

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

REPUBLIC OF CAMEROON

Paix-Travail-Patrie

**PERSONALIZED PROJECT REPORT**

**INSTITUT AFRICAIN DE**

**L’INFORMATIQUE – CAMEROUN**

**CENTRE TECHNOLOGIQUE DE L’EXCELLENCE PAUL BIYA**

B.P: 13719 Yaoundé

Tel: +237.242.729.957 Fax: 22729958

Courriel:contact@iaicameroun.com

**AFRICAN INSTITUTE OF COMPUTER SCIENCES - CAMEROON PAUL BIYA TECHNOLOGICAL CENTER OF EXCELLENCE**

P.O. Box: 13719 Yaoundé

Tel: +237.242.729.957 Fax: 22729958

E-mail:contact@iaicameroun.com

CONCEPTION AND REALIZATION OF A DIGITAL PLATFORM FOR PROFESSIONAL CONNECTIONS AND OPPORTUNITIES IN CAMEROON

Written By:

**NGATSING TAKAM FRANCK ARTHUR**

Level III student at AICS Cameroon

Academic year: 2024-2025

# DEDICATION

THIS WORK IS

DEDICATED

TO THE TAKAM’S FAMILY

# ACKNOWLEDGMENTS

This document would not have been done without the contribution of a number of people who gave me valuable help and support. To this end, we express our sincere gratitude to the following people:

* First, Thanks to the **Lord Almighty** for all his grace bestowed upon my family, my classmate, teacher and I during this academic year.
* Thanks to **Mr. ARMAND CLAUDE ABANDA,** Resident Representative of AICS Cameroon for his words of wisdom and encouragement to hard work as well as giving us students a good follow up and precious counselling during studies.
* Mr **Mr. FOUAPON HASSAN** the director of **MICKMAQ** and at the same time my professional supervisor for his kindness and youthfulness for permitting us to carry out our internship in their institution.
* We express our gratitude to our academic supervisor **Mr.NGENGE** for his constant availability and support for the realization of this project.
* To our academic teachers **Mr. AGBOR Anderson** AND **Mr. MESSIO** for their advices and assistance in realizing this document.
* My thanks also go to all the staff of AICS Cameroon who relentlessly have impacted knowledge for over two years to prepare us for this challenge.
* Thanks to my parents, family members, class mates, school mates and friends for supporting, collaborating, loving and sharing of ideas with me.
* The countless contributors of open-source programming community, for their great help in learning basic skills and detecting and solving bugs.
* To all my classmates for their collaborative work throughout the academic year.

# CONTENT

[DEDICATION ii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625553)

[ACKNOWLEDGMENTS iii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625554)

[CONTENT iv](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625555)

[LIST OF FIGURES vi](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625556)

[LIST OF TABLES viii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625557)

[GLOSSARY x](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625558)

[ABSTRACT xi](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625559)

[RESUME xii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625560)

[GENERAL INTRODUCTION 1](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625561)

[Preamble 4](#_Toc201625562)

[INTRODUCTION 5](#_Toc201625563)

[I. PRESENTATION OF THE PROJECT THEME 6](#_Toc201625564)

[B. Justification 19](#_Toc201625565)

[A. GENERAL OBJECTIVE 21](#_Toc201625566)

[B. SPECIFIC OBJECTIVES 21](#_Toc201625567)

[III. EXPRESSION OF NEEDS 22](#_Toc201625568)

[ Non-Functional Needs 23](#_Toc201625569)

[**a.** **Chronogram of activities** 26](#_Toc201625570)

[VII. LIST OF PARTICIPANTS AND DELIVERABLES 31](#_Toc201625571)

[a. LIST OF PARTICIPANTS 31](#_Toc201625572)

[ANALYSIS PHASE 32](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625573)

[**PART III** 33](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625574)

[**ANALYSIS PHASE** 33](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625575)

[Preamble 34](#_Toc201625576)

[I. METHODOLOGY 36](#_Toc201625577)

[**REALIZATION PHASE** 83](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625578)

[**I.** **ARCHITECTURE OF THE APPLICATION** 92](#_Toc201625579)

[CHAPTER VI: FONCTIONALITY TEST 98](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625580)

[CHAPTER VII: USER GUIDE 104](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625581)

[Installing MySQL on Windows 107](#_Toc201625582)

[**PART XXI** 109](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625583)

[**ANNEXES** 109](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201625584)

# LIST OF FIGURES

[Figure 1: Doctor Survey Question 1 7](#_Toc201686083)

[Figure 2: Doctor Survey Question 2 8](#_Toc201686084)

[Figure 3: Doctor Survey Question 3 8](#_Toc201686085)

[*Figure 4: Gantt Project* 27](#_Toc201686086)

[*Figure 5: UML 2.5 diagrams overview* 38](#_Toc201686087)

[*Figure6:Realization Branch* 41](#_Toc201686088)

[*Figure 7: Use Case Diagram Formalism* 44](#_Toc201686089)

[*Figure 8:General Use case Diagram* 51](#_Toc201686090)

[*Figure 9: view patient medical record specific Use case Diagram* 52](#_Toc201686091)

[*Figure 10:Manage accounts specific use case* 53](#_Toc201686092)

[*Figure 11:Communication Diagram Formalism* 54](#_Toc201686093)

[*Figure 12 : Communication Diagram of authentication* 56](#_Toc201686094)

[*Figure 13 : Communication Diagram of book an appointment* 57](#_Toc201686095)

[*Figure 14: Sequence Diagram formalism* 58](#_Toc201686096)

[*Figure 15: Sequence Diagram of authentication* 60](#_Toc201686097)

[*Figure 16 : Sequence Diagram of book appointment* 61](#_Toc201686098)

[*Figure 17 : Activity Diagram formalism* 62](#_Toc201686099)

[*Figure 18: Activity Diagram for authentication* 64](#_Toc201686100)

[*Figure 19: Activity Diagram of create medical record* 65](#_Toc201686101)

[*Figure 20:Hardware Architectural Diagram of the system* 70](#_Toc201686102)

[*Figure21:The* *MVC architecture* 72](#_Toc201686103)

[*Figure 22 :Class Diagram Formalism* 73](#_Toc201686104)

[*Figure 23:System class Diagram* 75](#_Toc201686105)

[*Figure 24:Formalism of a state machine diagram* 76](#_Toc201686106)

[*Figure 25:System State Machine Diagram for book appointment* 78](#_Toc201686107)

[*Figure 26:Package Diagram Formalism* 79](#_Toc201686108)

[*Figure 27:System Package Diagram* 81](#_Toc201686109)

[*Figure 28 Component Diagram Formalism* 86](#_Toc201686110)

[*Figure 29:Deployment Diagram Formalism* 89](#_Toc201686111)

[*Figure 30:System Deployment Diagram* 91](#_Toc201686112)

[*Figure31:The MVC* *architecture Source:https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvcarchitecture-and-frameworks-explained/* 94](#_Toc201686113)

[Figure 32 Postman test by email 102](#_Toc201686114)

[Figure 33 Create patient postman test 102](#_Toc201686115)

[*Figure 34: login Page* 108](#_Toc201686116)

[Figure 35: admin dashboard 109](#_Toc201686117)

# LIST OF TABLES

[Table 1: Limitations, Consequence and proposed solution 10](#_Toc201685303)

[*Table 2: Planning of the Project* 26](#_Toc201685304)

[*Table 3:Software Resources(*Source https://www.google.com/search?*)* 28](#_Toc201685305)

[*Table 4:.Hardware Resources Project estimation(Source: mercurial 2023)* 29](#_Toc201685306)

[*Table 5:: Human Resources* 29](#_Toc201685307)

[*Table 6:Total of the Project estimation* 30](#_Toc201685308)

[Table 7: List of participants 31](#_Toc201685309)

[*Table 8: Difference MERISE and UML* 39](#_Toc201685310)

[*Table 9:Use case diagram component* 46](#_Toc201685311)

[*Table 10:Actors of our system.* 46](#_Toc201685312)

[*Table 11:Nominal Scenario for book appointment* 47](#_Toc201685313)

[*Table 12:Nominal Scenario for Authentication* 48](#_Toc201685314)

[*Table 13:Nominal Scenario for create a medical record* 49](#_Toc201685315)

[*Table 14:Nominal Scenario for Authentication* 50](#_Toc201685316)

[*Table 15:Communication Diagram Components* 55](#_Toc201685317)

[*Table 16:Sequence Diagram Components* 59](#_Toc201685318)

[*Table 17:components of an Activity Diagram* 63](#_Toc201685319)

[*Table 18:Class Diagram Components* 74](#_Toc201685320)

[*Table 19:State Machine Diagram Components* 77](#_Toc201685321)

[*Table 20: Package Diagram Component* 80](#_Toc201685322)

[*Table 21: Component Diagram Components* 87](#_Toc201685323)

[*Table 22:Component Diagram Elements* 90](#_Toc201685324)

[*Table 23: Softwares used for the development of the system* 95](#_Toc201685325)

[*Table 24:Technologies used for the development of the system* 96](#_Toc201685326)

# GLOSSARY

* AICS: African institute of computer science
* CEO : **C**hief **E**xecutive **O**fficer
* CSS : **C**ascading **S**tyle **S**heets
* DBMS : **D**atabase **M**anagement **S**ystem
* HTML : **H**ypertext **M**arkup **L**anguage
* ICT : Information Communication Technologies.
* IDEs: **I**ntegrated **D**evelopment **E**nvironments.
* IT: **I**nformation **T**echnology.
* UML: **U**niversal **M**odeling **L**anguage.

# ABSTRACT

Technology has significantly reshaped various sectors, including healthcare, education, and business, enhancing efficiency and productivity. However, the integration of digital solutions in some fields remains underdeveloped. One such area is patient management within dental clinics, where traditional systems often fall short in addressing the growing demand for streamlined operations and improved patient care. This project, conducted in collaboration with the IT company Mickmaq, focuses on the "Design and Implementation of a Patient Management System for a Dental Clinic." The goal is to develop a web application tailored to this clinic’s needs, leveraging modern technology to automate and optimize patient management processes. By integrating real-time communication, automated appointment scheduling, and comprehensive patient record management, this system aims to enhance the efficiency and effectiveness of the clinic’s operations. The findings indicate that the digitalization of patient management can significantly improve clinic performance by reducing administrative errors, enhancing patient satisfaction, and increasing overall productivity. However, the adoption of such digital systems is often challenged by resistance to change, the need for proper training, and ensuring data privacy and security compliance. This project underscores the importance of digital transformation in healthcare and its potential to revolutionize patient management in dental clinics.

**KEYWORDS:**

* **Patient Management**
* **Dental Clinic**
* **Web Application**
* **Real-Time Communication**
* **Automation**
* **Digital Transformation**
* **Healthcare Technology**
* **Data Security**

# RESUME

Top of FormBottom of Form

La technologie a considérablement transformé divers secteurs, notamment la santé, l'éducation et les affaires, en améliorant l'efficacité et la productivité. Cependant, l'intégration de solutions numériques dans certains domaines reste sous-développée. L'un de ces domaines est la gestion des patients au sein d'une clinique dentaire, où les systèmes traditionnels peinent souvent à répondre à la demande croissante pour des opérations simplifiées et des soins améliorés. Ce projet, réalisé en collaboration avec l'entreprise informatique Mickmaq, se concentre sur la "Conception et Mise en Œuvre d'un Système de Gestion des Patients pour une Clinique Dentaire." L'objectif est de développer une application web qui exploite les technologies modernes pour automatiser et optimiser les processus de gestion des patients. En intégrant la communication en temps réel, la planification automatisée des rendez-vous et la gestion complète des dossiers patients, ce système vise à améliorer l'efficacité et l'efficience des opérations de la clinique dentaire. Les résultats montrent que la digitalisation de la gestion des patients peut considérablement améliorer les performances de la clinique en réduisant les erreurs administratives, en augmentant la satisfaction des patients et en accroissant la productivité globale. Toutefois, l'adoption de ces systèmes numériques est souvent confrontée à des défis tels que la résistance au changement, la nécessité d'une formation adéquate et le respect des normes de confidentialité et de sécurité des données. Ce projet souligne l'importance de la transformation numérique dans le secteur de la santé et son potentiel à révolutionner la gestion des patients au sein d'une clinique dentaire

.**MOTS-CLÉS:**

* **Gestion des Patients**
* **Clinique Dentaires**
* **Application Web**
* **Communication en Temps Réel**
* **Automatisation**
* **Transformation Digitale**
* **Technologie de Santé**
* **Sécurité des Données**

# GENERAL INTRODUCTION

Technology is advancing rapidly, transforming every sector and driving a shift toward digital solutions that streamline operations and improve service delivery. In the healthcare field, our dental clinic faces challenges with traditional patient management—manual processes, paper records, and limited communication often result in inefficiency and patient dissatisfaction. Recognizing the need for modernization, we have embraced digital transformation to enhance both patient care and clinic performance. By leveraging secure record management, automated workflows, and real-time communication, we aim to create a more efficient, patient-centered environment. This report details the unique needs of our clinic, the impact of digitalization, and the technical approach—using UML for design, Java Spring Boot for backend, React.js for frontend, and MySQL for data storage—to deliver a robust solution. This project focuses on the DESIGN AND IMPLEMENTATION OF A PATIENT MANAGEMENT SYSTEM SPECIFICALLY FOR A DENTAL CLINIC. The goal is to automate key processes like patient registration, appointment scheduling, treatment tracking, and communication, ultimately improving efficiency, accuracy, and patient satisfaction.

1. **Existing System:** Here, we shall present the already present system in place, that is the one used for consultation and follow-up purposes.
2. **Specification Book:** In this book, we specify the needs of the user taking into considerations the time and cost of the project.
3. **Analysis Document:** Here, we shall present the analysis method chosen together with the presentation of all the diagrams used for the analysis of this project.
4. **Conception phase:** This presents the generic and detailed conception of the project to bring out real world constituents.
5. **The Realisation phase:** This phase will permit us to visualize the implementation process of the solution.
6. **Test of functionalities:** In this phase, we shall present to you the different functionalities or modules of our application and how they work.
7. **The User Guide:** This elaborates on all conditions necessary to use the application and how to use it.

**PART I**

**EXISTING SYSTEM**

## Preamble

The existing system is a document that provides a view of the system currently in place, that is how it carries out its different activities, also it provides a deep understanding of this system associated to the various limitations, the problems that result from these and the solution we propose.

Content

INTRODUCTION

1. PRESENTATION OF THE THEME
2. STUDY OF THE EXISTING SYSTEM
3. CRITICISMS OF THE EXISTING SYSTEM
4. PROBLEMATICS
5. PROPOSED SOLUTION

CONCLUSION

## INTRODUCTION

As Engineers our role in the society is problem-solving which is targeted at making life easier for everyone and in order to bring forth a solution, we ought to ask ourselves some questions to know why the problem exist in the first place and how it is being dealt with at the time. The existing system refers to the system put in place to carry out the work done in the field on which our theme is based. Understanding this system is a great step in solving the problems that we might identify. It’s of great importance that we take into consideration this system before proposing a suitable solution that will ameliorate or replace the current one.

## PRESENTATION OF THE PROJECT THEME

Our theme is “DESIGN AND IMPLEMENTATION OF A PATIENT MANAGEMENT SYSTEM SPECIFICALLY FOR A DENTAL CLINIC.” This application aims to revolutionize the way our clinic manages patient care by integrating both in-person and remote interactions between patients and dental professionals. In circumstances such as the COVID-19 pandemic, where reducing physical contact is essential, the system enables patients to book appointments online, consult dentists through video calls or chat for less urgent concerns, and receive digital prescriptions following diagnosis. Beyond consultations, the platform offers comprehensive treatment history tracking and automated follow-up features that help monitor patient progress and adherence to prescribed care plans. Patients can also search for nearby pharmacies by medication availability and operating hours, facilitating easier access to required drugs.

When laboratory tests are necessary, patients can schedule appointments directly through the system at their convenience. Throughout the treatment period, patients fill out progress forms detailing their health status, and if their condition does not improve after a specified portion of medication is taken, the dentist is promptly alerted to reassess the case. By combining appointment scheduling, remote consultations, pharmacy searches, lab test bookings, and continuous patient monitoring, this system enhances operational efficiency, improves communication, and fosters a more patient-centered approach, ultimately leading to better dental health outcomes and increased patient satisfaction.

1. DESCRIPTION OF THE EXISTING SYSTEM

We notice more and more an overheating of dental clinics due to the exponential increase in patients, especially in urban areas. As a result, dental professionals are often faced with overloaded and poorly organized schedules. Patients frequently encounter challenges during their dental follow-up, including long waiting times, missed opportunities to see a dentist, and lack of timely information about appointment changes or cancellations. Even when appointments are scheduled, patients may arrive to find them canceled without prior notice, leading to frustration and gaps in care.

From a survey conducted in urban dental clinics in Cameroon, it was found that more than 70% of patients sought dental care only when symptoms became severe, often due to financial constraints, lack of awareness, or fear of the dentist. Many reported long waiting times, direct out-of-pocket payments, and a lack of insurance coverage, which limited their access to timely and preventive dental care. Additionally, late consultations frequently resulted in complications such as pulpitis and abscesses, highlighting the need for better patient management and communication systems within dental clinics. This survey and related studies reflect data collected up to August 2023, underscoring ongoing challenges in dental healthcare access and management in Cameroon

Figure 1: Doctor Survey Question 1

* 75% of the doctors work in a hospital while the 25% of the ones left are redistributed equally in a clinical center, health center and medical practice.

Figure 2: Doctor Survey Question 2

* 75% of doctors don’t attend by appointments.

Figure 3: Doctor Survey Question 3

* 66.4% of the doctors work in Yaounde.
* From the above results, we came out with the following observations
* Late presentation and emergency visits: Patients often delay seeking dental care until problems become severe, leading to emergencies.
* Overcrowding and long wait times: Too few dental professionals cause clinics to be overcrowded and patients to wait long periods.
* Poor appointment and follow-up systems: Lack of structured scheduling results in long waits, missed appointments, and poor continuity of care.
* Financial barriers: High out-of-pocket costs and limited insurance make preventive and restorative care inaccessible for many.
* Self-medication and delayed care: Patients frequently self-treat dental issues, worsening their conditions before seeking professional help.
* Limited preventive care and health education: Preventive visits and oral health education are insufficient, so most seek care only when in pain.
* Inadequate infrastructure and resources: Many clinics lack modern equipment, enough staff, and efficient management systems.
* Complications from late consultations: Delayed dental visits often lead to more severe and costly health complications.

1. LIMITATIONS OF THE EXISTING SYSTEM

At the end of our information gathering on the dental consultation process in clinics, we observed that, despite the friendly environment, the current system has significant weaknesses that cannot be ignored. Issues such as long waiting times, poor appointment management, inadequate follow-up, and limited patient communication were identified. Based on these observations, it is important for us to propose targeted solutions to address these challenges and improve the overall efficiency and quality of care within the dental clinic

Table 1: Limitations, Consequence and proposed solution

|  |  |  |
| --- | --- | --- |
| LIMITATIONS | CONSEQUENCE | PROPOSE SOLUTION |
| Long queues for dental consultations, even for minor issues like toothache or gum discomfort | Patients spend hours waiting, leading to frustration and delayed care | Implement an online appointment booking and queue management system to streamline patient flow |
| Patients often self-medicate or delay care due to cost, fear, or lack of nearby dental facilities | Worsening of dental conditions, increased emergencies, and complications | Enable remote dental consultations (video/chat) for early advice and triage, reducing unnecessary in-person visits |
| High transport costs for patients in remote areas without easy access to dental clinics | Financial burden and missed or delayed dental care | Offer tele-dentistry options so patients can consult dentists without traveling long distances |
| Poor communication between dentist and patient, with limited follow-up after procedures | Patients may not adhere to treatment plans, leading to poor outcomes | Integrate secure messaging and follow-up reminders to maintain ongoing communication and monitor recovery |
| Difficulty finding prescribed dental products or medications in local pharmacies | Delays in starting treatment and risk of complications | Develop a pharmacy locator that shows availability of dental products and open pharmacies near the patient |
| Manual appointment and schedule management | Double bookings, missed appointments, and administrative errors | Digitize appointment scheduling and automate reminders to improve accuracy and reduce confusion |
| Outpatient follow-ups require patients to return to the clinic for every review | Increased transport costs and missed follow-ups | Use daily online check-ins and symptom surveys, alerting dentists if issues arise for timely intervention |

1. PROBLEMATIC

Effective communication is the cornerstone of quality dental care, and it is crucial that interactions between patients and dentists are made as simple, clear, and accessible as possible, given the direct impact on patient health. Considering the challenges observed in dental clinics—such as long waiting times, poor appointment management, and inadequate follow-up—we ask: **HOW CAN WE FACILITATE ONLINE COMMUNICATION BETWEEN PATIENTS AND DENTISTS, IMPROVE PATIENT FOLLOW-UP, AND STREAMLINE ACCESS TO DENTAL** CONSULTATIONS AND SERVICES? Our application aims to provide a comprehensive solution by enabling real-time video consultations and secure messaging for clear dialogue, automating follow-up processes to monitor treatment progress, and managing appointment scheduling efficiently. By leveraging these features, the platform will help reduce overcrowding, improve continuity of care, and make dental services more accessible and patient-centered.

1. PROPOSED SOLUTION

After reviewing and analyzing the current workflow in our dental clinic, we propose the design of a web application specifically tailored for a single dental clinic environment. The application will provide the following features for both dental professionals and patients:

**For Dental Professionals**

* Electronic agenda to efficiently manage availability and schedule patient appointments.
* Secure chat system for direct communication with patients regarding appointments, treatment updates, and follow-up questions.
* Digital management of patient records, including treatment plans and prescriptions, for easy access and organization.

**For Patients**

* Ability to view available appointment slots and book, reschedule, or cancel appointments through the web application.
* Receive real-time notifications for appointment confirmations, reminders, and any schedule changes.
* Access to a secure messaging system to communicate with dental staff for inquiries or post-treatment follow-up.
* View and download their digital treatment plans and prescriptions for reference and adherence.

CONCLUSION

Having reached the end of this section, we conducted a detailed analysis of the current workflow in our dental clinic by engaging with dental professionals through targeted surveys. This approach allowed us to clearly identify the limitations of the existing system and understand the resulting challenges faced by both staff and patients. Based on these findings, we proposed practical solutions to address each limitation. Ultimately, we recommend the development of a web application specifically designed to streamline appointment management, enhance patient engagement, and improve communication within the clinic1. This step is essential, as a thorough understanding of the current system is necessary before implementing improvements that will optimize and modernize the clinic’s operations.

PART II

SPECIFICATION BOOK

PART II

SPECIFICATION BOOK

Preamble

This section of our report will cover details on the product to be delivered with the agreement of the client(s) and the solution provider. In this section, we are going to present the context in which we are to but in place a platform, what the system should do and how the system should do it.

Content

INTRODUCTION

1. CONTEXT AND JUSTIFICATION
2. OBJECTIVES OF THE PROJECT
3. EXPRESSION OF NEEDS
4. ESTIMATED COST OF THE PROJECT
5. PROJECT PLANNING
6. LIST OF PARTICIPAANTS AND DELIVERABLES.

CONCLUSION

INTRODUCTION

The specification book gives us the different directives on the product to be delivered with the agreement of the client(s) and the solution provider. In this section, we are going to present the context in which we are to but in place a platform, what the system should do and how the system should do it. These specifications are to avoid the production of inadequate results. We will also include project detail such as the team involve, constraints, the budget, deadlines, constrains and the deliverable.

1. CONTEXT AND JUSTIFICATION
2. Context

Cameroon, with a population of nearly 29 million, faces significant challenges in healthcare delivery, particularly during times of crisis or increased demand. The country has a very low density of healthcare professionals—about 1.1 physicians and 7.8 nurses/midwives per 10,000 people[3](https://data.who.int/countries/120). The shortage is even more acute in dental care, with only one dental surgeon per 100,000 inhabitants, and the majority of these professionals concentrated in urban centers like Yaoundé and Douala

x

We can see from the above that, of the 48 people who answered to the questions from the survey, 87,5% encounter queues when going to the hospital. Also,

We can also notice that 75% of those who answered require follow-up each time they are prescribed a drug or each time they are on medication.

## Justification

The dental healthcare sector, like much of the health industry, is increasingly overwhelmed by the growing number of patients. In many clinics, including ours, this leads to long waiting times, difficulty in scheduling timely consultations, and challenges in efficiently managing patient records and follow-ups. Patients often arrive without knowing the availability of the dentist, and many wait in queues for hours. After consultations, they may face uncertainty—especially when the prescribed medications are unavailable at the clinic’s pharmacy or difficult to find in nearby pharmacies.

To address these issues, we developed **WAMS**, a dedicated **web application** designed specifically to **digitalize and streamline the operations of our dental clinic**. With WAMS, patients can book appointments online based on their availability, reducing physical queues and improving time management for both patients and clinic staff. The platform also enables patients to attend **remote consultations** when needed, which is especially beneficial in minimizing clinic overcrowding and reducing the risk of infections.

WAMS goes beyond appointment scheduling. It offers centralized **patient data management**, making it easier for dentists to access medical histories, track ongoing treatments, and provide personalized follow-up care. Additionally, WAMS includes a smart **pharmacy search tool**, allowing patients to quickly locate nearby open pharmacies that stock their prescribed medications—helping them save time and avoid the frustration of going from pharmacy to pharmacy.

Furthermore, WAMS allows patients to **book lab tests online**, such as dental X-rays and oral screenings, ensuring a fully digital and seamless experience from diagnosis to treatment. By digitalizing these key processes, **WAMS enhances the quality, accessibility, and efficiency of care** within our dental clinic—ultimately improving the experience for both patients and healthcare professionals.

1. OBJECTIVES OF THE PROJECT

## 

## GENERAL OBJECTIVE

The general objective in this project is to develop a web application that provides computerized management and track keeping of the patients of a dental clinic. The goal is to optimize and improve the activities of a dental clinic.

## SPECIFIC OBJECTIVES

DENTIST:

* Access his account to manage user accounts, including creating, modifying, or deleting accounts for patients and nurses.
* Monitor the overall system including having access and manipulating the local database.
* Receive notifications or alerts regarding any critical issues or malfunctions within the system.
* View and analyze data collected from patients.
* Generate medical documents including numerized medical booklet to keep track of patients’ situations, and billing for services.
* Manage User’s accounts (enable, disable, modify) and generate user’s unique identification keys.

nurse:

* Login to the application using their account credentials.
* Access a dashboard displaying her interface
* Create new patient account
* View the dentist’s planning

patient:

* Login using their credentials.
* Access their interface that will display their personal information and function buttons
* View availability of dentist
* Request for an appointement
* Request a Postpone to a fixed appointments
* Contact dentist

The dentist is the administrator of the application, he manages his local database and all other users. Patients and nurses cannot access the confidential informations of other patients

## EXPRESSION OF NEEDS

* 1. Functional Needs

**Dentist:**

* User Management
* Register and manage user accounts, including authentication and access privileges.
* Handle user-related issues, such as password resets and account verifications.
* Planning management
* Makes a weekly planning within the application.
* Receives notifications within the app.
* Medical record management
* Update patient’s informations.
* Suspend or delete medical document.
* Appointments management
* Confirms an appointment booked by a patient.
* Rejects appointments requested by patients.

**Patient:**

* Book appointments
* Request for a appointment or postpone.
* View dentist availability.
* Communication with the dentist
* Communicate with the dentist through the WhatsApp App.
* Send feedbacks or report issues related to buccal health.

**Nurse:**

* Register new patients
* Create accounts for the patients of the app.
* Contact dentist
* Contact the dentist through chat.

## Non-Functional Needs

* **Performance:**
* Responsiveness: The application should respond quickly to user interactions and provide a smooth user experience.
* Efficiency: The application should be optimized to perform tasks efficiently, minimizing resource usage and response times.
* Scalability: The application should be capable of handling increasing user loads and data volumes without significant performance degradation.
* Reliability: The application should be reliable and available, minimizing downtime and disruptions.
* **Usability**
* Intuitive Interface: The application should have a user-friendly and intuitive interface that is easy to navigate and understand.
* Accessibility: The application should be accessible to users with disabilities, complying with accessibility standards and guidelines.
* Multilingual Support: The application should support multiple languages to cater to a diverse user base.
* Consistency: The application should maintain consistent design elements and interactions throughout different screens and modules.
* **Security**
* User Data Protection: The application should securely store and handle user data, following best practices for encryption and data privacy.
* Authentication and Authorization: The application should provide secure user authentication mechanisms and access controls to protect user accounts and data.
* Secure Transactions: The application should ensure secure transmission and handling of sensitive informations.
* Data Backup and Recovery: The application should have robust backup and recovery mechanisms to protect against data loss and enable quick recovery.
* **Compatibility**
* Cross-Platform Compatibility: The application should be compatible with different operating systems and devices.
* Browser Compatibility: The web application should be compatible with major web browsers, ensuring consistent functionality and appearance.
* API Integration: The application should be able to integrate with external systems or APIs seamlessly.
* **Performance Monitoring and Analysis**
* Logging and Monitoring: The application should log events and errors for monitoring and troubleshooting purposes.
* Analytics and Insights: The application should provide analytics and insights on user behavior, system performance, and usage patterns.
* Performance Optimization: The application should continuously optimize its performance based on analytics and monitoring data.
* **Maintainability**
* Modularity: The application should be designed with a modular structure, making it easier to maintain and enhance individual components.
* Code Quality: The application's code should follow best practices and coding standards, ensuring readability and maintainability.
* Documentation: The application should have comprehensive documentation, including installation instructions, user guides, and API documentation.
* **Integration and Interoperability**
* Integration Capabilities: The application should support integration with third-party services, platforms, or APIs to extend its functionality
* Inter-operability: The application should be able to exchange data and interact with other systems effectively. Project managers coordinate tasks and ensure timely delivery. Collaboration and teamwork among these professionals are crucial.

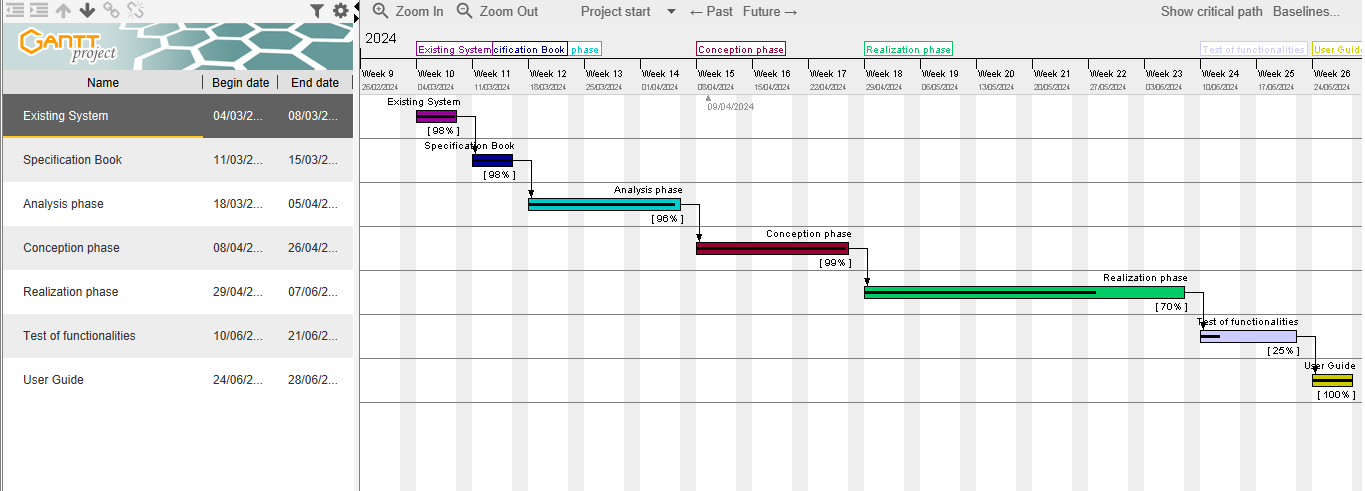
1. PLANNING THE PROJECT

### **Chronogram of activities**

*Table 2**: Planning of the Project*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Objective | | Outing | Duration | Period |
| EXISTING SYSTEM | | Study of the existing system | Existing System | 5 days | PERIOD |
| SPECIFICATION BOOK | | Specification of the user needs | Specification book | 5 days | PERIOD |
| ANALYSIS | | Capture of needs Use case and textual description Modelling | Analysis | 15 days | PERIOD |
| CONCEPTION | | Preliminary conception and Detailed conception | Conception document | 15 days | PERIOD |
| REALIZATION | Implementation Unitary test Integration Test Development, Deployment, Component diagrams | | Realization Document | 30 days | PERIOD |
| TESTING AND MAINTENANCE | Testing of the software and debugging | | User testing | 5 days | PERIOD |
| INSTALLATION AND USER GUIDE | Documenting software | | User Guide | 12 days | PERIOD |

Gantt Diagram Project Diagram



*Figure 4: Gantt Project*

1. ESTIMATE OF THE PROJECT
2. Software Resources

*Table 3**:Software Resources(*Source <https://www.google.com/search>?*)*

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Usage | Quantity | Price (FCFA) |
| Google chrome | A free and opensource web browser use to make  research on the project. | 1 | Freemium |
| Windows 10 | A system software uses to manage our computer  hardware and software. | 1 | 49,850 |
| laragon | A local server used to create and manage our  database. | 1 | Freemium |
| React.js | Front-end development | 1 | Freemium |
| Visual Paradigm | Modeling tool which was use for the modeling of  the system to be developed. | 1 | 165,540 |
| Visual studio Code | Front-End Development environment | 1 | Freemium |
| Postman | API development and testing tool | 1 | Freemium |
| TOTAL 1 | | | **215,390** |

1. Hardware Resources

*Table 4**:.Hardware Resources Project estimation(Source: mercurial 2023)*

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Usage | Quantity | Price (FCFA) |
| hp computer | The computer was used for the development of the application | 1 | 450,394 |
| Printer | Use to print out the report | 1 | 546 250 |
| Modern | For the downloading of resource and achievement of information | 1 | 126,500 |
| Removable Disk | Referring and testing app on different machine |  | 20,394 |
| TOTAL 2 | | | **1,143,538** |

1. Human Resources

*Table 5**:: Human Resources*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ROLE | NUMBER OF DAYS | QUANTITY | COST PER DAY | TOTAL PRICE (FCFA) |
| Project Manager | 90 | 01 | 30 000 | 2 700 000 |
| Analyst | 21 | 01 | 25 000 | 525 000 |
| UI/UX Designer | 07 | 01 | 20 000 | 140 000 |
| Programmer | 30 | 01 | 15 000 | 450 000 |
| Tester | 14 | 02 | 10 000 | 140 000 |
| Margin error | / | / | / | 1,500,000 |
| TOTAL 3 |  |  |  | **5,455,000** |

1. Total of the Project estimation

*Table 6:Total of the Project estimation*

|  |  |  |
| --- | --- | --- |
| TOTAL | RESOURCE | AMOUNT(FCFA) |
| TOTAL 1 | Software Resources | **215,390** |
| TOTAL 2 | Hardware Resources | **1,143,538** |
| TOTAL 3 | Human Resources | **5,455,000** |
| FINAL TOTAL |  | **6,813,928** |

1. CONSTRAINTS
2. **Quality**

For the development of our system, we have sufficiently robust tools to guarantee a minimum of security, extensible and excellent Scalability. Moreover, the programming phase will have to follow all the technical standards for a better performance in a reduced execution time, this is why the choice of the development technologies is crucial.

1. **Time Constraint**

The project will be realized in 13 weeks (3 months) starting from the beginning date, July 1st, 2024.

1. **Cost constraint**

The realization of our project will require expenditures in human resources, material and software a total cost of **6,925,528FCFA.**

## LIST OF PARTICIPANTS AND DELIVERABLES

### LIST OF PARTICIPANTS

Table 7: List of participants

|  |  |  |
| --- | --- | --- |
| NAME | FUNCTION | ROLE |
| NGASTING TAKAM FRANCK | AICS Student | Student at AICS |

1. DELIVRABLES
2. In project management, any component materializing the result of a realization service is called a deliverable. In the case of our project, the deliverables are: A report composed of the following document

* The application.
* The user guide.
* The powerpoint.

CONCLUSION

# ANALYSIS PHASE

The specifications book contains detailed information about the project's objectives, requirements (both functional and non-functional), team members, project plan, cost, and constraints. This document serves as a guide to develop a solution that meets the project's needs while avoiding errors and compatibility issues. The specifications book acts as a foundation for the analysis phase of the project, where we carefully examine the provided information to determine the best approach and ensure a successful outcome.

# PART III

# ANALYSIS PHASE

## Preamble

After specification book, we have the Analysis phase which permits us to represent a detailed analysis of the limitations identified in our context, and our solution, through a software development process and modelling language.

Content

NTRODUCTION

1. METHODOLOGY
2. COMPARATIVE STUDY OF UML AND MERISE.
3. COMPARATIVE STUDY OF UNIFIED PROCESSES.
4. CHOICE OF THE ANALYSIS METHOD
5. MODELLING OF THE PROPOSED SOLUTION

CONCLUSION

INTRODUCTION

The analysis book permits us to examine in an explicit way the existing system, it’s limitations and how we can remedy them. We will also describe in details the modeling language known as UML (Unified Modeling Language) which is coupled with Two Tract Unified Process (2TUP) to form a method and its justification why we decided to use it in preference of another. then we will dive directly into the modeling of the proposed solution consisting of diagrams that meets the requirements of the functional needs.

## METHODOLOGY

1. COMPARATIVE STUDY OF UML AND MERISE

We have studied in details some analysis methods by looking at its objectives, its structuring, its pros and cons, so as to choose an analysis method which is reliable and adaptive to our project, below are some analysis methods we studied and a brief explanation in order for you to understand why we made our choice;

* 1. **MERISE:** it is an information system design and development widely used in France. It was first early introduced in 1980s.
  2. **Agile:** this methodology is growing in popularity, thanks to highly competitive business environment and increased innovation. In general, agile methodologies prioritized shorter, interactive cycle and flexibility.
  3. **Scrum:** this is the most popular agile development framework because it is relatively simple to implement. It also solves so many problems that software developers struggle with in the past, convoluted development cycles, project plan, and shifting production schedules. This methodology allows for rapid development and testing, especially with small teams.
  4. **APF:**  which stands for ADAPTIVE PROJECT FRAMEWORK, it grows from the difficulty in managing most IT projects using traditional project management methods due to uncertain and changing requirement. APF begins with a requirement breakdown structure (RBS) to define strategic goals based on productive requirements, functions, sub-function and features. The project proceeds in iterative stages, and at the end of each step, teams evaluate previous results to improve performance and practices.

**XP:** which stands for Extreme Programming is a software development methodology that advocate frequent releases in short development cycles, which is intended to include checkpoints for the adoption of new customer requirements and improve productivity. This methodology takes its name from the idea that the traditional software engineering practices are taken to extreme levels.

1. MODELING WITH UML (2.5)

The unified modeling language (UML) is a general purpose, developmental modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The Unified Modeling Language (UML) was standardized in January 1997 by the Object Management Group (OMG) which is an American association created in 1989 and aims to promote and standardize the object model in all it forms. In 2005, UML was also published by the international organization for standardization (ISO) as an approved ISO standard. Uml since 2015 is in its version 2.5. This version consists of fourteen diagrams classified into structural and behavioral diagrams

1. Structural Diagrams:

Structural diagrams represent the static components of a system; they emphasize on what should be in the system we are modeling. They include:

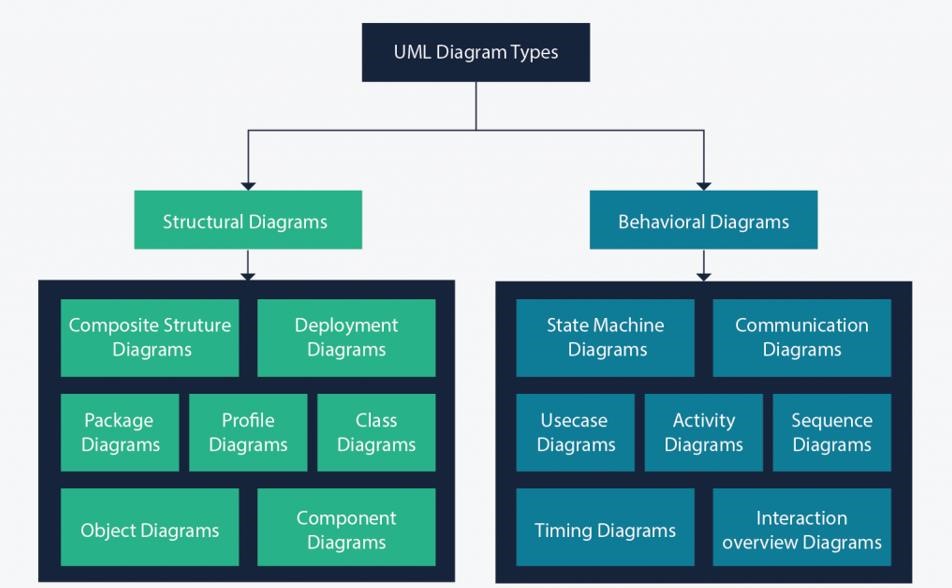
* Class diagram;
* Object diagram;
* Package diagram;
* Composite structural;
* Deployment diagram;
* Component diagram;
* Profile diagram.

1. Behavioral Diagrams:

The behavioral diagrams capture the dynamic state of a system; they emphasize on what should happen in the system we are modeling. They are:

* + Use case diagram;
  + Activity diagram;
  + State machine diagram;
  + Sequence diagram;
  + Communication diagram;
  + Interaction overview diagram;
  + Timing diagram

1. UML 2.5 diagrams overview:



*Figure 5**: UML 2.5 diagrams overview*

*Source:* [*https://creately.com/blog/diagrams/uml-diagram-types-examples/*](https://creately.com/blog/diagrams/uml-diagram-types-examples/)

It is important to note that UML is a modeling language and not a method or procedure. Hence, to give it an approach, we need to associate UML to a Unified process (UP) in order to give our conception a methodology to follow. A unified process is a generic method for developing software. This implies it is necessary to adapt the UP to the context of the project, team domain and or the organization. We will use the Two Tracks Unified Process (2TUP) throughout our project to implement our solution.

Difference between UML and MERISE

*Table 8**: Difference MERISE and UML*

|  |  |
| --- | --- |
| MERISE | UML |
| It stands for Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprises | Unified Modeling Language |
| MERISE is a systemic method of analysis and design of information systems. That is, it uses a system approach. | UML is however not a method but rather an object modeling language to which it is necessary to associate an approach to make it a method. This is the case with the 2TUP method; RUP and XP. |
| MERISE proposes to consider the real system from two points of view: - A static view (data) - A dynamic view (treatments). That is, with the MERISE method, we have a separate study of the data and the treatments. | UML offers a different approach from that of MERISE in that it combines data and processing. Because with UML, centralizing the data of a type and the associated processing makes it possible to limit the maintenance points in the code and facilitates access to information in the event of software development. In addition, UML describes the dynamics of the information system as a set of operations attached to the objects of the system. |
| Rational | Object |

1. COMPARATIVE STUDY OF UNIFIED PROCESS
2. The Process of Development of a Software

A process can be defined as a partially sequence of steps that permits us to obtain software systems or evolution of an existing one. The main objective of software development is the production of quality software that response to the needs of the users during a particular time and at a particular cost.

1. A Unified Process:

A unified Process is a process of development of software constructed on UML; it is iterative, incremental, centered on architecture, driven by use cases and requirements.

**Iteration** is distinct sequence of activities with a basic plan and evaluation criterion that produces an internal or external output. Either the content of an iteration is improved or the evolution of the system is evaluated by users.

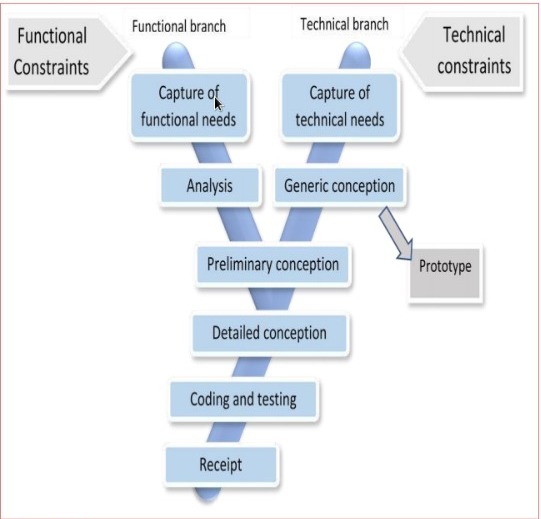
**An increment** is the difference between two released products at the end of two iterations. Each iteration that the group is capable of integrating the technical environment in order to develop a final product and give users the possibility of having tangible results.

**Centered on architecture** the different models derived during the establishment of system must be reliable and coherent.

**Driven by use case and requirements** enables the clear definition of a users’ needs and priorities respectively thereby minimizing the risk of project failure.

1. The Two Track Unified Process (2TUP)

2TUP is a unified process which is belt on UML and has as objective to bring solution to constraints of functional and technical changes imposed on information systems by strengthening controls on development capacities. It proposes a Y-sharped development life cycle that separates the functional aspect from the technical aspects, and the merging of these two forms the implementation aspect. 2TUP distinguishes therefore two branches: the functional and technical branches, the combination of the result of these two branches forms the third: the realization branch – where we realize our system. The diagram bellow illustrates the branches of 2TUP.



*Figure6**:Realization Branch*

*Source:https://image.slidesharecdn.com/2tup-presentation-121226144316-phpapp01/85/mthodologie-2-track-unified-process-7-320.jpg?cb=1666089460*

1. The left branch (functional branch)

It captures the functional needs of a system. This ensures the production of software that meets the needs/requirements of the user. The analysis here consists of studying precisely the functional specification in order to obtain an idea of what the system is going to realize and its result does not depend on any technology.

1. **The right branch (Technical branch)**

The technical branch enumerates the technical needs and proposes a generic design validated by a prototype. The technical needs include constraints and choices related to the conception of the system, the tools and equipment as well as the integration constraint with the existing system condition. The different diagrams are shown in the table below.

1. **The middle branch (Realization or implementation branch)**

In this branch, we study the preliminary conception, detailed conception, and documentation of the system. The realization branch supports the following:

**Preliminary conception:** This is the most sensitive step of 2TUP as it is the confluence of the functional and technical branch. It is completed when the deployment model, the operating model, the logical model, inter-phases and the software configuration model are defined.

**Detailed conception:** This is the detailed design of each feature of the system.

**Coding and testing:** This are the phase where we program the designed features and test the coded features.

**The recipe:** Also known as the deliverable is the validation phase of the functions of the developed system.

1. JUSTIFICATION OF METHOD OF ANALYSIS

The reason why we chose UML modelling language and the software development process 2TUP instead of many others that exist, include:

UML is the current standard for programming in an object-oriented language. For this reason, it is widely understood and well known making it easy for a new programmer to join the project and be productive from the very first day.

UML diagrams allow teams to visualize how a project is or will be working, and they can be used in any field, not just software engineering. The diagrams will allow teams to visualize together how a system or a process will work or did work. It can provide new ideas for how teams have to collaborate to achieve the goal of the work-flow process.2TUP is centered around the creation and maintenance of a model, rather than the production of mountain documents.

2TUP is user oriented as it permits the development of software that responds to the needs of the users through the study of the user needs.

2TUP is iterative and incremental, hence it enables the project team to produce refined amelioration if necessary and easily integrate it in the already existing system. 2TUP by permitting the project team identify and test the key functionalities of the system limits the risk related to building the system.

1. MODELING OF THE SYSTEM
2. Capture of Functional Needs:

The first step of the left (functional) branch of Two Track Unified Process (2TUP) is the capture of the functional needs. At this step, we capture the intended behavior of the system that maybe express as services, tasks or functions the system is required to perform.

1. *Use Case Diagram*
2. **Definition**

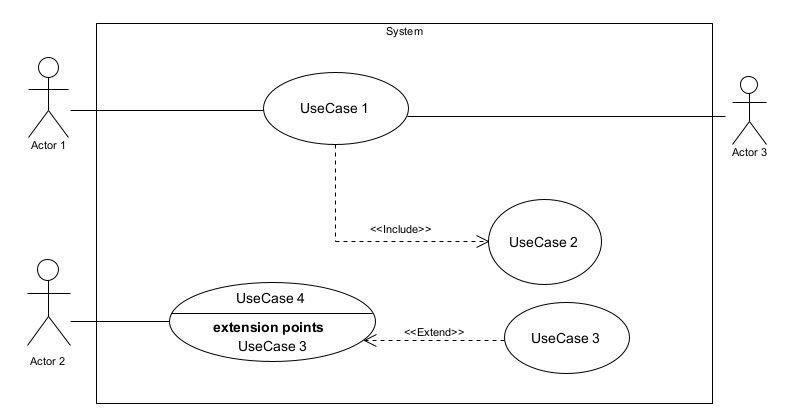
A use case diagram shows the functionalities of a system, their inter-dependencies and how they relate with actors of the system. A use case is a specification of behavior.

The main objectives of the use case diagram are:

* Provide a high-level view of the system;
* Identify the functions of the system.

Use case diagrams are completed with a textual description of each use case that is intended to define the use case in greater details.

1. **Formalism**



*Figure 7**: Use Case Diagram Formalism*

1. **Elements of a Usecase diagram**

|  |  |  |
| --- | --- | --- |
| Elements Notation Description | | |
| Actors |  | Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system. |
| Use case | Use Case  1 | A use case represents a functionality of the system. It is an action that can be performed by an actor. |
| Association |  | it indicates that an actor takes part in a use Case. |
| Include |  | An inclusion denotes that an included action must be performed before the including action can be performed. |
| Extend |  | An extension denotes that an extending action may be performed while an extended action is being performed. |
| Generalization |  | This shows that an actor or a use case is a kind of another abstract or concrete actors can be defined and later specialized using generalization relationship. |
| System |  | It is a container of use cases which interact with external actors |

*Table 9**:Use case diagram component*

1. **The Actors of Our System**
2. Primary Actors

*Table 10**:Actors of our system.*

|  |  |
| --- | --- |
| Actor | Role |
| Dentist | manage planning, manage medical document, manage appointments, billing. |
| Nurse | Record new patients, view the planning |
| Patient | Book appointment, contact doctor, view medical document |
| Admin | Manages accounts, manage roles and permissions |

Nominal Scenario for Book an appointment

*Table 11:Nominal Scenario for book appointment*

|  |  |
| --- | --- |
| TITLE | CONTROL |
| Summary | The user wants to book an appointment |
| Actors | patient |
| Precondition(s) | The actor has an account on the platform  The app is launched  Has Internet connection  The user has successfully authenticated |
| Triggers | The user clicks on button book appointment |
| Nominal Scenario | -The actor launches the application and clicks on book appointment button.  -The system displays form.  -The actor fills and submits the form.  -The system verifies conformity of the form fields  -The system sends the data to the DBMS.  -The system processes the query  -The DBMS returns result of the query.  -The system displays a success message to the actor. |
| Alternative Scenarios | -At step 4) of the nominal scenario, the user enters  mismatched or missing information  -The system displays an error message then returns to step 2 of the nominal scenario. |
| Postcondition of success | The actor has successfully booked an appointment and it is recorded in the database |
| Postcondition for failure | The user remains on the appointment booking form |
| Non-functional requirements | The system must handle at least 100 concurrent users appointment booking |

1. Nominal Scenario for Authentication

*Table 12:Nominal Scenario for Authentication*

|  |  |
| --- | --- |
| TITLE | CONTROL |
| Summary | The user needs to authenticate |
| Actors | Dentist, nurse, patient |
| Precondition(s) | The actor has an account on the platform  The app is launched  Has Internet connection |
| Triggers | The user clicks authentication button |
| Nominal Scenario | -The actor launches the application.  -The system displays the login form.  -The actor fills and submits the form.  -The system verifies conformity of the form fields  -The system sends the data to the DBMS.  -The DBMS returns result of the query.  -The system displays a success message to the actor. |
| Alternative Scenarios | -At step 4) of the nominal scenario, the user enters  mismatched or missing information  -The system displays an error message then returns to step 2 of the nominal scenario. |
| Postcondition of success | The actor has access to the dashboard, depending on the type of account |
| Postcondition for failure | The user does not have access to the platform and is sent back to the login form |
| Non-functional requirements | Entering of the password must not be visible on the screen |

Nominal Scenario for create a medical record

*Table 13:Nominal Scenario for create a medical record*

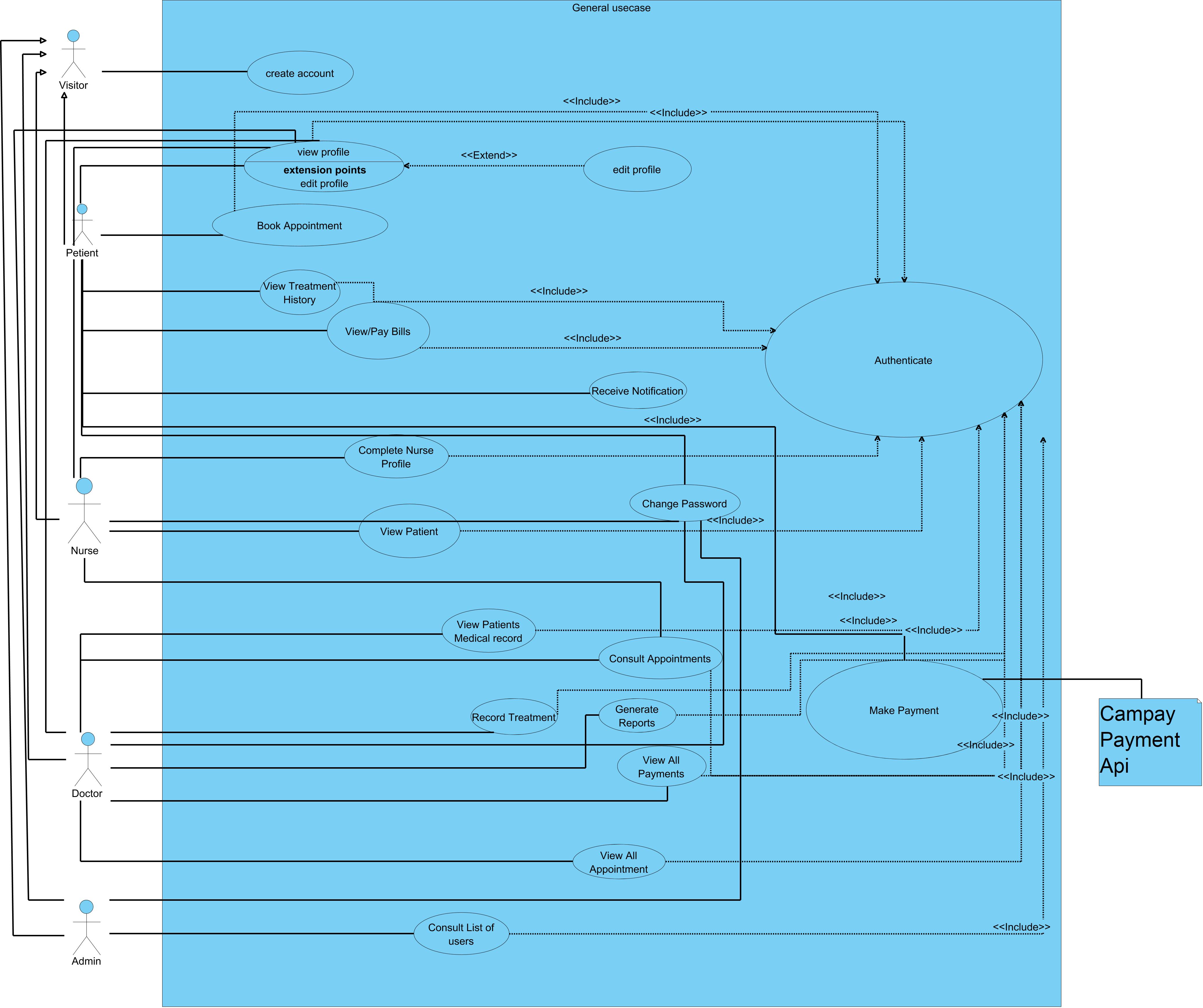
|  |  |
| --- | --- |
| TITLE | CONTROL |
| Summary | The user wants to create a medical record |
| Actors | nurse |
| Precondition(s) | The actor has an account on the platform  The app is launched  The user has successfully authenticated |
| Triggers | The user clicks on button create medical record |
| Nominal Scenario | -The actor launches the application and clicks on create medical button.  -The system displays form.  -The actor fills and submits the form.  -The system verifies conformity of the form fields  -The system sends the data to the DBMS.  -The system processes the query  -The DBMS returns result of the query.  -The system displays a success message to the actor. |
| Alternative Scenarios | -At step 4) of the nominal scenario, the user enters  mismatched or missing information  -The system displays an error message then returns to step 2 of the nominal scenario. |
| Postcondition of success | The actor has successfully create the medical and it is recorded in the database |
| Postcondition for failure | The user remains on the create medical record form |
| Non-functional requirements | The system should be able to accommodate an increasing number of patients and records without requiring significant re-engineering. |

1. Nominal Scenario for Authentication

*Table 14:Nominal Scenario for Authentication*

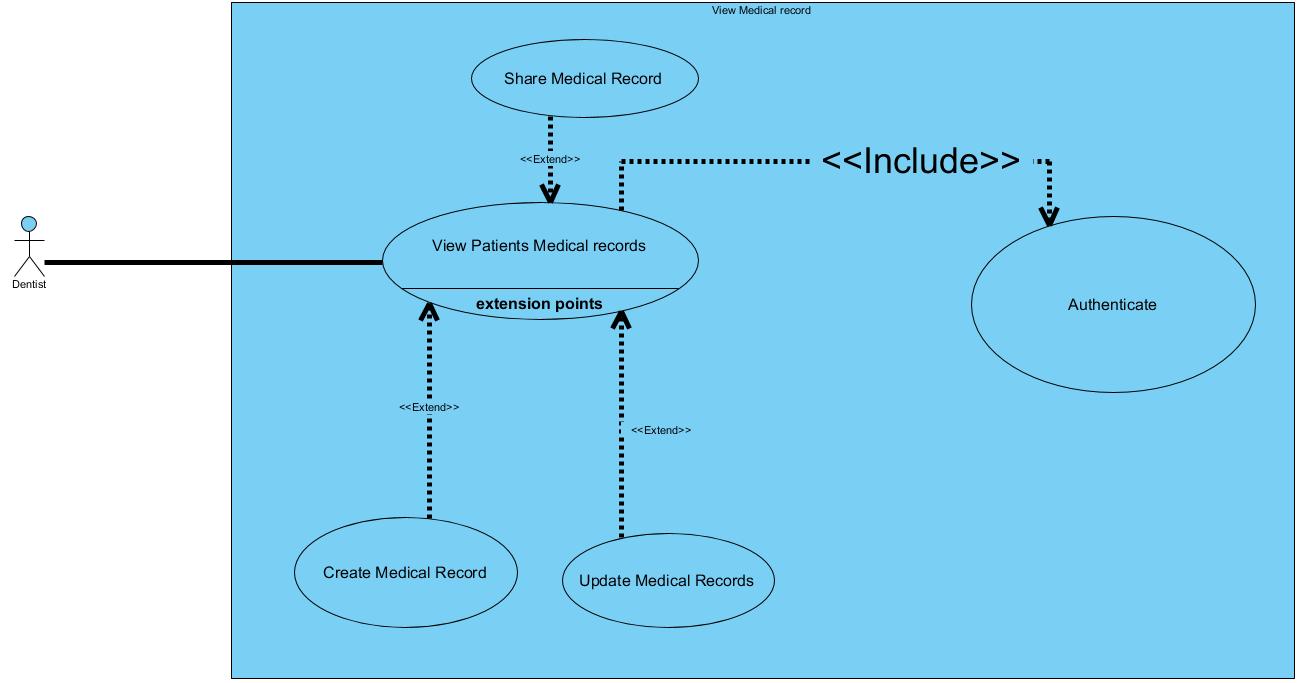
|  |  |
| --- | --- |
| TITLE | CONTROL |
| Summary | The user needs to authenticate |
| Actors | Dentist, nurse, patient |
| Precondition(s) | The actor has an account on the platform  The app is launched  Has Internet connection |
| Triggers | The user clicks authentication button |
| Nominal Scenario | -The actor launches the application.  -The system displays the login form.  -The actor fills and submits the form.  -The system verifies conformity of the form fields  -The system sends the data to the DBMS.  -The DBMS returns result of the query.  -The system displays a success message to the actor. |
| Alternative Scenarios | -At step 4) of the nominal scenario, the user enters  mismatched or missing information  -The system displays an error message then returns to step 2 of the nominal scenario. |
| Postcondition of success | The actor has access to the dashboard, depending on the type of account |
| Postcondition for failure | The user does not have access to the platform and is sent back to the login form |
| Non-functional requirements | Entering of the password must not be visible on the screen |

1. General Use Case Diagram



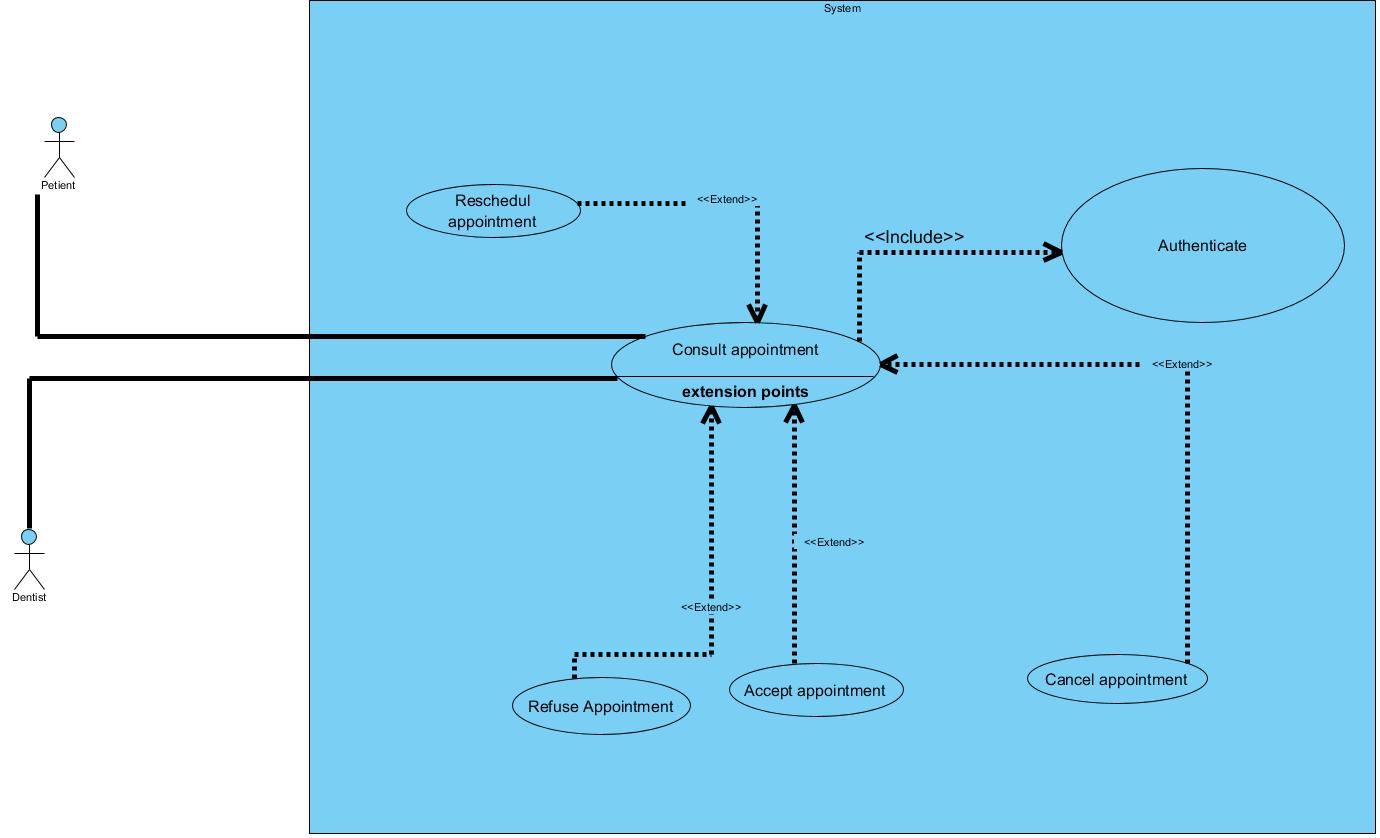
*Figure 8**:General Use case Diagram*

1. View patient medical record specific Use case Diagram



*Figure 9**: view patient medical record specific Use case Diagram*

1. Book appointment specific Use Case Diagram

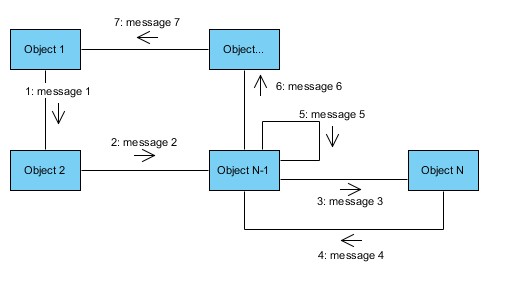


*Figure 10:Manage accounts specific use case*

Communication Diagram

* 1. Definition:

It is a diagram which is used to show the relationship between the actors of a system, both the sequence and the communication diagrams represent the same information but differently. Instead of showing the flow of message. It depicts the architecture of the object residing in the system as it is based on object-oriented programming.

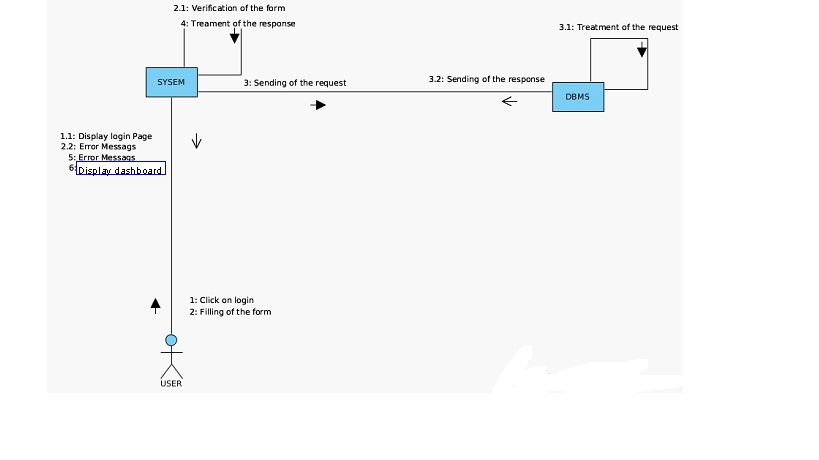
* 1. Formalism

*Figure 11**:Communication Diagram Formalism*

*Table 15**:Communication Diagram Components*

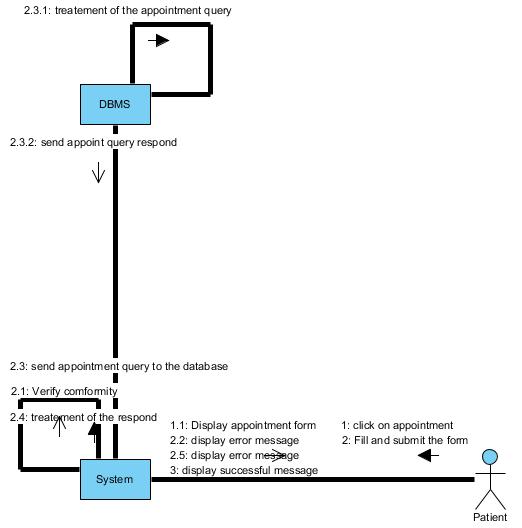
|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Object |  | An actor represents an individual participant in the interaction conversation. |
| Link |  | It initiates an association it connects two objects together for them to communicate. |
| Actor |  | A role play by an entity that interacts with the subjects. |
| Message |  | Defines a particular communication between lifelines in an interaction. |

Communication Diagram for Authentication



*Figure 12* *: Communication Diagram of authentication*

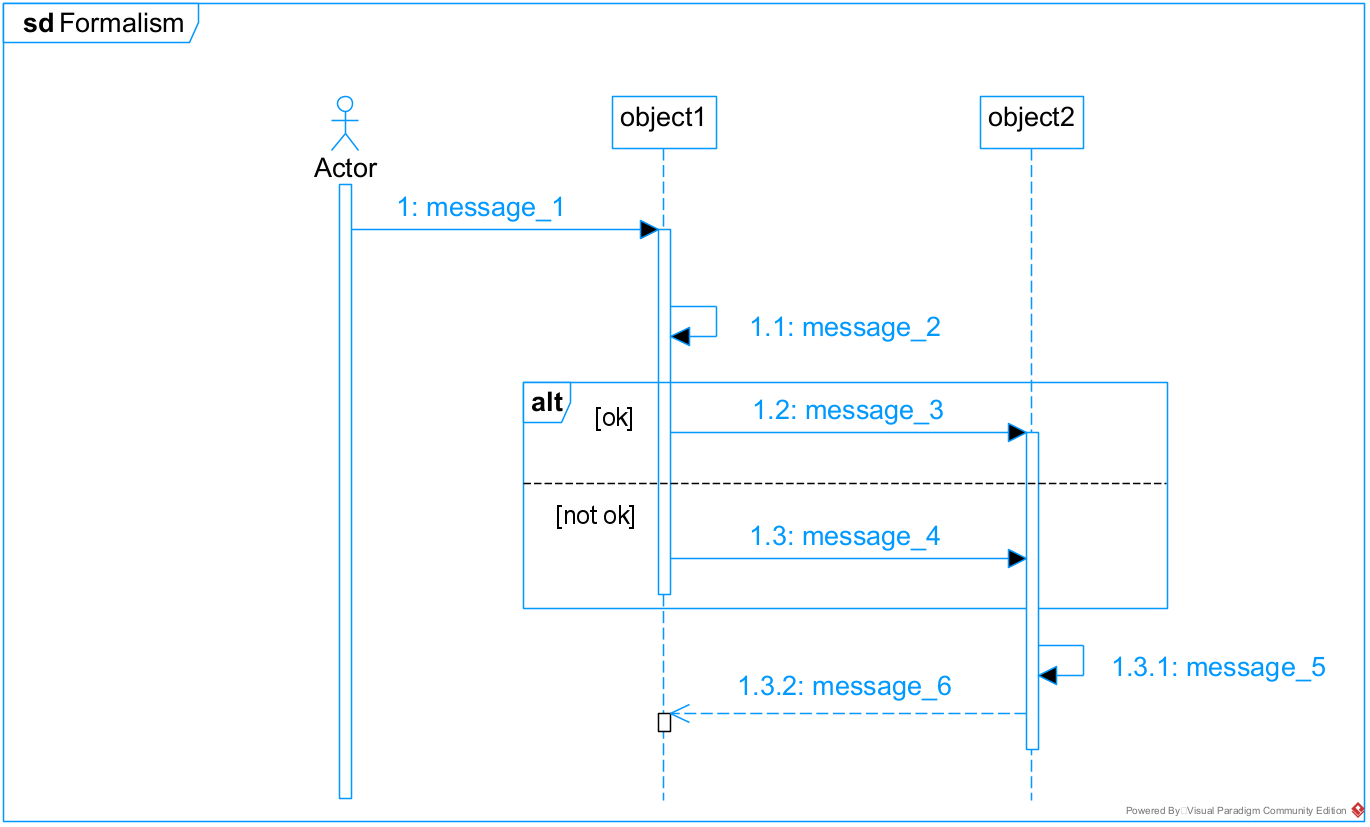
Communication Diagram for book an appointment



*Figure 13* *: Communication Diagram of book an appointment*

1. Sequence Diagram
2. Definition:

A sequence diagram is an interaction diagram which represents the flow of message between elements in a system, it is termed as an event diagram. It portrays the communication between any two lifelines as a time-ordered sequence of events.

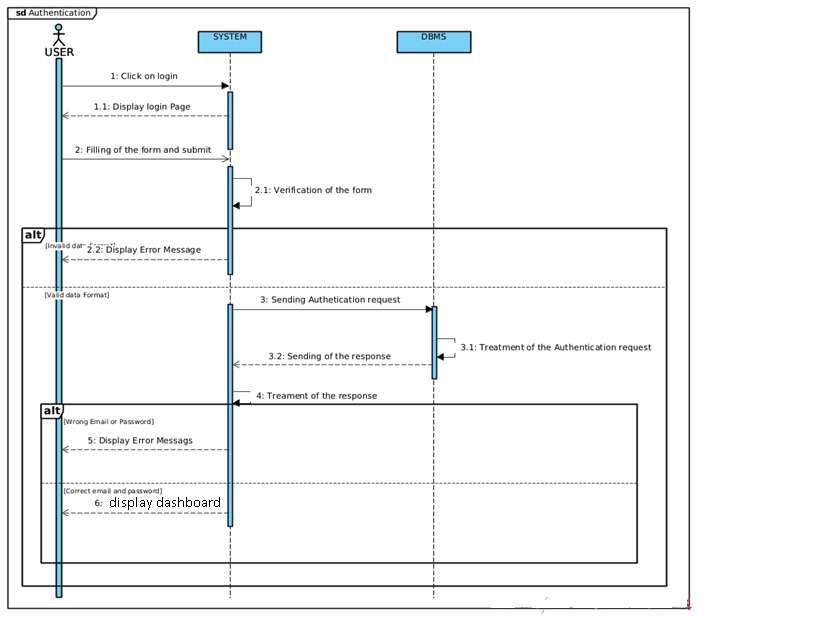
1. Formalism: 

*Figure 14**: Sequence Diagram formalism*

*Table 16:Sequence Diagram Components*

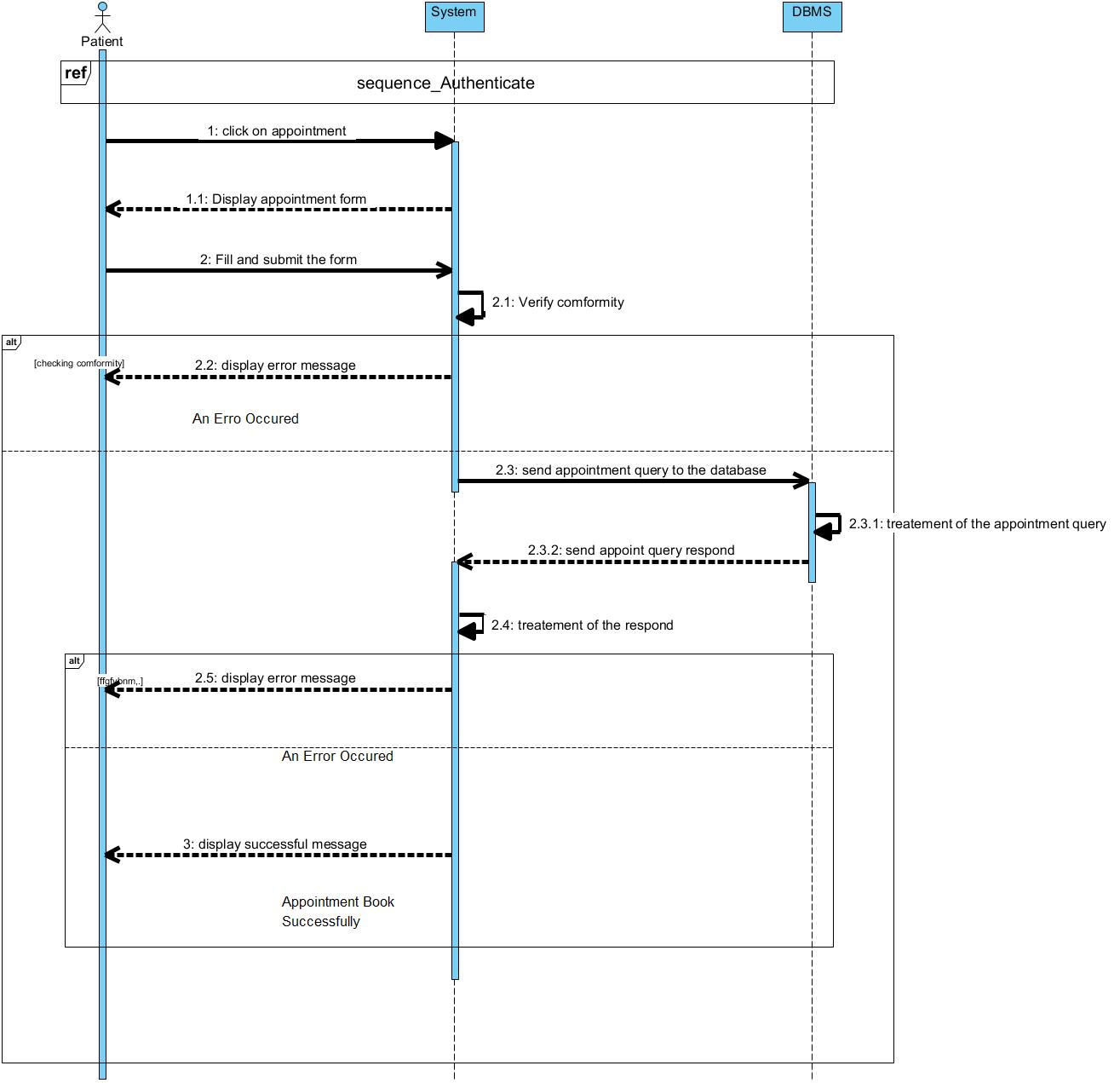
|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Lifeline |  | An individual participant in a sequence diagram, it is position at the top of the diagram. |
| Combined fragment |  | It represents a choice of behavior in which at most one operand will be chosen. |
| Messages |  | These are arrows which shows the direction of message flow. We have the synchronous, the asynchronous and the self messages. |
| Activation |  | It describes the time period in which an operation is performed by an element. |

Authentication Sequence Diagram



*Figure 15**: Sequence Diagram of authentication*

Book appointment Sequence Diagram

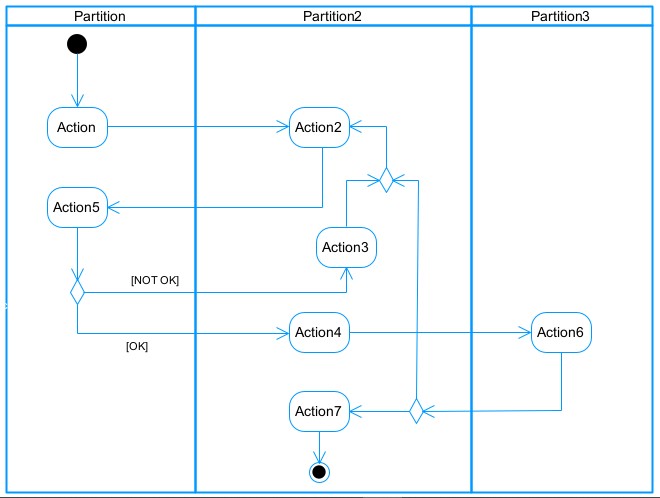


*Figure 16* *: Sequence Diagram of book appointment*

1. Activity Diagram
2. Definition

An activity diagram is a graphical representation of work-flows that show the steps needed in the realization of a process; showing the details from a start point to an end point through all decisions and actions that can possible be performed. Activity diagrams are intended to model both the computational and organizational process. They flow can be sequential, branched or concurrent. Below is an activity diagram formalism.

1. **Formalism**



*Figure 17* *: Activity Diagram formalism*

1. Activity Diagram Elements:

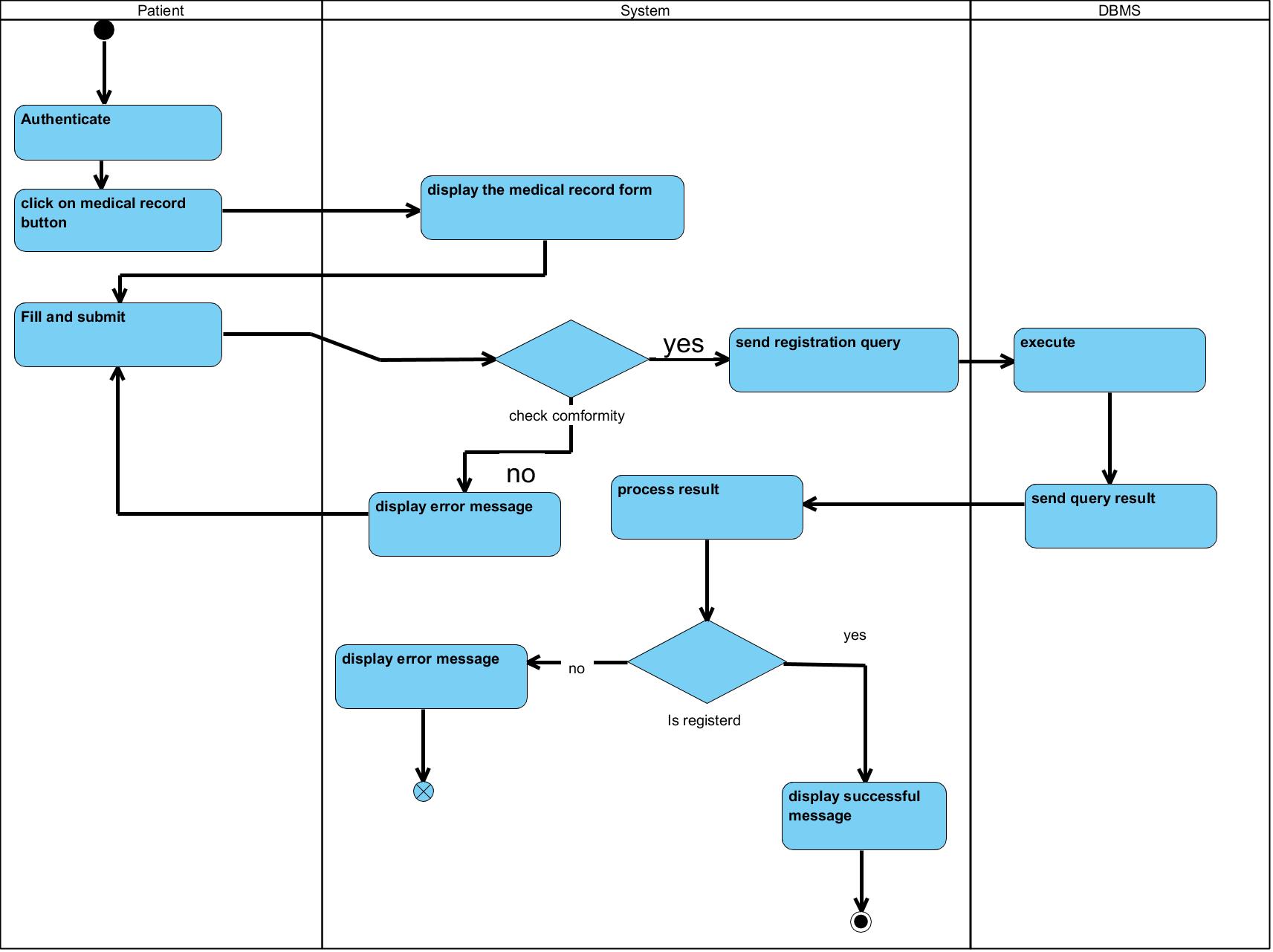
*Table 17**:components of an Activity Diagram*

|  |  |  |
| --- | --- | --- |
| Elements Diagrammatic Representation | | Description |
| Activity |  | Use to represent a set of actions. |
| Action |  | Represent a task  to be performed. |
| Activity edge |  | A directed connection between two activity nodes through which tokens may flow. |
| Initial node |  | Shows the beginning of an activity or set of actions. |
| Final node |  | Stops all controls and object flows in an activity. |
| Object node |  | Represents an object connected to a series of object flows. |
| Decision node |  | Represents a test condition that slits an incoming activity edge into opposite outgoing activity edges. |
| Merge node |  | Reunite different decision paths created using a decision node. |
| Fork node |  | Slits behavior into parallel or concurrent flows of activities  (or actions) |
| Join node |  | Unites a set of parallel or concurrent flows of activities or actions. | |
| Swimlane and partition |  | A way of grouping activities performed by the same actor in an activity diagram or to group actions in the same thread. | |

1. Activity Diagram for authentication

*Figure 18: Activity Diagram for authentication*

1. Activity Diagram to create medical record



*Figure 19**: Activity Diagram of create medical record*

CONCLUSION

In the analysis phase, we chose a software development process and modelling language, after which we presented the existing system, its limitation and our proposed solutions. We went forth explaining the functional need of our system, we saw the use case diagram which shows the relationship between the actors and use case (the action the actor can perform on the system), we saw the activity diagram which shows the work-flow of our system, and lastly the State machine diagram which shows the behavior of a single object in response to an event. We will now move to the conception phase in which we will present the other two branches of the 2TUP which are the Technical and the realization branch of our system.

**PART V**

**CONCEPTION PHASE**

Preamble

The conception phase will permit us to present in an orderly manner the components necessary for the good functioning of our software and also the architecture used for the proposed solution. It bridges the gap between the analysis phase and the realization phase

Content

INTRODUCTION

1. TECHNICAL BRANCH
2. GENERIC DESIGN
3. CAPTURES OF TECHNICAL NEEDS
4. RELATED UML DIAGRAMS
5. Class Diagram
6. State Transition Diagram
7. Package Diagram

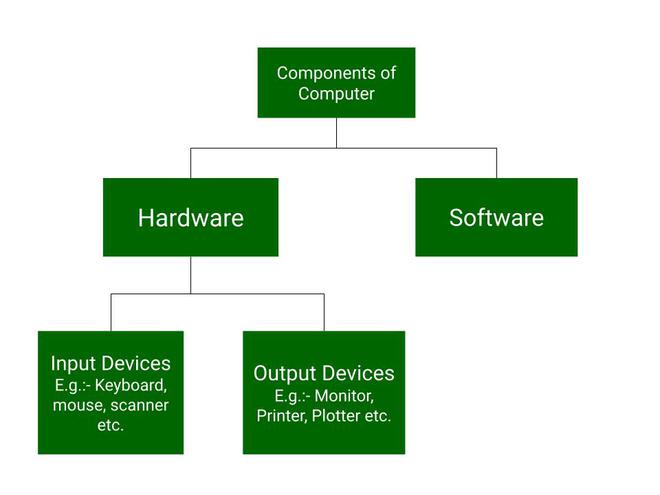
CONCLUSION

INTRODUCTION

The conceptual phase will describe in details the necessary specifications, features and operations that will satisfy the functioning requirements of the proposed system as modeled in the analysis phase. This phase is meant to identify and consider essential components (hardware /or software), structure (network capabilities), processes and procedures for the system to accomplish it objectives. We will look at some diagrams such as the component diagram, package diagram, communication diagram, class diagram and the object diagram.

1. TECHNICAL BRANCH
2. GENERIC DESIGN
3. Hardware diagram of the system

The hardware diagram simply shows how the system components of our system are deployed; it shows the positioning of each device into it right proportion.



*Figure 20**:Hardware Architectural Diagram of the system*

1. High Level Architectural Diagram of the Software

The high-level architecture diagram provides an overview of the entire system, identifying the main components that would be developed for the product and their interfaces.

1. CAPTURE TECHNICAL NEEDS
   1. Physical architecture

The design of the DBMS depends on its architecture. An n-tier architecture partitions on the whole system into related but separated n modules, which can be independently modified, altered, changed or replace. A large amount of data on web servers, personal computers and others are link with networks with the help of basic client or server architecture.

Within the scope of our project, we made use of the multi-tier architecture This architecture separates it tiers from each other based upon the user and the manipulated data in the database. It is important to note that with the multi-tier architecture, only neighboring layers can communicate. Each layer has a well-defined communication interface and the evolution of the layer is independent of the other. The multi-tier of our system is made up of:

* The hardware tier, which represents
* The client tier, which is also known as our presentation inter-phase.
* Application Tier, which represents our web-server.

❖ The data tier, which represents our DBMS server

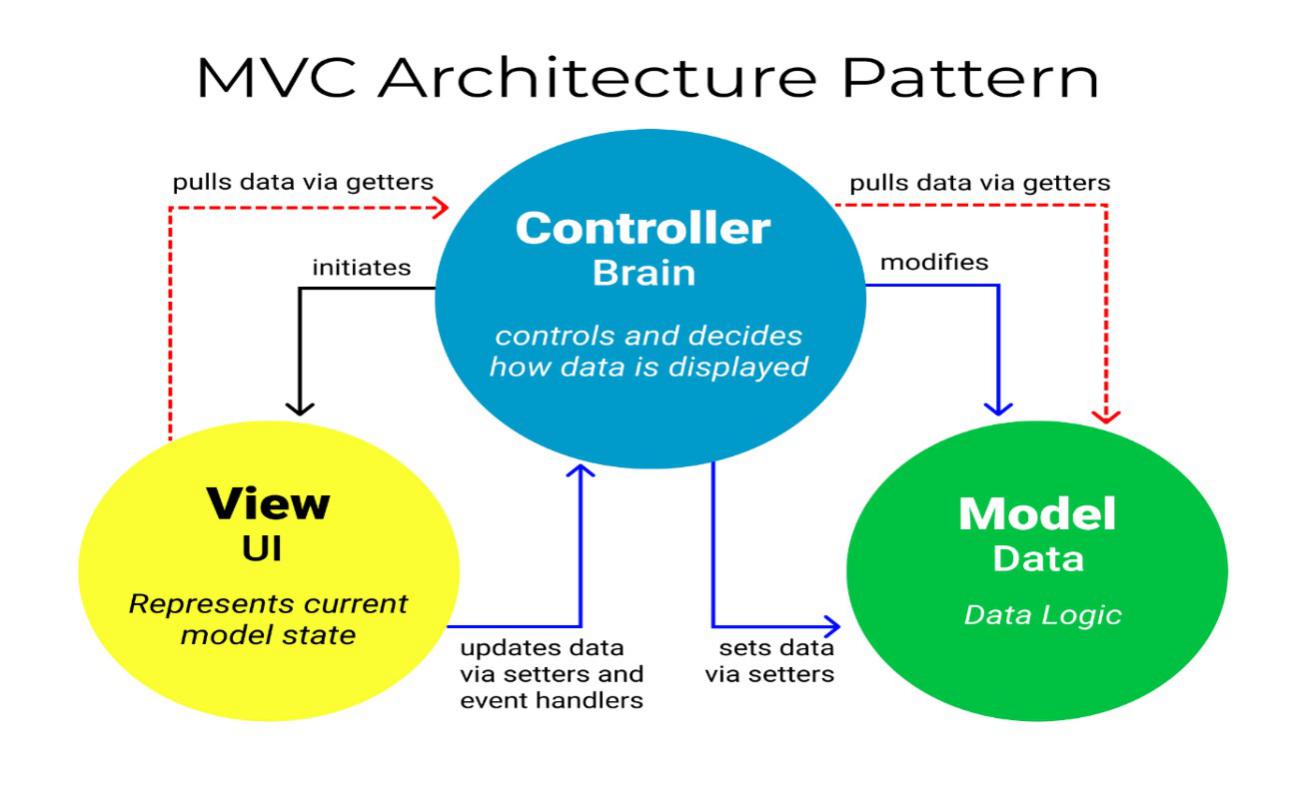
Présentation de l’architecture logique du système

1. Logical architecture:

Model View controller or MVC as it is popularly called, is a software design patten for developing application. A model view controller pattern is made up of the following three parts.

* Model: The lowest level of the patten which is responsible for maintaining data.
* View: This is responsible for displaying all or a portion of data to the user.
* Controller: It handles software codes that controls the interactions between the model and the view.

MVC is popular as it isolates the application logic from the user interface and supports separation of concerns. Here the controller receives all requests for the application then works with the model to prepare data needed by the view. The view then uses the data prepared by the controller to produce a final response. The MVC can be represented as follows.



*Figure21**:The MVC architecture*

*Source:https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvcarchitecture-and-frameworks-explained/*

1. **RELATED UML DIAGRAMS**
2. **Class Diagrams**
3. Definition:

A class diagram is a static diagram. It represents the static view of an application. class diagram is not only used for visualizing, describing and documenting different aspect of the system but also for constructing executable code of the software application. Class diagram describes the attribute and operation of a class and also constraints imposed on the system. Its purpose is to model the static view of an application.

1. Formalism:

multiplicity

private (accessible

only within the class

scope)

public

(

accessible out

of the class

scope)

0..1

0..\*



class name

-

Attribute

:

int



Classe\_2

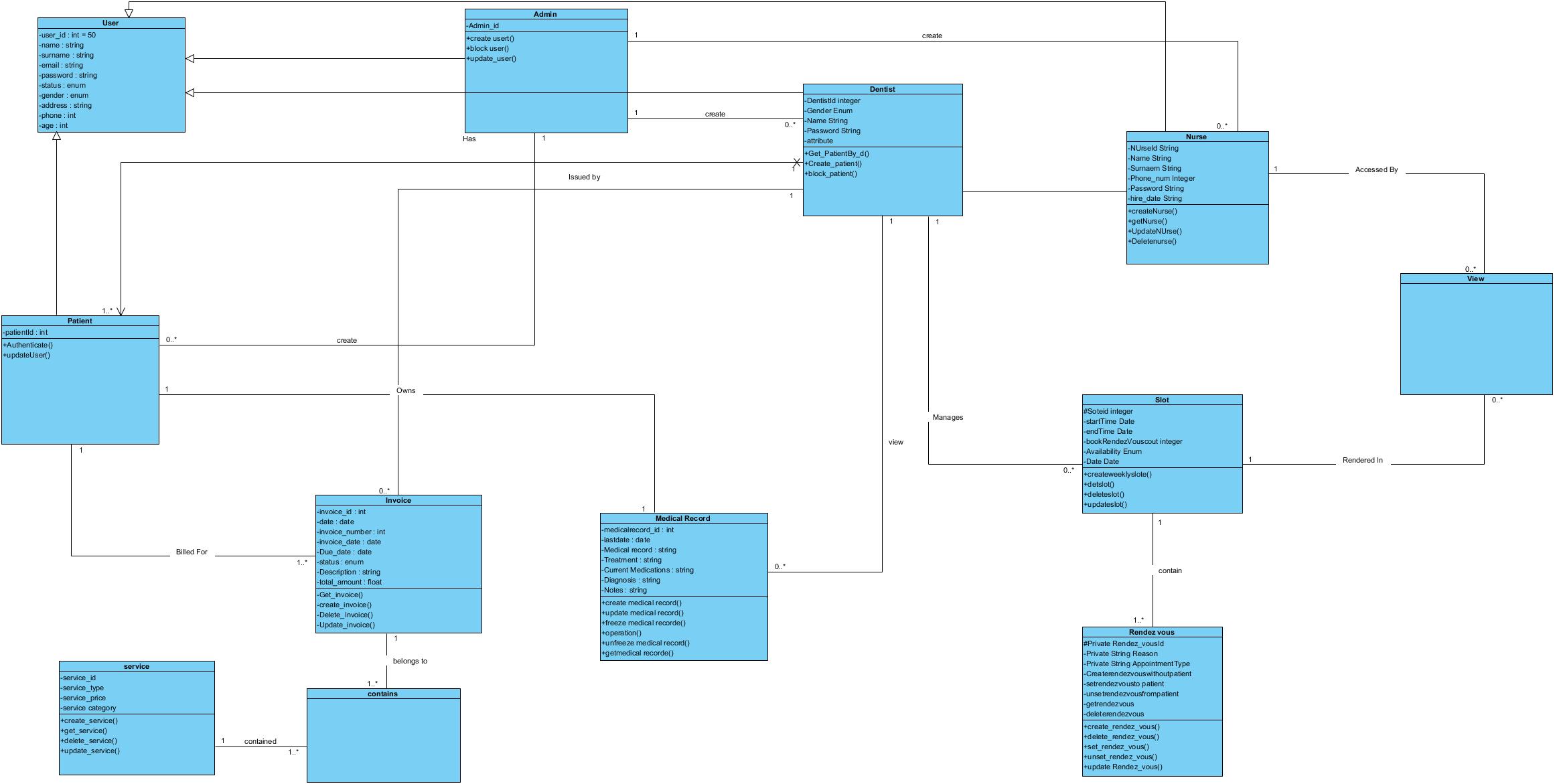
*Figure 22* *:Class Diagram Formalism*

Class Diagram Elements Description

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| class |  | A class is an element that defines the attributes and behaviors that an object is able to generate |
| Composition |  | If a parent of a composite is deleted, usually, all of its parts are deleted with it. |
| Aggregation |  | If the parent of the aggregate is deleted, usually the children are not deleted. |
| Dependency |  | It existed between two classes, if one changes it may cause the change in the order, but the other way around. |
| Generalization |  | it a relationship between a whole thing (called superclass) and a more specific thing (called subclass) |
| Association |  | It is a general type of relationship between elements, it may include cardinality, roles etc. |

*Table 18**:Class Diagram Components*

1. Class Diagram:



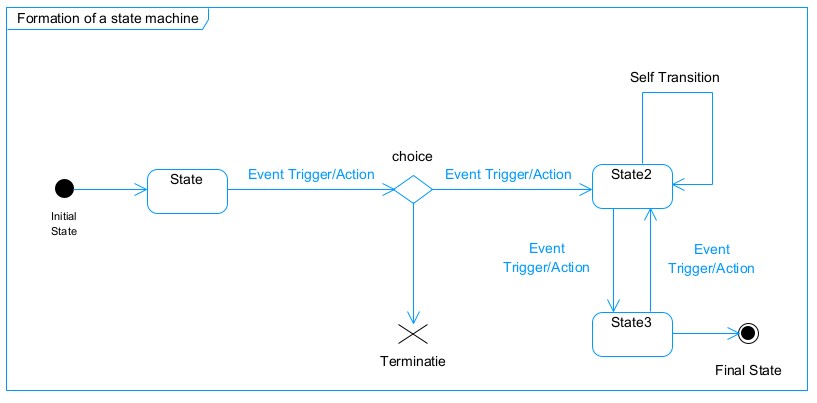
*Figure 23**:System class Diagram*

**State Machine Diagram**

1. Definition

A state machine diagram describes the behavior of a single object in response to a series of events in a system. Also known as the state machine diagram, it models the dynamic flow of control from the state of a particular object within a system.

1. Formalism:

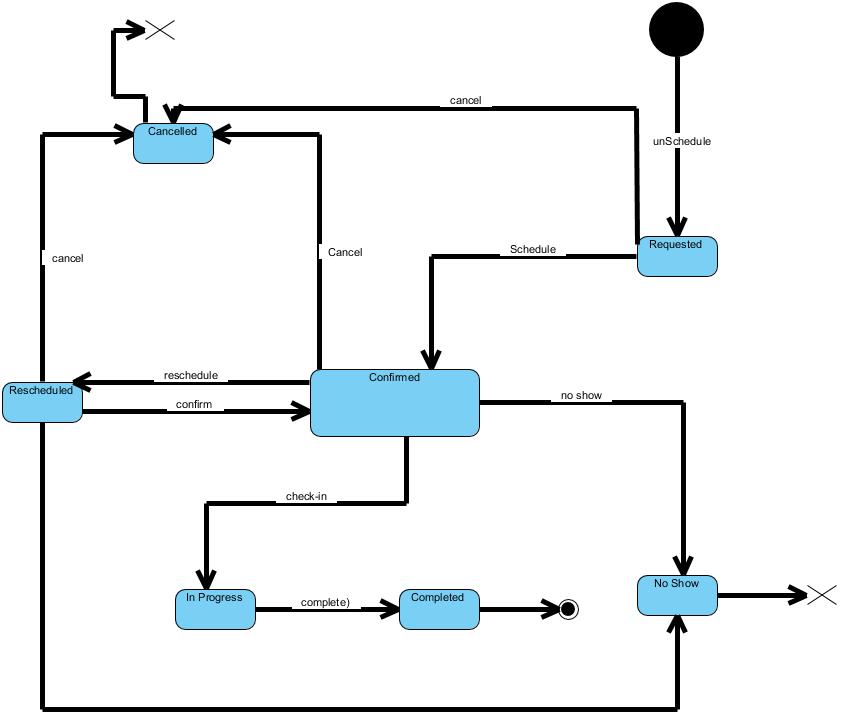


*Figure 24**:Formalism of a state machine diagram*

State Machine Diagram Elements

*Table 19**:State Machine Diagram Components*

|  |  |  |
| --- | --- | --- |
|  | Diagram Relationship | Description |
| State |  | Models a situation during which a certain invariant condition holds. |
| First (Initial) State |  | It represents a default vertex, that is, a source for a single transaction to the default or composite state. |
| Final State |  | A state specifying that the enclosing region is complete. |
| Transition |  | A direction relation between a source and a target vertex. |
| Choice pseudo State |  | A diamond symbol that indicates a dynamic condition with branched potential results |
| Terminate |  | Implies that the execution of a state by means of it context is terminated. |
| Diagram Overview |  | A placeholder for the linked states in a state machine diagram. |

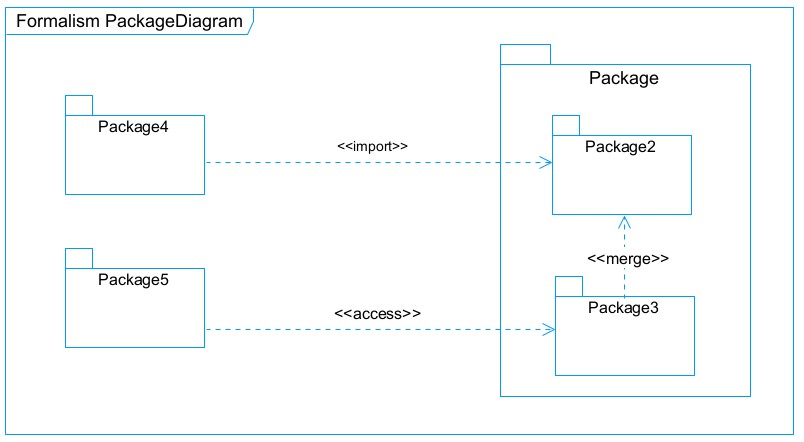


*Figure 25:System State Machine Diagram for book appointment*

1. **Package Diagram:**
   * + 1. Definition:

This is a structural diagram used to show the organization and arrangement of various model elements in the form of packages. A package diagram is the grouping of related uml elements such as classes, diagrams or eve other packages.

* + - 1. Formalism:

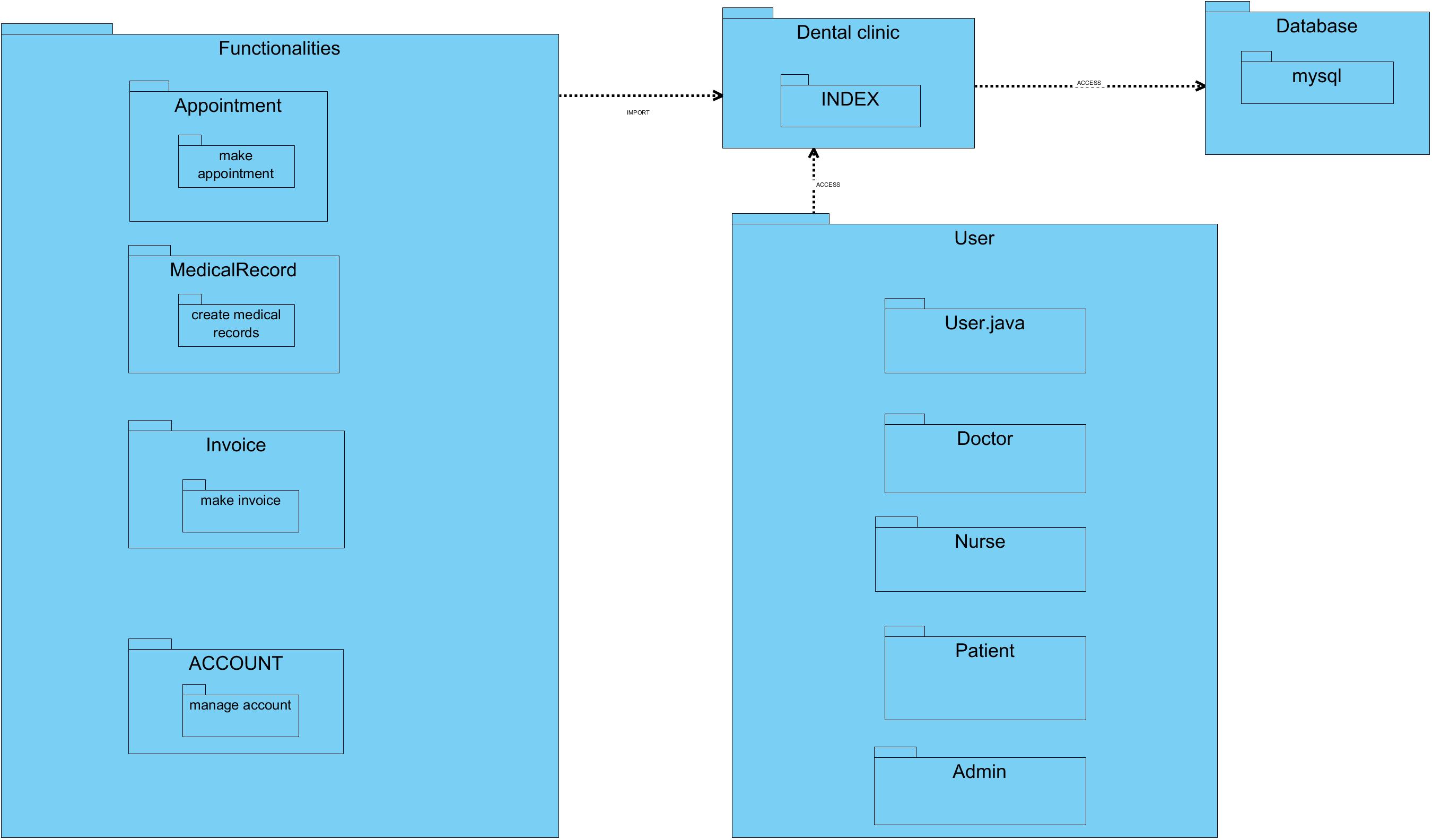


*Figure 26**:Package Diagram Formalism*

*Table 20**: Package Diagram Component*

|  |  |  |
| --- | --- | --- |
| NAME | Representation | Description |
| Package |  | A package is a names-pace use to group related elements; it is a mechanism used to group elements into a better structure in a system. |
| Package import | tup | A relationship Indicate that, functionality has been imported from one package to another. |
| Package access |  | A relationship Indicates that one package requires assistance from the function of another package. |
| Package merge |  | It is a relationship which shows that, the functionality of two packages are combines to a single function. |

System package Diagram :



*Figure 27**:System Package Diagram*

CONCLUSION

In the conception phase, we set as objective to plane the different aspect of our system by showing how it will be structure and deployed within existing technical architectures. We began by taking into account the technical constraints for our system, after which we proceeded to identify the components of our system, how they are grouped together and how they should be deployed on appropriate deployment targets. We finished this phase by looking at interactions between the various aspects and actors of our system. The next phase of our report is the realization phase where we will look at aspects concerning the implementation of our system.

# **REALIZATION PHASE**

Preamble

In this phase we will to straight forward in the implementation of our solution, we will base ourselves on the analysis and conception phases content

Content

INTRODUCTION

1. ARCHITECTURE OF THE APPLICATION
2. Logical architecture
3. Physical architecture
4. TECHNOLOGICAL CHOICE
5. Choice of programming language and framework
6. Choice of tools used

CONCLUSION

INTRODUCTION

Here in the realization phase, we will concentrate on building or implementing our solution, based on the different analysis and conception that we had carried out, which will help to facilitate our work, this phase is as critical as the previous phases. We are going to look at the relationship that exist between the entities of the entity relational diagram. Furthermore, we will see the choices of technologies used for the implementation of our system.

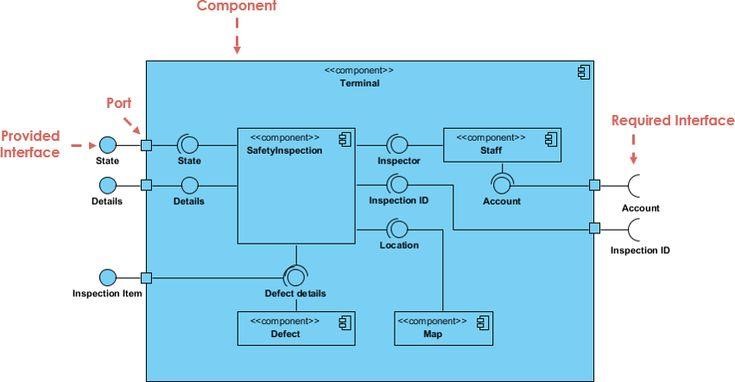
**IMPLEMENTATION BRANCH**

Here we will see the preliminary conception, detailed conception and documentation of

1. **Component Diagram:**
2. Definition:

Component diagrams are used to model the physical aspect of a system. Now the question is what are this physical aspect? They are elements such as Executables, libraries, files, document etc. which resides in a node. The component diagram does notupt describe the functionality of the system but it describes the components used to make those functionalities.

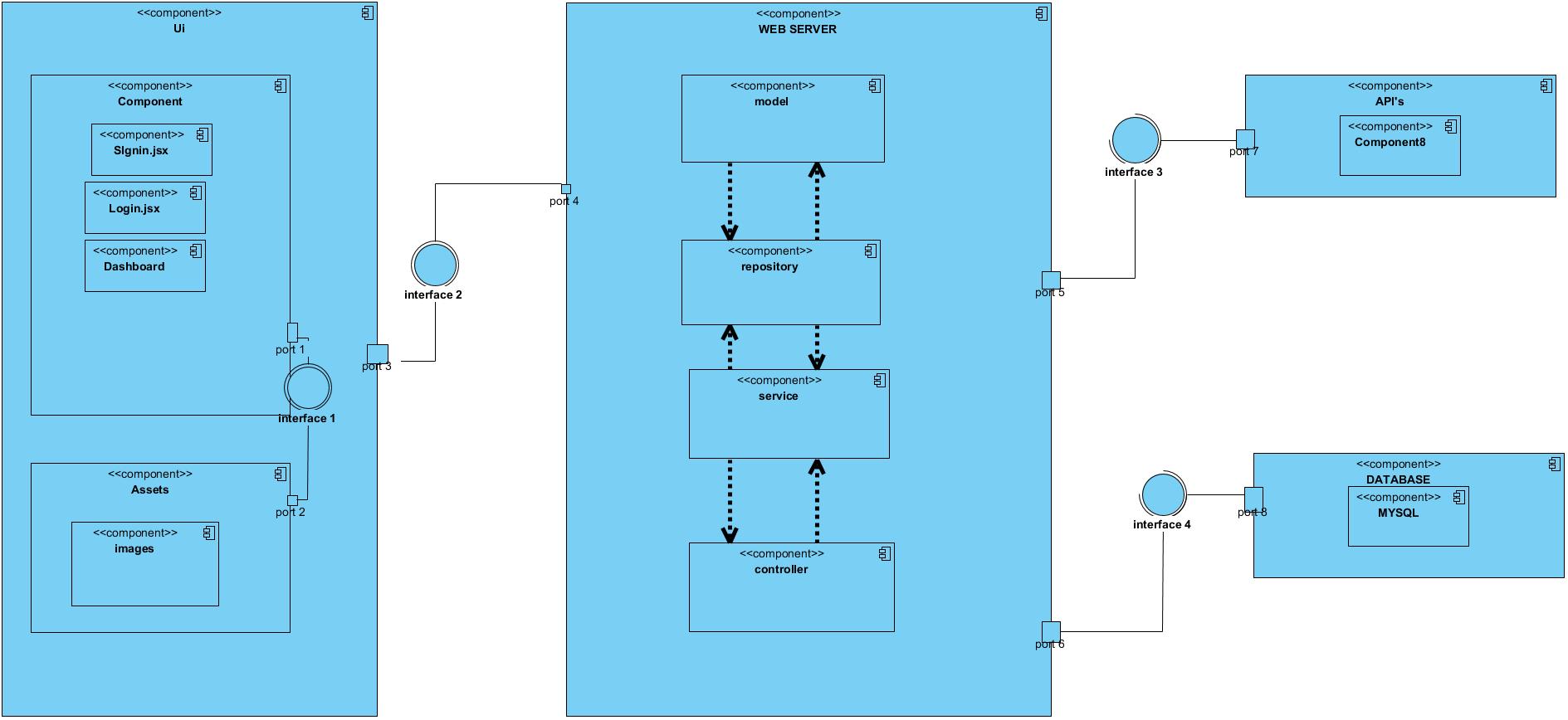
1. Formalism:



*Figure 28* *Component Diagram Formalism*

*Table 21: Component Diagram Components*

|  |  |  |
| --- | --- | --- |
| NAME | REPRESENTATION | DESCRIPTION |
| A component |  | A component is an abstract logical unit block of a system.it is represented as a rectangle with smaller rectangle in the upper right corner which saves as it icon for recognition. |
| Dependency |  | Dependency is a directed relationship which is used to show that some components are dependent on others for their correct functioning. |
| interface |  | An interface is a circle or a semicircle attached to a stick which looks like a lollipop. It describes groups of operations provided or required by components. |
| port |  | A port (represented by a small square at the end of a required or provided interface) is used when the components delegate the interfaces to an internal class. |

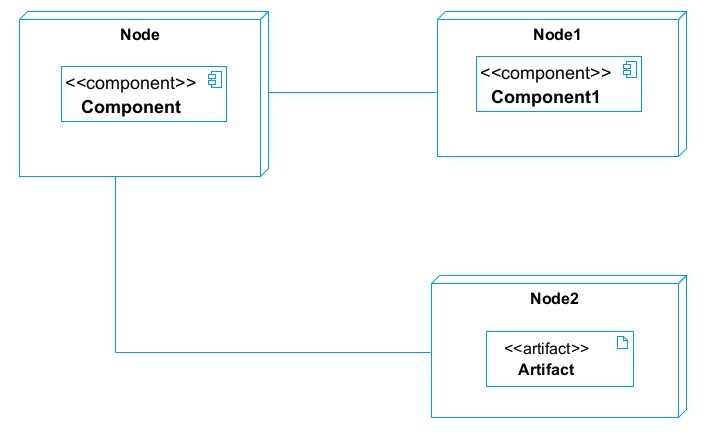


*Figure 1 : System Component Diagram*

1. **Deployment diagram:**
   * 1. Definition:

Deploymentdiagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application Deployment diagram is a structural diagram used to visualize the topology of the physical components of a system, where the software is deployed. They consist of nodes and their relationship. It is related to the component diagram because the components are deployedtup using the deployment diagram. A deployment diagram:

* + 1. Formalism

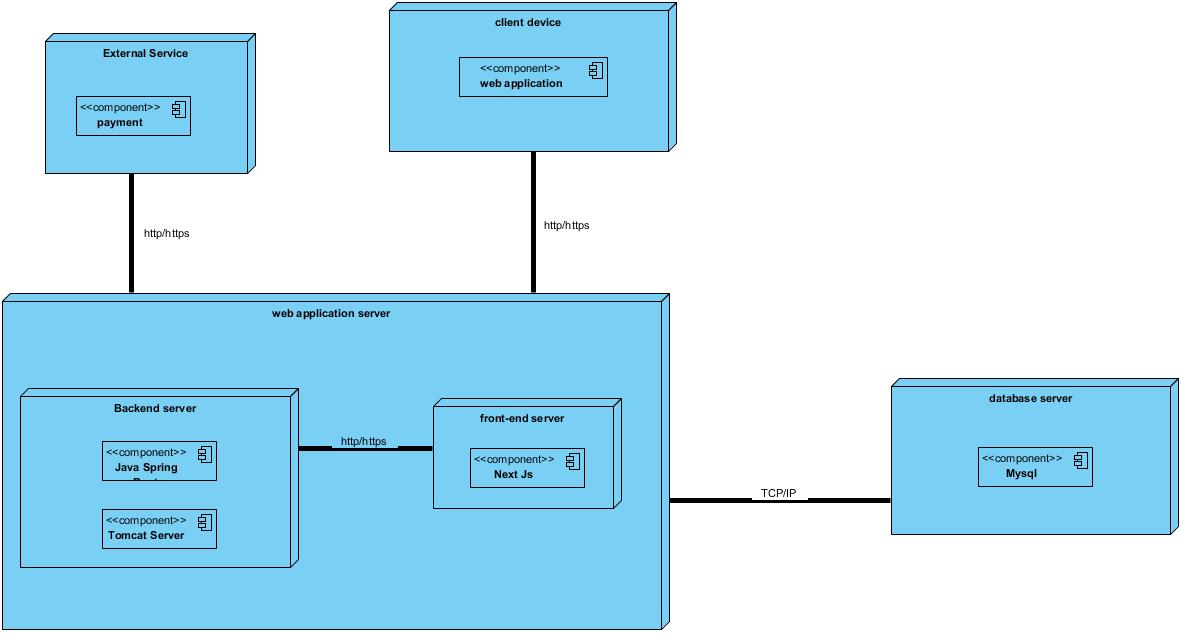


*Figure 29**:Deployment Diagram Formalism*

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Node |  | It is a hardware used to deploy the application |
| Artifact |  | An artifact is a major product, which is produced or used during the development of a software. E.g diagrams, data models, setup scripts |
| Component |  | It represents a modular part of a system that encapsulates its content and whose manifestation is replaceable within it environment. |
| Association |  | An association helps to connect two nodes together which permits them to communicate together |

*Table 22**:Component Diagram Elements*

* 1. System Deployment Diagram



*Figure 30**:System Deployment Diagram*

### **ARCHITECTURE OF THE APPLICATION**

* + - * 1. **Physical Architecture**

The physical structure layer described the system’s hardware, software and network environment. The physical architecture is a based on the non-functional requirements such as availability, performance and security. The physical architecture of our system follows the 3-tier client server architecture. Three-tier architecture of our system pattern in which the user interface(Presentation), functional process logic (“business rules”), computer data storage and access are developed maintained as independent modules, most often on separate platform. 3-tier architecture has the following layers:

**The user (Presentation) tier:**

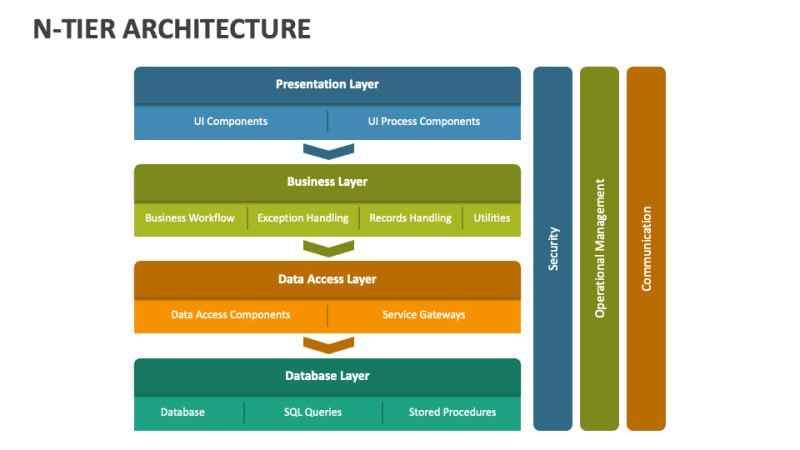
This is the tier which users can access directly. That is, this tier provides interfaces for the users and it is also responsible for transmitting users request to other tiers of the application and displaying the results of their actions back to them. It may be web page or an operating system’s GUI(That is our case).

**The application (logic or middle) tier:**

Here reside the application server and the programs in charge the data storage tier and providing result back to the presentation tier. It controls application functionality by performing detailed processing and reinforcing business rules.

**The database (data) tier:**

At this layer, we found the database which holds all or almost all the applications data along with its query processing programs.

*Figure33: n-tier architecture diagram*

* + - * 1. **Logical Architecture**

The application logic of our project is based on MVC design pattern which is a very practical way to organize its different classes in an application development project facilitating error detection, collaboration and code reusability.

The MVC(Model View Controller) design pattern is an architecture and a design method that organizes the human-machine interface(HMI) of a software application. This paradigm divided the HMI in to a model (data model) a view (user interface) and a controller (control logic, event management, synchronization),each having a specific role in the interface.

The MVC design patten allows you to organize your source code well.it will help you to know which files to create, but especially to define their role. The goal of the MVC is to separate the logic of the code in to three parts that are found in separate files

* **Model:** This part manages the data of your application. Its role is to retrieve ‘raw’ information from a local or remote database, organize it and assemble it so that the controller can process it. This data can be stored in databases, in the cloud in binary files
* **View:** this part focuses on the display. It does almost no computation and just retrieves variables to know what to display. In our application we find mainly flutter code but also some very simple loops and dart condition.
* **Controller:** this part manages the logic of the code the make decision. It is the analyze it, make decisions and send the text to be displayed to the view . The controller contains exclusively dart code in particular, it determines whether the user has the right to view the page or not (access right management). Representation of an MVC architecture:

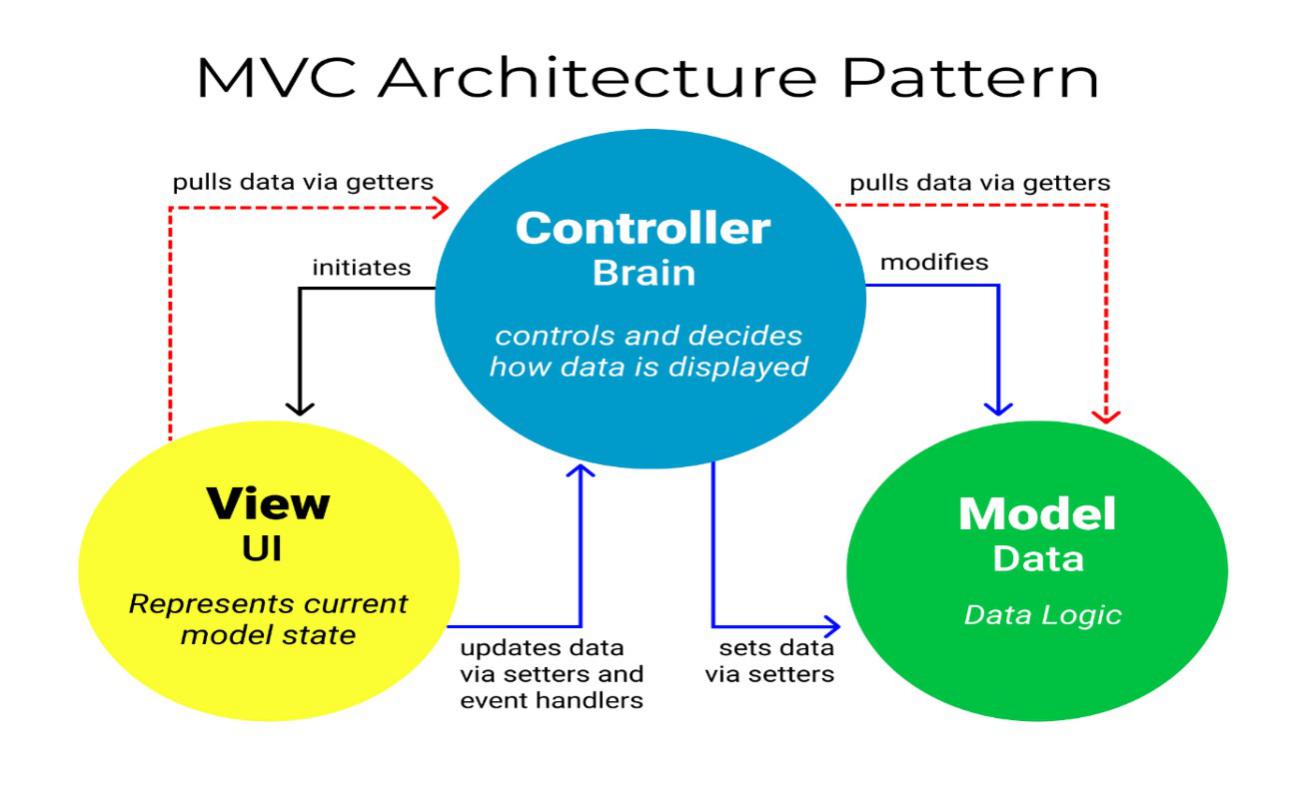


Figure31:The MVC architecture Source:https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvcarchitecture-and-frameworks-explained/

Software used for the development of the application

*Table 23**: Softwares used for the development of the system*

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Version | Role | Image |
| OS WINDOWS  10 | 18 362 | The operating system required to run the software is the one on which we have worked. |  |
| Visuel studio code | 1.70.0.8 | Éditeur de texte utilisé pour saisir les lignes de code qui seront interprétées par le navigateur. |  |
| Visual Paradigm |  | Visual Paradigm |  |
| Postman | 9.4 | It is a platform that simplifies every step of the API lifecycle and streamlines collaboration, making it easier and faster to create better APIs. |  |
| chrome | 129.0.6 | It is a navigator where the output of the code is being tested. |  |
| GitHub | 3.0.6 | It is a website and cloud service that helps developers store and manage their code. |  |
| Laragon | 8.0.30 | It is the use the database management |  |

Technologies used for the development of the system

|  |  |  |
| --- | --- | --- |
| Image | Name | Spring boot |
|  | Java sprint boot | Backend development |
|  | Framework Next.js | Front-end development |

*Table 24**:Technologies used for the development of the system*

CONCLUSION

In our realization phase, we set as objective to plane the different aspect of our system by showing how it will be structure and deployed within existing technical architectures. We began by taking into account the technical constraints for our system, after which we proceeded to identify the components of our system, how they are grouped together and how they should be deployed on appropriate deployment targets. We finished this phase by looking at interactions between the various aspects and actors of our system. The next phase of our report is the realization phase where we will look at aspects concerning the implementation of our system.

# CHAPTER VI: FONCTIONALITY TEST

Preamble

Once a solution is analysed, conceived and developed, it is now necessary to produce a functionality test. This part of our document concerns the functionalities of the implemented application.

Context

**INTRODUCTION**

1. **LOGIN**
2. **BOOK APPOINTMENT**
3. **ADD PATIENT**
4. **CREATE MEDICAL RECORDS**
5. **CREATE INVOICE**

**CONCLUSION**

INTRODUCTION

Functionality testing is a crucial aspect of software development and quality assurance process. It involves testing the individual functions or features of a software application to that the user meets his or her specific requirements. The primary goal of functionality test is to verify that the software functions correctly and delivers the expected result under various scenarios.

APPLICATION FUNCTIONNALITIES.

**LOGIN**  
This functionality allows users (such as doctors and administrative staff) to log in to the platform. Depending on the role assigned to the user, the platform will authenticate and provide access to respective features and permissions set by the administrator in the back end.

II.

**BOOK** **APPOINTMENT**  
The book appointment feature enables users (patients) to schedule appointments with medical professionals. The user selects an available date, time, and healthcare provider, and this information is stored in the database for the healthcare provider to review and confirm.

III.

**ADD PATIENT**  
This functionality permits medical staff (doctors, nurses, or administrative staff) to create new patient profiles. The staff member enters the patient’s personal details, including contact information, medical history, and other necessary data. The information is then stored in the database for future reference and interaction with the patient’s medical care.

IV.

**CREATE MEDICAL RECORDS**  
Medical professionals (doctors, nurses) use this feature to create detailed medical records for patients. These records can include diagnosis, treatment plans, medications, and other relevant medical information. The data is securely saved to the patient’s file, ensuring that the patient’s medical history is well-documented and accessible when needed.

V.

**CREATE INVOICE**  
This functionality enables administrative staff to generate invoices for patients based on services rendered. The invoice includes details of medical treatments, consultations, medications, and other relevant charges. It is saved in the system for billing purposes, and the patient can receive a copy for their reference or payment.

TEST OF FUNCTIONNALITIES

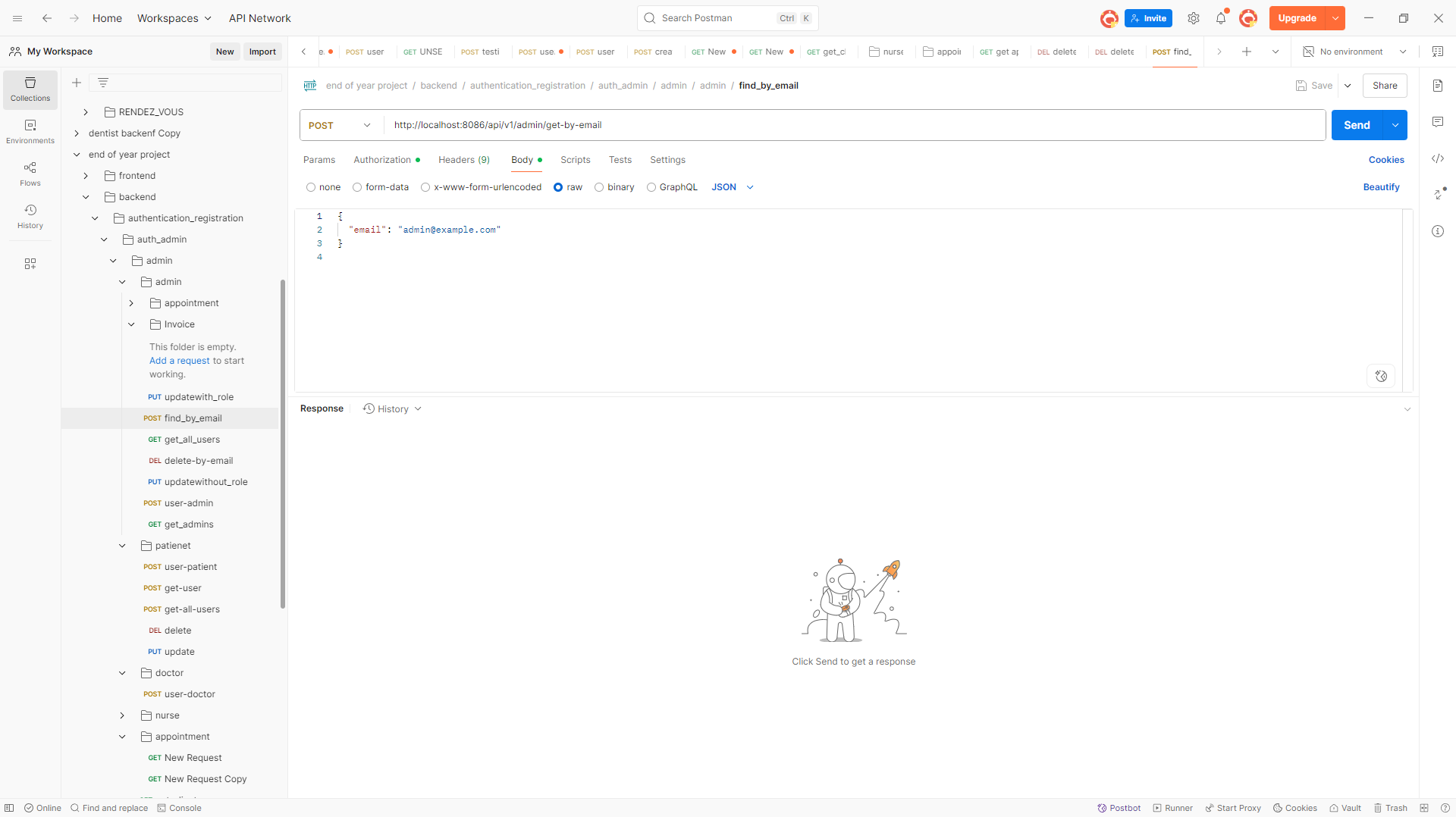


Figure 32 Postman test by email

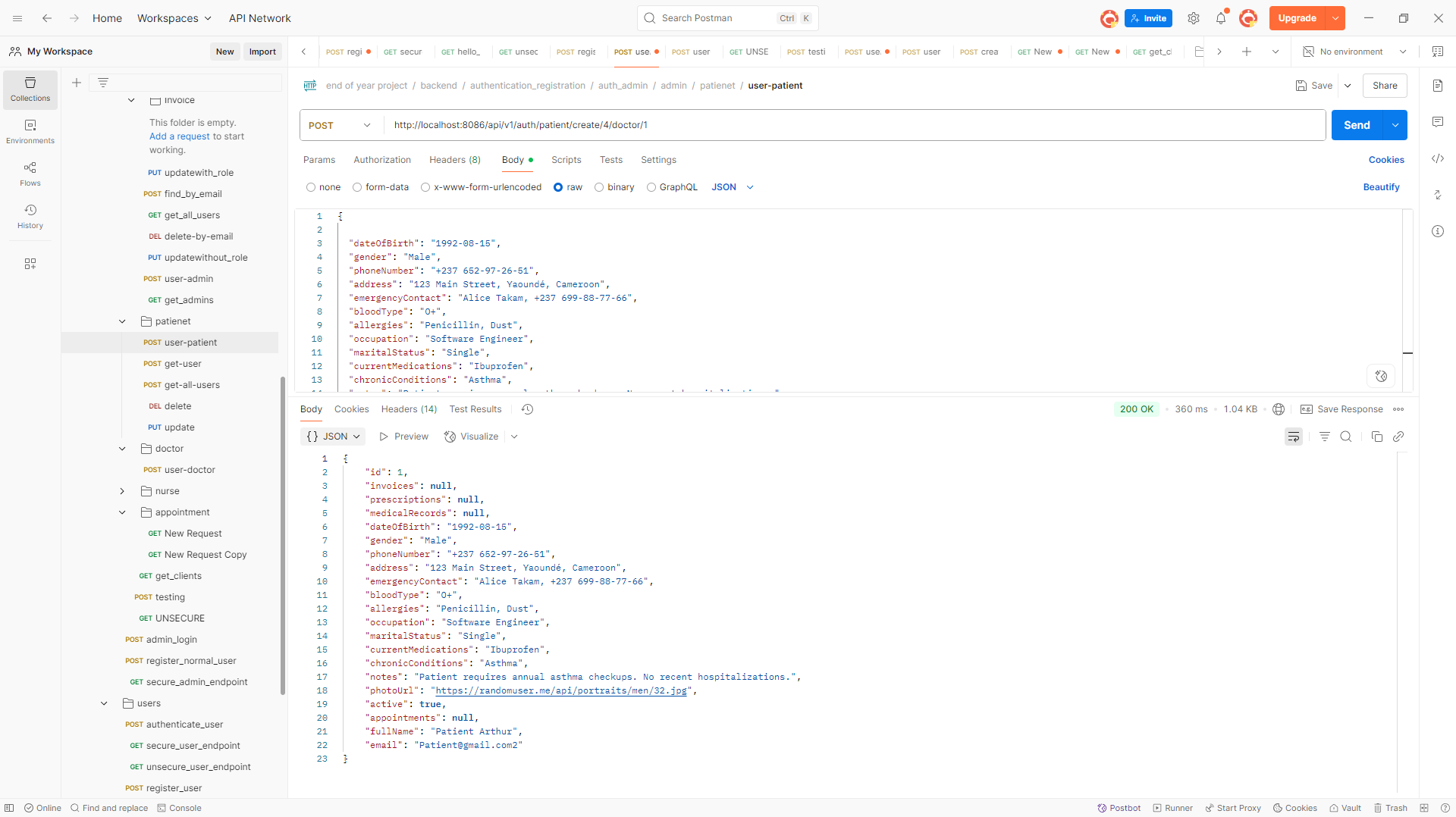


Figure 33 Create patient postman test

CONCLUTION

We have listed the main functions of our application and this is in order to help the user to get used to our application and perform his or her actions at ease and securely.

Next, we will talk of the user guide in order to describe how the user will use our application.

# CHAPTER VII: USER GUIDE

Preamble

The purpose of the user guide is to provide users of our platform with step-by-step instructions on how to install and use the system.

Context

**INTRODUCTION**

1. **INSTALLATION**
2. **USER GUIDE**

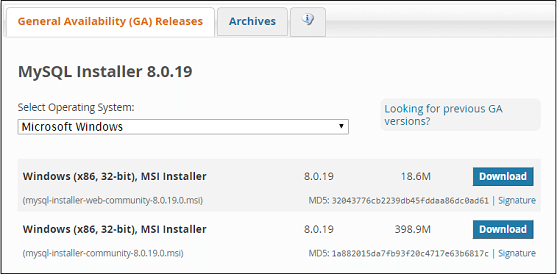
**CONCLUSION**

INTRODUCTION

The user manual is the document created after the implementation of a software platform or application. It answers the questions “how to use the software application or platform that is in front of us?”. It is therefore crucial for us to offer the different users of our plat form a guide allowing them to easily perform various operation on our platform. To achieve our goal, it will be necessary for us to use Microsoft edge web browser and Laragon which will be done on windows 11 operating system.

1. **INSTALLATION**

### Installing MySQL on Windows



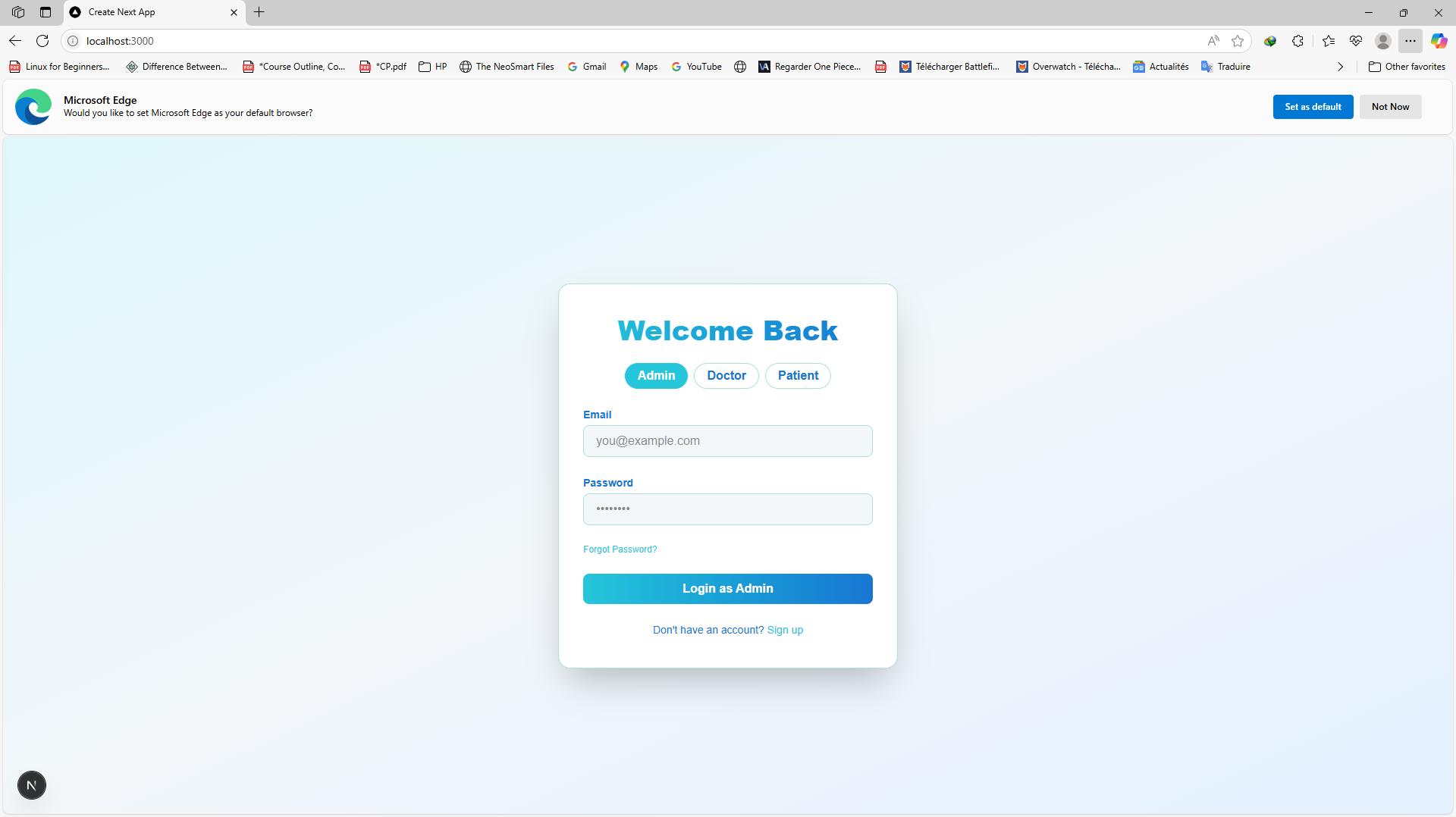
*Fig33:Mysql installation*

1. **USER GUIDE**

Here we have the first interface displayed when the application is opened. Here we have to click on “Get Started” to proceed in the application.

This interface is our login page. If we already have an account, we can simply enter our credentials (email and password) and click on “Login” to submit

*Figure 34: login Page*



Admin Dashboard

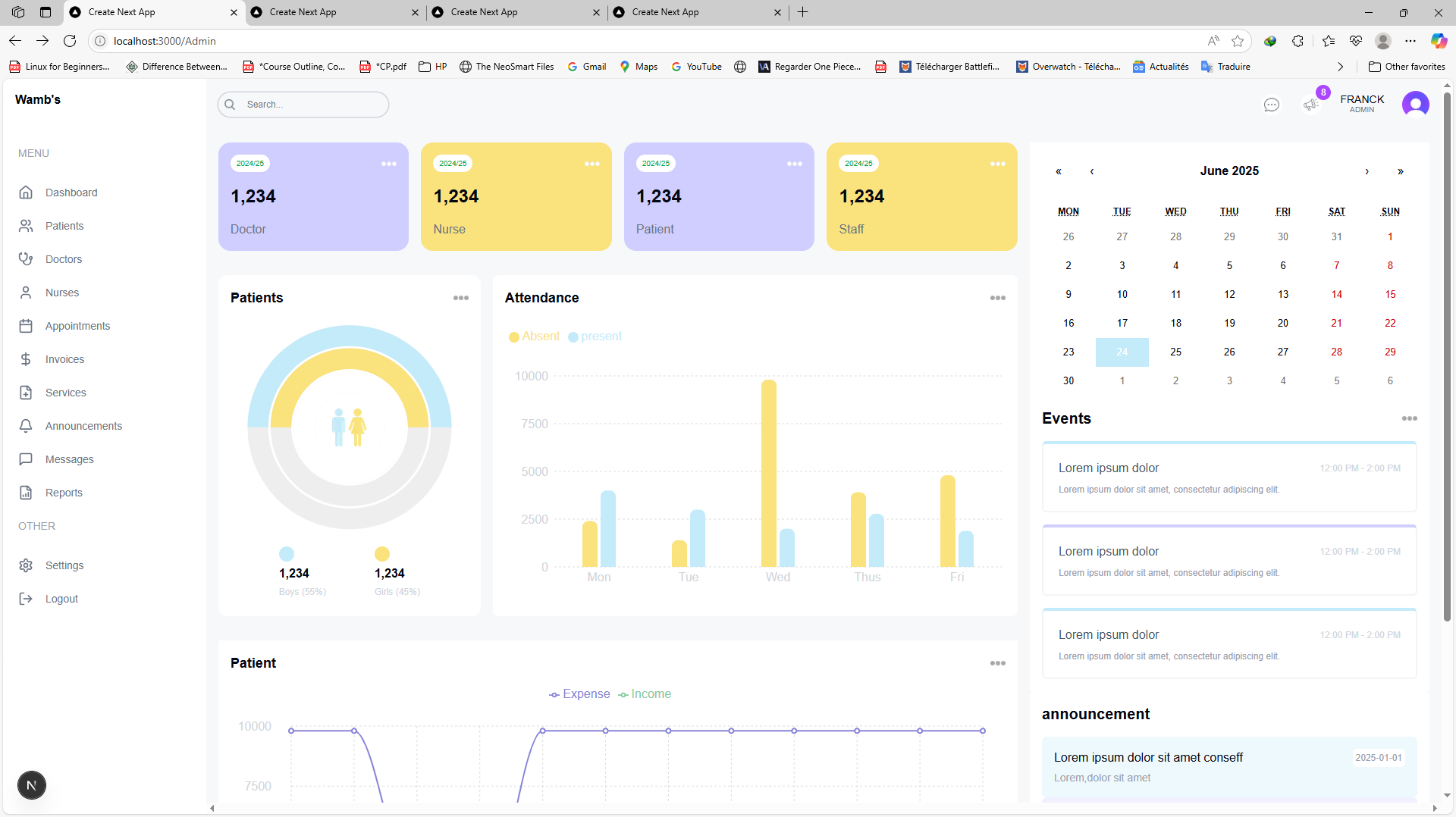


Figure 35: admin dashboard

GENERAL CONCLUSION

The "Design and Implementation of a Patient Management System for a Dental Clinic" project represents a significant step toward modernizing patient management within dental practices. In collaboration with Mickmaq, this initiative underscores the critical role of digital solutions in enhancing operational efficiency, improving patient care, and addressing the challenges posed by traditional management systems. Through the development of a tailored web application, we have successfully automated key processes such as patient registration, appointment scheduling, and treatment history tracking. By integrating real-time communication and secure record management, the system not only streamlines clinic operations but also fosters a more responsive and patient-centric service environment. Our findings highlight the positive impact of digital transformation in healthcare, particularly in dental clinics, where improved workflows can lead to reduced administrative errors, enhanced patient satisfaction, and increased overall productivity. However, we also acknowledge the challenges associated with such transitions, including resistance to change, the necessity for adequate training, and the importance of maintaining data privacy and security. The methodologies employed throughout the project, from UML system design to the use of Java Spring Boot, React.js, and MySQL, have ensured a robust and scalable solution that meets the specific needs of the dental clinic. Each phase of the project from specification and analysis to implementation and user guidance has been meticulously executed to guarantee a seamless transition to this new digital system. In conclusion, this project not only illustrates the potential of technology to revolutionize patient management in dental clinics but also serves as a roadmap for future digital transformation initiatives within the healthcare sector. By embracing such innovations, dental clinics can enhance their operational capabilities and significantly improve the patient experience, ultimately contributing to better health outcomes.

**BIBLIOGRAPHY**

* OMG, OMG Unified Modelling Language 2.5, Object Management Group, March 2015, 794 pages.
* React.js course (from basics to advanced) from W3Uschool 2022.
* JavaScript course from W3Uschool website 2023.

# PART XXI

# ANNEXES

**WEBOGRAPHY**

* https://react.dev/learn: react documentation
* https://www.memoireonline.com/  : learn more on 2TUP
* <https://stackoverflow.com/>  :  interactive website for QA concerning IT.
* <https://openclassroom.com/> UML documentation
* [https://sparxsystems.com/resources/tutorials/uml/part1 .html](https://sparxsystems.com/resources/tutorials/uml/part1%20.html): UML tutorials
* <https://www.slidescarnival.com/tag/tech/page/2?amp> : helped us better organize our presentation

**TABLE OF CONTENT**

Table of Contents

[DEDICATION ii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686050)

[ACKNOWLEDGMENTS iii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686051)

[CONTENT iv](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686052)

[LIST OF FIGURES vi](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686053)

[LIST OF TABLES viii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686054)

[GLOSSARY x](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686055)

[ABSTRACT xi](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686056)

[RESUME xii](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686057)

[GENERAL INTRODUCTION 1](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686058)

[Preamble 4](#_Toc201686059)

[INTRODUCTION 5](#_Toc201686060)

[I. PRESENTATION OF THE PROJECT THEME 6](#_Toc201686061)

[B. Justification 19](#_Toc201686062)

[A. 21](#_Toc201686063)

[B. GENERAL OBJECTIVE 21](#_Toc201686064)

[C. SPECIFIC OBJECTIVES 21](#_Toc201686065)

[III. EXPRESSION OF NEEDS 22](#_Toc201686066)

[2. Non-Functional Needs 23](#_Toc201686067)

[**a.** **Chronogram of activities** 26](#_Toc201686068)

[VII. LIST OF PARTICIPANTS AND DELIVERABLES 31](#_Toc201686069)

[a. LIST OF PARTICIPANTS 31](#_Toc201686070)

[ANALYSIS PHASE 32](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686071)

[PART III 33](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686072)

[ANALYSIS PHASE 33](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686073)

[Preamble 34](#_Toc201686074)

[I. METHODOLOGY 36](#_Toc201686075)

[**REALIZATION PHASE** 83](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686076)

[**I.** **ARCHITECTURE OF THE APPLICATION** 92](#_Toc201686077)

[CHAPTER VI: FONCTIONALITY TEST 98](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686078)

[CHAPTER VII: USER GUIDE 104](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686079)

[Installing MySQL on Windows 107](#_Toc201686080)

[PART XXI 112](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686081)

[ANNEXES 112](file:///H:\Users\takam\OneDrive\Documents\TAKAMs.docx#_Toc201686082)