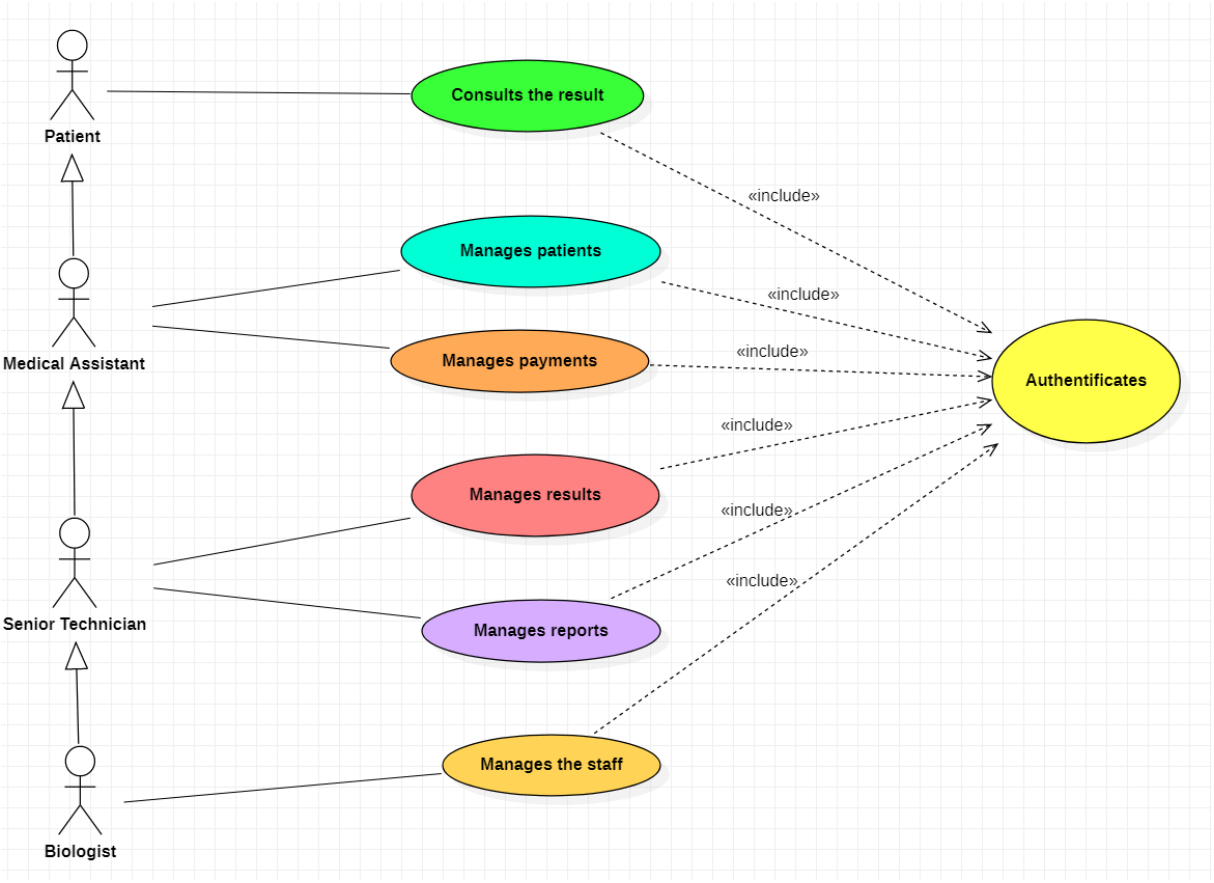

Figure 4 General Use Case

Conclusion

After giving an overview of the existing system, we announced the proposed solution. Then we defined the functional requirement that can't be only supported by the architecture. The non-functional requirements need to be explicitly taken into account too. To more understand our system, in the next chapter, we will see the detailed solution conception.

Chapter Three: Conceptual Study

Chapter 3: Conceptual Study

Introduction

A good detailed design is more likely to lead to a good project deliverable. If the detailed design is poor, the project deliverables are much less likely to meet requirements!

We provide the static view of our application through the use case diagram and the class diagram and the dynamic view by the sequence diagram and the activity diagram.

1. Design Methodology

Unified Modelling language (**UML**) is a standardized modelling language enabling developers to specify, visualize, construct and document artefacts of a software system. Thus, **UML** makes these artefacts scalable, secure and robust in execution. **UML** is an important aspect involved in object-oriented software development.

2. Detailed Design:

2.1. Detailed description of use cases:

The main objective of creating Use Case Diagram is for describing the behaviour of the target system from an external point of view. Besides drawing the diagram, Visual Paradigm allows you to detail the document's requirements through the Use Case Description. All these information can be output into HTML/ PDF/ MS Word formats.

The following are the use case diagrams of the main flows and sub-flows. Every use case diagram is been followed by the respective flow-of-event document.

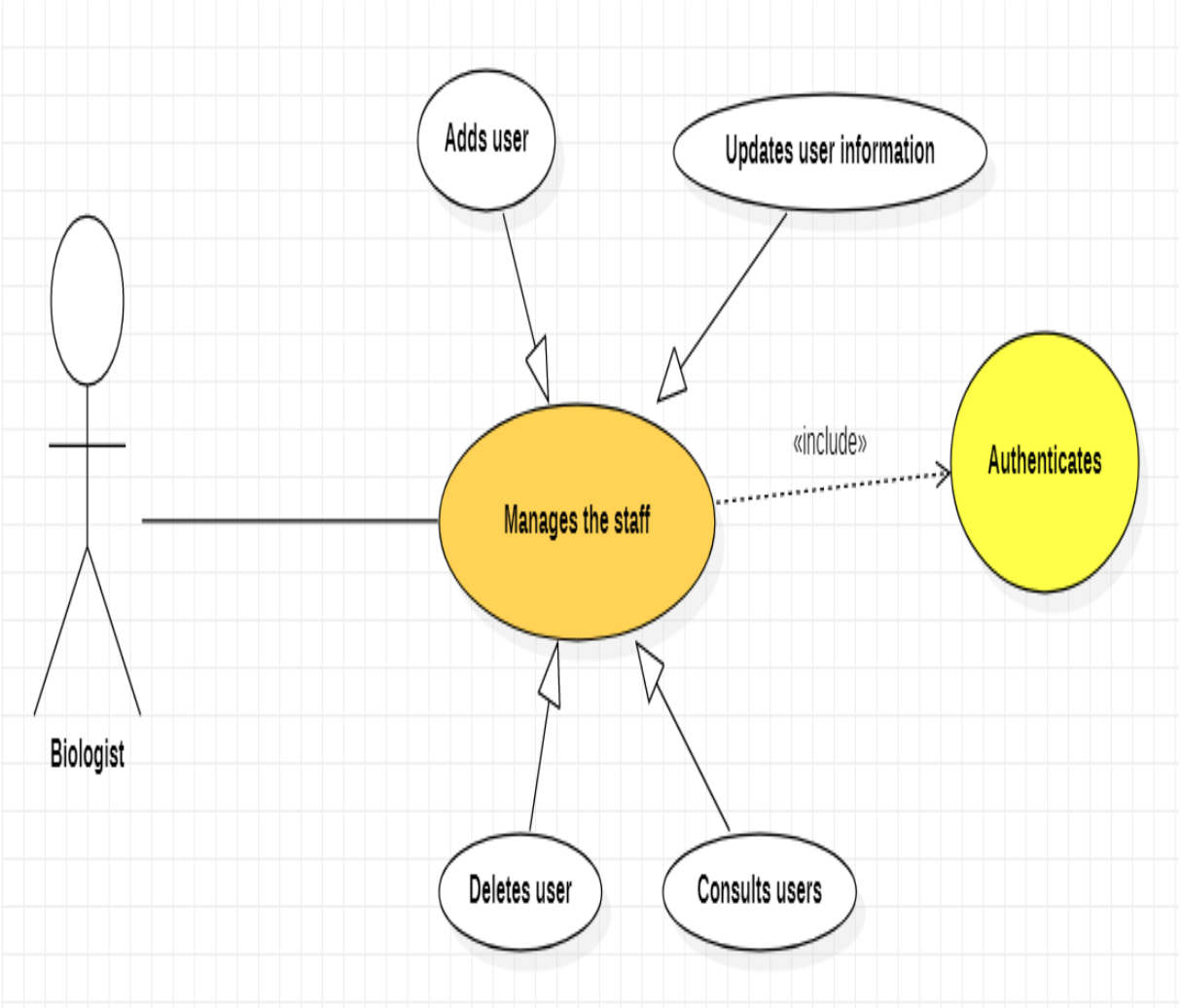

Figure 5 Manage Staff Use Case

Table 1 Manage Staff Use Case Description

1.		**Use Case : Manage the staff**
	1.1	**Precondition** Only the appointed supervisors that have the username and password can login into the application. The data entry is done daily.
	1.2	**Main Flow** This use case starts when the appointed supervisor login into the web application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, supervisor is logged into application and prompts supervisor to manage the staff list. This main flow consists of several sub flows that are: ▪ If the activity selected is ADD, the S-1: Add a user subflow is performed. ▪ If the activity selected is DELETE, the S-2: Delete a user subflow is performed. ▪ If the activity selected is CONSULT, the S-3: Consult a user information subflow is performed. ▪ If the activity selected is UPDATE, the S-4: Update a user subflow is performed.
	1.3	**Subflows** **S-1: Add a user.** Fill the associed form to add a user.
		S-2: Delete a user. The admin can delete a user.
		S-3: Consult a user. The supervisor can search and consult a defined user or an information.
		S-4: Update a user Based on a given form, the supervisor may change or update an information related to a saved user.
	1.4	**Alternative Flows** **E-1**: Supervisor forgot his/her username or/and password. Supervisor needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to supervisor. **E-2:** When invalid username and/or password entered, supervisor can re-enter the combination three times, then the system will be locked and the use case ends. Supervisor need to wait 30 seconds after the application lock out in order to re-login.

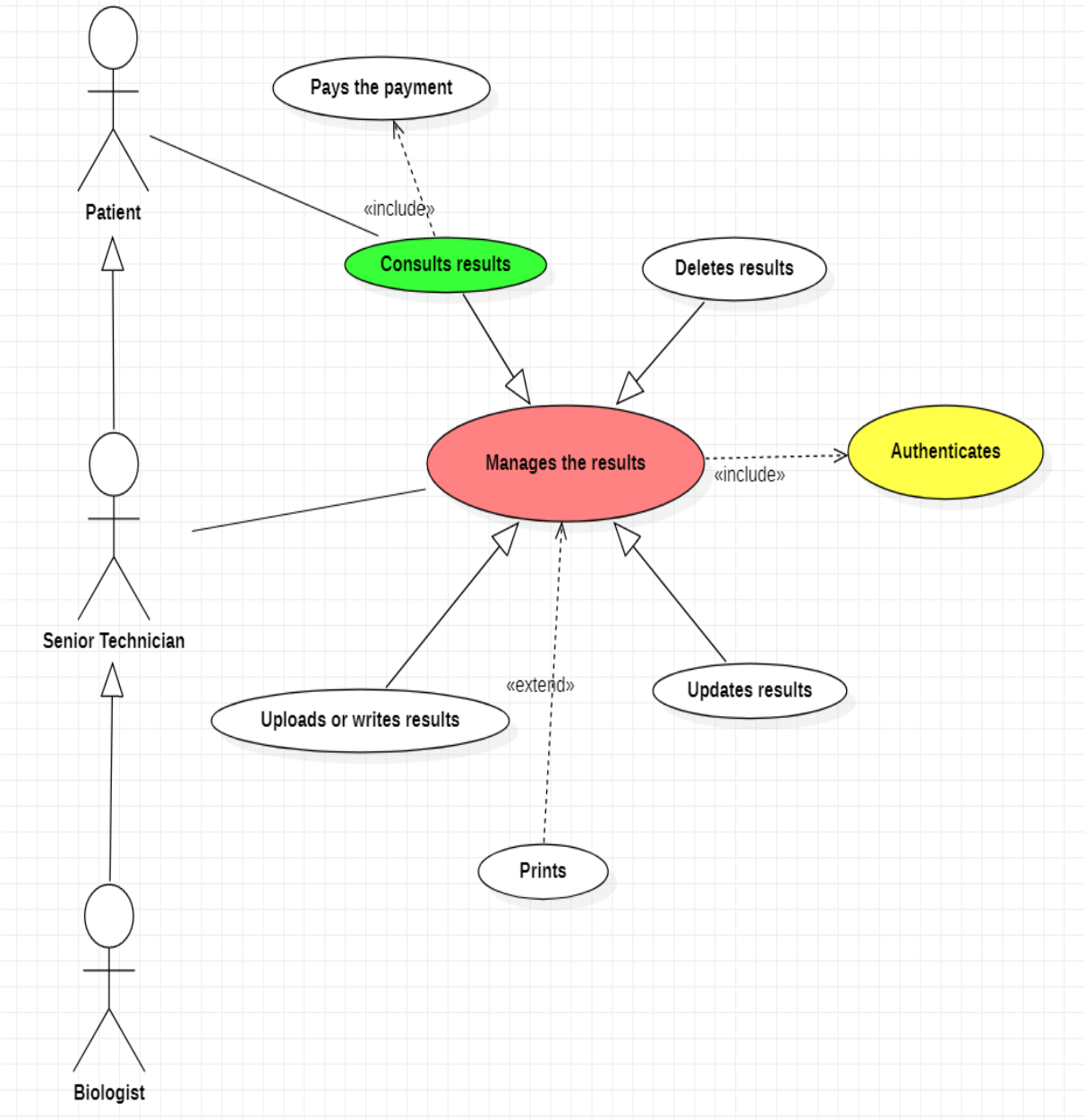

Figure 6 Manage Results Use Case

Table 2 Manage Results Use Case Description

2.		**Use Case : Manage Results**
	2.1	**Precondition** Only the appointed senior technician/the patient that have the username and password can login into the application. The data entry is done daily (for the senior technician).
	2.2	**Main Flow** This use case starts when the appointed The senior technician/the patient login into the web application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, he is logged into application and prompts him to manage the results/consult only. The patient have to complete the payment (E-3) to could enter through the application. This main flow consists of several sub flows that are: ▪ If the activity selected is ADD, the S-1: Add a result subflow is performed. ▪ If the activity selected is DELETE, the S-2: Delete a result subflow is been performed. ▪ If the activity selected is CONSULT, the S-3: Consult a result information subflow is performed. ▪ If the activity selected is UPDATE, the S-4: Update a result subflow is performed. ▪ If the activity selected is PRINT, the S-5: Print a result subflow is performed.
	2.3	**Subflows** **S-1: Add a result.** Upload the result of the patient on the indicated form.
		S-2: Delete a result. The senior technician delete a result.
		S-3: Consult a result. The senior technician can search and consult a defined result. About the patient, he can only access to see his result.
		S-4: Update a result. Based on a given form, the senior technician may change or update an information related to a result.
		S-5: Print a result. The senior technician can print a result. The patient can print his own result.
	2.4	**Alternative Flows** **E-1**: The senior technician/the patient forgot his/her username or/and password. He needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to the senior technician/ the patient. **E-2:** When invalid username and/or password entered, the senior technician/the patient can re-enter the combination three times, then the system will be locked and the use case ends. The senior technician/the patient need to wait 30 seconds after the application lock out in order to re-login. **E-3:** The patient have to complete the payment task with the medical assistant, to could access to the application to consult his result after an indicated time.

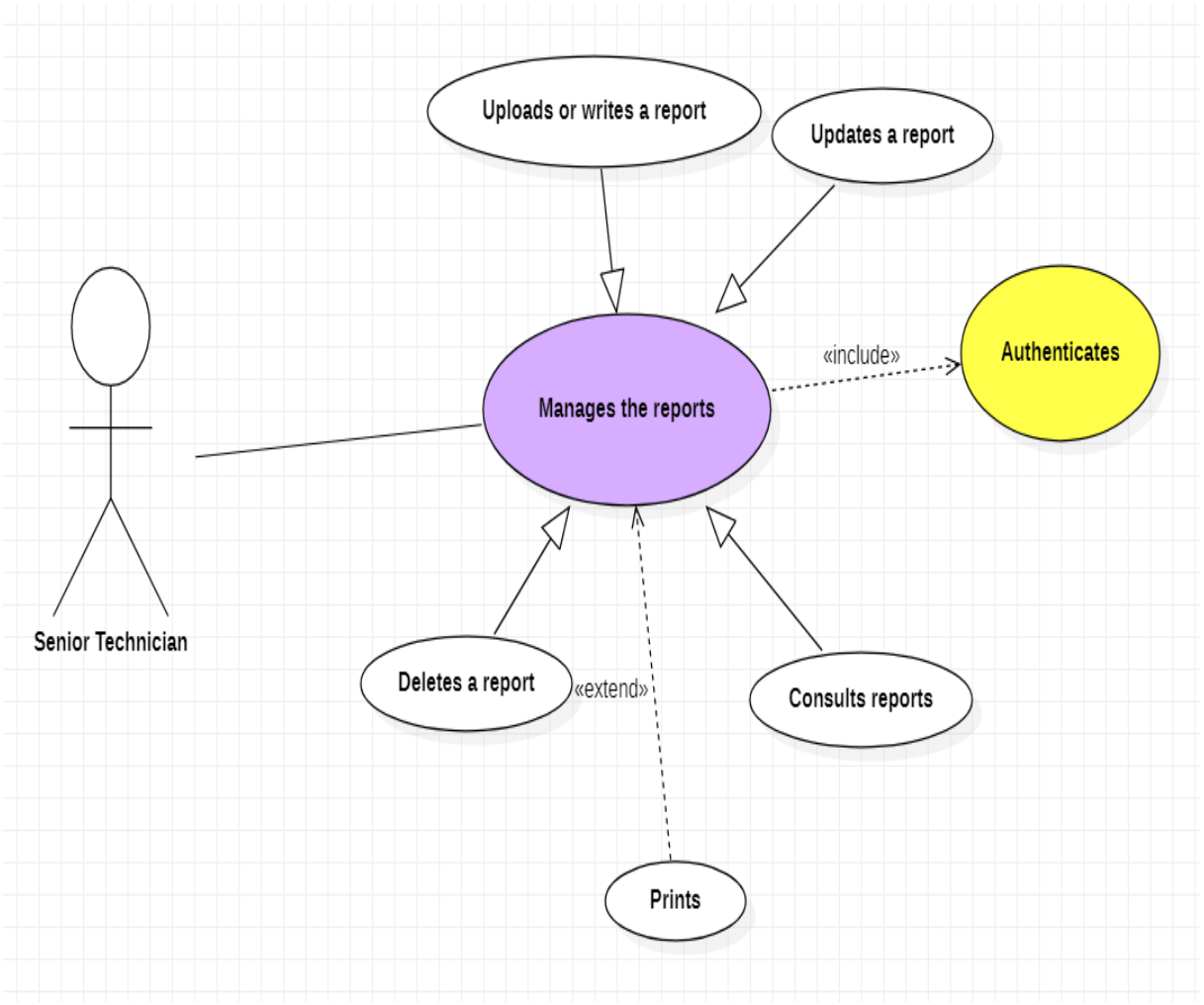

Figure 7 Manage Reports

Table 3 Manage Reports Use Case Description

3.		**Use Case : Manage Reports**
	3.1	**Precondition** Only the appointed senior technician that have the username and password can login into the application. The data entry is done daily.
	3.2	**Main Flow** This use case starts when the appointed senior technician login into the web application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, he is logged into application and prompts him to manage the reports. This main flow consists of several sub flows that are: ▪ If the activity selected is ADD, the S-1: Add a report subflow is performed. ▪ If the activity selected is DELETE, the S-2: Delete a report subflow is been performed. ▪ If the activity selected is CONSULT, the S-3: Consult a report information subflow is performed. ▪ If the activity selected is UPDATE, the S-4: Update a report subflow is performed. ▪ If the activity selected is PRINT, the S-5: Print a report subflow is performed.
	3.3	**Subflows** **S-1: Add a report.** Upload the report of the patient on the indicated form.
		S-2: Delete a report. The senior technician delete a report.
		S-3: Consult a report. The senior technician can search and consult a defined report.
		S-4: Update a report. Based on a given form, the senior technician may change or update an information related to a report.
		S-5: Print a report. The senior technician can print a report.
	3.4	**Alternative Flows** **E-1**: The senior technician forgot his/her username or/and password. He needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to the senior technician. **E-2:** When invalid username and/or password entered, the senior technician can re-enter the combination three times, then the system will be locked and the use case ends. The senior technician need to wait 30 seconds after the application lock out in order to re-login.

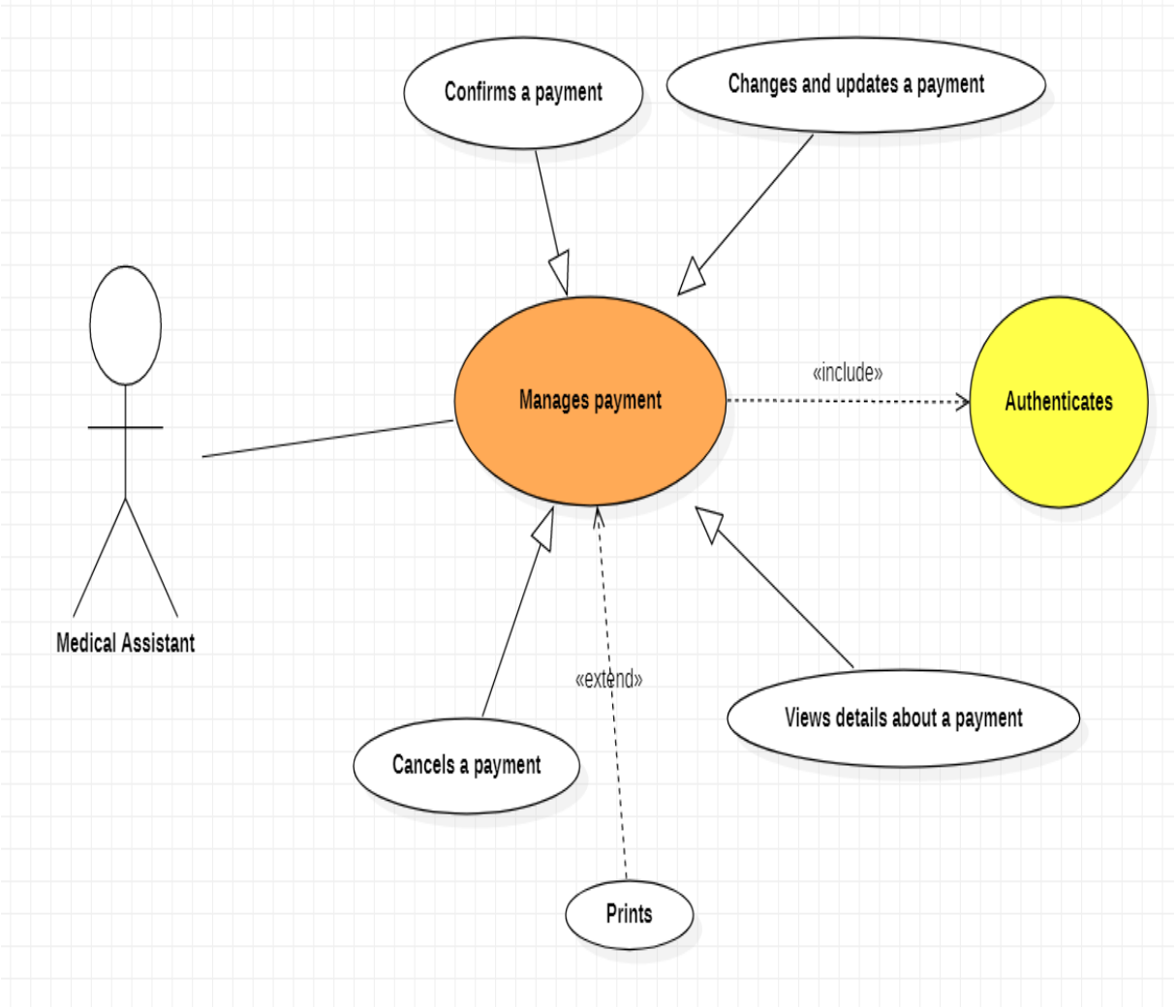

Figure 8 Manage Payment Use Case

Table 4 Manage Payments Use Case Description

4.		**Use Case : Manage payment**
	4.1	**Precondition** Only the appointed medical assistant that have the username and password can login into the application. The data entry is done daily.
	4.2	**Main Flow** This use case starts when the appointed medical assistant login into the web application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, he is logged into application and prompts him to manage payments. This main flow consists of several sub flows that are: ▪ If the activity selected is CONFIRM, the S-1: Add and confirm a payment subflow is performed. ▪ If the activity selected is CANCEL, the S-2: Cancel a payment subflow is been performed. ▪ If the activity selected is CONSULT, the S-3: Consult a payment information subflow is performed. ▪ If the activity selected is UPDATE, the S-4: Update a payment subflow is performed. ▪ If the activity selected is PRINT, the S-5: Print a paymentt subflow is performed.
	4.3	**Subflows** **S-1: Confirm a payment.** The medical assistant make the payment of the patient by filling the indicated form. The patient should pay the indicated sum.
		S-2: Cancel a payment. The medical assistant cancel the payment in progress for a patient.
		S-3: Consult a payment. The medical assistant can search and consult a defined payment.
		S-4: Update a payment. Based on a given form, the medical assistant may change or update an information related to a payment.
		S-5: Print a payment. The medical assistant can print a payment.
	4.4	**Alternative Flows** **E-1**: The medical assistant forgot his/her username or/and password. He needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to the medical assistant. **E-2:** When invalid username and/or password entered, the medical assistant can re-enter the combination three times, then the system will be locked and the use case ends. The medical assistant needs to wait 30 seconds after the application lock out in order to re-login.

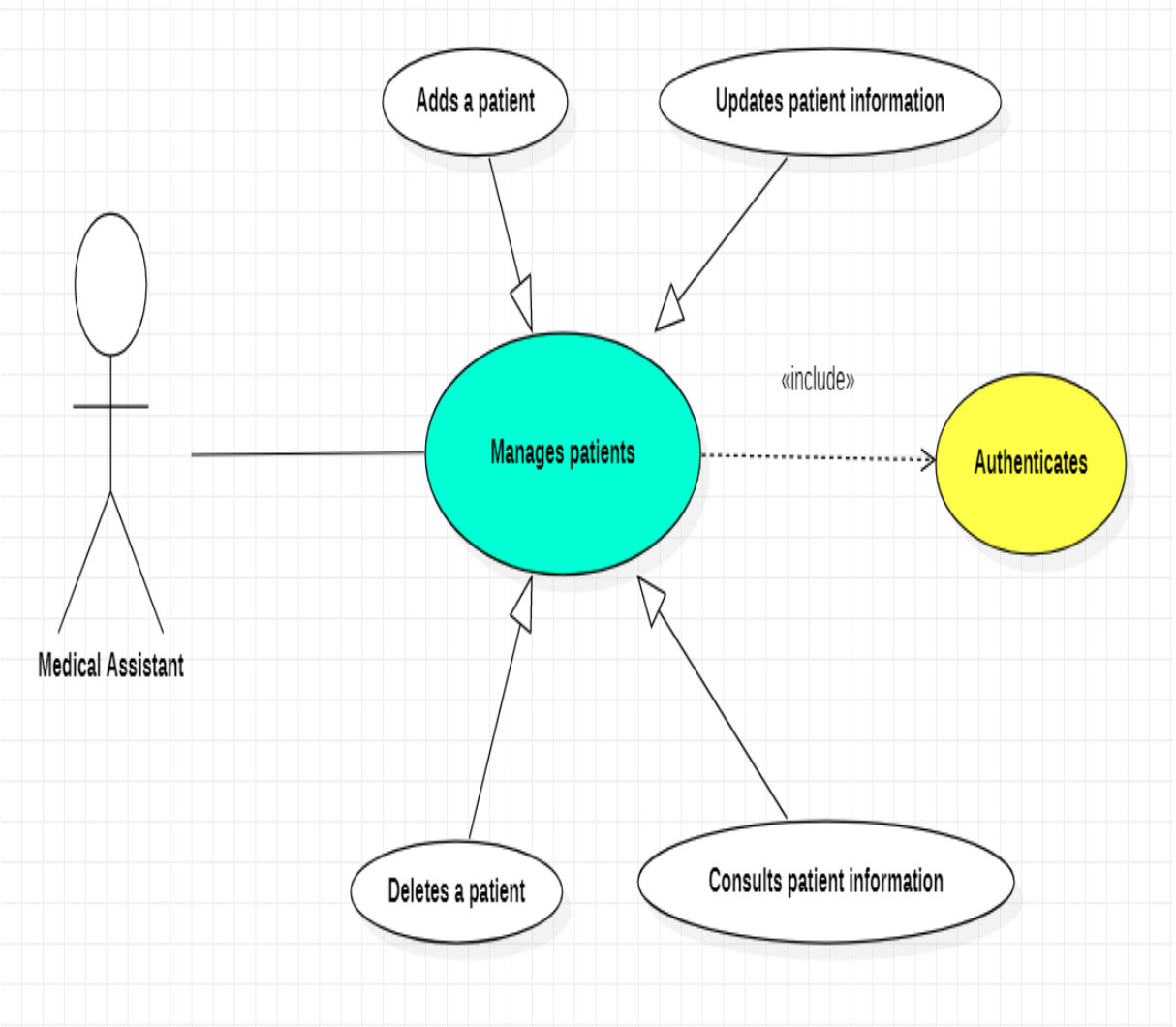

Figure 9 Manage Patients Use Case

Table 5 Manage Patients Use Case Description

5.		**Use Case : Manage Patients**
	5.1	**Precondition** Only the appointed medical assistant that have the username and password can login into the application. The data entry is done daily.
	5.2	**Main Flow** This use case starts when the appointed medical assistant login into the web application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, he is logged into application and prompts him to manage patients. This main flow consists of several sub flows that are: ▪ If the activity selected is ADD, the S-1: Add a patient subflow is performed. ▪ If the activity selected is DELETE, the S-2: Delete a patient subflow is been performed. ▪ If the activity selected is CONSULT, the S-3: Consult a patient information subflow is performed. ▪ If the activity selected is UPDATE, the S-4: Update a patient subflow is performed.
	5.3	**Subflows** **S-1: Add a patient.** The medical assistant fill the form of adding a patient.
		S-2: Delete a patient. The medical assistant delete a patient from the database.
		S-3: Consult a patient. The medical assistant can search and consult a defined pateint.
		S-4: Update a patient. Based on a given form, the medical assistant may change or update an information related to a patient.
	5.4	**Alternative Flows** **E-1**: The medical assistant forgot his/her username or/and password. He needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to the medical assistant. **E-2:** When invalid username and/or password entered, the medical assistant can re-enter the combination three times, then the system will be locked and the use case ends. The medical assistant needs to wait 30 seconds after the application lock out in order to re-login.

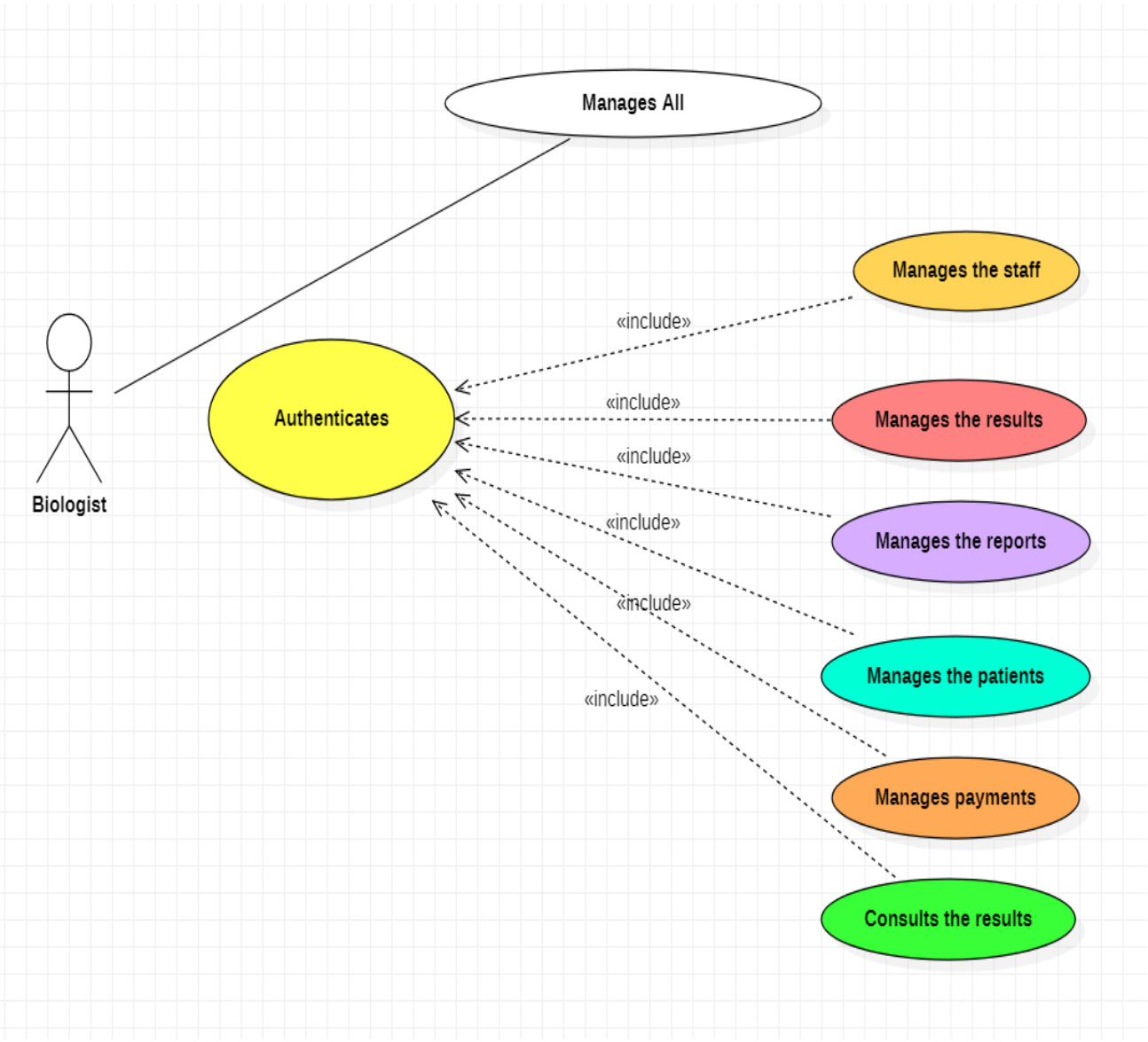

Figure 10 Authentication Use Case

Table 6 Authentication Use Case Description

6.		**Use Case : Authenticate User**
	6.1	**Precondition** Only the designated or appointed users that have username and password can login into application.
	6.2	**Main Flow** This use case starts when user login into the application by entering his/her username and password (E-1). The system verifies the username and password combination validity (E-2). If valid, the users will have access to the application. This main flow contains the following components: ▪ If the activity selected is LOGIN, the S-1: Login system sub flow is performed. ▪ If activity selected is LOGOFF, S-2: Logoff system (E3) subflow is performed.
	6.3	**Subflows** **S-1: Login system** From the login screen the designated user need to key-in his/her username and password in the given fields. This task is to make sure only the authorized person can access the system.
		S-2: Logoff system After completing his/her tasks, user has to logoff from the system to prevent unauthorized access.
	6.4	**Alternative Flows** **E-1**: User forgot his username or/and password. User needs to fill in his/her email address in the provided field and the use case ends. The system will email the username or password to the user. **E-2:** When invalid username and/or password entered, user can re-enter the combination three times, then the system will be locked and the use case ends. User need to wait an hour after the system lock out in order to re-logon. **E-3:** If user forgets to logoff, the system will auto-logoff after in idle condition for 30 seconds.

We proceed to the next stage, finding classes and building the sequence diagrams.

2.2. Class Diagram Definition:

In software engineering, a class diagram in the Unified Modeling Language (UML) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

The following section list down all the found classes in our project. It is our first attempt and no collaboration shown here. We have eight classes:

1. User
2. Biologist
3. Senior Technician
4. Medical Assistant
5. Patient
6. Result
7. Report
8. Payment
9. Analysis Category
10. Room

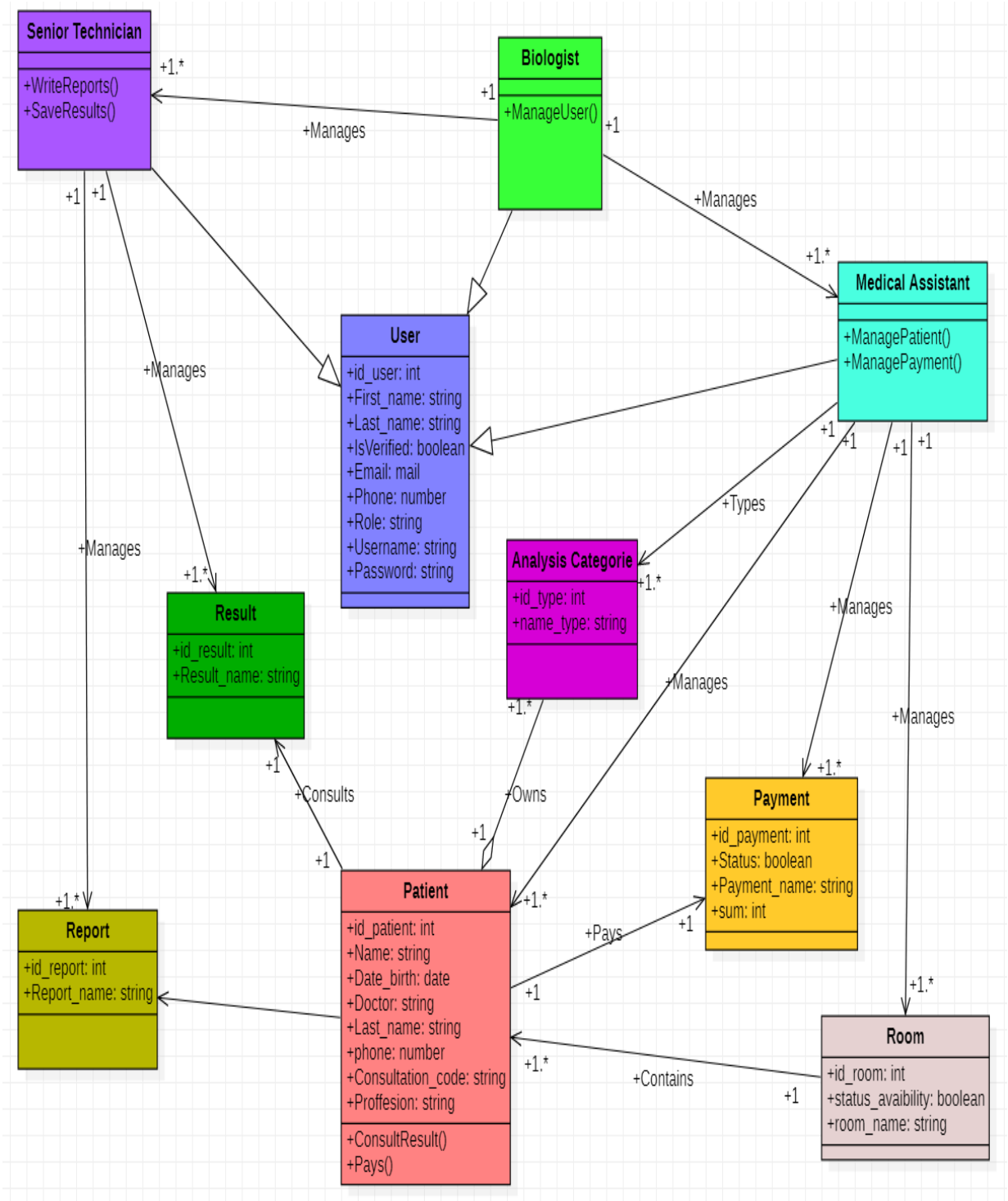

Figure 11 Class Diagram

2.3. Sequence Diagram Definition

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.
Every use case in the use case diagram will be converted to the sequence diagram. The following are the sequence diagrams from our first attempt. After eliminating the duplicated diagrams, we have five sequence diagrams as listed below.

1. Sequence Authentication diagram
2. Sequence Add Result diagram
3. Sequence Delete User diagram
4. Sequence Update Report diagram
5. Sequence Consult Patient diagram

- **Authenticate sequence diagram:**

  To access our application, the user must authenticate by entering his login and password. The authentication process can be summarized in the following activity diagram:

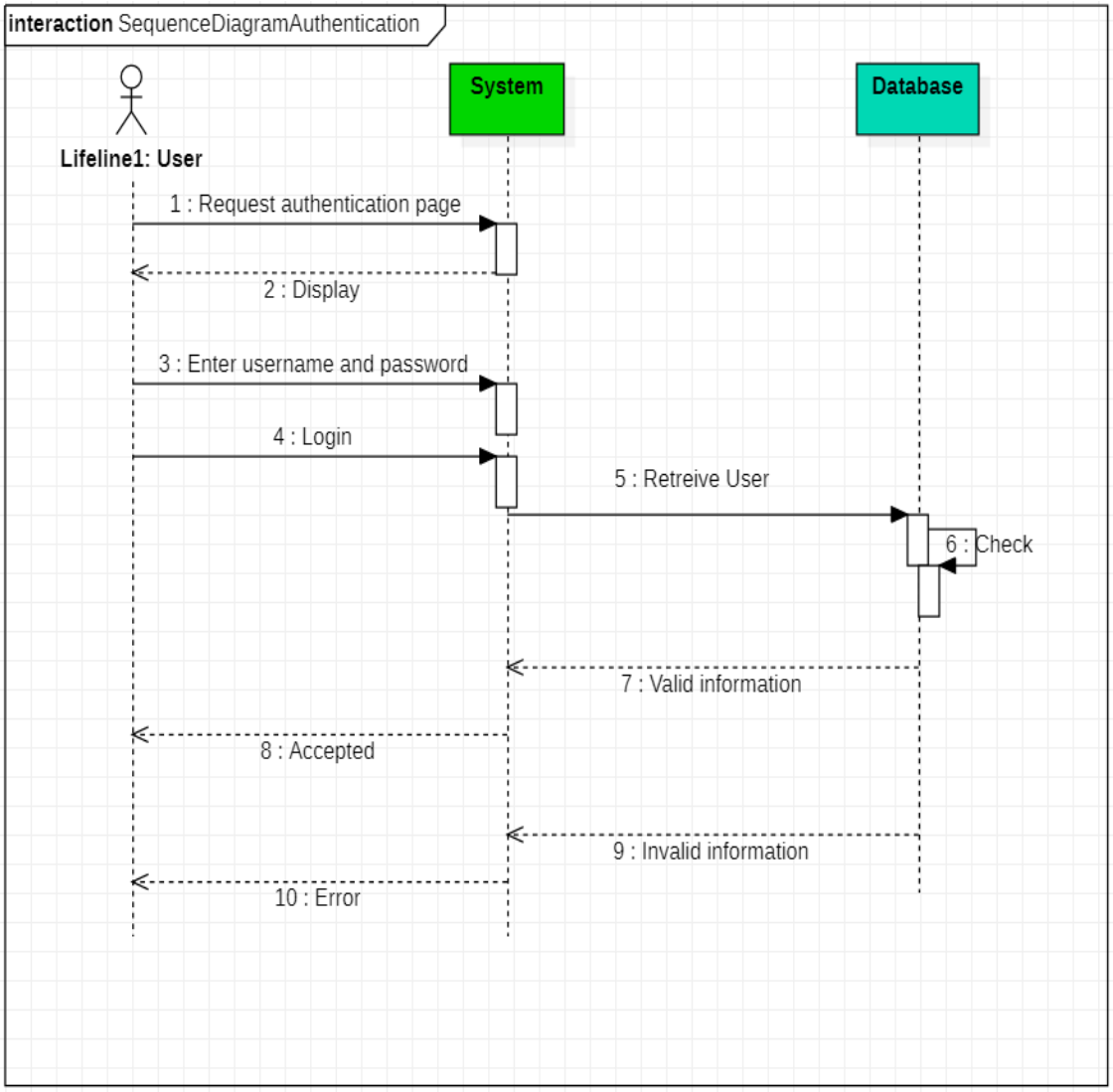

Figure 12 Sequence Diagram Authentication

- **Add Result Sequence diagram:**

To add student, the user must open the results Entry interface. Then he fill the form or to upload a result file. After that the system verify the data and the result will be registered.

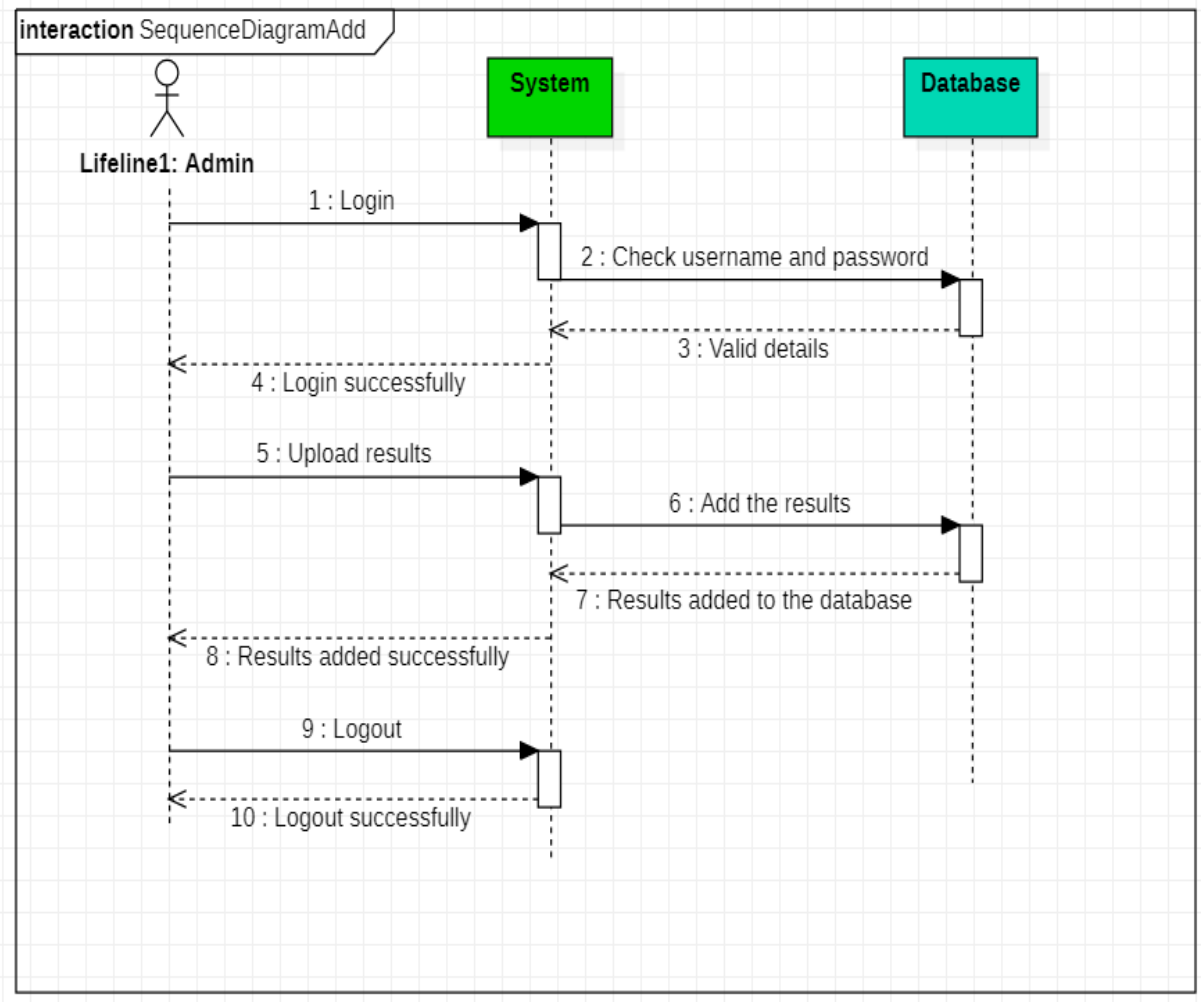

Figure 13 Add Result Sequence Diagram

- **Delete User Sequence Diagram:**

To delete a user, the admin must open the users interface. Then he search for a specific user type and delete him. The process of deleting a user will be shown in the following figure.

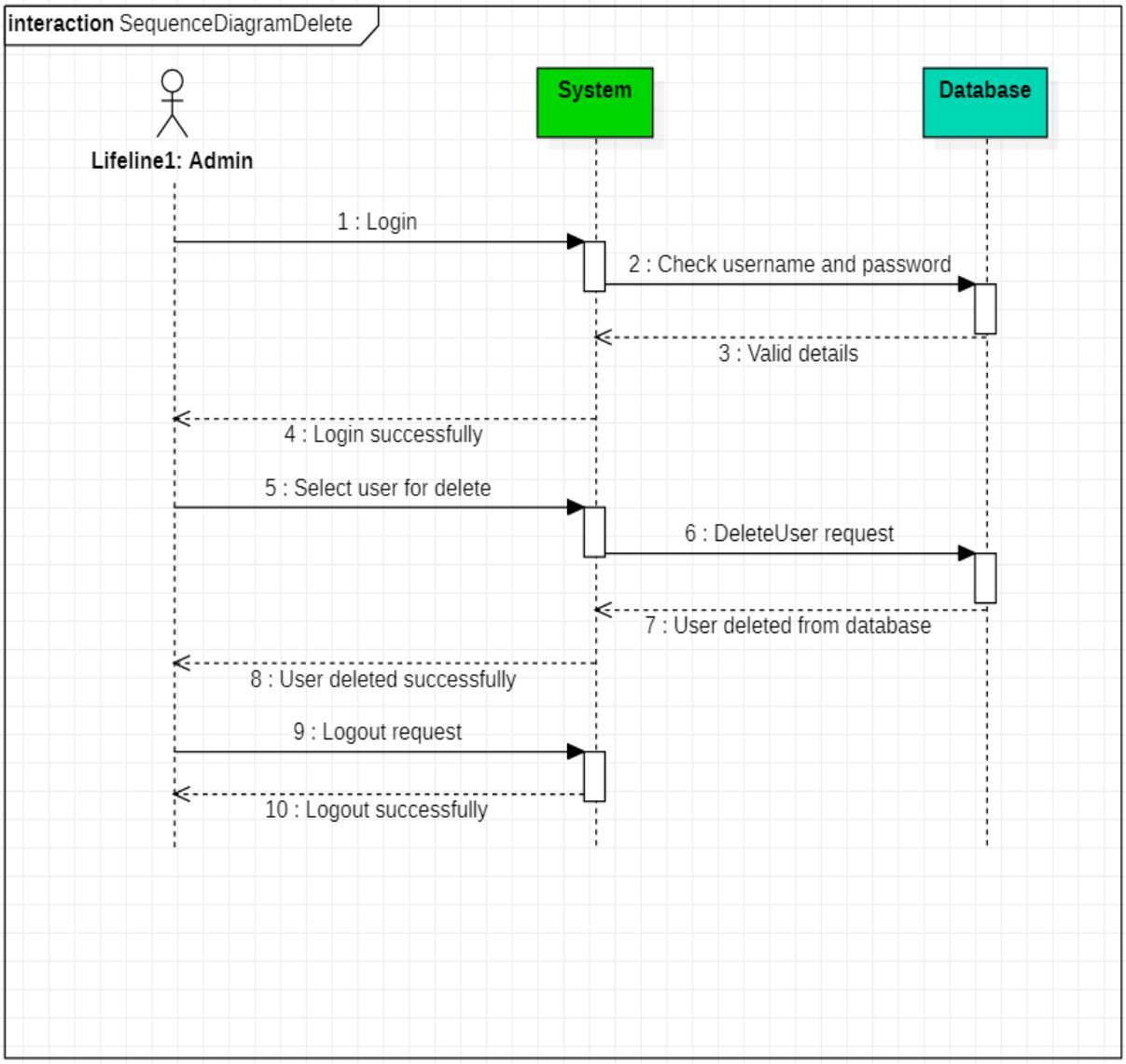

Figure 14 Delete User Sequence Diagram

- **Update Report Sequence Diagram:**

The process of updating a report will be shown in the following figure.

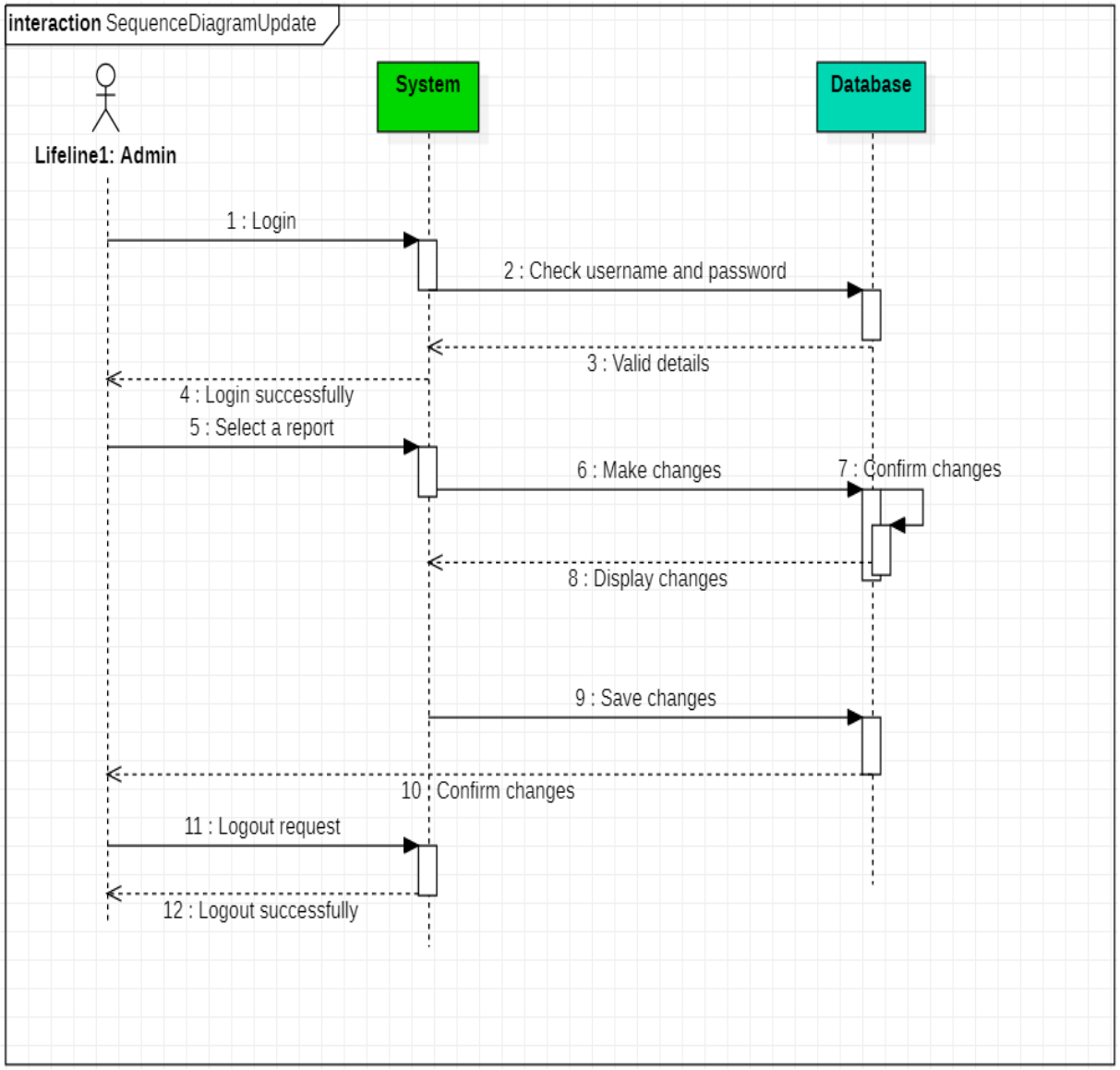

Figure 15 Update Report Sequence Diagram

- **Consult Patient Sequence Diagram:**

The process of updating a patient will be shown in the following figure.

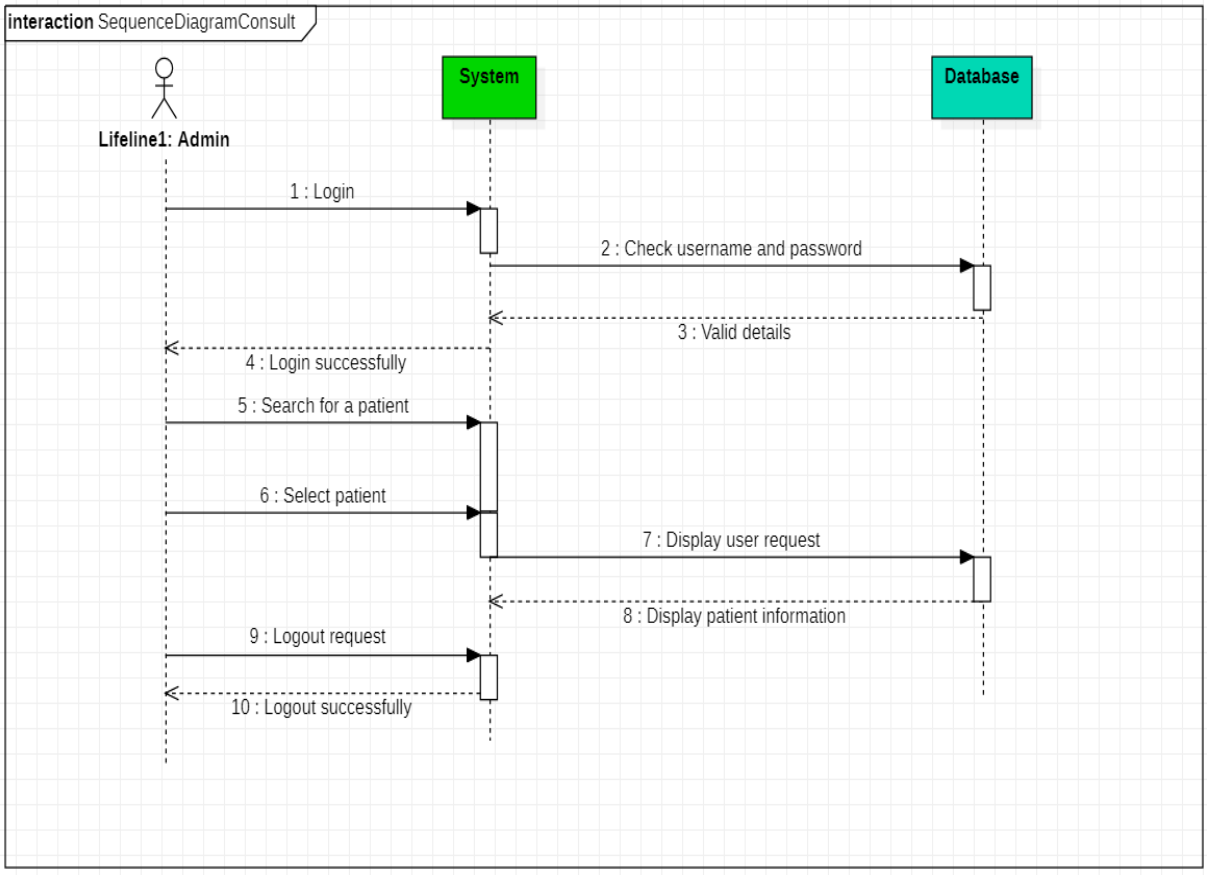

Figure 16 Consult Patient Sequence Diagram

2.4. Activity diagrams

Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity.

The following are the activity diagrams of Admin and Patient using Star UML, with the CRUD and Logout Activity Diagrams using the web application Creately.

- **General activity diagram**

The following activity diagram summarizes the different workflows of our application (admin side).

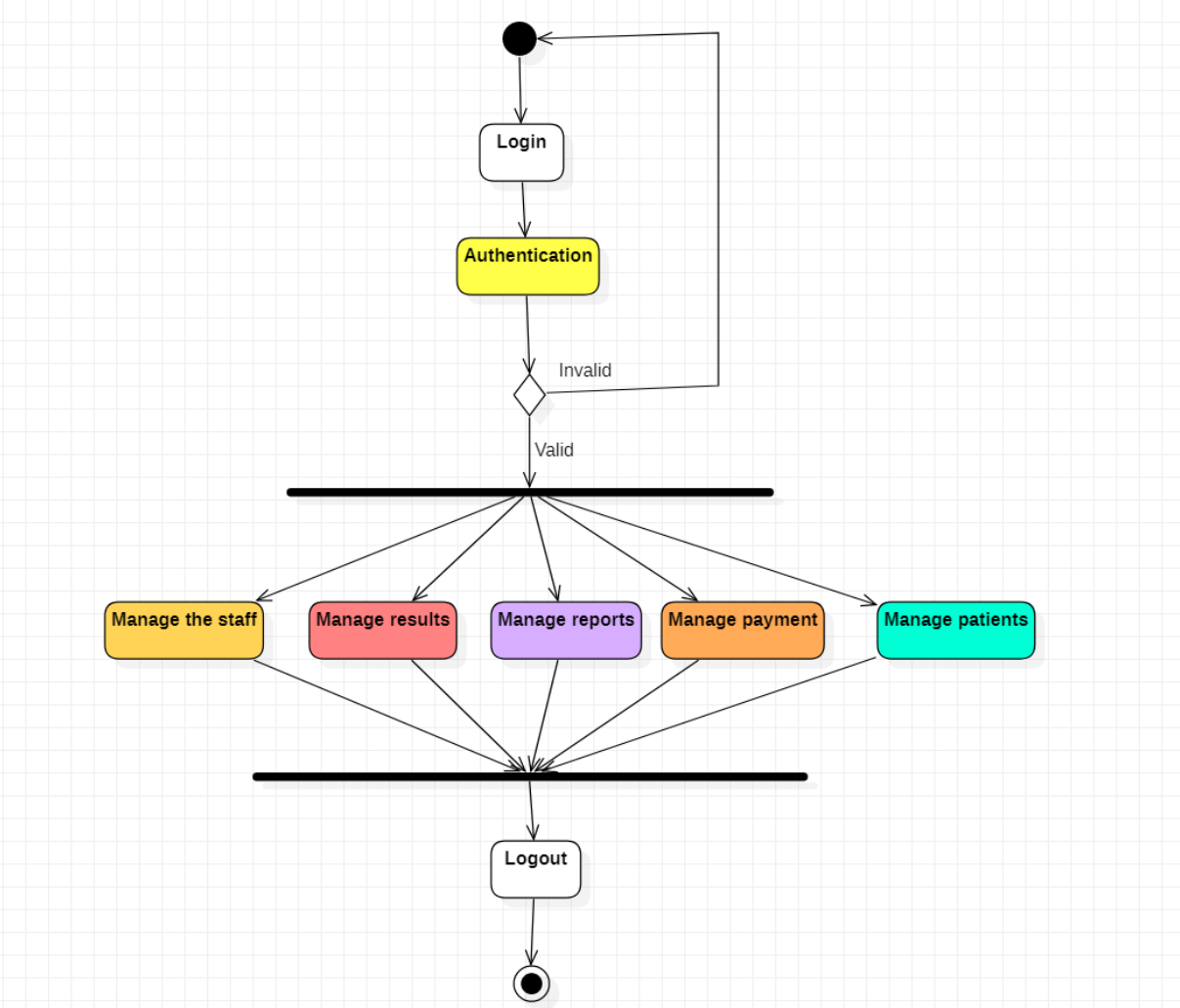

Figure 17 General Activity Diagram

- **Add User activity diagram**

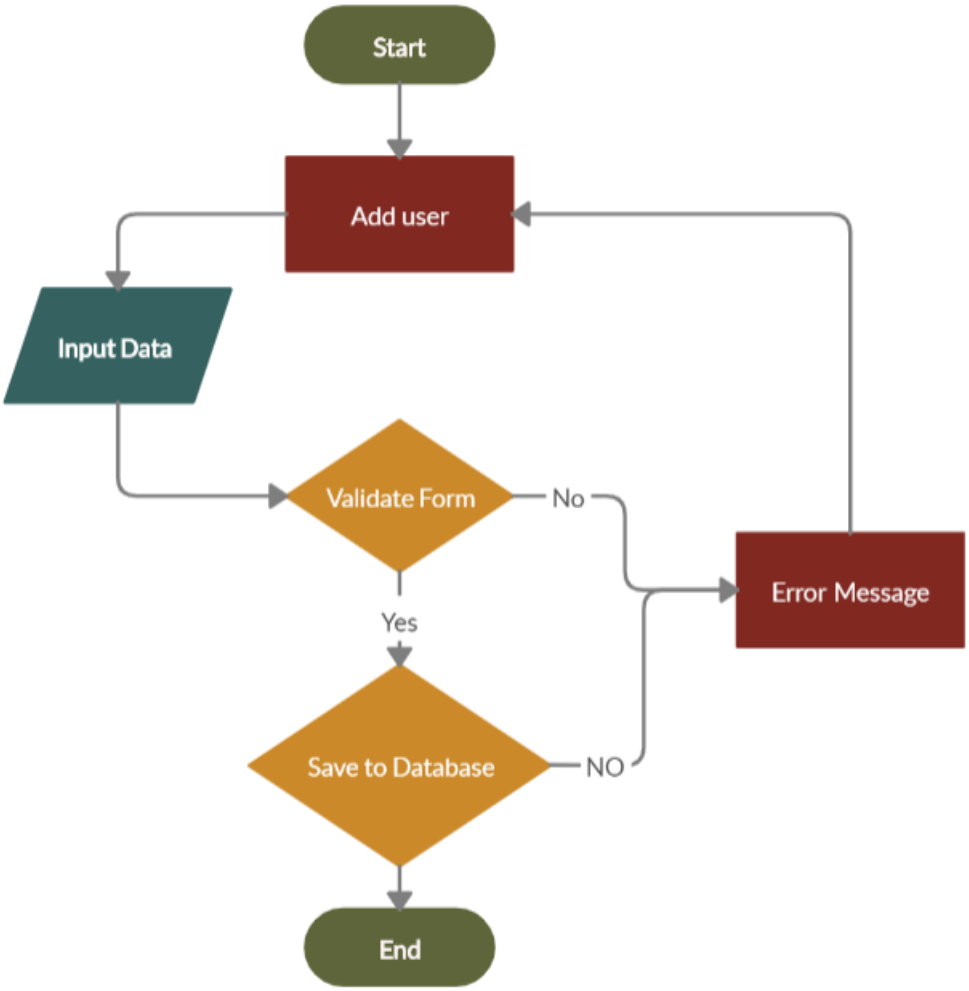

Figure 18 Add User Activity Diagram

- **Delete Report activity diagram**

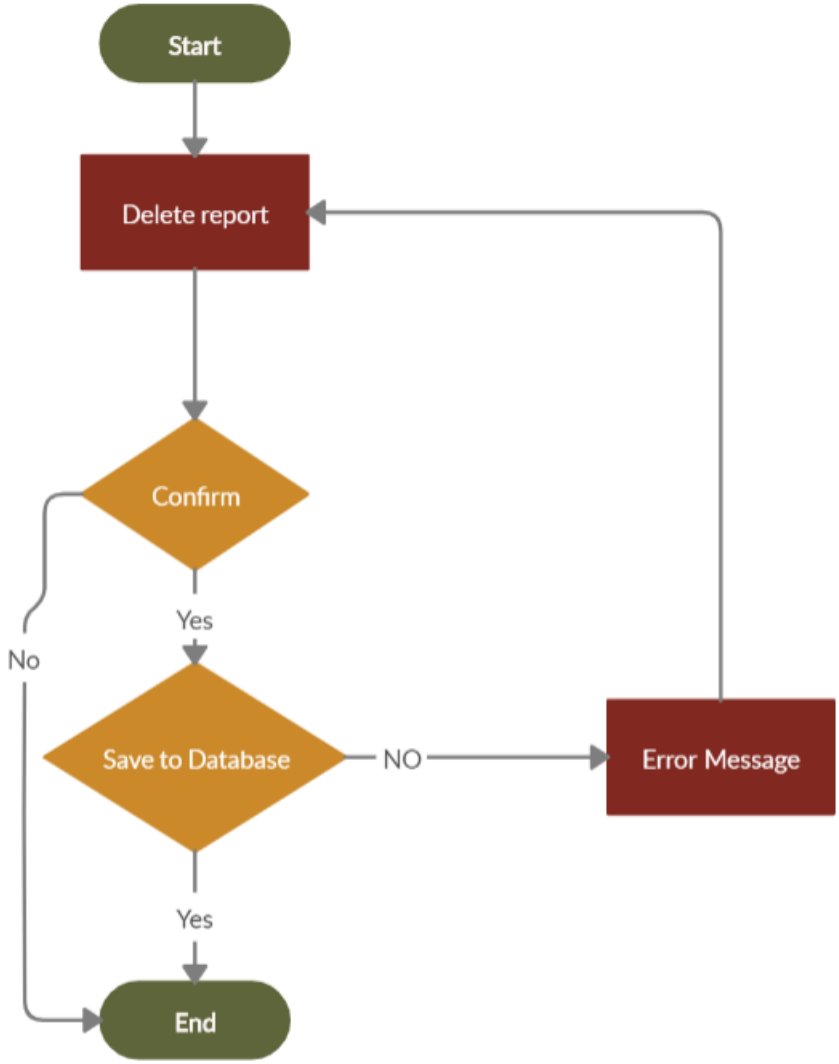

Figure 19 Delete Report Activity Diagram

- **Update Result activity diagram**

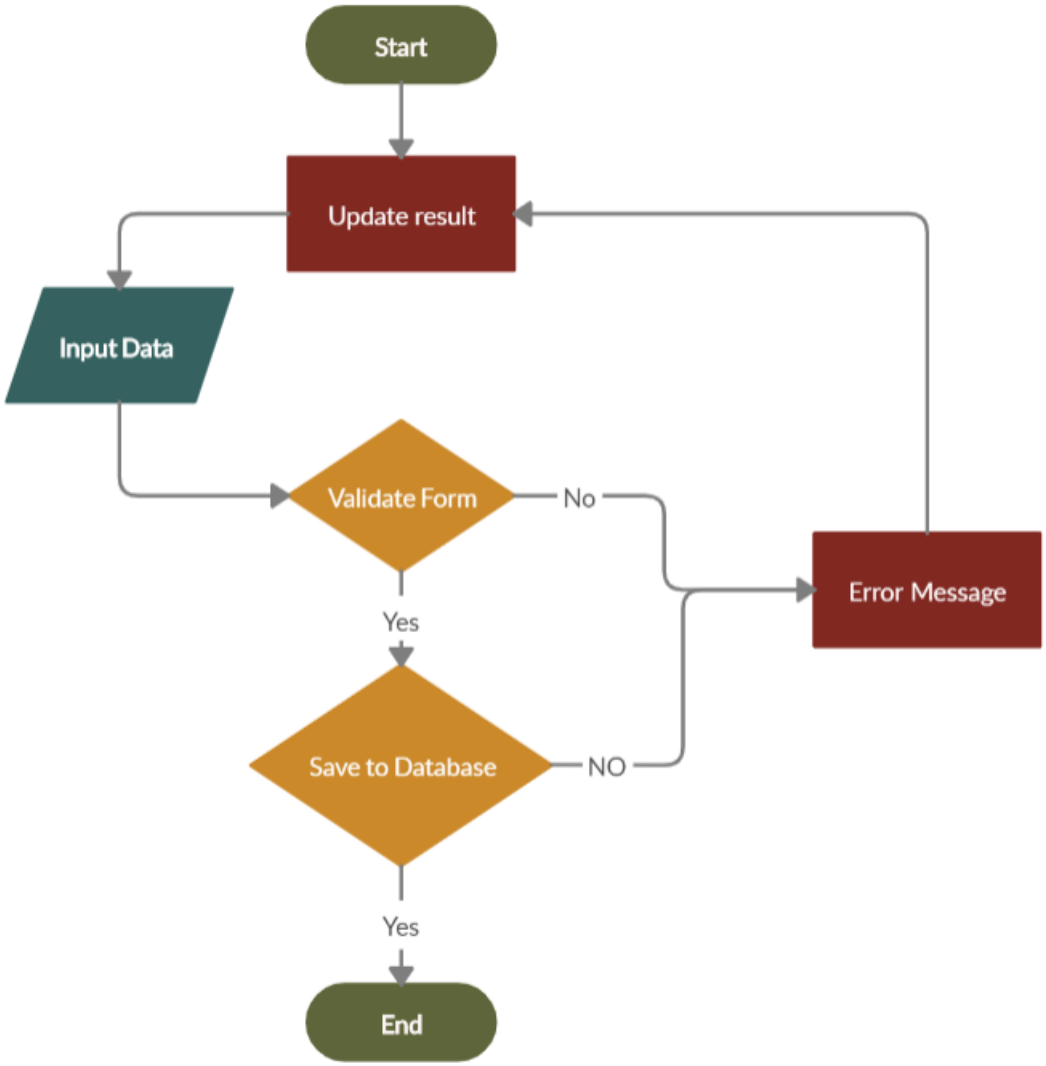

Figure 20 Update Result Activity Diagram

- **Consult Patient activity diagram**

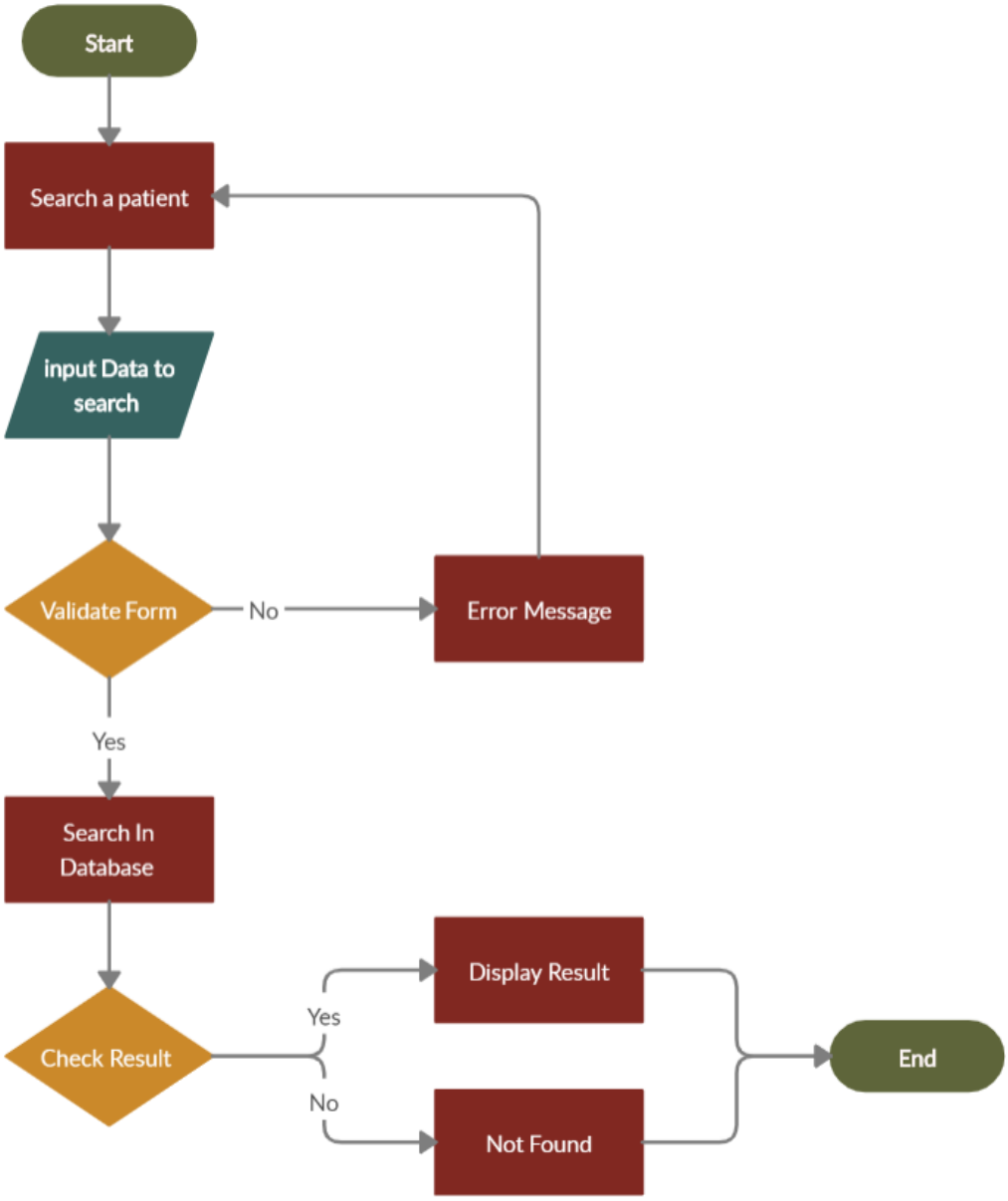

Figure 21 Consult Patient Activity Diagram

Conclusion

Throughout this chapter, we detailed the relationships between objects, system and components in order to better understand how the adopted solution works. In the next chapter, we will deal with the implementation phase, in which we will describe the final result.

Chapter Four: Realization

Chapter four: Realization

1. Development environment:

1.1. Hardware environment

During my internship, I used my personal computer. Following are shown main characteristics of my machine a Lenovo brand:

- Processor: Intel® Core i3 (6th Gen) 6100U / 2.00 GHz
- System type: 64-bit Operating System, x64-based processor
- 15.6” LED Screen
- 1TB Hard Disk

1.2. Software environment:WAMP -Apache Server

- **Wamp- Apache Server**

Acronym for **W***indows/***A***pache/***M***ySQL/***P***HP,* **P***ython, (and/or)* **P***ERL*

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL/MariaDB handle the database components, while PHP, Python, or PERL represents the dynamic scripting languages.

Figure 22 WampServer

- **StarUML**

StarUML is an open source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling. ... It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

Figure 23 StarUml

- **Symfony**

Symfony is a PHP web application framework and a set of reusable PHP components/libraries. It was published as free software on October 18, 2005 and released under the MIT license.

Figure 24 Symfony

- ***Bootstrap***

Bootstrap is a free and open-source CSS **framework** directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Figure 25 Bootstrap

- **Git**

Git (/gɪt/) is a distributed version-control system for tracking changes **in** source code during **software** development. It is designed for coordinating work among programmers, but it can be used to track changes **in** any set of files.

Figure 26 Git

- **Composer**

Composer (software) **Composer** is an application-level package manager for the **PHP** programming language that provides a standard format for managing dependencies of **PHP** software and required libraries. ... **Composer** runs from the command line and installs dependencies (e.g. libraries) for an application.

Figure 27 Composer

- **PHP**

PHP (**PHP**: Hypertext Preprocessor) is a scripting language that helps people make web pages more interactive by allowing them to do more intelligent, complex things. **PHP** code runs on the web server. ... Many languages like C, Perl, Java, C++, and even Python influenced its structure.

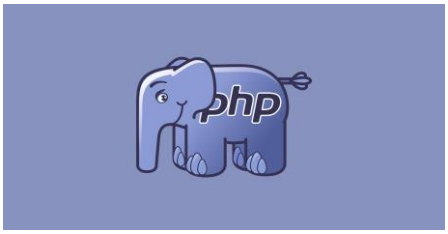

Figure 28 PHP

- **jQuery**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019, **jQuery** is used by 73% of the 10 million most popular websites.

Figure 29 jQuery

- **JavaScript**

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. ... Like server-side scripting languages, such as PHP and ASP, **JavaScript** code can be inserted anywhere within the HTML of a webpage.

Figure 30 Javascript

- **HTML5**

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and latest major version of HTML that is a World Wide Web Consortium (W3C) recommendation. ... **HTML5** is intended to subsume not only HTML 4 but also XHTML 1 and DOM Level 2 HTML.

Figure 31 HTML5

- **CSS3**

CSS is the acronym of “Cascading Style Sheets”. **CSS** is a computer language for laying out and structuring web pages (HTML or XML).

Figure 32 CSS3

2. Project implementation

In this chapter, we will describe different parts of the developed solution answering the main tasks requested and necessary for a professional management of a medical laboratory.

2.1. The tree structure

The tree structure, or site map, generally takes the form of a diagram that organizes the site pages into sections and defines the different levels of navigation. It is usually the first step in designing a website.

2.2. Login page

To secure our system and securing the information, before realizing any task the manager must identify himself by entering his identifiers (Login, Password) respectively.

Of course, with the forgot password link and remember me check box.

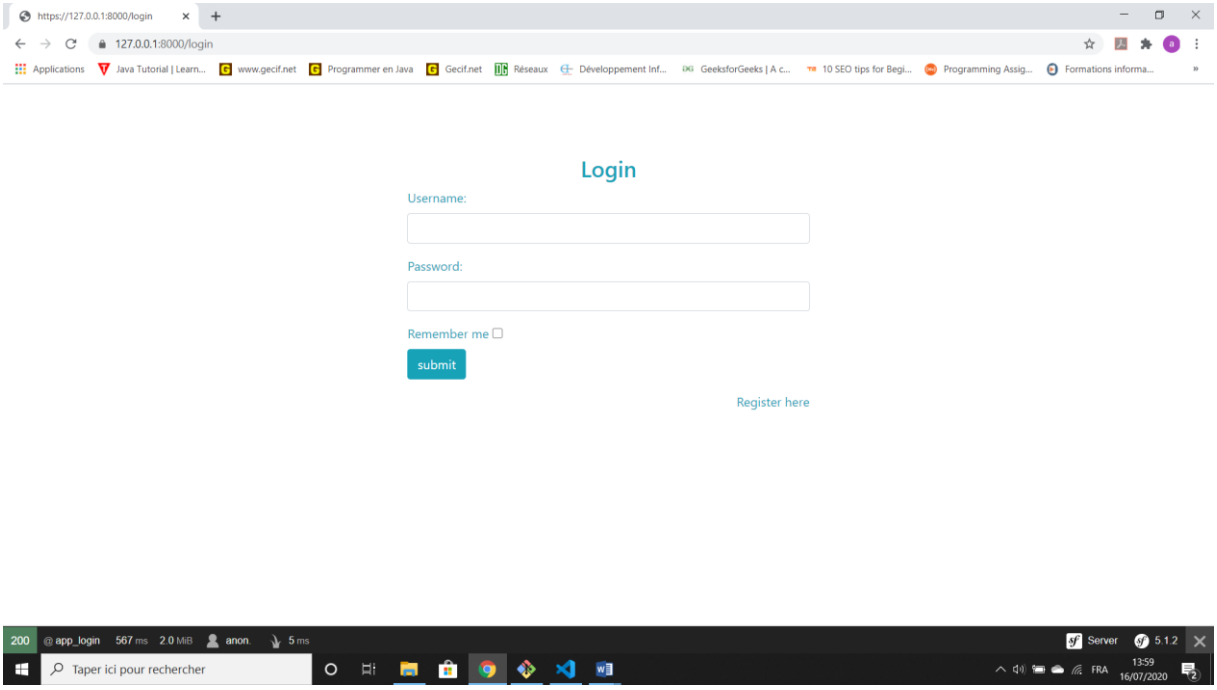

Figure 33 Login Page

2.3. Client side page

“Home page” it’s the page that anyone can access.

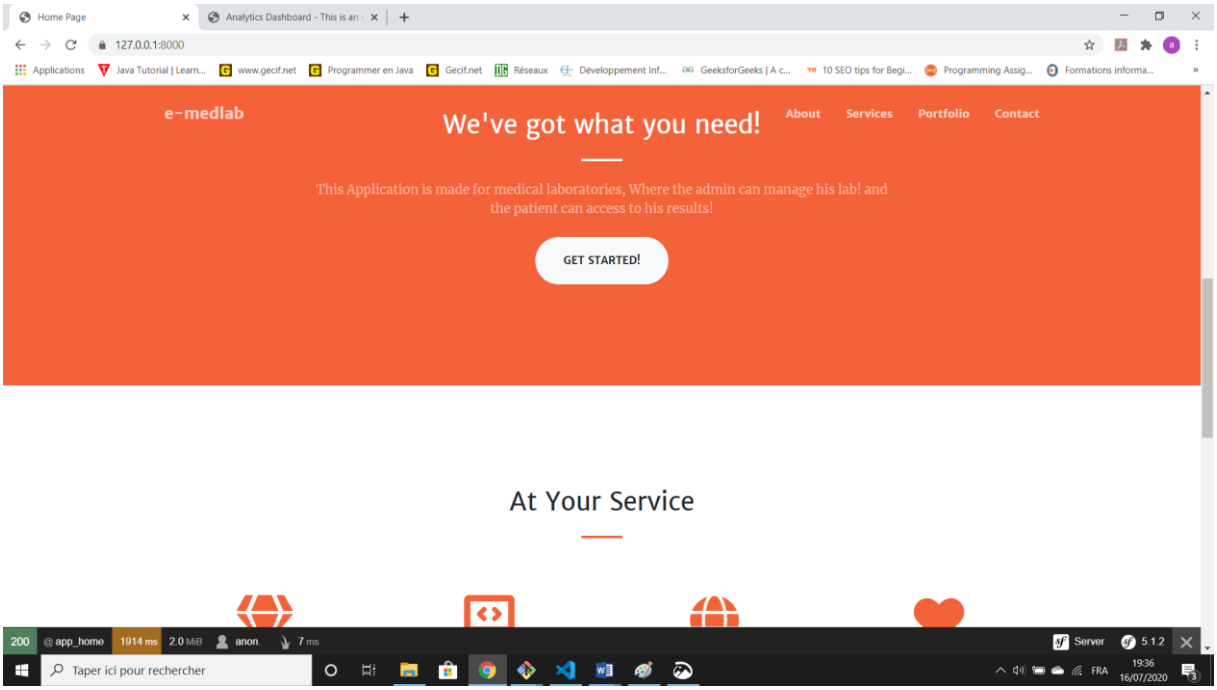

Figure 34 Client Side

2.4. Admin Panel

From this page, the staff of the laboratory can manage it.

They can manage patients, results, reports and payments, but only the biologist how can get into the staff management.

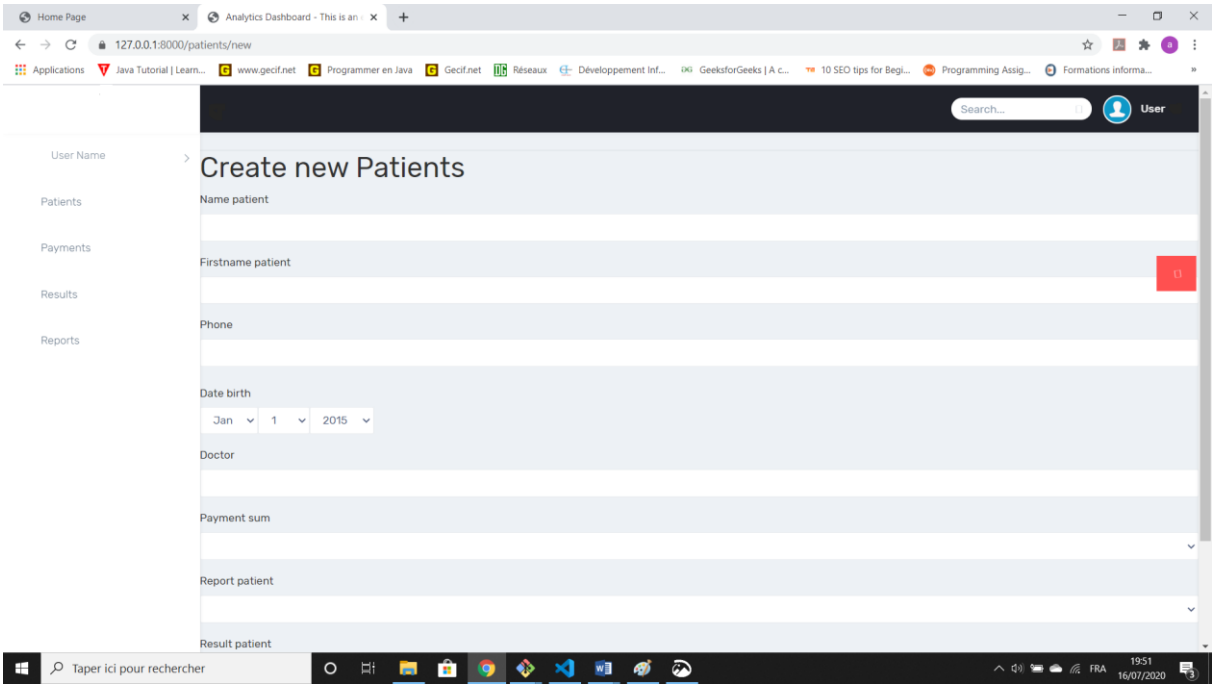

Figure 35 Admin Side

General Conclusion

This project offered me a great opportunity at different levels theoretical improvement of my knowledge firstly, about design and implementation of databases; such as and MySQL, secondly about web scripting design tools including HTML, CSS and PhP.

I have learnt that even if you do not have a client to do the application for, you can imagine anything and build it, and really, the imaginable applications are more known and trended than specific ones.

Moreover, I discovered the huge amount of business possibilities offered by Google and its marvellous tools and Apps; trying to master into Symfony framework with his great documentation.

Finally, this project offered me the opportunity to improve my English language.

Netography

https://symfony.com/

https://www.wrappixel.com/

https://getbootstrap.com/

https://packagist.org/packages/symfony/symfony

https://getcomposer.org/

https://git-scm.com/

https://github.com/