UNIVERSITY HASSIBABENBOUALI DE CHLEF

FINAL YEAR PROJECT

THEME

Conception and Implementation of Medical Clinic Management Application(Cardiology)

Presented By :

Bekouassa Mohammed

Aouameur Riadh

Supervised by :

Mrs.Balkadi Khadidja

Table Of Contents

List of Figures

List of Tables

Thanks

To Allah

for giving us the strength, the health, the will, and courage that accompanied us throughout our studies and preparation of this licence's thesis, please guide our steps in the right way .

A special thank to our parents, for their love, their sacrifices and their patience, And to our friends for their eternal support in particular: Benhami Samir, Amellel Sofiane, Bachich Ahmed.

A huge thank to the teachers for the quality of teaching they gave us during these 3 years spent at Hassiba BenBouali De Chlef University, in particular : Mr.Gharbaoui, Mr.Aridj and Mrs.Aboura .

We would particularly like to thank our supervise Mrs.Balkadi for giving us the guidance and the orientation during our work on our project.

Also a huge thanks to Mr.Fellahi for helping us to solve the hurdles we faced thanks to his programming experience.

We thank the members of jury for having devoted their time to evaluate our work.

GENERAL INTRODUCTION

To this day, the medical institution has made use of technology to improve its management methods, relying on different mechanisms and concepts Provided by Computer Science.

The concept of databases for example, defined as *"an organized set of data, which is generally stored and accessed electronically from a computer system"* has ensured for the medical clinic a convenient way to store and also manipulate patients medical records, which now requires less space and less time compared to the methods used before the database concept appeared.

However, computer science is more than just "databases" and if such solutions prove possible, then the potential for improving it is not in doubt, there is still a lot of development that can be made within the medical establishment.

The overall objective of the project is to simplify medical clinic management, and the secret of "simplicity" lies in achieving "better patient care".
In real life, the patients number , may exceed the number of appointments a clinic can create or handle, on the other hand, when the patients miss their scheduled appointments, they don't just waste valuable time, they hurt clinic revenue too.

So a balance must be struck between these two units, in order to control the functioning of the medical clinic, and the balance mechanism will be explained in detail later in this chapter.

In fact, it's true that Patient care should be the number one priority,but it will remain elusive, unless the methods of communication between the doctor and his assistant are improved and their privileges and restrictions are defined.
When challenging situations arise in doctor assistant interactions, how can we best manage them?.
This project came with a new network-based approach to connect the doctor and the assistant applications using Local Area Network (LAN) Wire(Ethernet) or Wireless(WiFi), allowing information in databases to be shared according to some specified set of privileges.

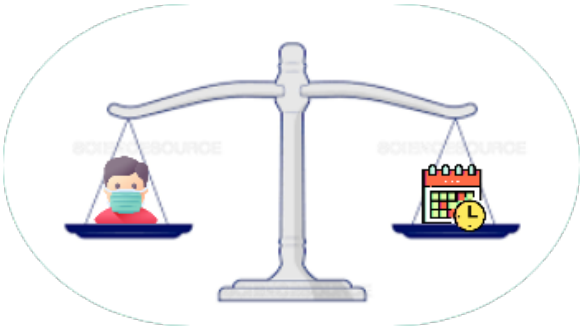


Figure 1: The balance between patients and their appointments

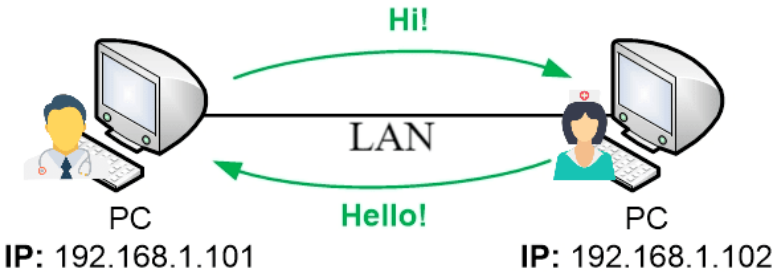


Figure 2: Representation of the connection using local area notwork(LAN)

Chapter 1
Medical Cabinet Representation

1.1 Introduction :

This chapter is a representation of the important units within the medical clinic, starting from the medical record, through scheduling appointments, and ending with the clinic's members and their tasks.
From all that, step by step, all the necessary specifications will be revealed, which are an essential stage for understanding and building the project.
By the end of this chapter, the objective will be clear and the project will be ready for the conception and implementation phase.

1.2 What is cardiology?

Cardiology is the study and treatment of disorders of the heart and the blood vessels. A person with heart disease or cardiovascular disease may be referred to a cardiologist.

1.3 Medical records:

1. Introduction:

Every time a patient enters the clinic, a new medical record is created.
Doctors use patients medical records to document patient care, and their primary purpose is to routinely record clinical information and communicate the patients, like pre-existing conditions, risk factors, patient characteristics, diagnoses, life

style, treatments, and procedures. and it's important to take matters of privacy and legality seriously.

When the patient is discharged from the clinic, the medical record data is stored and converted later into a code, this code will be used to identify diseases patterns, improve quality of care, and determine the cost of providing this care.
In short, an incomplete or ambiguous medical record can lead to inaccurate code, and due to strict coding rules, code can't make assumptions about diagnosis, and procedures ...etc. So an unclear medical record is simply unhelpful.

2. Contents of the medical record:

- **Patient Profile**:
  It contains the patient's identifying information such as his fullname and age, additional information that the doctor needs during the consultation, such as the patient's weight and height and also some contact information.

- **History Of illness**:
  Having information about the history of the patient and his family with the illness allows the doctor to make an important opinion about his patient condition. we summarize it in: allergies, medical history, surgical history, and family history.

- **Life Style**:
  The lifestyle can inform the doctor about the general psychological and physical condition of his patient. It generally revolves around diet, physical activity, and addiction.

- **Reasons of Consultation**:
  The doctor explains the reasons that led to the consultation based on the initial dialogue with the patient.

- **Current treatment**:
  After the consultation,the doctor records the medications that are given to the patient to consume, with all the necessary instructions that the patient must follow while taking it as well as leaving notes about the duration(how long) and patient compliance.

- **Examinations and Tests Results**:
  The examinations and tests that were performed on the patient, whether inside or outside the clinic,are recorded and stored because they are very essential for the next consultations.

the Table below will represent a summarize about Medical records contents:

Medical record Contents	
Patient Profile	**Identifying info :** • Full Name. • Age and Birth date. • Gender. • Profession. • Marital Status. • Children. **Consultation info :** • weight. • weight. • waist circumference. • body mass index(BMI). **Contact info :** • Email adress or Phone number.
History Of illness	**Summarized in** : • allergies. • medical history. • surgical history. • family history.

Life Style	**summarized in :** • Diet. • physical activity . • addiction.
Clinical Tests	**summarized in :** • ECG(Electrocardiogram). • TTE(Transthoracic Echocardiogram). • Examination of the neck. • Examination of the chest. • Examination of the abdomen. • Examination of the ankles and calves.
Current Treatment	**contain :** • Medication name and Type. • Dosage(mg). • Frequency eg.(once per day). • Duration. • Compliance.

Table 1.1: Summerize of Medical record important contents.

1.4 Appointments Management

Introduction:

Daily, the majority of health care clinics are overworked and overstretched to meet demand, however, many different challenging interactions occur between them and patients because of bad appointments booking system.
To be fair, it is difficult to build a good and organized scheduling system in an imperfect and disorganized work environment, how is that?.
Appointments Management is a task that depends always and essentially on the positive interaction and the constant communication between both the doctor and the assistant, which many current medical applications do not provide.

It became evident that there are two types of interactions within the medical clinic that reflect the quality of the appointment scheduling system :

1. Doctor-Assistant Interaction.
2. Clinic-Patient Interaction.

Doctor-Assistant Interaction

LAN Network-Based Oproach:

Local Network Area:

A local area network is a computer network that interconnects computers within a limited area such as a school, or office building.
The local network **LAN** allows connected devices within it to share files, printers, and the Internet as well.

Local Network Area In Medical Application:

The ability to share data and information is the most important feature of a local network, how does this help build an organized appointment reservation system, and what is the improvements in the methods of communication between the doctor and his assistant?

Examples of some current medical applications will be studied, analyzed and also compared to obtain an application that ensures the best communication and interaction for the doctor and the assistant.

Study Explanation:

Two general types of applications in our study :

1. **Application 1** :
   Using separated Applications(could be the same) installed on the doctor and the assistant devices without connection between them.
2. **Application 2** :
   Using one Application installed on both the doctor and the assistant devices in addition to provide LAN connection between them.

Conditions and rules :

The application will be evaluated according to :

- **the privileges of both the doctor and the assistant to access some important and critical tasks inside medical clinic application.**

References :

- **The Doctor will be referred to as "Dr".**
- **The Assistant will be referred to as "Ast".**
- **Green Color refers to Allowed privilege.**
- **Red Color refers to Not Allowed privilege.**

The Privileges required are :

- **Medical record :**
  Medical record is private, Only the **Doctor** has the privilege to access.
- **Patient List :**
  Its important to allow not just **the Assistant** but also **the Doctor** to access into patients list to keep him updated.
- **Appointment schedule :**
  Appointments management require the access of both **the doctor** and **The Assistant** to insure an organized schedule.

Access to :	Medical Record	Patients List	Appoint. schedule
App 1	Dr	Ast	Ast
App 2	Dr + Ast	Dr + Ast	Dr + Ast

Table 1.2: Table represent the study results.

Table analysis :

From the above table, the results look very interesting, when the LAN approach is implemented on the medical applications, communication between the doctor and the assistant becomes possible, allowing both parties to easily access each other's information, which greatly improves work efficiency inside medical institutions. However, the use of the local network alone fails to take into account the required privileges while accessing the medical record or any task that requires only the doctor access such as the consultation or any patient private information.

Access and Privileges Control :

The MySQL Privilege System allows assigning specific database privileges to each user, whether the doctor or the assistant using queries.
The queries can grant easily the ultimate access for the doctor and revoke the assistant access to the private tasks.

Access to :	Medical Record	Patients List	Appoint. schedule
App 2	Dr	Dr + Ast	Dr + Ast

Table 1.3: Table represent Importance of priveleges control.

Clinic-Patient Interaction

Now, communication between the doctor and the assistant has been enhanced, it is time to take care of the balance between appointment supply and patient request, and the protocol that reduces patient absenteeism, as mentioned in the general introduction.

Tasks to complete :

1. **Improve the Patient Access Improvements** :
   (a) The advanced Access scheduling.
   (b) The Patients Pressure.
2. **Reduce The Number of Missed Appointments.**

The Patient Access Improvements:

1. The Patient Access Methods:

Introduction :

The first interaction between the clinic and patients is when they call to book an appointment, and here the doctor or his assistant often replies: "Come next week" or "next month", and this may disappoint the patient who needs to see his doctor as soon as possible, especially if he is in dire need of medical care.

The Traditional Access scheduling Model:

The traditional reservation systems have the principle of providing most of the appointments for future dates, resulting in a patient care appointment being reserved in advance, where the schedule is already full before the start of the workday, which is the reason behind prolonged timeswait for appointements.
Studies have shown that delays in care and long waiting times for appointments have very serious consequences. For example, for heart and cancer patients,the prolonged wait times have been associated with an increased risk of disease and death due to delays in care [2, 3] .

What is really needed is to match the patients requests correctly with the scheduled appointments, in other words, it means seeing patients when needed, not pushing them to another day or another week.

As a predictable response after the traditional access model failed, experts came up with new alternatives models to reduce the patients access problems.

a.The Carve-out Access scheduling model :

Since patients in need of urgent care are the most affected by the problems of the traditional model, 50% same-day appointment slots are reserved for urgent care and the rest 50% booked.

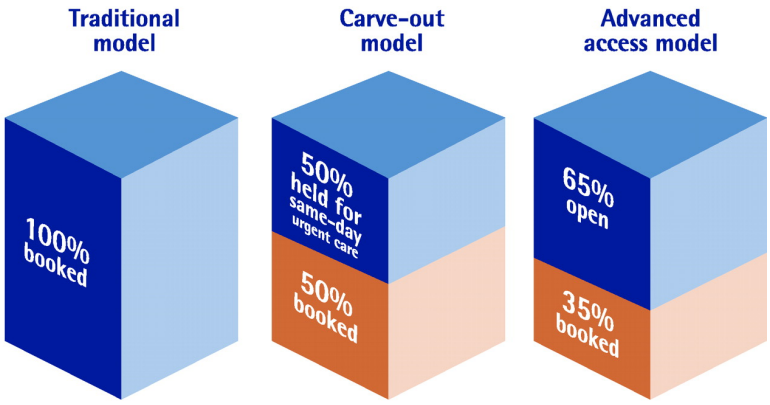
b.The Advanced Access scheduling model:

The core principle of the advanced access or open access or also known as same day access [4] is that patients calling to schedule a doctor visit are offered an appointment the same day regardless of the reason for the visit(routine, urgent or preventive visits), this method leaves approximately about 65% of the day open, eliminating booking the appointments weeks or months in advance, yet, under the Advanced Access people should still be able to book ahead if they prefer (35%).

Conclusion:

The medical clinic must realize that it is not obligated to follow a specific model because each model has been specifically designed to solve specific type of problems that other models cannot solve, which means that the best protocol is to choose the model according to the circumstances and medical clinic capabilities.

Figure 1. Comparison of patient booking models



H Adapted from Murray and Tantau.[2]

Figure 1.1: Comparison of patient booking models

2. The Patients Pressure :

How Many Patients Can Doctor Manage?
It is important to realize that there are limits to the number of patients a doctor can be accountable for, and these limits must be define using **Panel Size**.

Panel size is simply the number of individual patients under the care of the doctor.
Doctor or Assistant should not take more work than they can manage.
If the panel is too large,The doctor will create "overwork" which means "work cannot be done", and also "overtime" which means "needs constant support for overtime".
If the board is too small, the doctor will not generate enough revenue to cover the expenses.
So the demand of the patients must equal the supply of appointments if timely service is desired, and the clinic should always determine the size of the panel based on her work capacities [5].

Reduce the missed Appointments :

Why Do Patients Miss their Appointments?

The Medical Group Management Association (MGMA) [6], conducted a survey of over 143 patients to understand how medical offices can optimize their efforts to schedule the oppointments.

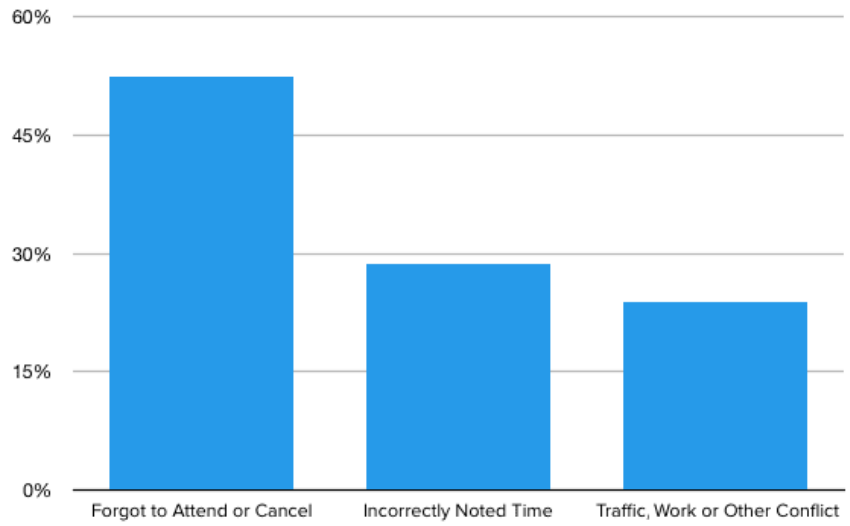


Figure 1.2: The graph represents the reasons behind missed appointments.

The graph shows that most patients forgot to attend or cancel their scheduled appointments, and the rest cited that they wrote down the wrong time or date, or they had to miss the appointment due to other issues such as traffic or work.
Source: Maximizing Patient Access and Scheduling Report by MGMA.

Reduce The Number of Missed Appointments

Since forgetting to attend or canceling scheduled appointments and typing the wrong time or date are the two main reasons for missing an appointment,it seems like using patient **reminder** appears to be the most appropriate solution.

The perfect reminder :

In fact, there are different types of reminders that can be used, but unfortunately most of them are not sufficient to guarantee the most appropriate success for the reminder task,and here some examples :

Postcards : Unless you sort the mail every day, you're lost in a sea of junk mail, and as the world is digital, the clinic has to use digital reminders as well.
Phone calls : This can be inconvenient and badly timed, so the patient may not answer, or the patient's phone is closed for some reason, so the reminder will not reach him.
Email alerts : It's still not available to everyone, and often the inbox is already full causing too many emails to be missed or ignored.

Fortunately,Once again, the Medical Management Association is conducting a survey of how people want to be reminded of appointments.
That is the Response :

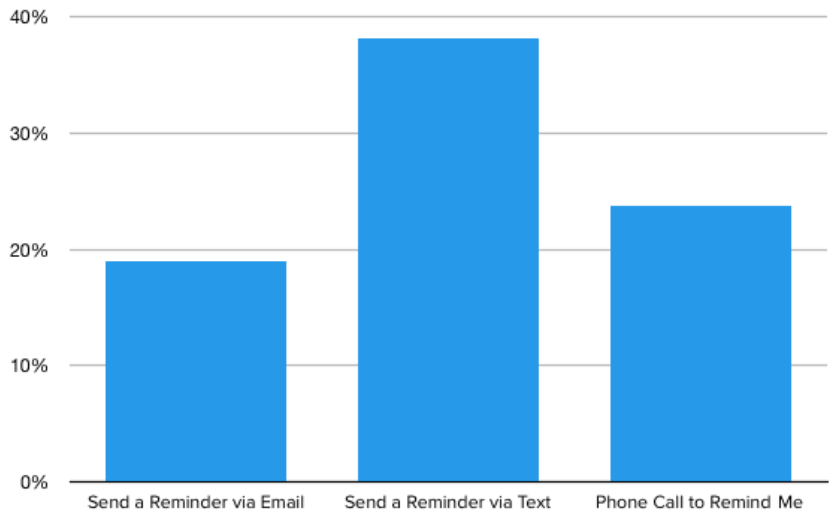


Figure 1.3: graph represent best reminder method according to the patients.

The data is clear: text messages (SMS) are the preferred way to inform the patients about upcoming appointments.
This is not surprising, given that text messaging is quite easy to use, accessible and cannot be ignored.
In fact, 90% of texts are read within 3 minutes of receiving them!

TEXT MESSAGES (SMS):

- Texts are read instantly.
- Texts are friendly and unobtrusive.
- Texts are never ignored.

Summarize

The appointment reservation schedule is a very sensitive system that basically reflects the quality of interaction between the doctor, the assistant and the patient within the medical clinic.
The local network has proven its success in making communication between members of the medical clinic better than before, so interaction with the patient becomes much easier.
On the other hand, the appointment schedule must match the number of patients and aim to satisfy them and interact with them using new methods such as the reminders to reduce their absenteeism.

Figure 1.4: Appointments Schedule

1.5 The Medical Clinic Members

Introduction :

Every institution has its own representatives (the actors), who are distinguished from each other by the tasks assigned to them.
The medical clinic consists generally of two main members:

- The doctor (cardiologist).
- The Assistant.

The Members Activities :

Medical Clinic Members Activities	
The Doctor Manage :	**The Assistant Manage :**
• Consultation . • Medical Record . • Prescriptions .	• Patients List. • Appointments Schedule. • the payment.

Table 1.4: Summerize of Medical record important contents.

1.6 Work objectives:

After studying the types of interaction within the medical clinic and working to improve them using new technologies and models that basically aim to make the clinic a better place capable of developing the level of patient care and keeping the management simple as possible, the work objectives are now clear and the specification stage has been completed and it's the time for the conception phase.

1.7 Conclusion:

In this chapter, the focus is on the medical clinic and its different units, how each one of them works, and the impact that its quality reflects. Preparing medical records and appointments and defining tasks correctly in a way that matches the clinic's capacity allows providing comfort and satisfaction of the actors inside the medical clinic, in addition to insure a good and positive interaction between them.

In a summary of the techniques and models provided by the project to improve the efficiency of these units, the interaction between the doctor and his assistant benefited from the effectiveness of the local area network(LAN)in enhancing the communication and defining their privileges,For the patients, the use of reminders to avoid the absenteeism and the "panel size" concept to reduce the pressure on the clinic, as well as the introduction of new technologies to manage appointments booking system such as the advanced access and the Carve-out models.

Chapter 2

UML

2.1 Definition of UML :

UML is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems [1].

2.2 Design

UML offers a way to visualize a system's architectural blueprints in a diagram, including elements such as:

- any activities (jobs).
- individual components of the system.
- and how they can interact with other software components.
- how the system will run.
- how entities interact with others (components and interfaces).
- external user interface.

2.3 UML application:

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy. UML itself finds different uses in software development and business process

2.4 Blueprint:

In such a case, the UML diagram serves as a complete design that requires solely the actual implementation of the system or software. Often, this is done by using CASE tools (Computer Aided Software Engineering Tools). The main drawback of using CASE tools is that they require a certain level of expertise, user training as well as management and staff commitment.

2.5 Pseudo Programming Language:

UML is not a stand-alone programming language like Java, C++ or Python, however, with the right tools, it can turn into a pseudo programming language. In order to achieve this, the whole system needs to be documented in different UML diagrams and, by using the right software, the diagrams can be directly translated into code. This method can only be beneficial if the time it takes to draw the diagrams would take less time than writing the actual code.
Despite UML having been created for modeling software systems, it has found several adoptions in business fields or non-software systems.

2.6 Types of UML Diagrams:

There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation).
The two most broad categories that encompass all other types are **Behavioral** UML diagram and **Structural** UML diagram.

1. **Behavioral UML Diagram:**
    - Activity Diagram.
    - Use Case Diagram.
    - Interaction Overview Diagram.
    - Timing Diagram.
    - State Machine Diagram.
    - Communication Diagram.
    - Sequence Diagram.
2. **Structural UML Diagram:**
    - Class Diagram.
    - Object Diagram.
    - Component Diagram.
    - Composite Structure Diagram.
    - Deployment Diagram.
    - Package Diagram.
    - Profile Diagram.

Not all of the 14 different types of UML diagrams are used on a regular basis when documenting systems and/or architectures.
The most frequently used ones in software development are: Use Case diagrams, Class diagrams, and Sequence diagrams.

2.7 Use Case Diagram:

A cornerstone part of the system is the functional requirements that the system fulfills. Use Case diagrams are used to analyze the system's high-level requirements. These requirements are expressed through different use cases. We notice three main components of this UML diagram:

- **Functional requirements**: represented as use cases; a verb describing an action.
- **Actors**: they interact with the system; an actor can be a human being, an organization or an internal or external application.
- **Relationships**: between actors and use cases – represented using straight arrows.
  The example below depicts the use case UML diagram for an inventory management system. In this case, we have the owner, the supplier, the manager, the inventory clerk and the inventory inspector.

An example of a use-case diagram
Following use case diagram represents the working of the student management system:

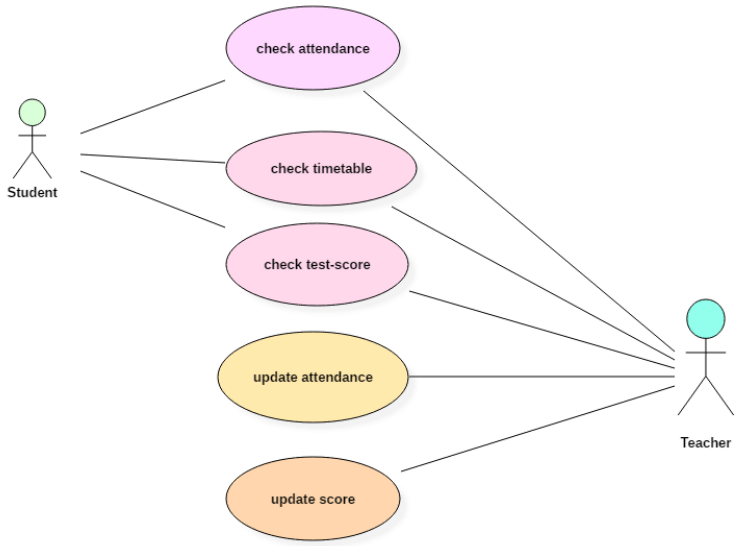


Figure 2.1: UseCase diagram exemple

Within the circular containers, we express the actions that the actors perform. Such actions are: purchasing and paying for the stock, checking stock quality, returning the stock or distributing it. As you might have noticed, use case UML diagrams are good for showing dynamic behaviors between actors within a system, by simplifying the view of the system and not reflecting the details of implementation.

2.8 Sequence UML Diagram:

Sequence diagrams are probably the most important UML diagrams among not only the computer science community but also as design-level models for business application development. Lately, they have become popular in depicting business processes, because of their visually self-explanatory nature.
As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active only when needed or when another object wants to communicate with them. All communication is represented in a chronological manner. To get a better idea, check the example of a UML sequence diagram below.
As the name suggests, structural diagrams are used to depict the structure of a system. More specifically, it is used in software development to represent the architecture of the system and how the different components are interconnected (not how they behave or communicate, simply where they stand).
Below you can see an example of a sequence diagram, depicting a course registration system.

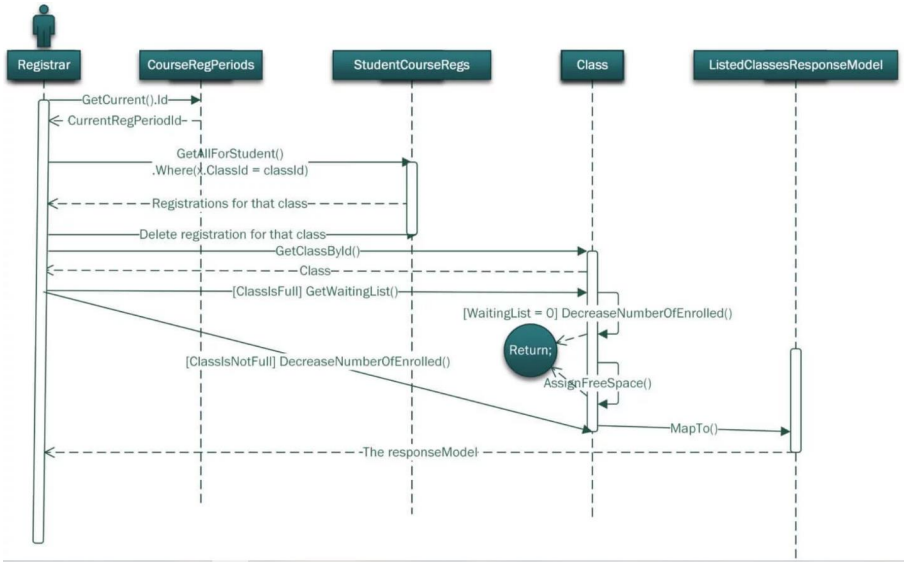


Figure 2.2: Sequence diagram exemple

2.9 Class Diagram:

Class UML diagram is the most common diagram type for software documentation. Since most software being created nowadays is still based on the Object-Oriented Programming paradigm, using class diagrams to document the software turns out to be a common-sense solution. This happens because OOP is based on classes and the relations between them.
In a nutshell, class diagrams contain classes, alongside with their attributes (also referred to as data fields) and their behaviors (also referred to as member functions). More specifically, each class has 3 fields: the class name at the top, the class attributes right below the name, the class operations/behaviors at the bottom. The relation between different classes (represented by a connecting line), makes up a class diagram.

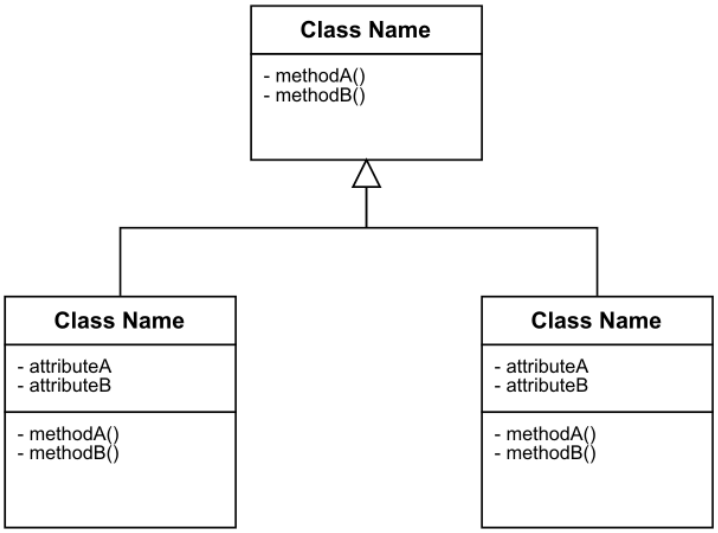


Figure 2.3: Class diagram exemple

The example above shows a basic class diagram. The 'Checking's Account' class and the 'Savings Account' class both inherit from the more general class, 'Account'. The inheritance is shown using the blank-headed arrow. The other class in the diagram is the 'Client' class. The diagram is quite self-explanatory and it clearly shows the different classes and how they are interrelated.

Chapter 3
Implementation and realization

3.1 Introduction

In this final chapter, following what have been mentioned in the previous chapter about the Unified Modeling Language(UML), and Unified Software Development Process (USPD) [7], will make it much easier to begin the implementation phase. Specifically, 3 diagrams will be studied and implemented in this chapter which are Use Case Diagram, Sequence Diagram and Class Diagram.
Next,the languages and the tools used in building the project will be represented. Finally, represent a collection of screenshots from the implemented application.

3.2 The Environment Realisation :

3.2.1 introduction

In this part we will represent the designs of our system using :

- usecase diagram .
- sequence diagram.
- class diagram.

3.2.2 UseCase Diagram Representation :

Introduction :
The use case diagram is important for defining the roles of Actors who are the Doctor and the Assistant within the Medical Clinic Environment.

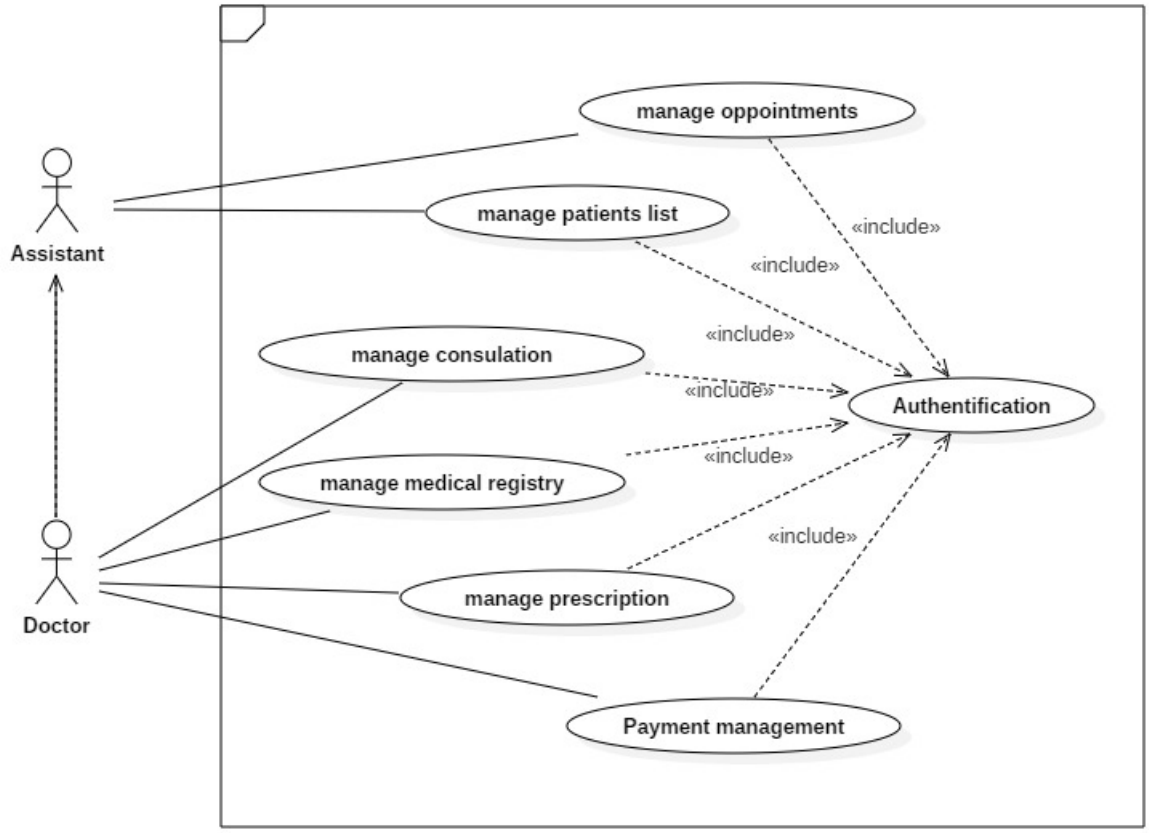


Figure 3.1: UseCase Diagram Representation

3.2.3 Sequence Diagram Representation :

Introduction :

their will be several Sequence diagrams to represent most and important events in our system :

- Login System(security) .
- Add and register Patients .
- Appointement Management .

3.2.3.1 Login System Management:

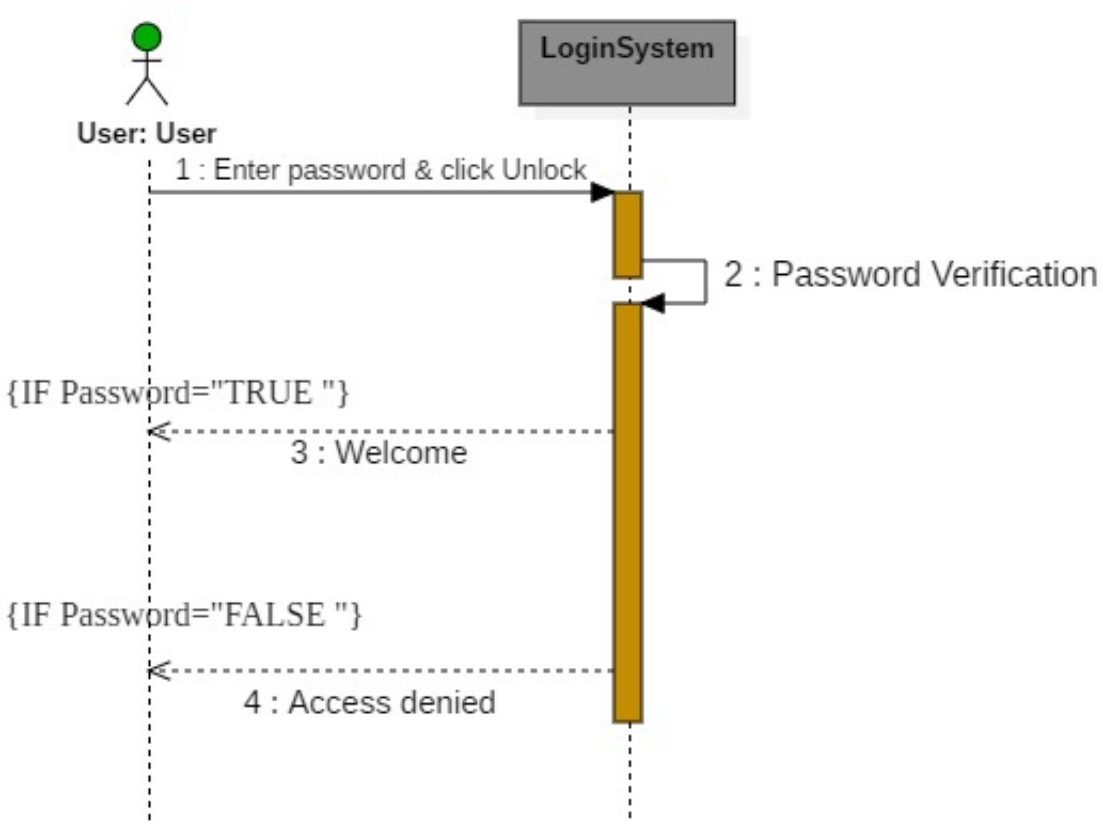


Figure 3.2: Sequence Diagram Representation for login System

3.2.3.2 Add Patient Management:

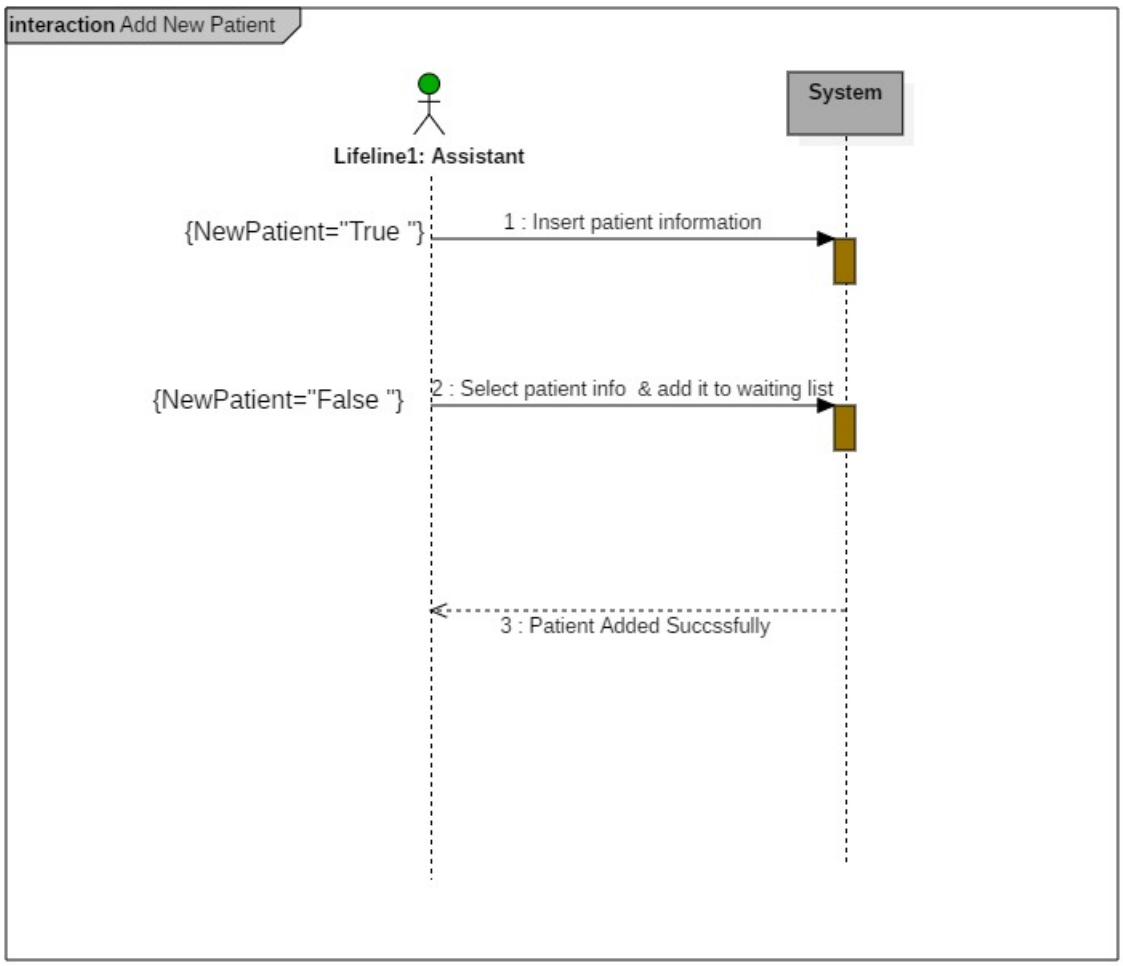


Figure 3.3: Sequence Diagram Representation for adding patient

3.2.3.3 Appointement Management:

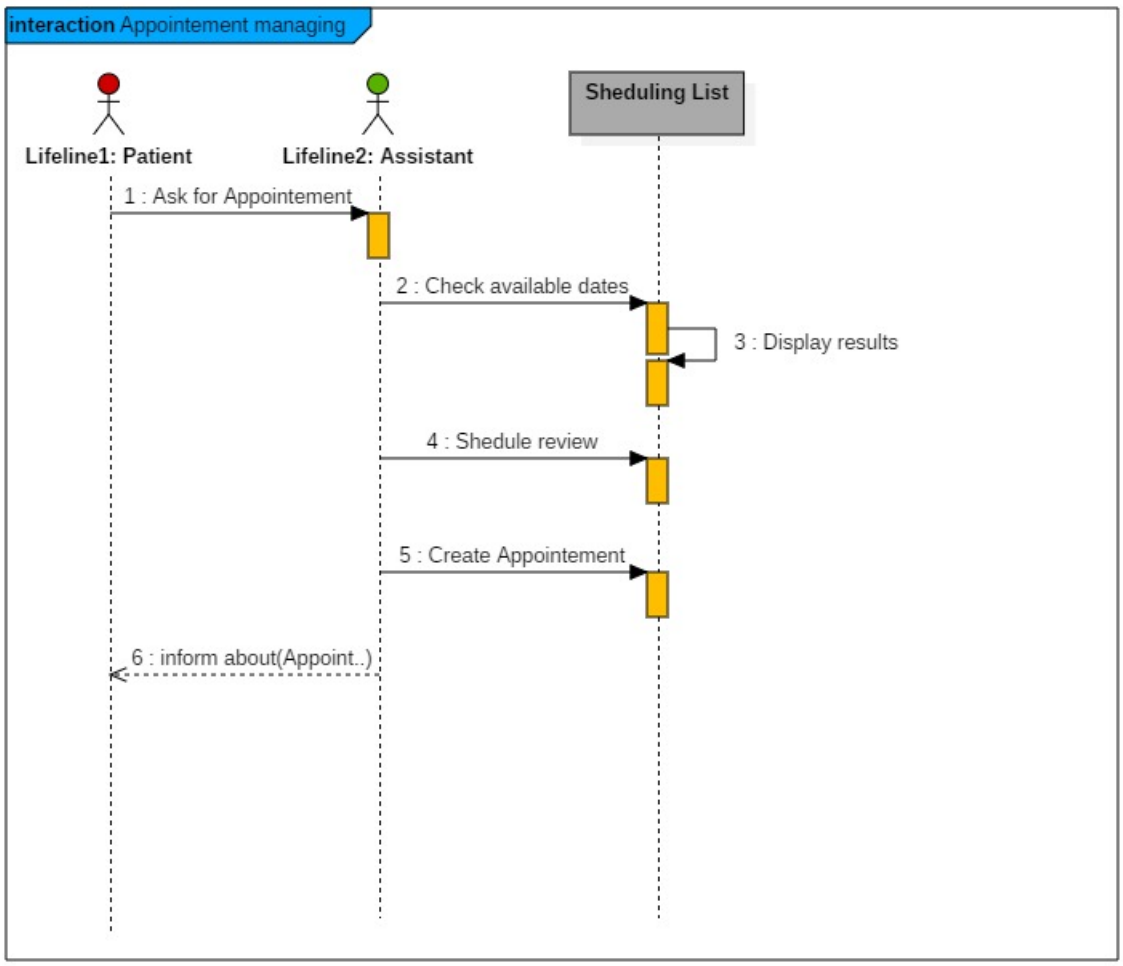


Figure 3.4: Sequence Diagram Representation for appointement management

3.2.4 Class Diagram Representation :

Introduction :

The purpose of class diagram is to model the static view of an application, and can be directly mapped with object-oriented languages.

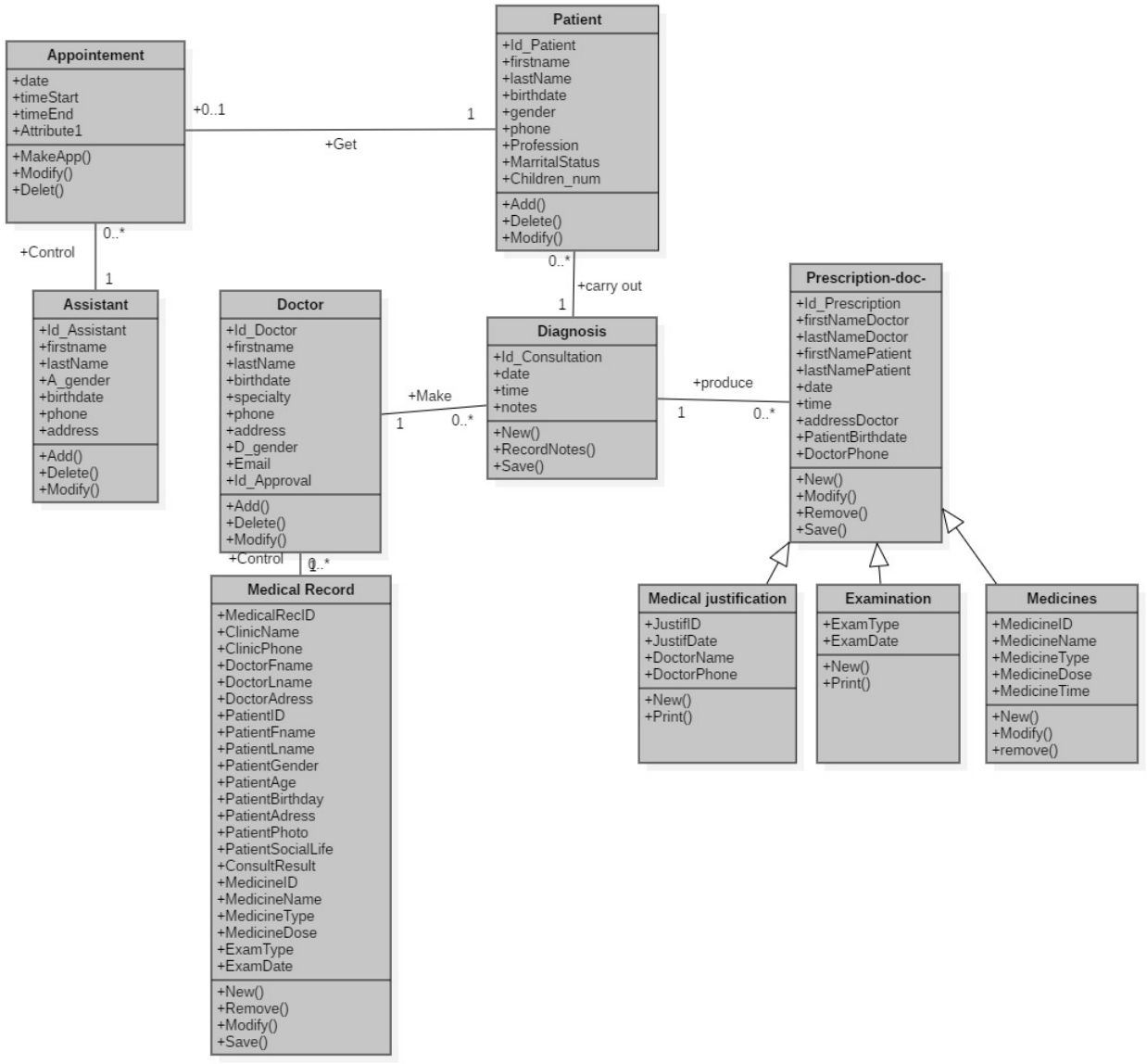


Figure 3.5: Class Diagram Representation

3.3 Tools and languages used :

3.3.1 *Tools :*

IntelliJ IDEA :

IntelliJ IDEA is a private programming environment or integrated development environment (IDE) that is largely dedicated to Java.
This environment is used especially for software development.

GitHub:

GitHub is an open source version management and collaboration platform for software developers.
The goal from using it is to store the code for our project and track the complete record of all changes made to this code.

Github generally consists of two important concepts :

1. ***Version control*** :
   the word "git" referring to an open-source version control system.
   Version control is a system that records changes to a file or group of files over time so that you can call specific versions Later.

2. ***Team collaboration***:
   The word "hub" refers to the social network built around the Git system wich provide the "team collaboration".
   Team collaboration means that the platform access control and collaboration custom features like bug tracking, feature requests, and task management for each project.

Gradel Build Tool :

Gradle is an open source building automation tool designed to be flexible enough to build almost any type of program.
We decided to use it in our project due to its important features like:

- High performance and Extensibility.
- JVM foundation and IDE support .

JasperReports Library :

JasperReports is a powerful open source reporting tool that has the ability to deliver rich content onto the screen, to the printer, or into PDF, HTML, XLS, RTF, ODT, CSV, TXT and XML files. It is entirely written in Java and can be used in a variety of Java-enabled applications to generate dynamic content. Its main purpose is to help create page-oriented, ready-to-print documents in a simple and flexible manner.

3.3.2 *Languages* :

3.3.2.1 JAVA(JavaFX Library):

Java is an easy language to set up different applications, It is possible to program powerful and error-free applications using it.

JavaFX is a very huge library developed to build applications with a Java based user interface. As developers, we were able to build very powerful applications quickly and easily because they provide ready-made technologies for the programmer to use in applications without any complication.
according to the javafx overview page on the oracle technology network "it is designed to provide a lightweight hardware accelerated java UI platform..." [8]

JavaFX features used :

- **FXML:**

FXML is a Markup Language primarily based on XML, which is specifically designed to be used in building the user interface for applications shown using the JavaFX library.
So when designing the user interface you can write the design code for each window in a special FXML file in it and link it to it.

- **Scene Builder:**

When designing the user interface, you can write the FXML code yourself or using the Scene Builder program, which allows you to easily add objects in the window.
Scene Builder is a program found within the Netbeans program.

- **CSS:**

JavaFX allows you to write CSS (Cascading Style Sheets) code to easily develop UI design.

3.3.3 *Implementation Of the Database:*

3.3.3.1 Database Management System(DBMS):

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database.
A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure.it also defines rules to validate and manipulate this data.
To implement our database, we use MYSQL Server which we will link with our interfaces (Java).

MYSQL Server :

The SQL in "MySQL" means "Structured Query Language": the standard language for database processing.
MYSQL is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database.

3.4 Project implementation

3.4.1 introduction:

This section will contain screenshots for the most important project intefaces.

3.4.2 Authentification System:

The login interface is a security measure designed to prevent unauthorized access to confidential data, which prompted us to provide effective logi interface for both users the doctor and the assistant(the importance of login in "bibliography").
Registration interface allows the user to enter his identification information which will in turn be stored in the installed application database.
The Licence Key :
To complete the acount registration a Licence Key is required, to insure that only the principle user can create an acounte to access his application.

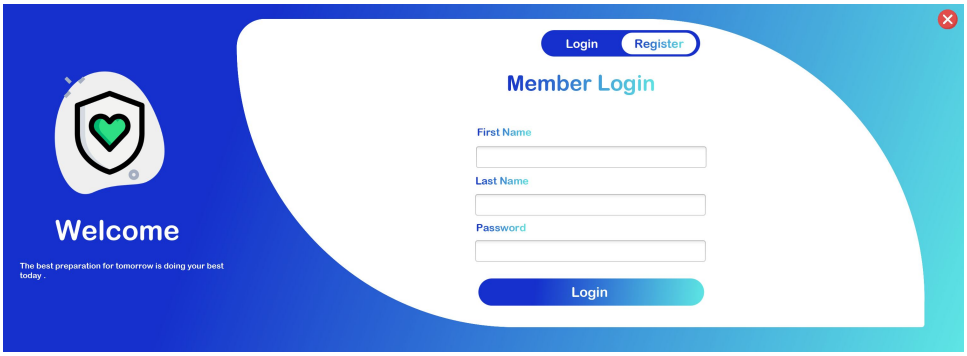


Figure 3.6: The Login Interface

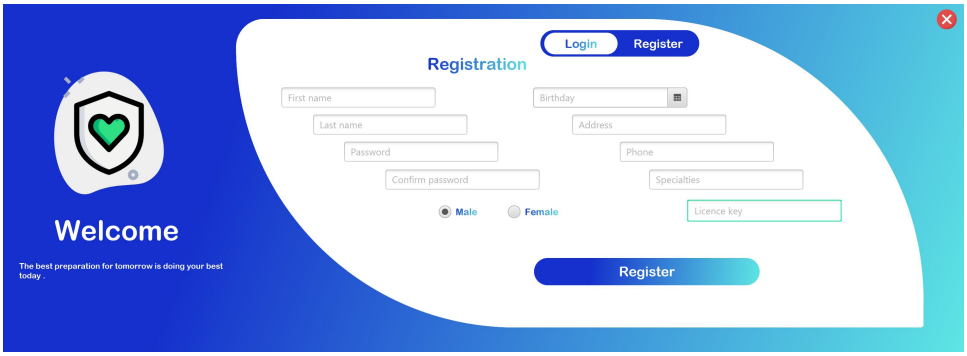


Figure 3.7: The Registration Interface

3.4.3 New Patient Creation:

The interface below existe in the doctor application since the assistant can't access to all patient information.

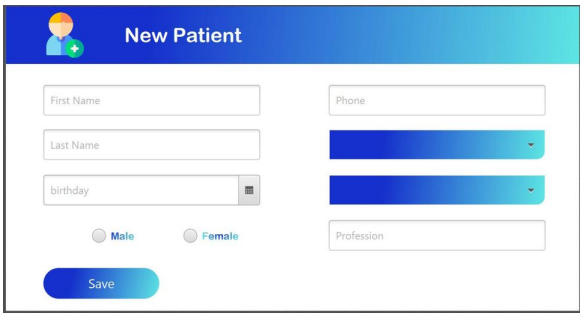


Figure 3.8: The Patient Creation Interface

3.4.4 Prescription management:

This interface represent the medications management(add , modify, delete..) .

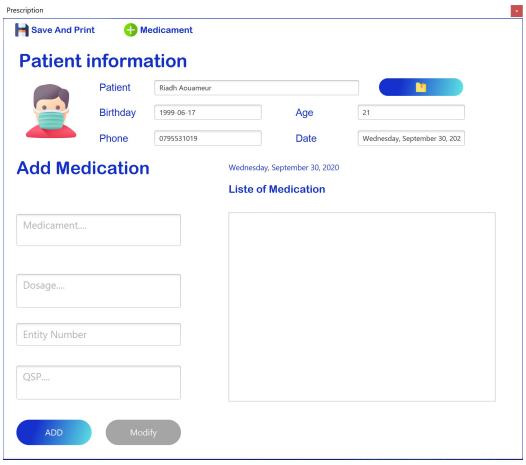


Figure 3.9: The Prescription Interface

3.5 Conclusion :

In the end of this chapter, the language and also the tools used to insure integrated application have been well defined, and the uml diagrams have been represented, in addition to some screenshots of the implemented application interfaces.

GENERAL CONCLUSION

In this final project thesis, the medical application has passed throught the most important required phases to insure its readiness and completeness from the specifications to the design and implementation stage.
First, it was important to represente the environment of the medical institution and define its essential units and members, in addition to the techniques provided by the project to improve the units work like enhancing the communication between the doctor and the assistant using the local network aproach, and also provide alternatives for the appointements traditional booking system such as the advanced access and carve-out models, and use the reminders to reduce patients absence.
Thanks to Uml diagrams, the conception phase became possible to implement by representing the Usecase,Sequence and Class diagram to better understand the relations and events between the running entities inside the medical clinic.
In the implementation phase, after choosing a suitable language (java)and defining all the used tools in the project,the application is finally successfully built.

Bibliography

[1] G. Booch, *The unified modeling language user guide*. Pearson Education India, 2005.

[2] B. Barua, N. Esmail, and T. Jackson, *The effect of wait times on mortality in Canada*. Fraser Institute Vancouver, 2014.

[3] N. Fahmy, A. Aprikian, M. Al-Otaibi, S. Tanguay, J. Steinberg, S. Jeyaganth, M. Amin, and W. Kassouf, "Impact of treatment delay in patients with bladder cancer managed with partial cystectomy in quebec: a population-based study," *Canadian Urological Association Journal*, vol. 3, no. 2, p. 131, 2009.

[4] M. Murray and C. Tantau, "Same-day appointments: exploding the access paradigm," *Family practice management*, vol. 7, no. 8, p. 45, 2000.

[5] M. Murray, M. Davies, and B. Boushon, "Panel size: how many patients can one doctor manage?," *Family practice management*, vol. 14, no. 4, p. 44, 2007.

[6] T. L. Adams, "Mgma: Who we are," *Healthcare Financial Management*, vol. 51, no. 2, pp. 64–65, 1997.

[7] I. Jacobson and S. Bylund, *The road to the unified software development process*, vol. 18. Cambridge University Press, 2000.

[8] M. Heckler, G. Grunwald, J. Pereda, S. Phillips, and C. Dea, *Javafx 8: Introduction by example*. Apress, 2014.

[9] H. W. Lie and B. Bos, *Cascading style sheets: Designing for the web, Portable Documents*. Addison-Wesley Professional, 2005.

`https://tallyfy.com/uml-diagram`

Cardiology definition : `https://www.healio.com/news/cardiology/20120629/what-is-cardiology`