# DEDICATION

WE DEDICATE THIS WORK TO MY FAMILY

# ACKNOWLEDGEMENT

Drafting this document would not have been possible without the contribution of some people who took upon themselves to see this work being accomplished. Our gratitude goes to the following people:

* **Mr. Armand Claude ABANDA**, for his support, words of encouragement and the different advices on how to approach situations.
* The chief executive officer of HB Train, **Mr. AMOMNO Ngongo Stephane** and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Our professional supervisor, **Mr. AMOMBO Ngongo Stephane** for his moral support, advice and professional expertise she shared with us.
* Our academic supervisor **Mr. AGBOR Donald Anderson** for his moral support, effort and advices throughout the academic year.
* To our academic teachers **Mrs. TCHOUTOUO Isabelle, Mrs. ONGUENE AND Mr. MESSIO** for their advices and assistance in realizing this document.
* To **Mr. NGONO Belong Christian** for the tough training, advices and support.
* To all my teachers of AICS CAMEROON for the knowledge they impacted me with for me to be where I am today. For my brothers and sister for their love and support.
* 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 and friends for their collaborative work throughout the academic year

# CONTENT

[DEDICATION i](#_Toc179099398)

[ACKNOWLEDGEMENT ii](#_Toc179099399)

[CONTENT iii](#_Toc179099400)

[GLOSSARY v](#_Toc179099401)

[LIST OF FIGURES vi](#_Toc179099402)

[LIST OF TABLES viii](#_Toc179099403)

[ABSTRACT ix](#_Toc179099404)

[RESUME x](#_Toc179099405)

[GENERAL INTRODUCTION 1](#_Toc179099406)

[PART ONE: INSERTION PHASE 2](#_Toc179099407)

[INTRODUCTION 4](#_Toc179099408)

[I. WELCOME AND INTEGRATION 5](#_Toc179099409)

[II. GENERAL PRESENTATION OF HB TRAIN 5](#_Toc179099410)

[III. ORGANISATION OF HB TRAIN 7](#_Toc179099411)

[IV. HARDWARE AND SOFTWARE RESOURCES OF HB TRAIN 11](#_Toc179099412)

[CONCLUSION 13](#_Toc179099413)

[PART TWO: TECHNICAL PHASE 14](#_Toc179099414)

[INTRODUCTION 16](#_Toc179099415)

[I. PRESENTATION OF THEME 16](#_Toc179099416)

[II. DESCRIPTION OF EXISTING SYSTEM 16](#_Toc179099417)

[III. CRITICISM OF THE EXISTING SYSTEM 17](#_Toc179099418)

[IV. PROBLEMATIC 17](#_Toc179099419)

[V. PROPOSED SOLUTION 18](#_Toc179099420)

[P 20](#_Toc179099421)

[PART THREE: SPECIFIACTION BOOK 20](#_Toc179099422)

[INTRODUCTION 22](#_Toc179099423)

[I. CONTEXT AND JUSTIFIACTION 23](#_Toc179099424)

[II. Objective of the project 24](#_Toc179099425)

[III. Expression of needs 25](#_Toc179099426)

[IV. ESTIMATED COST OF THE PROJECT 27](#_Toc179099427)

[V. PROJECT PLANING 30](#_Toc179099428)

[VI. LIST OF PARTICIPANTS DELIVERABLES 32](#_Toc179099429)

[CONCLUSION 33](#_Toc179099430)

[PART FOUR: ANALYSIS PHASE 34](#_Toc179099431)

[INTRODUCTION 36](#_Toc179099432)

[I. PRESENTATION OF THE ANALYSIS METHODS 37](#_Toc179099433)

[II. JUSTIFICATION OF THE ANALYSIS METHOD 43](#_Toc179099434)

[III. MODELLING OF THE PROPOSED SOLUTION 44](#_Toc179099435)

[CONCLUSION 65](#_Toc179099436)

[PART FIVE: CONCEPTION PHASE 66](#_Toc179099437)

[INTRODUCTION 68](#_Toc179099438)

[PART SIX: REALIZATION PHASE 80](#_Toc179099439)

[INTRODUCTION 82](#_Toc179099440)

[Deployment diagram 85](#_Toc179099441)

[CONCLUSION 92](#_Toc179099442)

[PART SEVEN: USER GUIDE 93](#_Toc179099443)

[INTRODUCTION 95](#_Toc179099444)

[I. DEPLOYMENT OF THE WEB APPLICATION 95](#_Toc179099445)

[II. SHOWCASE 99](#_Toc179099446)

[CONCLUSION 102](#_Toc179099447)

[PERSPECTIVES 103](#_Toc179099448)

[GENERAL CONCLUSION 104](#_Toc179099449)

[ANNEXES 105](#_Toc179099450)

[BIBLIOGRAPHY 106](#_Toc179099451)

[WEBOGRAPHY 107](#_Toc179099452)

[TABLE OF CONTENT 108](#_Toc179099453)

# GLOSSARY

* 2TUP: TWO TRACK UNIFIED PROCESS
* AICS: AFRICAN INSTITUTE OF COMPUTER SCIENCE
* UML: UNIFIED MODELLING LANGUAGE.
* OMT: object-Modelling Technique
* MVC: Model View controller
* API : APPLICATION PROGRAMMING INTERFACE
* 2TUP: TWO TRACK UNIFIED PROCESS
* DBMS : DATABASE MANAGEMENT SYSTEM
* UP: UNIFIED PROCESSES
* HTTP: HYPERTEXT TRANSFER PROTOCOL

# LIST OF FIGURES

[*Figure 1: Geographical location (Source: HB TRAIN)* 7](#_Toc179099454)

[*Figure 2: organization chart* 9](#_Toc179099455)

[*Figure 3: GANTT CHART* 31](#_Toc179099456)

[*Figure 4: UML diagrams overviews (Source: https://creatly.com/log/diagram-types-examples/)* 39](#_Toc179099457)

[*Figure 5:2TUP diagram (Source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)* 41](#_Toc179099458)

[*Figure 6: formalism of a use case diagram* 45](#_Toc179099459)

[*Figure 7: general usecase diagram* 48](#_Toc179099460)

[*Figure 8: manage course use case* 49](#_Toc179099461)

[*Figure 9: take assessment usecase* 49](#_Toc179099462)

[*Figure 10 Communication diagram formalism* 51](#_Toc179099463)

[*Figure 11: add assessment communication diagram* 52](#_Toc179099464)

[*Figure 12:communication diagram authenticate* 53](#_Toc179099465)

[*Figure 13: submit assessment communication diagram* 54](#_Toc179099466)

[*Figure 14 Formalism of a sequence diagram* 55](#_Toc179099467)

[*Figure 15:Sequence diagram authenticate* 57](#_Toc179099468)

[*Figure 16: add assessment sequence diagram* 58](#_Toc179099469)

[*Figure 17: submit assessment sequence diagram* 58](#_Toc179099470)

[*Figure 18 Formalism of an activity diagram* 59](#_Toc179099471)

[*Figure 19: create account activity diagram* 62](#_Toc179099472)

[***Figure 20: add assessment activity diagram*** 63](#_Toc179099473)

[*Figure 21: submit assessment activity diagram* 64](#_Toc179099474)

[*Figure 22: Formalism of a class diagram* 71](#_Toc179099475)

[*Figure 23: system class diagram* 72](#_Toc179099476)

[*Figure 24: state machine diagram formalism* 73](#_Toc179099477)

[*Figure 25: manage student state machine diagram* 75](#_Toc179099478)

[*Figure 26: search state machine diagram* 76](#_Toc179099479)

[*Figure 27: Package diagram formalism* 77](#_Toc179099480)

[*Figure 28:system package diagram* 79](#_Toc179099481)

[*Figure 29: deployment diagram formalism* 83](#_Toc179099482)

[*Figure 30: deployment diagram* 85](#_Toc179099483)

[*Figure 31: Formalism of a component diagram* 86](#_Toc179099484)

[*Figure 32: system component diagram* 88](#_Toc179099485)

[*Figure 33: launching the server* 95](#_Toc179099486)

[*Figure 34: open database manager* 96](#_Toc179099487)

[*Figure 35: create database* 96](#_Toc179099488)

[*Figure 36: create app database* 97](#_Toc179099489)

[*Figure 37: import database structure* 97](#_Toc179099490)

[*Figure 38: importation of database* 98](#_Toc179099491)

[*Figure 39: successful importation* 98](#_Toc179099492)

[*Figure 40: launching the app* 99](#_Toc179099493)

[*Figure 41: sign in as admin* 99](#_Toc179099494)

[*Figure 42: admin dashboard* 100](#_Toc179099495)

[*Figure 43: fill form* 100](#_Toc179099496)

[*Figure 44: account created* 100](#_Toc179099497)

[*Figure 45: course display* 101](#_Toc179099498)

# LIST OF TABLES

[*Table 1: ORGANISATION STAFF* 8](#_Toc179099499)

[*Table 2: material resources of HB TRAIN* 11](#_Toc179099500)

[*Table 3: software resources of HB TRAIN* 12](#_Toc179099501)

[*Table 4: Hardware resources(source: mercurial)* 27](#_Toc179099502)

[*Table 5: Software resources(source; mercurial)* 28](#_Toc179099503)

[*Table 6: Human Resources(source: mercurial)* 29](#_Toc179099504)

[Table 7 Total Estimated Cost 29](#_Toc179099505)

[*Table 8: Chronogram of activity* 30](#_Toc179099506)

[*Table 9: Components of a usecase diagram* 45](#_Toc179099507)

[*Table 10: Identification of actors and their use cases* 47](#_Toc179099508)

[*Table 11: Textual description of use case Authenticate* 49](#_Toc179099509)

[*Table 12:Components of a communication diagram* 52](#_Toc179099510)

[*Table 13: components of a sequence diagram* 56](#_Toc179099511)

[*Table 14: Components of an activity diagram* 60](#_Toc179099512)

[*Table 15: components of a class diagram* 72](#_Toc179099513)

[*Table 16: component of a state machine diagram* 74](#_Toc179099514)

[*Table 17: components of a package diagram* 78](#_Toc179099515)

[*Table 18: components of a deployment diagram* 84](#_Toc179099516)

[*Table 19: Component of a component diagram* 87](#_Toc179099517)

[*Table 20: material resources* 89](#_Toc179099518)

[*Table 21: technological stacks* 89](#_Toc179099519)

# ABSTRACT

In today's fast-paced educational environment, there is a growing need for tools that enable continuous learning beyond the classroom. This project, themed **"CONCEPTION AND DEVELOPMENT OF AN EDUCATIONAL SUPPORT AND PERSONALIZED STUDENT PROGRESS PLATFORM,"** addresses this challenge by providing a solution that assists teachers in monitoring student progress outside of regular working hours and supports students in continuing their learning after class. The platform is designed to extend learning beyond school hours by offering personalized feedback, tracking academic progress, and providing resources tailored to each student's needs. It empowers teachers to stay connected with their students' development remotely and offers students the opportunity to maintain learning momentum at their own pace. With integrated communication tools and progress tracking, the platform facilitates seamless collaboration between teachers and students. This project follows agile methodologies and employs UML (Unified Modeling Language) for a structured and user-centered system design. The iterative development process ensures the platform evolves to meet the needs of both educators and learners effectively.

Keywords

* Teacher Assistance
* Student Progress
* Continuous Learning
* Educational Technology
* Remote Support

# RESUME

Dans l'environnement éducatif rapide d'aujourd'hui, il y a un besoin croissant d'outils qui permettent un apprentissage continu au-delà de la salle de classe. Ce projet, intitulé "CONCEPTION ET DÉVELOPPEMENT D'UN SUPPORT ÉDUCATIF ET D'UNE PLATEFORME DE PROGRÈS PERSONNALISÉ DES ÉTUDIANTS", répond à ce défi en proposant une solution qui aide les enseignants à suivre les progrès des élèves en dehors des heures de travail habituelles et soutient les élèves dans la poursuite de leur apprentissage après les cours. La plateforme est conçue pour prolonger l'apprentissage au-delà des heures scolaires en offrant des retours personnalisés, en suivant les progrès académiques et en fournissant des ressources adaptées aux besoins de chaque élève. Cela permet aux enseignants de rester connectés au développement de leurs élèves à distance et offre aux étudiants l'opportunité de maintenir leur élan d'apprentissage à leur propre rythme. Avec des outils de communication intégrés et un suivi des progrès, la plateforme facilite une collaboration fluide entre les enseignants et les étudiants. Ce projet suit des méthodologies agiles et utilise le UML (Langage de Modélisation Unifié) pour une conception de système structurée et centrée sur l'utilisateur. Le processus de développement itératif garantit que la plateforme évolue pour répondre efficacement aux besoins des éducateurs et des apprenants.

Mots-clés:

* Assistance aux enseignants
* Progrès des étudiants
* Apprentissage continu
* Technologie éducative
* Assistance à distance

# GENERAL INTRODUCTION

The evolution of education has been shaped by numerous technological advancements, enabling more effective learning and teaching processes. While traditional education systems focused primarily on in-person, classroom-based learning, the rise of digital tools and platforms has introduced new possibilities for supporting students outside formal learning environments. These developments have significantly enhanced educational experiences, allowing students to learn at their own pace and enabling teachers to track their students' progress beyond regular school hours. In the modern educational landscape, personalized learning platforms have become vital in fostering student growth and teacher-student collaboration. In alignment with this trend, the African Institute of Computer Sciences (AICS) continues to fulfill its mission of equipping future African engineers with the skills necessary to meet the demands of evolving industries. As part of this mission, AICS provides students with a three-month internship program that allows them to apply their acquired skills in real-world scenarios, further honing their problem-solving abilities. This report covers the development of a project carried out during out internship at **“HB TRAIN”** which focused on the theme **"CONCEPTION AND DEVELOPMENT OF AN EDUCATIONAL SUPPORT AND PERSONALIZED STUDENT PROGRESS PLATFORM."** The goal of this project is to provide teachers with a platform that assists them in monitoring students' academic progress outside of working hours, while also empowering students to continue learning autonomously after school hours. By using this platform, educators can deliver tailored feedback and resources based on each student's unique needs, fostering personalized learning pathways. The project was carried out in six main phases:

**The insertion phase,** where the host company and the intern's integration process are detailed;

**The specification phase,** identifying user needs and outlining the project's constraints and requirements;

**The analysis phase,** presenting the methodology used to design the system and initial models of the platform;

**The conception phase,** which delves into the detailed study of the platform’s real-world components;

**The realization phase,** where the system is developed and made operational for the users;

**The user guide,** providing instructions on how to install and use the platform.

RTION PHASE

# PART ONE: INSERTION PHASE

**Preamble**

This document provides an overview of our enterprise, covering the following key aspects welcoming and Introduction, discovering the working environment, organizational structure, and partnership and localization plan.

**Content Overview**

INTRODCUTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF HB TRAIN
3. ORGANISATION OF HB TRAIN
4. HARDWARE AND SOFTWARE RESOURCES
5. BRIEF PRESENTATION OF OUR THEME

CONCLUSION

# INTRODUCTION

The insertion phase is the period during which a student must become familiar with our working environment, the staff and the other interns. Here, we will begin by discussing our first week in the company, how we were welcomed and how we began adapting to our internship environment. We will proceed by exploring the history of the company discovering its missions, learning what its major activities are and witnessing some of its key realisations. We will also get to understand how the company is structured administratively and functionally so that it operates effectively and accomplishes its goals. Furthermore, we will look at the hardware equipment used by the company, and the software resources used in its daily operations. We will then introduce our chosen theme for the internship period, briefly explaining what it is all about.

# WELCOME AND INTEGRATION

On July 1ST, 2024, we visited the premises of HB Train. Upon arrival at the reception, we were warmly welcomed by Mrs. NDIOMO MBAZOA Jacinta. We were then taken to the developer's workspace and the CEO's office, Mr. NGONGO Amombo, which was intended to familiarize us with the new environment and facilitate our adaptation after the initial introduction; we were asked to assist the interns already present in the enterprise. Furthermore, on the same day, we were tasked with researching potential themes for our defense presentation the next day; we began working on the project assigned to us by some of the interns. Simultaneously, we continued our research on potential themes for the defense. By the end of the second week, we had successfully delivered the assigned task. During the first two weeks, we primarily focused on developing the project using HTML, CSS, JavaScript, Python, and PHP. Toward the end of the two-week period, we were assigned our respective defense themes based on the guidelines provided by the Resident Representatives of AICS-Cameroon.

# GENERAL PRESENTATION OF HB TRAIN

* **History**

**HB TRAIN**, born on January 1, 2024, is not just a startup; it's an extraordinary human adventure. Founded by Ms. Amombo Ngongo Stefan, a visionary with a compassionate heart, HB TRAIN is the embodiment of a fierce determination to make the world a better place through technology.

* **The Spark of an Idea**

The story of HB TRAIN begins with a chance encounter. While observing a blind man struggling to hail a taxi, Ms. Amombo was deeply touched by the daily challenges faced by the visually impaired. This experience set in motion a series of events that would change her life and the lives of many others.

* **From Innovation to Impact**

Inspired by this encounter, Ms. Amombo brought to life a revolutionary invention: a smart helmet for the blind equipped with artificial intelligence. This ingenious device not only facilitated mobility for the visually impaired but also paved the way for a multitude of other innovations.

* **A Shared Vision, Amplified Success**

Convinced of the transformative potential of technology, Ms. Amombo decided to share her knowledge and expertise with others. Partnering with Ms. Ndiomo Mbazoa Jacinta, an astute financial expert, she founded HB TRAIN, a startup dedicated to training young people passionate about IT and eager to pursue their own projects.

* **A Spectrum of Services for a Connected World**

HB TRAIN didn't just stop at training talent. Under the visionary leadership of Ms. Amombo Ngongo Stefan (CEO), Ms. Ndiomo Mbazoa Jacinta (CFO), Ms. Biaka Tony (Internal Controller), and Ms. Ayissi Atangana Jean Marie (Marketing Director), the company developed a comprehensive range of innovative services, including:

* **Smart Home Implementation:** HB TRAIN transforms ordinary homes into intelligent and comfortable spaces, giving users greater control over their environment.
* **Blind Helmets:** HB TRAIN's flagship invention continues to evolve, offering increased autonomy and freedom of movement for the visually impaired.
* **Custom AI for Businesses:** HB TRAIN leverages its AI expertise to develop customized solutions that meet the specific needs of businesses.
* **IT Security for Businesses:** In an increasingly complex digital world, HB TRAIN protects businesses from cyber threats and ensures the security of their data.
* **Applications:** HB TRAIN designs and develops custom mobile and web applications to meet the needs of individuals and businesses.
* **Flyers and Video Editing:** HB TRAIN provides visual communication services to help businesses stand out and reach their target audience.
* **Recognition and Awards**

HB TRAIN's talent and innovation have been recognized on multiple occasions. Ms. Amombo Ngongo Stefan has won prestigious awards, including for her smart helmet for the blind, at competitions such as the University Games. These accolades are a testament to HB TRAIN's commitment to developing cutting-edge technological solutions that have a positive impact on people's lives.

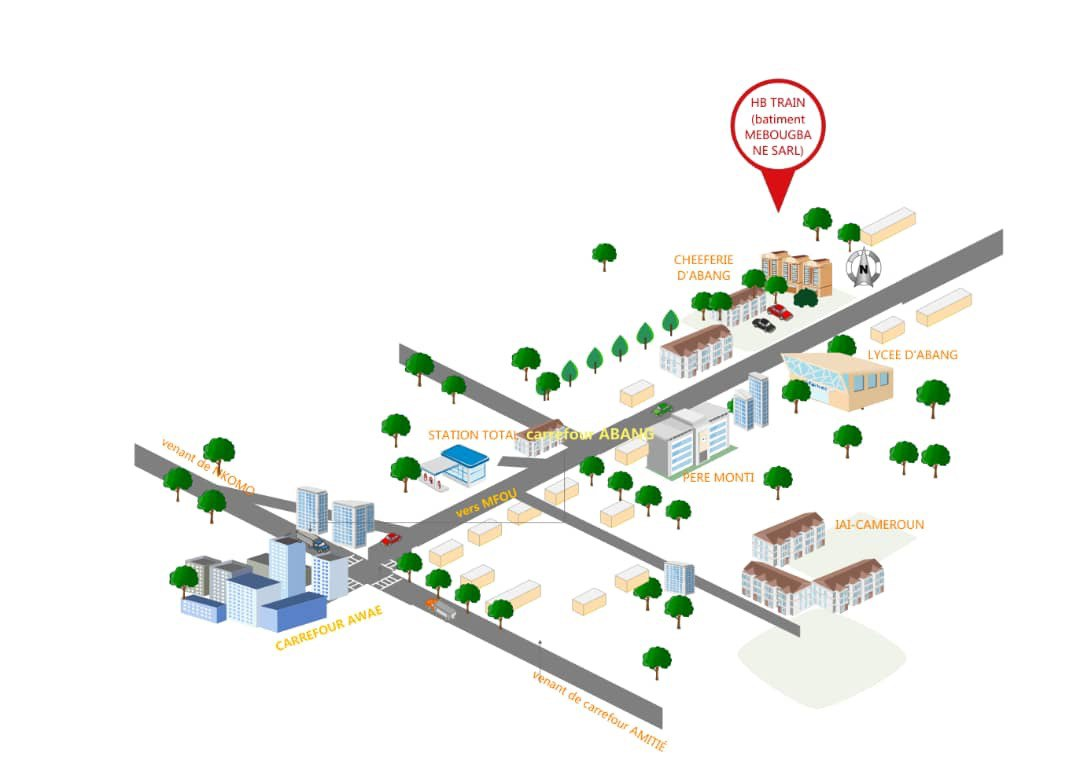
* **HB TRAIN: More Than a Business, a Mission**

HB TRAIN is not just a thriving business; it's an inspiration and a model of success. The story of Ms. Amombo Ngongo Stefan and her team demonstrates that with passion, perseverance, and a commitment to innovation, it's possible to transform ideas into reality and create a positive impact on the world.

# ORGANISATION OF HB TRAIN

## GEOGRAPHICAL LOCATION

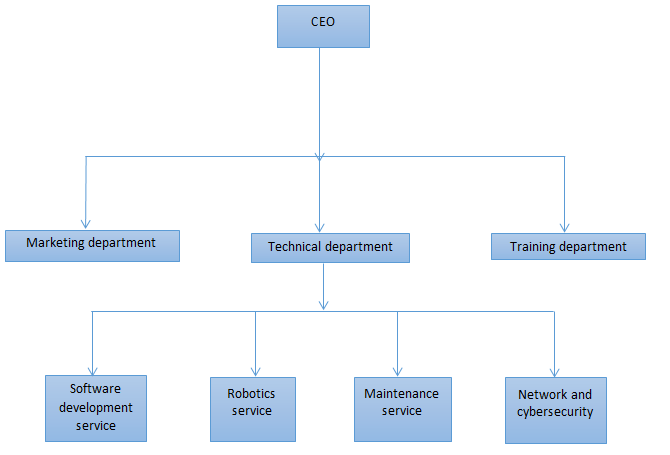
HB-TRAIN is located in Yaoundé, in the Centre region of Cameroon, in the Abang district.

*Figure 1: Geographical location (Source: HB TRAIN)*

*Table 1: ORGANISATION STAFF*

|  |  |
| --- | --- |
| Company Name | HB TRAIN |
| Type Of Enterprise | Start-Up |
| Chief Executive/Developers/Project Manager | Mr. Amombo Ngongo Stephane |
| RESPONSABLE COMMUNICATION / MARKETING | Mr. Ayissi Atangana Jean Marie |
| PHONE NUMBER | 691-56-99-75/671-51-89-25 |
| WEBSITE | Hbtrain.com |
| LANGUAGE | English and French |
| LOGO |  |

## ORGANIZATIONAL CHART OF HB Train



*Figure 2: organization chart*

* **GENERAL MANAGEMENT**: Its function is to define the company's objectives, plan and choose actions to be taken, monitor their execution, and take any necessary corrective and developmental measures. It ensures that all priorities defined within the company are respected.
* **MARKETING MANAGEMENT:** Its role is to develop and implement marketing plans and actions to boost the sale of the company's goods or services.
* **TECHNICAL MANAGEMENT:** Responsible for the efficiency of the production process within the company. The technical director collaborates with general management and analyses available resources to make recommendations regarding organization, safety, and equipment investment.
* **TRAINING MANAGEMENT:** It is the department responsible for training staff on the use of new technologies and computer equipment. This department ensures that the staff is capable of using the computer tools at their disposal.

## MISSIONS AND ACTIVITIES OF HB TRAIN

### MISSIONS

The primary mission of HB-TRAIN is to provide Cameroonian society with technological solutions to improve their living conditions in medical, educational, or professional fields. To achieve this goal, they have set objectives, which include:Développer des applications et des outils technologiques innovants pour faciliter l'accès aux soins de santé dans les zones rurales et urbaines.

* Developing innovative applications and technological tools to facilitate access to healthcare in rural and urban areas.
* Establishing online learning platforms and digital educational resources to promote education and professional training.
* Designing software solutions and technological equipment tailored to the specific needs of Cameroonian businesses and workers.
* Organizing training programs and support to help communities adopt and effectively use new technologies.
* Promoting digital inclusion by ensuring that the solutions developed are accessible to everyone, including the most vulnerable people.
* Investing in research and development to design technologies adapted to the local context and evolving needs.

### ACTIVITIES

HB-TRAIN is a startup specializing in IT services, which include:

* **Website Development:** They offer dynamic and static websites, paying attention to detail and perfection. Their expert and professional team will accompany clients from the expression of their needs to the realization of a modern and professional website.
* **IT Maintenance:** They ensure the proper functioning of the IT park (a place consisting of several electronic devices such as computers, printers, etc.). For example, when a computer breaks down, they take care of its repair. On the software side, they ensure that the software works correctly, handle installations and updates of antivirus programs, constantly check the Microsoft Park and other software, update obsolete applications, and more.
* **IT Security and Networking:** They handle the setup of network equipment (computers, etc.) to share resources (printers, scanners, files, documents, modems, etc.) and secure networked computers against external attacks. They also create restrictions within networks, such as preventing an intern from accessing social networks without prior authorization, among other measures.
* **IT Project Analysis:** They use two analysis methods (UML and MERISE) to analyze projects from the beginning to the delivery of the project document. This analysis involves evaluating the objectives to be achieved for the project's realization and setting up all the factors that will help in its completion, including identifying potential problems.
* **Graphic Design and Digital Marketing:** They design flyers, business cards, logos, personalized advertisements to solve visibility issues and manage online marketing to meet clients' needs.

# HARDWARE AND SOFTWARE RESOURCES OF HB TRAIN

### Material Resources

The hardware of this structure consists of the following elements:

*Table 2: material resources of HB TRAIN*

|  |  |  |
| --- | --- | --- |
| Type of Resource | Description | Quantity |
| Computer | Ordinateur de bureau Dell | 10 |
| Printer | Dell desktop computers | 3 |
| Tableau | HP multifonction printers | 4 |
| Whiteboard | Erasable whiteboards | 2 |
| Screen | ND smart flat screen | 1 |
| Large format printer | Yingher large format printer | 1 |

### Software Resources

The software resources are composed of the following elements:

*Table 3: software resources of HB TRAIN*

|  |  |
| --- | --- |
| Software Type | Examples |
| Browser | Google Chrome, Microsoft Edge |
| Antivirus | Kaspersky anti-virus |
| Operating System | Windows 10 professional |
| IDE | Visual studio code, Arduino |
| Office Suite | Microsoft package |
| Graphic Tools | Adobe package |

## **CONCLUSION**

The insertion phase at HB Train lasted for two(02) weeks, we discovered an amazing environment where I found myself in. this environment gave room for hard work, perseverance and collaboration, what most students are looking for. I learnt about the missions of the company and its activities. We met and got familiar with the staff and other interns at HB Train. Finally, we chose my theme **“ CONCEPTION AND DEVELOPMENT OF AN EDUCATIONAL SUPPORT AND PERSONALIZED STUDENTS PROGRESS PLATFORM”.**

# PART TWO: TECHNICAL PHASE

**Preamble**

In this chapter, we shall give a detailed explanation of our theme. That is, what the theme is all about and to give more precision concerning the application we are to produce. This chapter will also describe the problematic that led to our proposed solution.

**Overview**

INTRODUCTION

I THEME PRESENTATION

II STUDY OF THE EXISTING SYSTEM

II CRITICISM OF THE EXISTING SYSTEM

IV PROBLEMATIC

V PROPOSED SOLUTION

CONCLUSION

V PROPOSED SOLUTION

CONCLUSION

INTRODUCTION

I THEME PRESENTATION

II STUDY OF THE EXISTING SYSTEM

II CRITICISM OF THE EXISTING SYSTEM

IV PROBLEMATIC

V PROPOSED SOLUTION

CONCLUSION

# INTRODUCTION

The existing system part of our report is very important. Here, we shall present our theme to improve our understanding of it and increase the likelihood of succeeding, present the study of the existing system, which will guide us on the approach to take. After studying the existing system, we shall present our different critics concerning it. We shall also give the problematic i.e., the different problems phased by the existing system and finally we shall propose a solution to solve these problems.

# PRESENTATION OF THEME

In the context where online learning is becoming increasingly import, it is essential to provide tools that offers continuous educational support and personalized academic management of students. This platform aims to be an interactive space where students can access online courses, ask questions, track their progress, and receive tailored recommendations based on their needs. The primary goal is to ensure constant monitoring via resources available 24/7, while allowing teachers to guide and evaluate their learners. The platform also integrates innovative features that enhance student autonomy when working out of regular instructional hours. Without revealing all aspects from the outset, this solution also holds a surprise in terms of intelligent assistance, always ensuring a seamless and support learning experience.

# DESCRIPTION OF EXISTING SYSTEM

We noticed there was little or no teacher assistance; every student was left on his own, a task is given but not controlled, for shy students, it was difficult to express themselves looking at their mates’ rapid ability to master and understand the course. It was such a way that only those who rapidly understand the course could go to the next level; the slow learners were left on their own to understand and follow the lead. In addition, there was language barrier; since there was more Anglophones than francophone, the courses where given in French despite the presence of English-speaking students. Moreover, we were leaving very early, giving us less time to student or revise; when the time comes, everybody leaves.

# CRITICISM OF THE EXISTING SYSTEM

At the end of our collection of information concerning the teaching and learning process in HB Train, we find that this system has weaknesses that should be considered despite their warm welcome. However, it should be noted that some observations have been made within the start-up and it would be wise for us to suggest solutions to the various problems noted.

|  |  |  |
| --- | --- | --- |
| Limitation | Consequence | Proposed solution |
| Overcrowded Classroom | Difficulty in monitoring student progress | Use the platform’s analytics to identify at-risk students and provide targeted support. |
| Insufficient personalized attention from teachers | Students may feel isolated and disengaged | Offers personalized learning pathways based on individual assessment the platform. |
| Lack of follow-up mechanism for student performance | Lower academic achievement and higher dropout rates. | Implement real-time progress tracking and feedback systems within the platform. |

# PROBLEMATIC

Despite the growing of digital learning tools, traditional and physical classes still face significant challenges that hinder effective and inclusive education. Through these observations, we ask ourselves **“ HOW CAN ONLINE EDUCATIONAL PLATFORM ADDRESS THE CHALLENGES IN TRADITIONAL PHYSICAL CLASSES? ”**

# PROPOSED SOLUTION

To improve education in Cameroon involves developing a comprehensive educational support platform that centralizes personalized learning. This platform would create individualized learning pathways based on student assessments, provide access to high quality educational resources, and enable real-time progress tracking. It would also include communication tools to foster collaboration between students and educators. By addressing these key areas, the platform aims to promote educational equity and improve student engagement and outcomes across the country.

## CONCLUSION

In conclusion, the development of a comprehensive educational support platform represents a vital step towards addressing the challenges faced by the educational system in Cameroon. By personalizing learning experiences and providing accessible resources, the platform can bridge the gaps in educational equity and enhance students’ engagement. Furthermore, equipping educators with the necessary tools and training will foster a more effective teaching environment. Ultimately, this initiative has the potential to transform the educational landscape, empowering students to achieve their full potential and contributing to the overall development of the nation.

# P

# PART THREE: SPECIFIACTION BOOK

**Preamble**

The specification book outlines the goals to be achieved through this project and the responsibilities of the different parties involved in the project. It specifies and describes the subject and the needs of the users, as well the conditions necessary to realize the project.

**Content Overview**

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 PARTICIPANTS

CONCLUSION

# INTRODUCTION

The specification book of our reports helps us provide details about our theme, to improve our understanding of it and increase the likelihood of it succeeding. To delimitate the scope of our project, we will specify the context of our theme. From the context, we will list the problems we have identified in our context and that we have decided to address throughout the project. After presenting our solution, we will talk about the objective we have set for ourselves for the project. Also, we will explore the needs to which our system will respond both at the functional and non-functional level. We will then look at the estimated financial requirements for our project, and establish a plan we will follow to complete our project on time. From here we will discuss what is expected of us by the end of the project under the project deliverables.

# CONTEXT AND JUSTIFIACTION

Context refers to the circumstances or conditions in which something exists or occurs. It encompasses the background information environment or setting that helps to clarify the meaning of an event, statement or idea. Context can include various factors such as:

* Cultural: the social norms and values that influence understanding.
* Historical: the period and events that shape the situation.
* Situational: the specific condition or environment at a given moment
* Linguistic: the words and phrases surrounding a particular statement that provide meaning.

Understanding context is crucial for effective communication and interpretation, as it influences how information is perceived and understood.

## Context

Cameroon has a mix of public and private schools, with varying quality standards. Taking publics for example, there are numerous students per class for only one(01) teacher making it difficult for teachers to resolve every students’ problems. Overcrowded Classrooms; In many public schools, classrooms can have an average of 90 to 176 students per teacher. This overcrowding results from limited resources and insufficient teacher recruitment to meet the growing demand for education. Such large class sizes make it nearly impossible for teachers to provide individualized attention to each student, leading to a one-size-fits-all approach to instruction.

## JUSTIFICATION

Education in Cameroon is witnessing a significant influx of youth across various regions, with numerous schools established to accommodate this growing demand. However, this rapid expansion does not come with a guarantee of effective follow-up and support for students. Many schools struggle to provide the necessary resources and personalized attention needed for meaningful learning experiences. As a result, despite the increasing number of educational institutions, challenges such as overcrowded classrooms and limited access to quality educational materials persist. This lack of adequate follow-up means that teachers often cannot monitor student progress effectively, leaving many students without the guidance they need to succeed. Consequently, while the desire for education is evident, the system's inability to ensure comprehensive support undermines the potential for academic achievement and personal development among the youth.

# Objective of the project

## General Objectives

To develop a comprehensive educational support platform that personalizes student learning experiences, fosters academic growth, and enhances overall student engagement through tailored resources and progress tracking.

## Specific Objectives

* Personalized Learning Pathways:

Create individualized learning plans based on students' strengths, weaknesses, and learning styles.

* Progress Tracking:

Implement a robust system for monitoring and reporting student progress in real-time, allowing for timely interventions.

* Resource Accessibility:

Provide a centralized repository of educational resources, including tutorials, articles, and interactive materials, tailored to different learning needs.

* Engagement Features:

Develop interactive tools such as quizzes, gamified learning modules, and discussion forums to enhance student participation and motivation.

* Feedback Mechanisms:

Establish channels for students and educators to give and receive feedback, fostering a collaborative learning environment..

* User Training and Support:

Provide training resources and ongoing support for both educators and students to maximize platform effectiveness.

By focusing on these objectives, your project can effectively support students' educational journeys and improve their academic outcomes.

# Expression of needs

## Functional Needs

In this context, functional requirements describe what the system or application should do. The modules are as follows;

1. **User Registration and Profiles:** Secure user registration for students, educators, and administrators. Customizable user profiles that include academic information, learning goals, and preferences.
2. **User Authentication:** Implement secure login mechanisms for users to ensure only authorized access. Support for multi-factor authentication (MFA) to enhance security.
3. **Role-Based Access Control:** Define user roles (e.g., student, teacher, admin) with specific permissions for appropriate access to resources and functionalities. Allow administrators to manage user roles and permissions easily.
4. **Personalized Learning Pathways:** Algorithms to generate personalized learning pathways based on assessments and learning styles. Ability for users to adjust their learning goals and preferences.
5. **Progress Tracking Dashboard:** Interactive dashboards that visually display student progress, achievements, and areas needing improvement. Notifications and alerts for milestones or required actions.
6. **Resource Management:** A centralized library for educational resources, including videos, articles, quizzes, and interactive simulations. Search and filter options for easy access to relevant materials.
7. **Interactive Learning Tools:** Tools for creating quizzes and assessments that adapt to student performance. Gaming features, such as badges and leaderboards, to enhance engagement.
8. **Communication and Collaboration:** Messaging and discussion forums for students and educators to interact. Group project features that facilitate collaboration among students.
9. **Feedback and Assessment Tools:** Mechanisms for educators to provide feedback on assignments and progress. Self-assessment tools for students to reflect on their learning.

## Non-functional needs

They specify the quality attribute of a software system. They judge the software system of application based on Performance, Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to its success. Failing to meet non-functional requirements can result in a system that fails to satisfy user need. **Performance**, **Scalability** and **Security**

**Performance** defines how fast a software system , or its piece responds to certain user’s action under certain workload. In most cases, this metric explains how much a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it’s not always like that. Performance requirement may describe background processes invisible to users. Our goal will be to provide our users with the best performance as it affects the overall user experience.

**Scalability** accesses the highest workloads under which the system will still meet the performance requirements. In this project we will mainly leverage the power of cloud storage and third-party API’s. The application should have a friendly user interface (UI) and should be easy to use. The code should be clear to facilitate future development and improvement. The web application should be resizable when opened on any device (android phone, computer, tablet web browsers)

**Security** is also one of the most important aspects of any system, especially the ones dealing with sensitive user’s information. The application should provide a strong security mechanism to reassure user’s that they can trust it for their sensitive information. Some of the security principles include reassuring confidentiality, accountability, Integrity, Authentication.

# ESTIMATED COST OF THE PROJECT

### Hardware Resources

*Table 4: Hardware resources(source: mercurial)*

| RESOURCES | HARDWARE | USAGE | QUANTITY | UNIT COST(FCFA) |
| --- | --- | --- | --- | --- |
| COMPUTER | DELL  Windows 10 pro 64 bits,  Latitude E6520, 8GB RAM,  500GB hard disk; intel core i5. | main resource for the accomplishment of this project; report writing, designs, analysis, coding | 1 | **300, 000** |
| Removable disk | 8GB USB key | For file transfer from one computer to another. | 1 | **8000** |
| CD | CD-ROM |  | 3 | **2500** |
| Local Server | Mid-rage server(example; intel xeon, 16GB RAM, 1TB HDD) | Hosts the platform for testing and initial deployment | 1 | **400 000** |
| MODEM | 4g LTE-advanced mobile Wi-Fi hotspot mq531 150mps | Used to share internet access amongst devices | 1 | **15000** |
| Total |  |  |  | **725500** |

### Software Resources

*Table 5: Software resources(source; mercurial)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | HARDWARE | USAGE | QUANTITY | UNIT COST(FCFA) |
| Development tool | Visual studio code | Code editing | 1 | Open source |
| Project planner | Gantt project | Project planning tool | 1 | Free software |
| Operating system | Microsoft windows 10 pro | Computer operating system | 1 | 103000 |
| Text Editor | Microsoft office word 2019 | For typing and formatting of report | 1 | 161000 |
| Presentation | Microsoft office PowerPoint 2019 | For making presentations | 1 | 84230 |
| Modelling tool | Visual-paradigm Modeler version | Modelling the system in uml | 1 | 3350 |
| Geo-location plan designing tool | Microsoft Visio 2007 | To draw location plan | 1 | Free version |
| Web browser | Google chrome | For running and testing application | 1 | Free version |
| Database management system | MYSQL | Communing with the database | 1 | Open source |
| Total |  |  |  | 351,580 |

### Human Resources

*Table 6: Human Resources(source: mercurial)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | NUMBER | COST PER DAY | NUMBER OF DAYS | COST (FCFA) |
| Project manager | 1 | 30 000 | 90 | 2 700 000 |
| Analyst | 1 | 25 000 | 30 | 750 000 |
| Designer | 3 | 20 000 | 25 | 500 000 |
| Programmer | 3 | 25 000 | 30 | 750 000 |
| Security expert | 1 | 25 000 | 10 | 250 000 |
| Network administrator | 1 | 20 000 | 10 | 200 000 |
| Program Tester | 5 | 10 000 | 10 | 500 000 |
| Total | 15 | 155 000 |  | 5 650 000 |

### Total Project Estimated Cost

Table 7 Total Estimated Cost

|  |
| --- |
| Overall total (FCFA)  6 727 080 |

# PROJECT PLANING

### Chronogram of activities

*Table 8: Chronogram of activity*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE | OBJECTIVE | OUTPUT | DURATION | PERIOD |
| Insertion | Welcome and installation into the into the company, and the attribution of themes | Insertion report | 2weeks | 01st to 14th July 2023 |
| Specification | Assessment of functional needs | Specification book | 1week 6days | 15th to 28th July 2023 |
| Analysis | Analysis of the system | Analysis book | 3weeks | 29th July to 19th August 2023 |
| Conception | Hardware and software conception | Conception book | 2weeks | 20th August to 09th September 2023 |
| Realization | Realization of the system | Realization document | 4weeks 2days | 10th September to 20th September |
| Deployment | Deployment of the system | Testing of the software and debugging | 1week 2days | 20th to 5th September |

### GANTT CHART

*Figure 3: GANTT CHART*

# LIST OF PARTICIPANTS DELIVERABLES

## List of participants

|  |  |  |
| --- | --- | --- |
| NAME | TITLE | ROLE |
| Mr. AGBOR Anderson | Lecturer at AICS-CAMEROON | Academic supervisor |
| Mr. NGONGO Amombo | CEO at HB TRAIN | Professional supervisor |
| PAULINE SIDONIE BAMSECK | software engineering student at AICS-Cameroon | Analyst and developer |

### Deliverables

We are supposed to submit a report at the end of this project that includes;

* An insertion phase book;
* A specification book;
* An analysis book;
* A conception book;
* A realization book;
* A user guides.

# CONCLUSION

The specification book helps us to site the different needs we need in order to implement our mobile application for student follow-up, we saw list of participants and deliverables for our project. In time allocated for this section, we were able to accomplish the different task, we will move directly to the next phase, which is the analysis phase. In the analysis phase, we will study the existing system in detail and model our system with a modelling language and process.

# PART FOUR: 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 Overview**

INTRODUCTION

1. PRESENTATION OF THE ANALYSIS METHOD
2. JUSTIFICATION OF THE ANALYSIS METHOD
3. MODELING OF THE SYSTEM

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.

# PRESENTATION OF THE ANALYSIS METHODS

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;

* Merise:It is an information system design and development widely used in France. It was first early introduced in 1980s.
* 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.
* 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.
* 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.

### MODELING WITH UML 2.5

The unified modeling language (UML) is a general purpose, developmental modeling language in the field of computer engineering that is intended to provide a standard way to visualize the design of a system. It was born from the merge of the three object-oriented programming methods namely:

The BOOCH Method (a method for object-oriented software development that has an object 3 modeling language, an iterative object-oriented development process and a set of recommended practices) developed by Grady brooch, James Rumbaing’s Object Modeling Technique (OMT) (an object modeling approach for software modeling and design) and Ivar Jacobson’s OBJECT SOFTWARE ENGINEERING (OOSE) (software design technique used in object-oriented programming). UML was created to forge a common, semantically and syntactically rich visual modeling language for the architecture, design and implementation of complex software systems both structurally and behaviorally.

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 its 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.

#### 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:

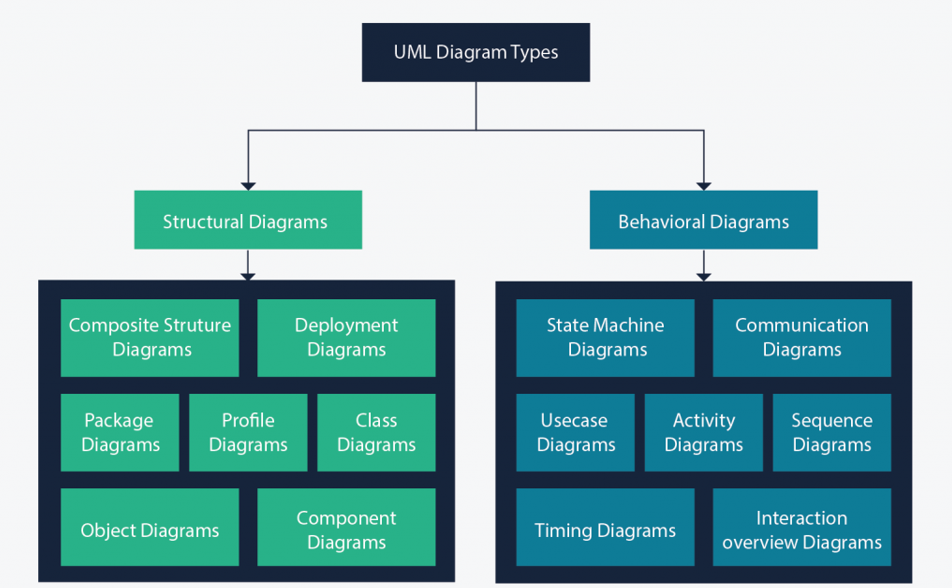
* Class diagram;
* Object diagram;
* Package diagram;
* Composite structural;
* Deployment diagram;
* Component diagram;
* Profile diagram.

#### 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

### UML 2.5 diagrams overview



*Figure 4: UML diagrams overviews (Source: https://creatly.com/log/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.

### 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.

### A Unified Process

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

**Iteration** are 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 users evaluate the evolution of the system.

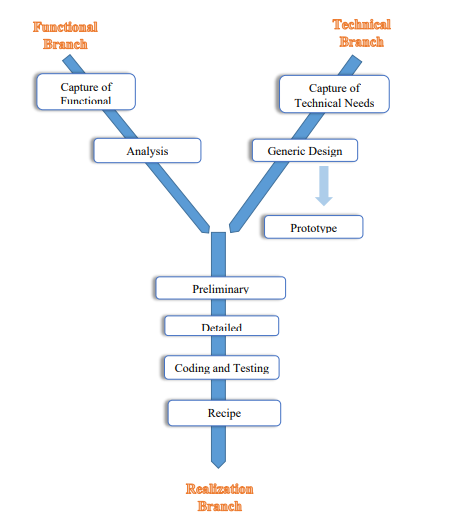
**An increment** is the difference between two released products at the end of two iterations. Each iteration that the group is capable of integrating the technical environment in order to develop a final product and give users the possibility of having tangible results.

**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.

### 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.



*Figure 5:2TUP diagram (Source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)*

#### 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.

#### 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, interphases and the software configuration model are defined.

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

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

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

# JUSTIFICATION OF THE ANALYSIS METHOD

Some developers have been successful at developing small systems without the use of diagrams or other features of modelling languages. However, as their systems become larger and larger, such developers have increasingly difficulties to seeing the ‘big picture’ of their system and are liable to create poor designs and take much longer time in their work. Most system are therefore documented with the use of diagrams. These provide views of structure and functionality that would be difficult to grasp by looking at code or textual description alone. In other words, diagrams provide abstraction.

Our choice for the UML modelling language and the 2TUP development process are based on the following criteria:

* UML is a language which is centered on user’s needs
* UML is based on the object-oriented approach
* 2TUP offers a development cycle (the Y shape development cycle) which dissociates the technical aspects from the functional aspects
* UML produces good standards for software development
* UML has large visual elements to construct and easy to follow

2TUP is a process based on object approach and is constructed on UML.

# MODELLING OF THE PROPOSED SOLUTION

### Capture of the functional needs

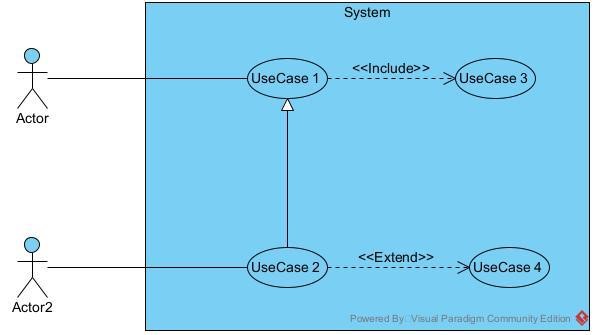
The functional requirements of a system specify the desired behavior or basic functionalities that the system is required to perform. In this section, we will present the following diagrams: a use case diagram along with textual descriptions of use cases, an activity diagram, a communication diagram, an interaction overview diagram, and a state machine diagram.

#### Use Case Diagram

##### Definition

A use case diagram is the primary form of a system/ software requirements for a new software program under development. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be donated both textual and visual representation (such as UML). A key concept of use case modelling is that it helps us design a system from the end user’s perspective. It is an effective technique for communicating system behaviour in the user’s terms by specifying all externally visible system behaviour.

##### Formalism



*Figure 6: formalism of a use case diagram*

##### Components of a use case diagram

##### 

*Table 9: Components of a usecase diagram*

|  |  |  |
| --- | --- | --- |
| Element Description and Main properties | | Notation |
| Actor | Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system. |  |
| Use case | It corresponds to the objective/function of the system motivated by the need for one or more actors. The set of use cases describes the goal of a system. They are described in the form of infinitive verbs plus objects. | Name of usecase 1 |
| Association | It indicates that an actor takes part in a use case. | B  Actor\_1 |
| Inclusion | An inclusion relation denotes that an included action must be performed before another action can be performed. A includes B signifies that B is a compulsory part of A. | <<  include  >>  A  B |
| Exclusion | It denotes that an action may be performed while another one is being performed. A use case B extends A means B is an optional part of A. | <<  extend  >>  A  B |
| Inheritance | It is the only possible relation between actors. | Acteur\_3  Acteur\_4 |
| Generalization | This shows that a uses case is a kind of another. This relation also permits to decompose a complex case into smaller and simple cases. | A  B  Actor\_1  Actor\_2 |
| System | This is the representation of our system |  |

##### Identification of actors and their use cases

We will now list the actors likely to interact with the system, but at first, we will define an actor.

An actor represents the abstraction of a role played by external entities that interact directly with the system studied. The actors of the system identified at first are:

*Table 10: Identification of actors and their use cases*

|  |  |
| --- | --- |
| Actor | Role |
| visitor | His roles are to view app details and create account. |
| student | His roles are to follow courses by taking assessments, consulting his results, takes quizzes, paying courses, and sending feedbacks.  He also has the possibility to manage his account and receive notifications. |
| instructor | His roles are to manage courses by giving assessments, creating virtual classes and receiving notifications. |
| Administrator | His roles are to manage virtual classes and users. |

##### General use case diagram

##### 

*Figure 7: general usecase diagram*

##### manage course



*Figure 8: manage course use case*

##### Take assessment



*Figure 9: take assessment usecase*

**case textual description**

As we mentioned earlier, a use case represents a transaction. Here, it is a question of breaking down the use cases, making a textual description and then emphasizing on the different detailed scenario. This will be done using diagrams known as the sequence diagram and the activity diagram.

##### Use case authenticate

*Table 11: Textual description of use case Authenticate*

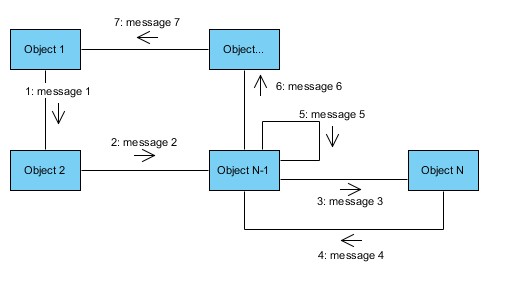
|  |  |
| --- | --- |
| **Goal in context:** | Grant the user access to the application |
| **Pre-condition(s):** | The system should be loaded |
| Nominal scenario: | 1. The user clicks the login button 2. The system provides the login form 3. The user fill and submit the form 4. The system verifies the credentials 5. The system verifies the information in the database 6. The system starts the user’s session |
| Alternative scenario (invalid credentials) | 4.1 The system displays the error message  4.2 The system returns to step (2) of the nominal scenario |
| Alternative scenario (user does not exist) | 5.1 The system displays the error message  5.2 The system returns to step (2) of the nominal scenario |
| Post-conditions (s) | The user’s session is launched. |

#### Communication Diagram

##### 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.

##### Formalism



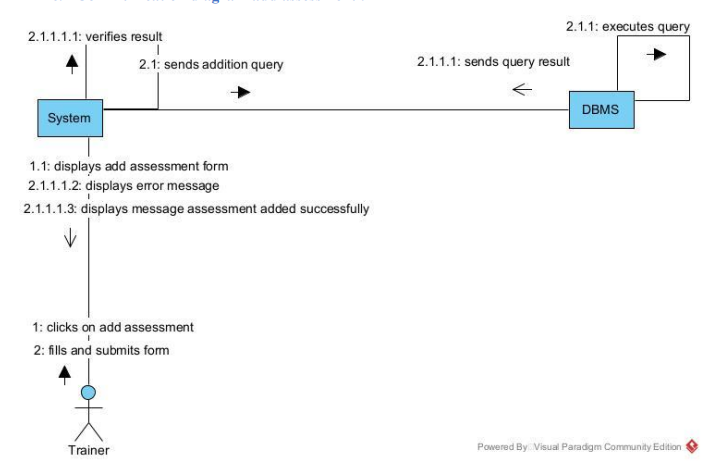
*Figure 10 Communication diagram formalism*

**c. components of a communication****diagram**

*Table 12:Components of a communication diagram*

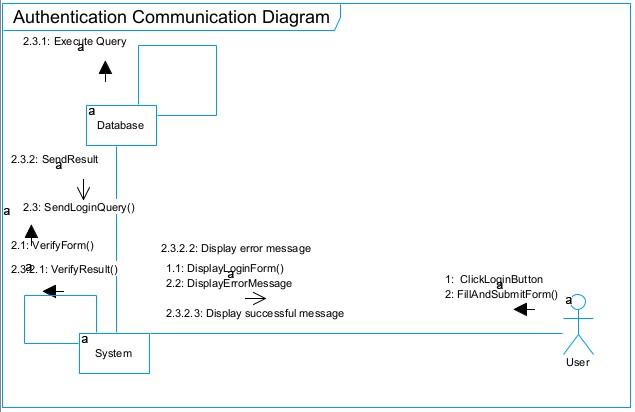
|  |  |  |
| --- | --- | --- |
| **Element** | **Representation** | **Description** |
| Object |  | An actor represents an individual participant in the interaction conversation. |
| link |  | It initiates an association it connects two objects together for them to communicate. |
| Actor |  | A role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

* 1. **Add assessment communication diagram**



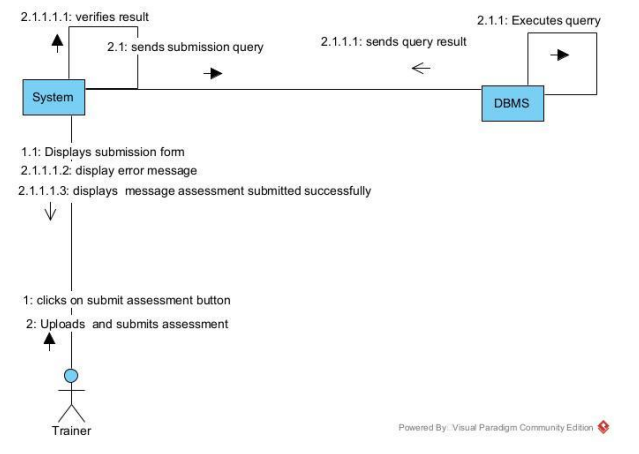
*Figure 11: add assessment communication diagram*

**ii. authentication communication diagram**



*Figure 12:communication diagram authenticate*

**iii. submit assessment communication diagram**



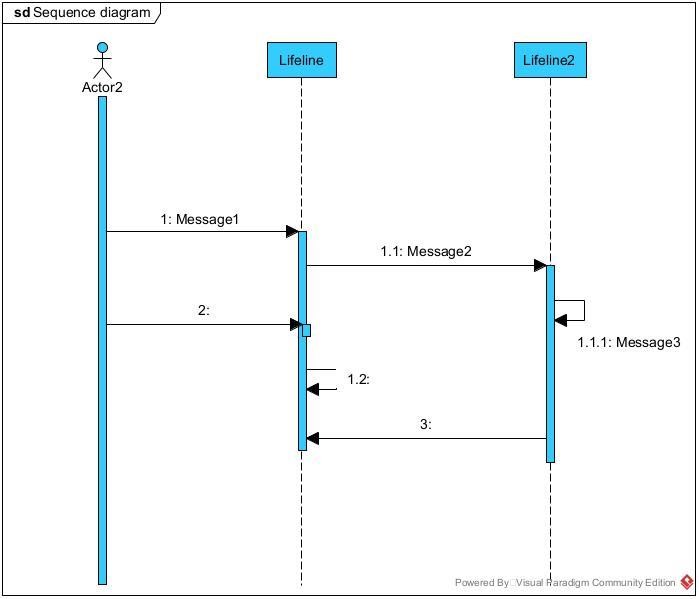
*Figure 13: submit assessment communication diagram*

#### Sequence Diagram

##### Definition

A sequence diagram is a form of interaction diagram, which shows objects as lifelines running down the page and with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. Sequence diagrams are good at showing, which objects communicate with which other objects and what messages trigger those communications. Sequence diagrams are not intended for showing complex procedural logic.

##### Formalism



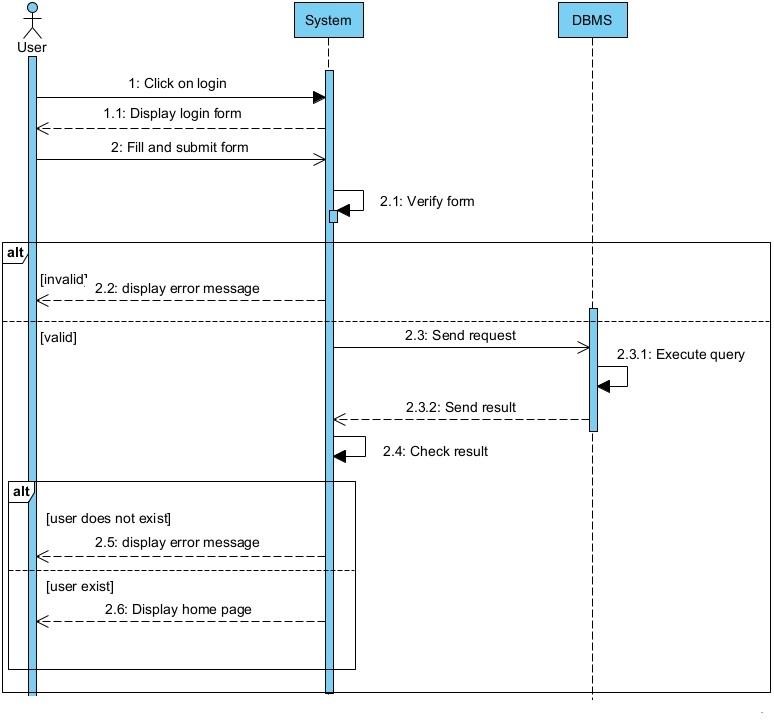
*Figure 14 Formalism of a sequence diagram*

##### Component of a sequence diagram

*Table 13: components of a sequence diagram*

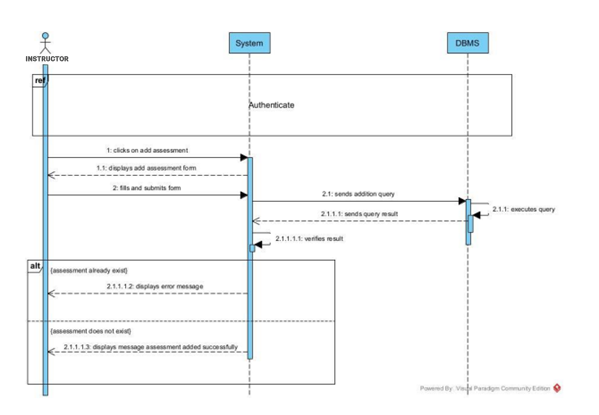
|  |  |  |
| --- | --- | --- |
| **Element** | **Representation** | **Description** |
| **Lifeline** |  | An individual participant in a sequence diagram, it is position at the top of the diagram. |
| **Combined fragment** |  | It represents a choice of behavior in which at most one operand will be chosen. |
| **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. |

**i. sequence diagram authenticate**



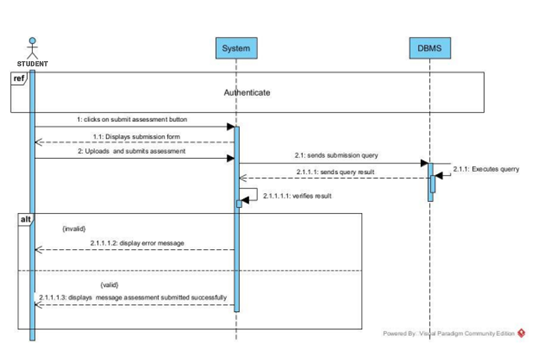
*Figure 15:Sequence diagram authenticate*

**ii. add assessment sequence diagram**



*Figure 16: add assessment sequence diagram*

***iii. Submit assessment sequence diagram***

**

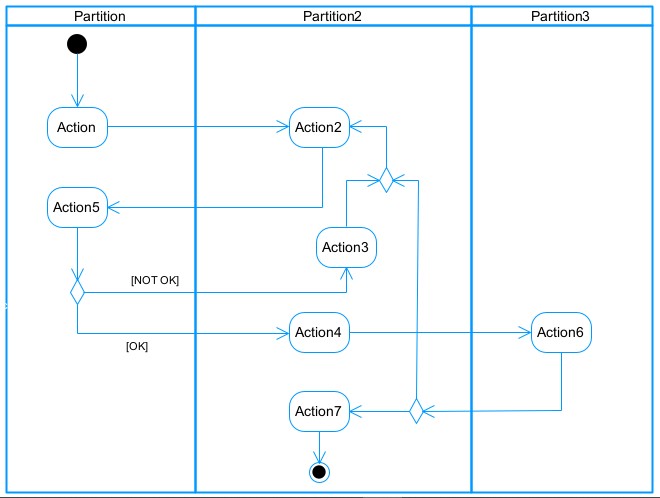
*Figure 17: submit assessment sequence diagram*

#### Activity diagrams

##### Definition

An activity diagram is a graphical representation of workflows 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.

##### Formalism



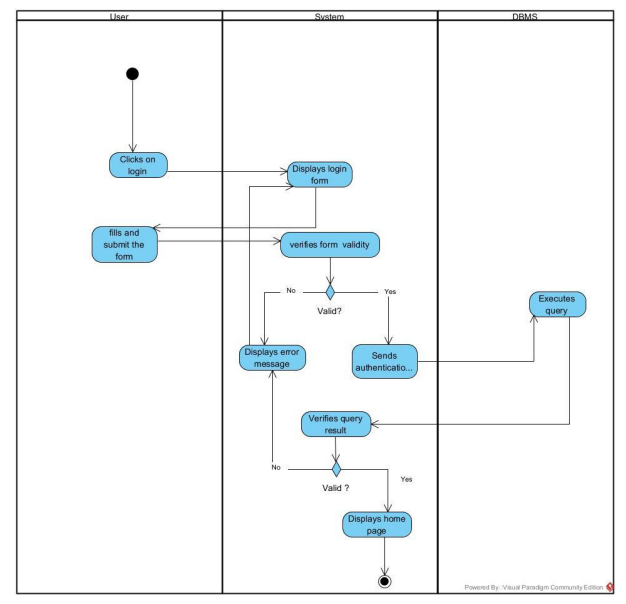
*Figure 18 Formalism of an activity diagram*

##### Components of an activity diagram

*Table 14: Components of an activity diagram*

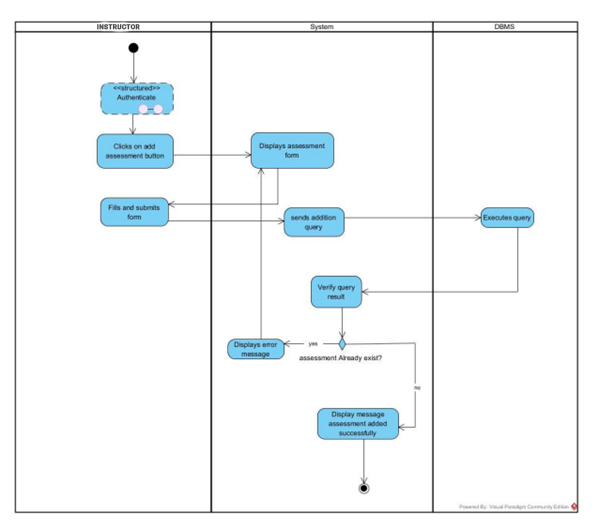
|  |  |  |
| --- | --- | --- |
| **Elements Diagrammatic Representation** | | **Description** |
| **Activity** |  | Use to represent a set of actions. |
| **Action** |  | Represent a task to be performed. |
| **Activity edge** |  | A directed connection between two activity nodes through which tokens 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 behaviour 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. |

**i. create account activity diagram**



*Figure 19: create account activity diagram*

***ii. add assessment activity diagram***

**

***Figure 20: add assessment activity diagram***

**iii. submit assessment activity diagram**



*Figure 21: submit assessment activity diagram*

# 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 workflow 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 FIVE: 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 Overview**

INTRODUCTION

1. TECHNICAL BRANCH
2. IMPLEMENTATION BRANCH

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 modelled 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.

## TECHNICAL BRANCH

### A. Generic Design

The generic conception consists of developing a solution that response to the technical specification. This conception is independent of the functional aspect specified in the functional branch of the 2TUP.

### a. Physical architecture

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

* Presentation layer (your PC, Tablet, mobile, etc.)
* Application layer (server)
* Database Server

#### Logical architecture

We will be using the Model View Controller (MVC22). The MVC is a design pattern for developing application that separate the application/business logic from the rest of the user interface, thus making it easier to solve problems related to object-Oriented programming (OOP29). MVC separates the application in three parts as shown in the figure below.

##### Model

Model is when the application’s data objects are stored. The model doesn’t know anything about views and controllers. When a model changes, typically it will notify its observer that a change has occurred.

##### View

The view is what is presented to the users and how users interact with the application. It the means the presentation of data in a particular format, triggered by a controller’s decision to present the data. They are script-based systems like JSP, PHP and very easy to integrate with AJAX technology.

##### Controller

The controller is responsible for responding to the user input and perform interactions on the data model 23 objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

## IMPLEMENTATION BRANCH

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

### A. Preliminary Design

#### Class diagram

##### a. 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. It purpose is to model the static view of an application.

##### b. 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: Formalism of a class diagram*

***c.******components of a class diagram***

*Table 15: components of a class diagram*

|  |  |  |
| --- | --- | --- |
| 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. |

**System class diagram**



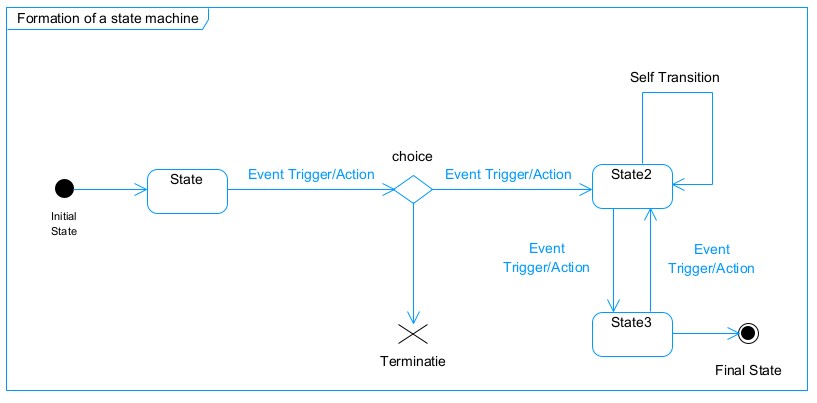
*Figure 23: system class diagram*

#### State machine diagram

##### a. 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 chart diagram, it models the dynamic flow of control from the state of a particular object within a system.

**b. Formalism**



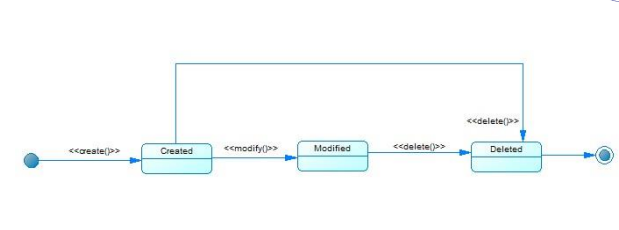
*Figure 24: state machine diagram formalism*

##### c. Component of a state machine diagram

*Table 16: component of a state machine diagram*

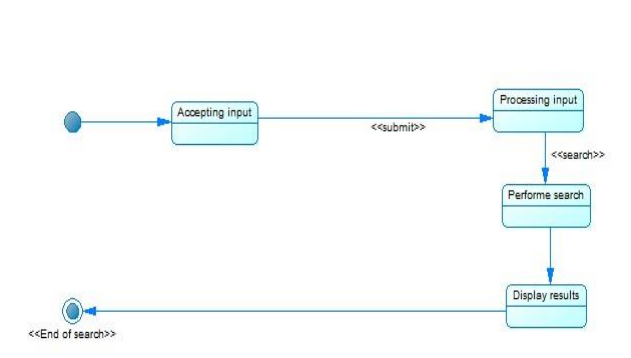
|  |  |  |
| --- | --- | --- |
| **Element** | **Diagram Relationship** | **Description** |
| **State** |  | Models a situation during which a certain invariant condition holds. |
| **First (Initial) State** |  | It represents a default vertex, that is, a source for a single transaction to the default 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. |

a**. manage student state machine diagram**



*Figure 25: manage student state machine diagram*

**b. search state machine diagram**



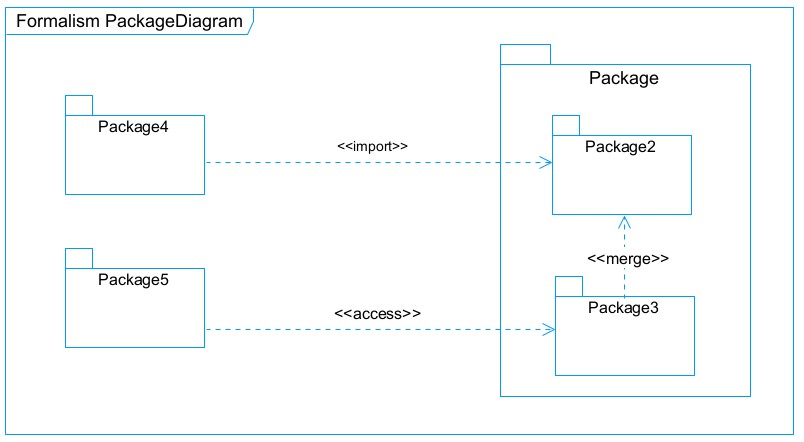
*Figure 26: search state machine diagram*

#### Package diagram

##### 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.

##### Formalism

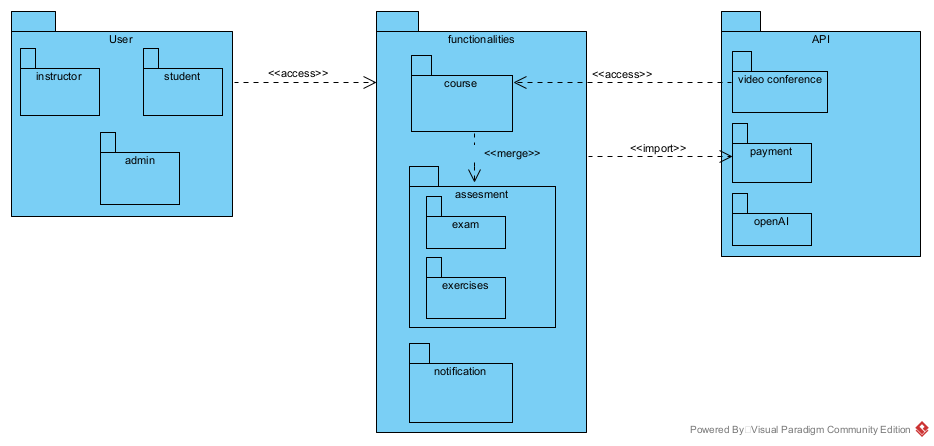
*Figure 27: Package diagram formalism*

##### Components of a package diagram

*Table 17: components of a package diagram*

|  |  |  |
| --- | --- | --- |
| **NAME** | **Representation** | **Description** |
| **Package** |  | A package is a namespace use to group related elements; it is a mechanism used to group elements into a better structure in a system. |
| **Package import** |  | 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 28:system package diagram*

# PART SIX: 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.

**Context**

INTRODUCTION

1. THE ENTITY RELATIONAL DIAGRAM
2. PRESENTATION OF THE DEVELOPMENT TOOLS

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.

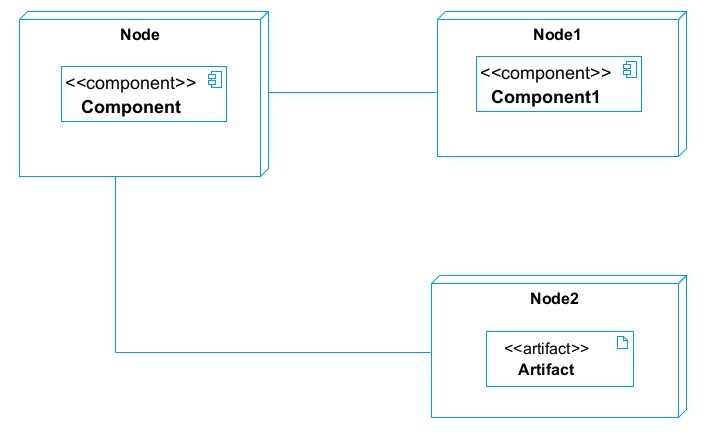
## ENTITY RELATIONAL DIAGRAMS

#### Deployment Diagram

##### Definition

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 deployed using the deployment diagram. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

##### Formalism



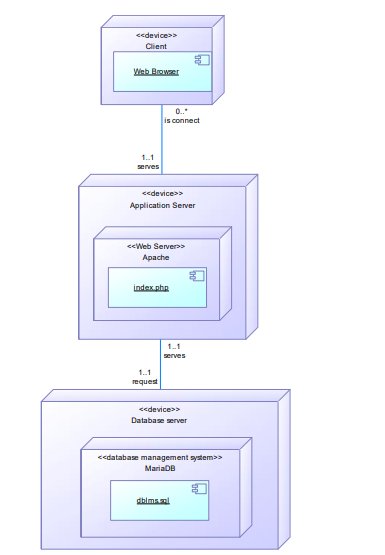
*Figure 29: deployment diagram formalism*

**c. components of a deployment diagram**

*Table 18: components of a deployment diagram*

|  |  |  |
| --- | --- | --- |
| 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 |

# Deployment diagram



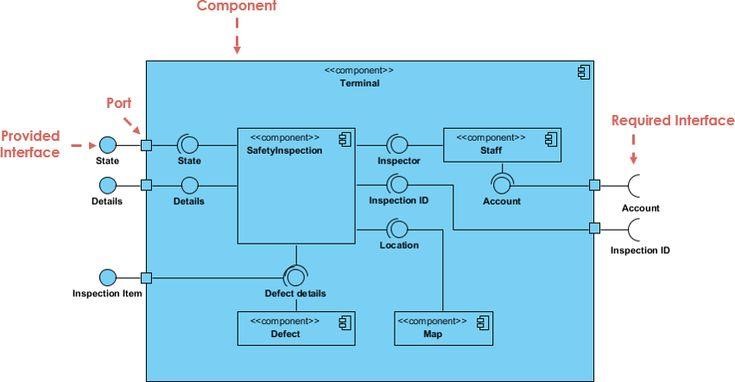
*Figure 30: deployment diagram*

#### Component diagram

##### 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 not describe the functionality of the system but it describes the components used to make those functionalities.

##### Formalism



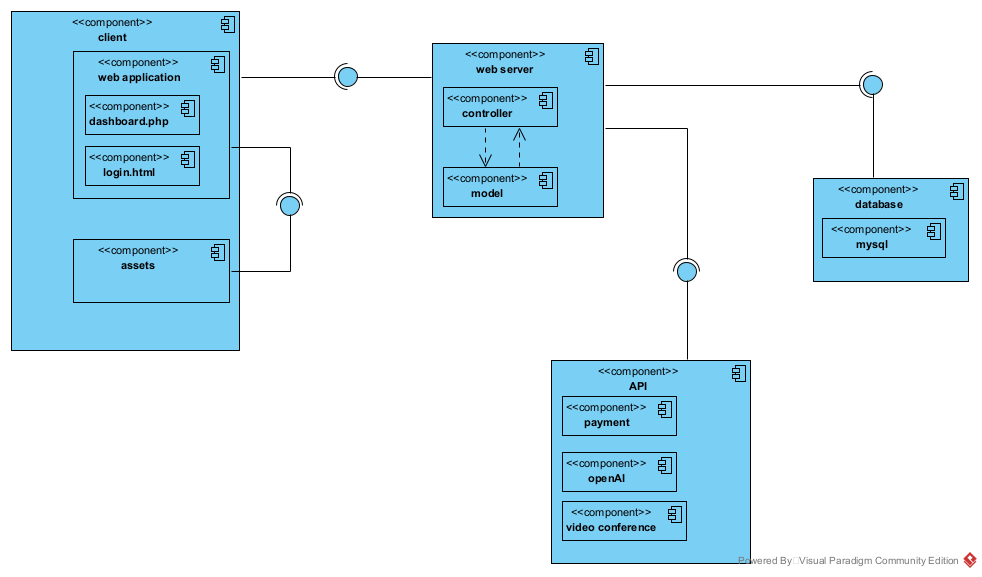
*Figure 31: Formalism of a component diagram*

##### Components of a component diagram

*Table 19: Component of a component diagram*

|  |  |  |
| --- | --- | --- |
| **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. |

**System component diagram**



*Figure 32: system component diagram*

## PRESENTATION OF DEVELOPMENT TOOLS

### Material resources

A development tool is a hardware or a software that supports the rapid implementation of software application. That is used to create, debug, maintain, or otherwise support other programs and applications.

*Table 20: material resources*

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | Parameters | Specification | Quantity |
| Computer(Lenovo ideapad) | RAM | 8GB | 1 |
| ROM | 128GB SSD |
| Graphic card | AMD RADEON(2GB dedicated)  Intel(128MB dedicated) |
| processor | Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.60 GHz |
|  |  |
|  |  |

### TECHNOLOGICAL STACKS

*Table 21: technological stacks*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resource** | **Version** | **License Owner** | **Function** |  |  |
| **Microsoft Visual Studio Code** | 1.60 | Microsoft Corporation | Text editing coding | and |  |
| **Xampp** | 7.3.10 | Apache | Web server |  |  |
| **PowerAMC** | 15.1.0.2850 | Sybase | Design UML diagrams |  |  |
| **Visual pradigm** | 17.2 | Visual paradigm community | Design UML diagrams |  |  |
|  |  |  |  |  |  |
| **Google chrome** | 129.0.66688.90 | Google | To do research |  |  |
| **Windows** | 10 | microsoft | Operating system |  |  |
| **youtube** |  | Google | Watch tutorials |  |  |
| **Gantt Project** | 3 | GNU General public | To design the Gantt chart |  |  |
| **PHP** | 7.3.10 | Apache | For the backend |  |  |
| **html** | 5 | whatwg | For the creation of web pages |  |  |
| **CSS** | 3 | CSS working group | For the styling and layout of the web pages |  |  |
| **MySQL** | 8.0.30 | Apache | For database management |  |  |

# CONCLUSION

In our realization phase, we implemented our application. In other to accomplish this phase, we made used of our analysis and conception phase. The Entity-Relational diagram. helped us to establish our database. We then presented the development tools we used for the development or realization of our system, we had material resource (visual studio code) and technology. We will move to the user guide phase where we shall elaborate on how to install and use **EasyBosh**.

# PART SEVEN: 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. Content Overview

INTRODUCTION

I. DEPLOYMENT OF THE WEB APPLICATION

II. SHOW CASE

CONCLUSION

# INTRODUCTION

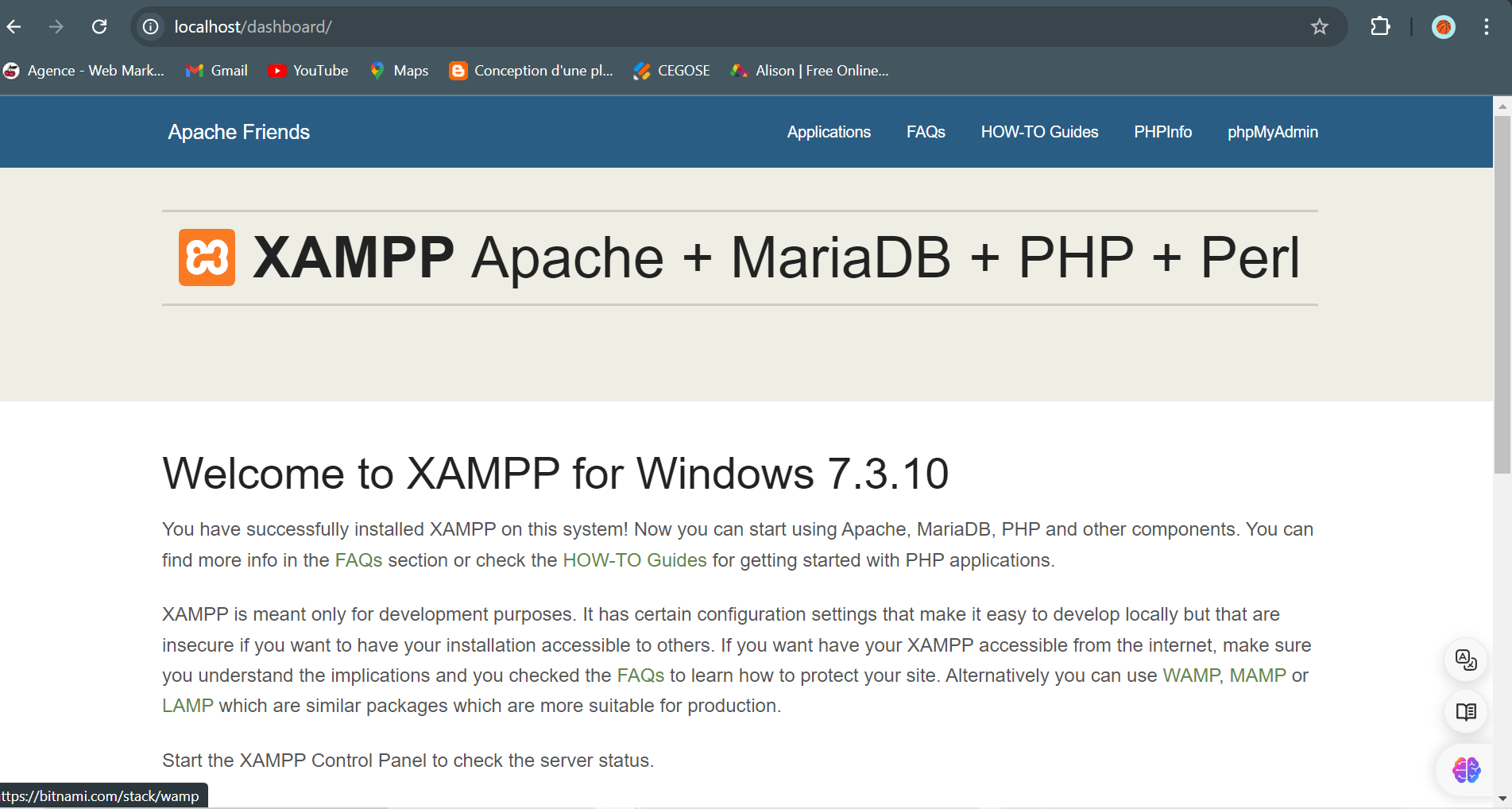
This is the final phase of our report. In this phase we will walk through the requirements for our system, the necessary installation process, accessing our system and it futures, all this in a step-by-step manner to facilitate the setting up of the platform for the first time users. The steps of different processes will be accompanied by images. After we will showcase our application by viewing, the different screens and some brief explanation.

# I. DEPLOYMENT OF THE WEB APPLICATION

**A. Database importation**

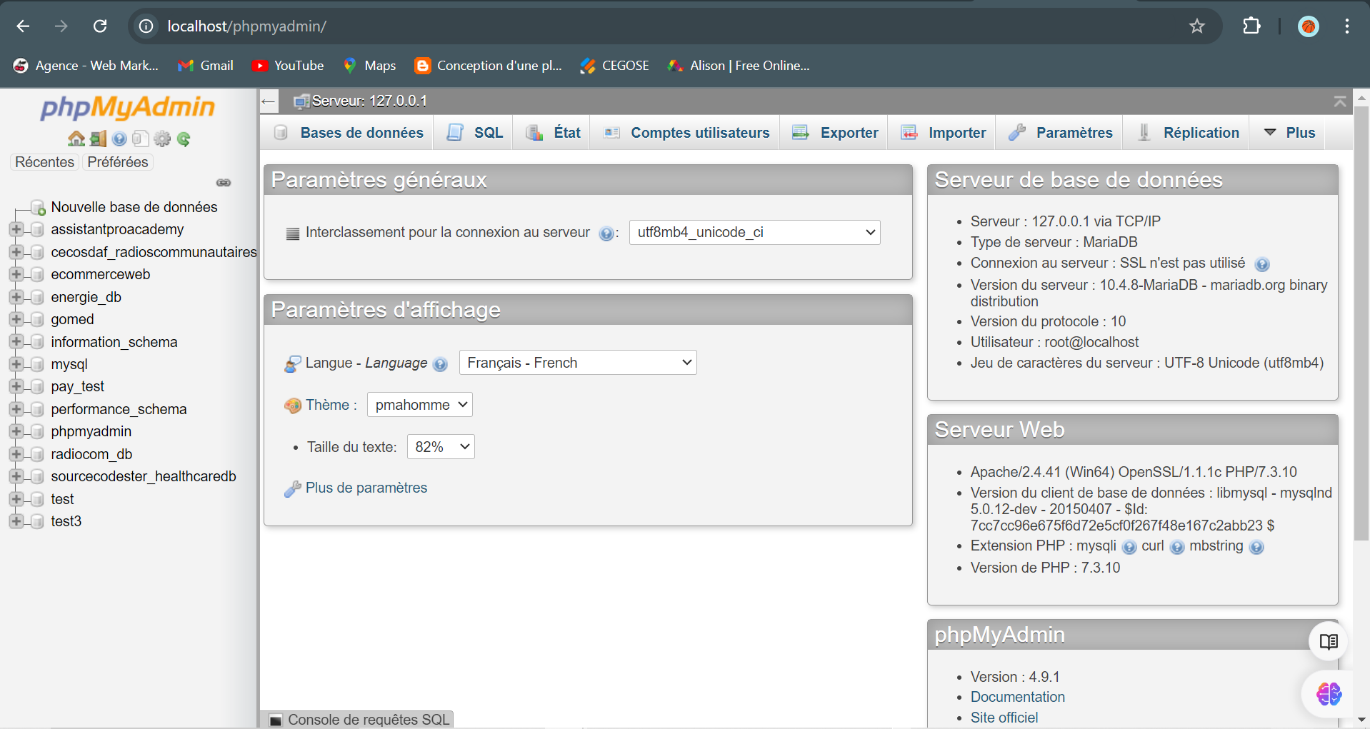
to deploy the software, we need to import the database on our server. In our case, we used Xampp.

1. **launching the server**



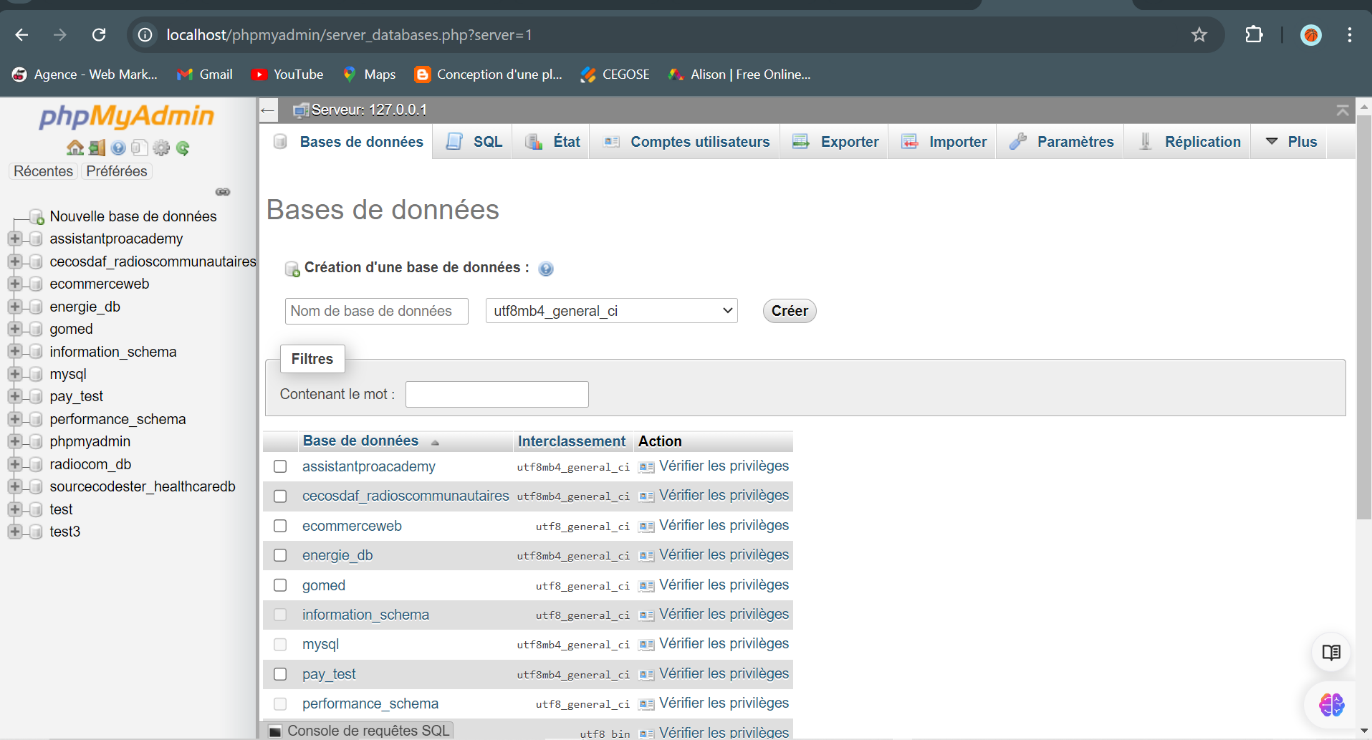
*Figure 33: launching the server*

1. **opening the database manager**

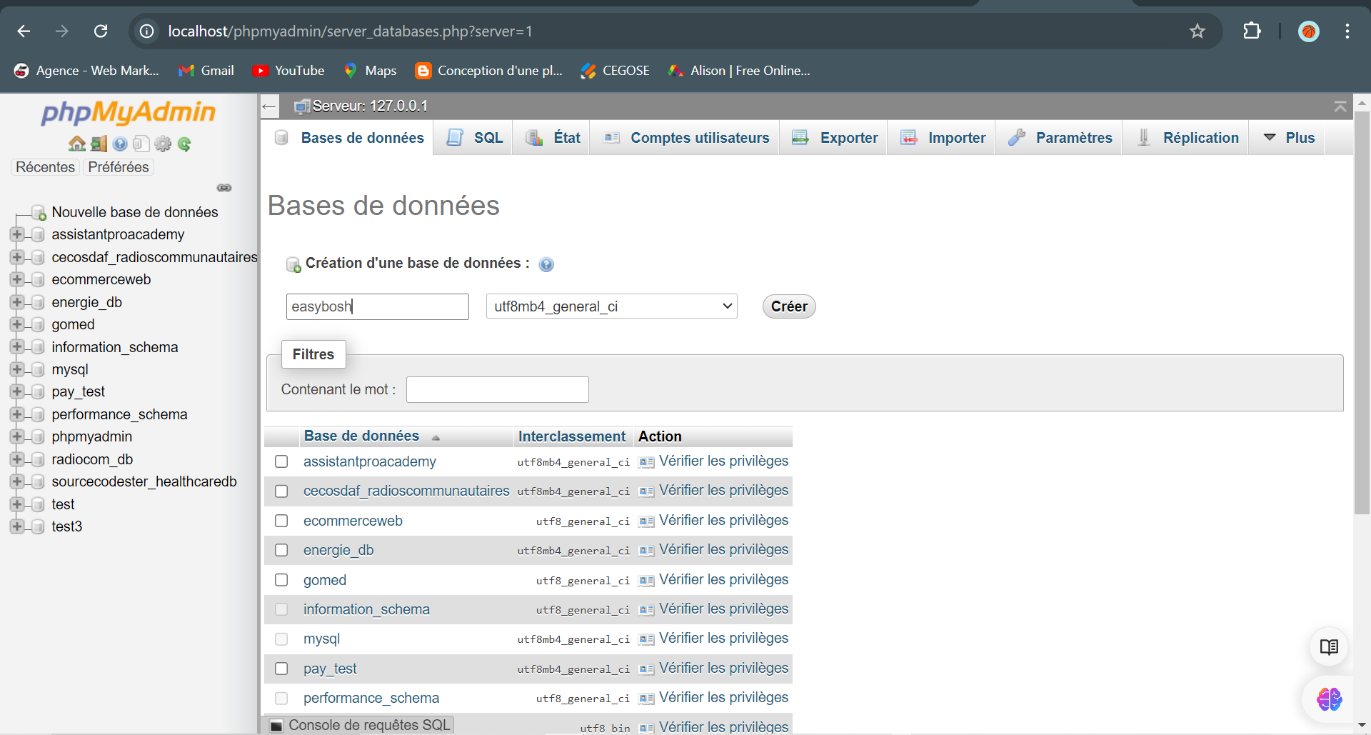


*Figure 34: open database manager*

1. **creating the database**

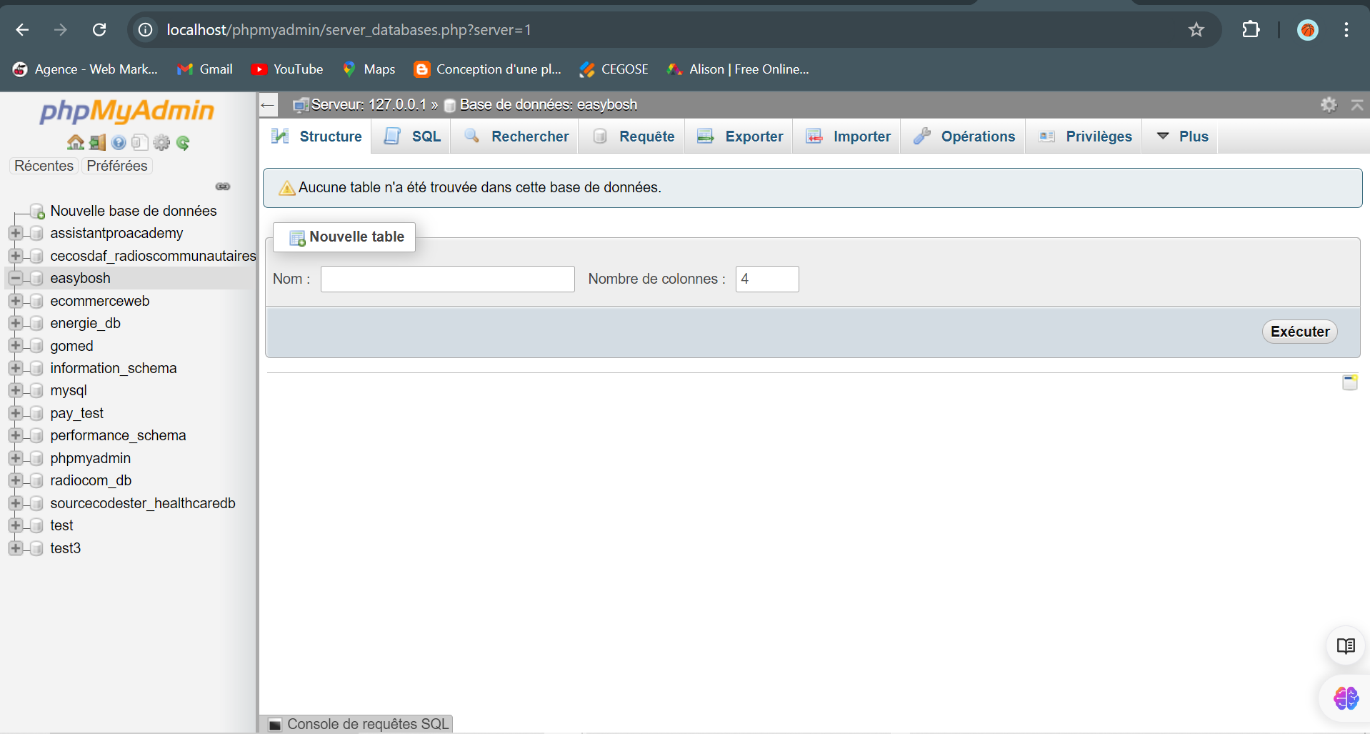


*Figure 35: create database*

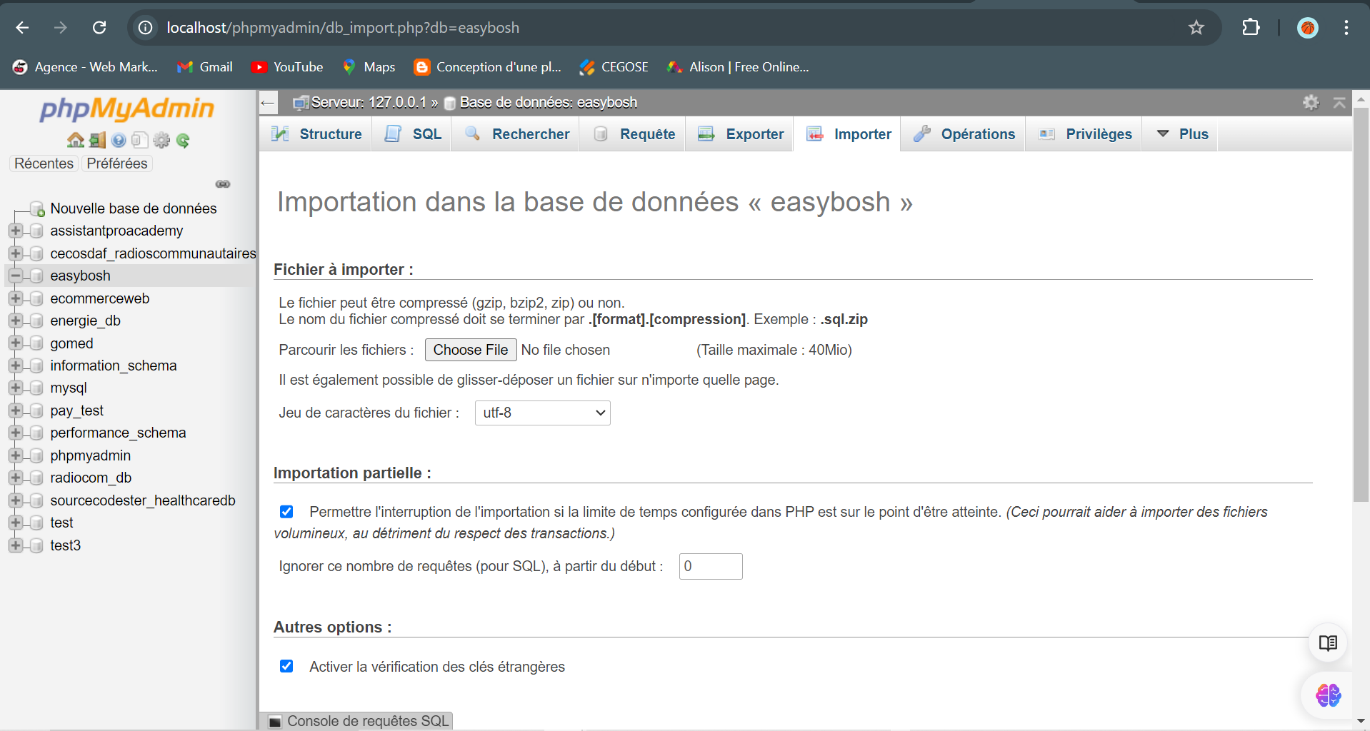


*Figure 36: create app database*

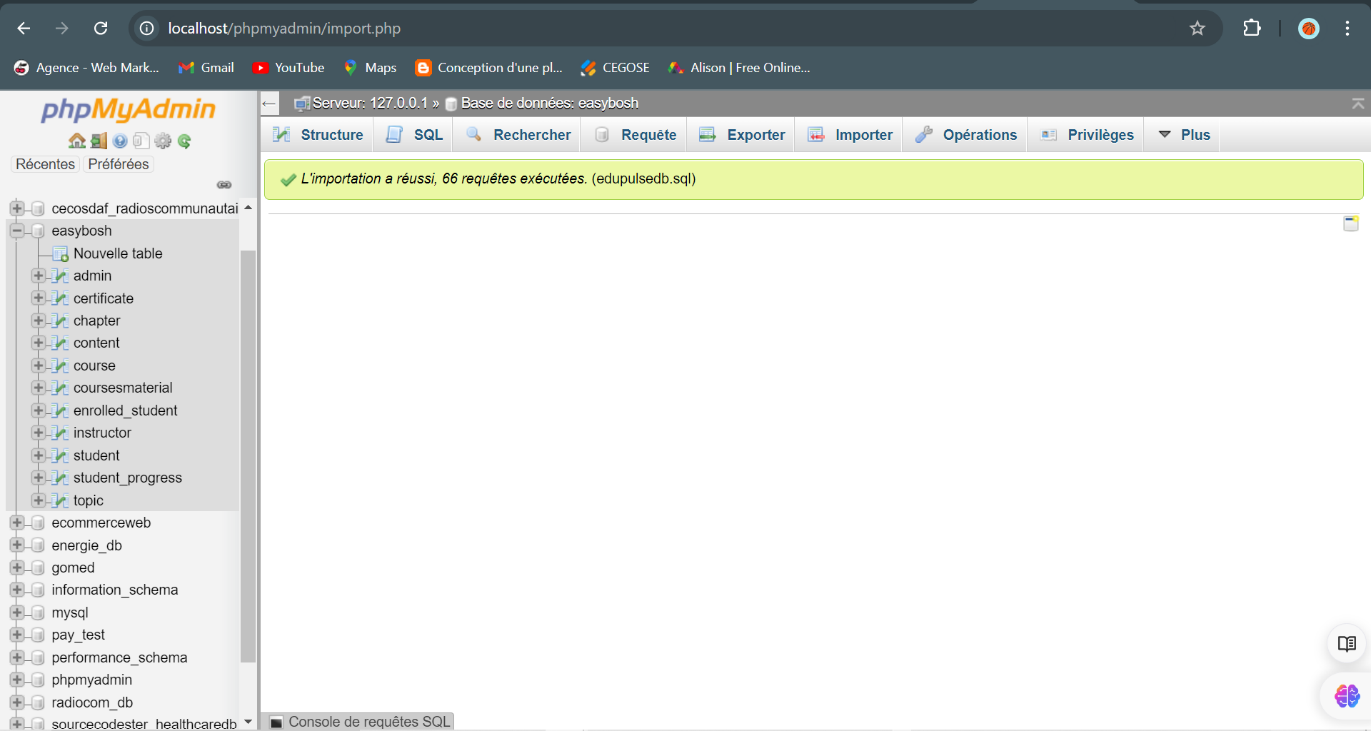
1. **import the database structure**



*Figure 37: import database structure*



*Figure 38: importation of database*



*Figure 39: successful importation*

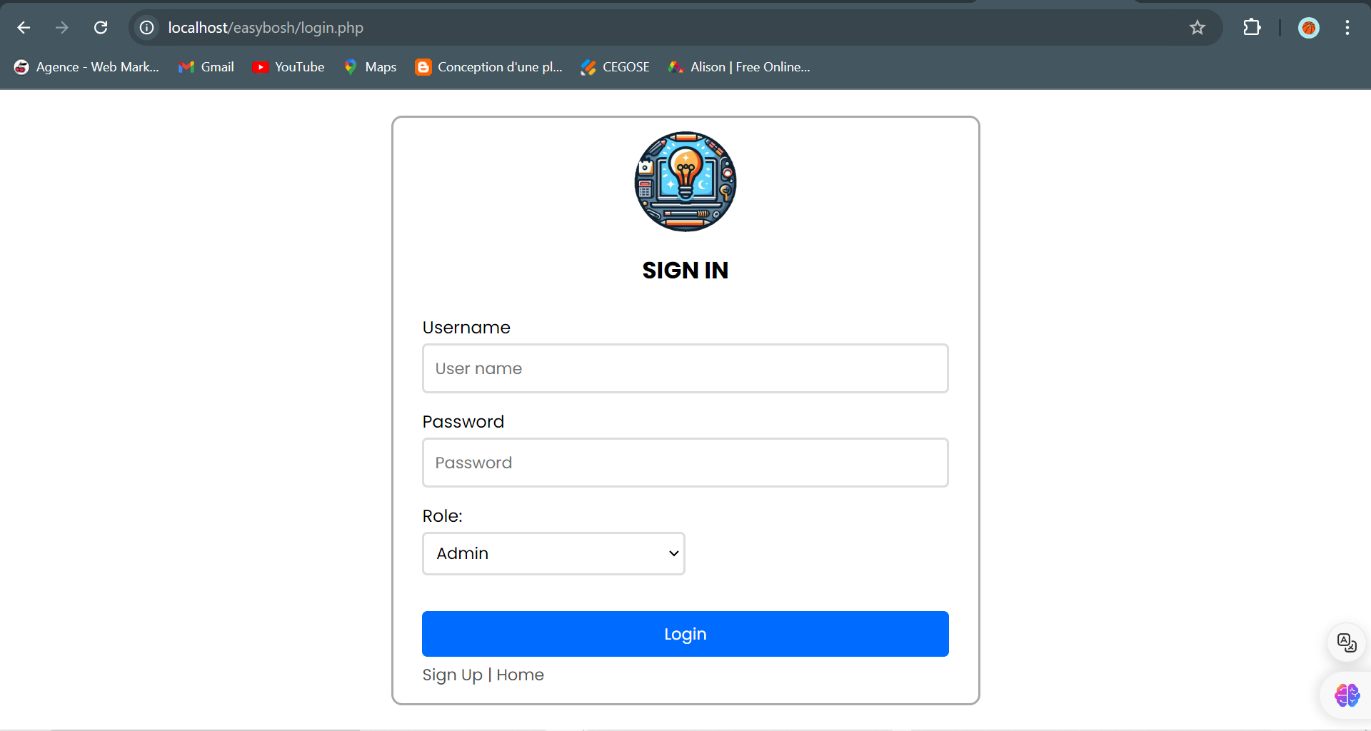
1. **launching the application**



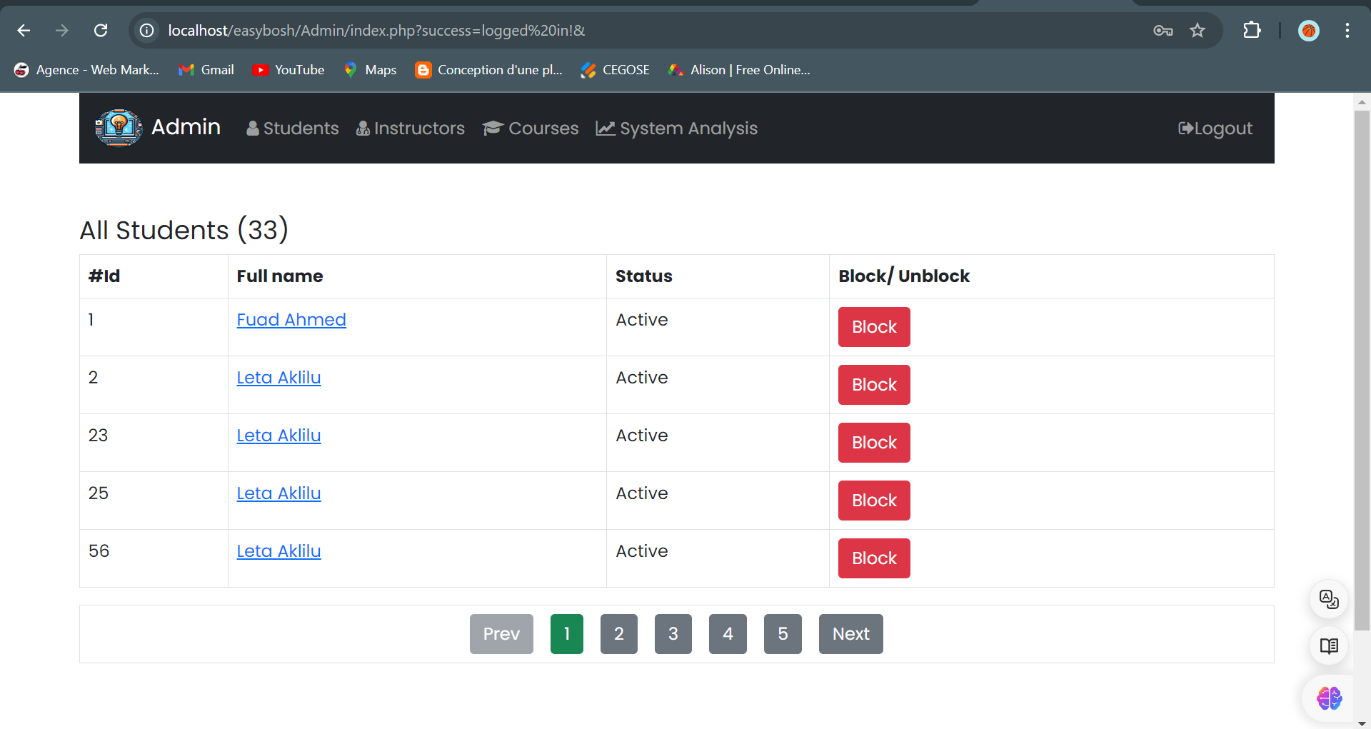
*Figure 40: launching the app*

# II. SHOWCASE

1. **logging in as an admin**



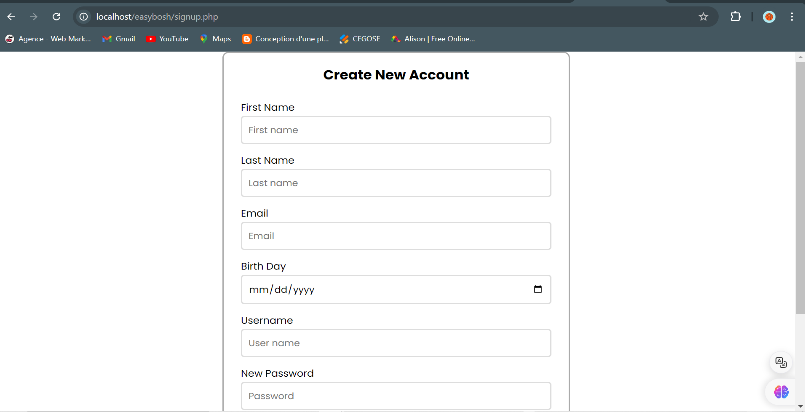
*Figure 41: sign in as admin*



*Figure 42: admin dashboard*

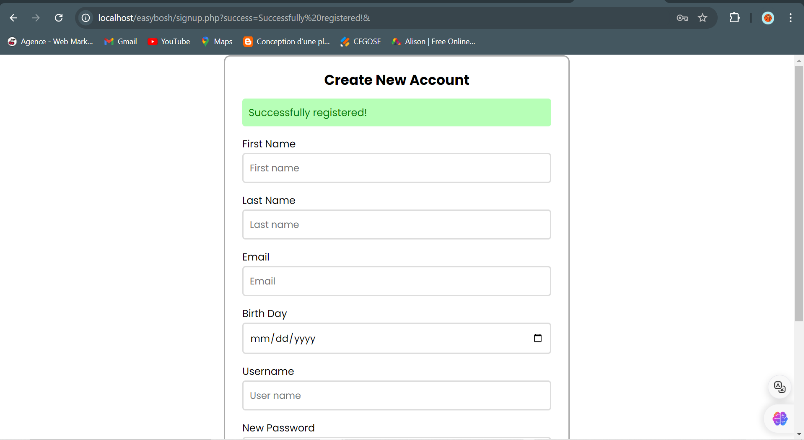
1. **create account**

* Step1: fill form

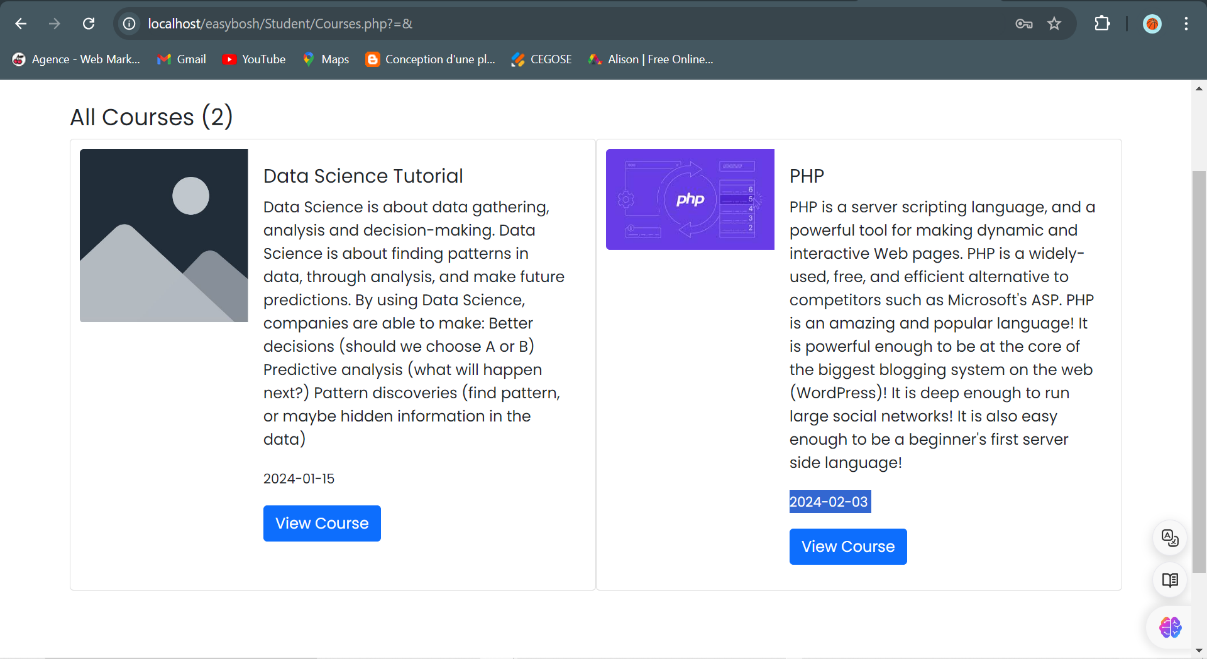


*Figure 43: fill form*

* Step2:



*Figure 44: account created*



*Figure 45: course display*

# CONCLUSION

Having put in place the application, it was not sufficient we had to produce a manual that will help its various users. That is why we presented the way the application was deployed and all the functionalities of this application. It also guides the users on how to use the application.

# PERSPECTIVES

* **Implementation of an AI:** The AI will permit the students to ask questions; after every class, the student may have supplementary questions but the instructor may not be available. The AI will answer their questions.
* **Implementation of the video conference:** Every instructor has a period to chat with the students virtually by a video conference, for a specific time. This API will enable their virtual class to occur.
* **Implementation of a payment API:** Each course in the system has a monetary value depending on the chosen course. Using OM, MoMo or even PayPal to receive the amount sent
* **Implementation of a quiz game:** To make the learning process more interesting, some games maybe implemented; these games are related to school so as not to forget the objectives of the platform
* **Implementation of rewards and badges:** This is to encourage the students to study; the more you answer questions, the more you earn rewards and badges.

# GENERAL CONCLUSION

Having met the end of our project and internship period, we can affirm with certainty that the internship was a very beneficial experience to us. The challenges faced, problem solved and knowledge acquired in the professional milieu have given us a taste of life as professional Software Engineer in a company. This experience will with no doubt enable us to face the upcoming challenges in the professional world.

Our theme **“CONCEPTION AND DEVELOPMENT OF AN EDUCATIONAL SUPPORT AND PERSONALIZED STUDENT PROGRESS PLATFORM”** has permitted us get acquainted with the analysis, realization and deployment of software Applications. Our analysis phase was all about structure and defining the needs of the software using UML diagrams. The realization phase was more concerned with the modelling of the software’s database and the technologies used for the implementation of the software.

Our ACADEMY application is finally ready for deployment and use. We are convinced that this application will be of great use to students increasing the respect of barrier measures and reduce the manual execution of tasks.

It is known that no human realization is perfect and as such we are sorry for any errors, be it on the report itself or the software. Nevertheless, we are hopeful of a favorable outcome from the readers of this report as it represents are first step in to domain of scientific research

# ANNEXES

# BIBLIOGRAPHY

* Core PHP programming using php to build to build dynamic websites supervision by JAN H SWHARTZ 2002.
* Ivar Jackobson, Grady Boosh, James Rambaugh, the unified process of developing a software. Edition Eyrolles, 1999.
* Juliard UML Unified Modelling Language, Journal from the university of Southern Britain UFR SSI-IUP Vannes, 2001-2002.
* Ivar Jackobson, Manus Cristerson, Patrick Jonson, Gunar Overgaard, the genius software object oriented, Edition Addison Wesley, 2000.
* Joseph Gabay and David Gabay, Implementation guide and study case, Edition Dunod, Paris 2008.
* Nathalie Lopez, Jorge Migueis and Emmanuel Pichon, Integrate UML in your projects, Edition Eyrolles, 2000.
* Alexandre Durain, « UML 2.0 et le processus 2TUP »,
* Internship report realized by PAULINE SIDONIE BAMSECK of BA2B.
* Mr. MESSIO UML course AICS-Cameroon 2022-2023
* Mercurial 2023
* Mercurial 2024

# WEBOGRAPHY

* [*https://www.visual-paradigm.com/VPGallery/diagrams/Component.html*](https://www.visual-paradigm.com/VPGallery/diagrams/Component.html) This site helped us get the definition and formalism of component diagram. (visited on the 23rd July 2024 at 11:45 AM)
* *https://www.tutorialspoint.com/struts\_2/basic\_mvc \_architecture.htm:* Here, we got the explanation of the MVC architecture. (visited on the 25th July 2024 at 4:30 PM)
* *https://en.wikipedia.org/wiki/Class\_diagram* Here, we got the definition of a class diagram (visited on the 25th July 2024 at 3:30 PM)
* *https://www.uml-diagrams.org/communication-diagrams.html:* Helped with the definition of communication diagram. (visited on the 20th August 2024 at 7:45 PM)
* *https://www.smartdraw.com/uml-diagram/* In which we learned more on UML Diagrams (visited on 10th August 2024 at 10:30 AM)
* *https://www.uml-diagrams.org* We got the various definitions of each UML Diagram (visited on 02nd September 2024 at 09:30 AM)
* *https://en.wikipedia.org/wiki/Applications\_of\_UML https://www.uml* (visited on the 23rd July 2024 at 11:45 AM)
* *diagrams.org/use-case-diagrams.htmlhttps://en.wikipedia.org/wiki/Sequence\_diagram* (visited on the 29th July 2024 at 10:00 AM)
* *https://www.smartsheet.com/how-write-smart-project-objective* (visited on the 23rd July 2024 at 17:45 AM)
* UML class diagram representing data structure | Download Scientific Diagramwww.researchgate.net(visited on the 29th august 2024 at 10:35 Am

# TABLE OF CONTENT

[DEDICATION i](#_Toc179099520)

[ACKNOWLEDGEMENT ii](#_Toc179099521)

[CONTENT iii](#_Toc179099522)

[GLOSSARY v](#_Toc179099523)

[LIST OF FIGURES vi](#_Toc179099524)

[LIST OF TABLES viii](#_Toc179099525)

[ABSTRACT ix](#_Toc179099526)

[RESUME x](#_Toc179099527)

[GENERAL INTRODUCTION 1](#_Toc179099528)

[PART ONE: INSERTION PHASE 2](#_Toc179099529)

[INTRODUCTION 4](#_Toc179099530)

[I. WELCOME AND INTEGRATION 5](#_Toc179099531)

[II. GENERAL PRESENTATION OF HB TRAIN 5](#_Toc179099532)

[III. ORGANISATION OF HB TRAIN 7](#_Toc179099533)

[I. GEOGRAPHICAL LOCATION 7](#_Toc179099534)

[II. ORGANIZATIONAL CHART OF HB Train 9](#_Toc179099535)

[III. MISSIONS AND ACTIVITIES OF HB TRAIN 10](#_Toc179099536)

[1. MISSIONS 10](#_Toc179099537)

[2. ACTIVITIES 10](#_Toc179099538)

[IV. HARDWARE AND SOFTWARE RESOURCES OF HB TRAIN 11](#_Toc179099539)

[1. Material Resources 11](#_Toc179099540)

[2. Software Resources 12](#_Toc179099541)

[CONCLUSION 13](#_Toc179099542)

[PART TWO: TECHNICAL PHASE 14](#_Toc179099543)

[INTRODUCTION 16](#_Toc179099544)

[I. PRESENTATION OF THEME 16](#_Toc179099545)

[II. DESCRIPTION OF EXISTING SYSTEM 16](#_Toc179099546)

[III. CRITICISM OF THE EXISTING SYSTEM 17](#_Toc179099547)

[IV. PROBLEMATIC 17](#_Toc179099548)

[V. PROPOSED SOLUTION 18](#_Toc179099549)

[CONCLUSION 19](#_Toc179099550)

[P 20](#_Toc179099551)

[PART THREE: SPECIFIACTION BOOK 20](#_Toc179099552)

[INTRODUCTION 22](#_Toc179099553)

[I. CONTEXT AND JUSTIFIACTION 23](#_Toc179099554)

[A. Context 23](#_Toc179099555)

[B. JUSTIFICATION 23](#_Toc179099556)

[II. Objective of the project 24](#_Toc179099557)

[A. General Objectives 24](#_Toc179099558)

[B. Specific Objectives 24](#_Toc179099559)

[III. Expression of needs 25](#_Toc179099560)

[A. Functional Needs 25](#_Toc179099561)

[B. Non-functional needs 26](#_Toc179099562)

[IV. ESTIMATED COST OF THE PROJECT 27](#_Toc179099563)

[A. Hardware Resources 27](#_Toc179099564)

[B. Software Resources 28](#_Toc179099565)

[C. Human Resources 29](#_Toc179099566)

[D. Total Project Estimated Cost 29](#_Toc179099567)

[V. PROJECT PLANING 30](#_Toc179099568)

[A. Chronogram of activities 30](#_Toc179099569)

[B. GANTT CHART 31](#_Toc179099570)

[VI. LIST OF PARTICIPANTS DELIVERABLES 32](#_Toc179099571)

[A. List of participants 32](#_Toc179099572)

[B. Deliverables 32](#_Toc179099573)

[CONCLUSION 33](#_Toc179099574)

[PART FOUR: ANALYSIS PHASE 34](#_Toc179099575)

[INTRODUCTION 36](#_Toc179099576)

[I. PRESENTATION OF THE ANALYSIS METHODS 37](#_Toc179099577)

[ Merise: 37](#_Toc179099578)

[ Agile: 37](#_Toc179099579)

[ Scrum: 37](#_Toc179099580)

[ APF: 37](#_Toc179099581)

[ XP: 37](#_Toc179099582)

[A. MODELING WITH UML 2.5 38](#_Toc179099583)

[a. Structural Diagrams: 38](#_Toc179099584)

[b. Behavioral Diagrams: 39](#_Toc179099585)

[B. UML 2.5 diagrams overview 39](#_Toc179099586)

[C. The Process of Development of a Software 40](#_Toc179099587)

[D. A Unified Process 40](#_Toc179099588)

[E. The Two Track Unified Process (2TUP) 40](#_Toc179099589)

[a. The left branch (functional branch) 42](#_Toc179099590)

[b. The middle branch (Realization or implementation branch) 42](#_Toc179099591)

[II. JUSTIFICATION OF THE ANALYSIS METHOD 43](#_Toc179099592)

[III. MODELLING OF THE PROPOSED SOLUTION 44](#_Toc179099593)

[A. Capture of the functional needs 44](#_Toc179099594)

[1. Use Case Diagram 44](#_Toc179099595)

[2. Communication Diagram 51](#_Toc179099596)

[3. Sequence Diagram 54](#_Toc179099597)

[4. Activity diagrams 59](#_Toc179099598)

[CONCLUSION 65](#_Toc179099599)

[PART FIVE: CONCEPTION PHASE 66](#_Toc179099600)

[INTRODUCTION 68](#_Toc179099601)

[I. TECHNICAL BRANCH 69](#_Toc179099602)

[A. Generic Design 69](#_Toc179099603)

[a. Physical architecture 69](#_Toc179099604)

[Logical architecture 69](#_Toc179099605)

[II. IMPLEMENTATION BRANCH 71](#_Toc179099606)

[A. Preliminary Design 71](#_Toc179099607)

[1) Class diagram 71](#_Toc179099608)

[2) State machine diagram 73](#_Toc179099609)

[3) Package diagram 76](#_Toc179099610)

[PART SIX: REALIZATION PHASE 80](#_Toc179099611)

[INTRODUCTION 82](#_Toc179099612)

[I. ENTITY RELATIONAL DIAGRAMS 83](#_Toc179099613)

[1. Deployment Diagram 83](#_Toc179099614)

[Deployment diagram 85](#_Toc179099615)

[5. Component diagram 86](#_Toc179099616)

[II. PRESENTATION OF DEVELOPMENT TOOLS 89](#_Toc179099617)

[A. Material resources 89](#_Toc179099618)

[B. TECHNOLOGICAL STACKS 89](#_Toc179099619)

[CONCLUSION 92](#_Toc179099620)

[PART SEVEN: USER GUIDE 93](#_Toc179099621)

[INTRODUCTION 95](#_Toc179099622)

[I. DEPLOYMENT OF THE WEB APPLICATION 95](#_Toc179099623)

[II. SHOWCASE 99](#_Toc179099624)

[CONCLUSION 102](#_Toc179099625)

[PERSPECTIVES 103](#_Toc179099626)

[GENERAL CONCLUSION 104](#_Toc179099627)

[ANNEXES 105](#_Toc179099628)

[BIBLIOGRAPHY 106](#_Toc179099629)

[WEBOGRAPHY 107](#_Toc179099630)

[TABLE OF CONTENT 108](#_Toc179099631)