**REPUBLIC OF CAMEROON**

Peace - Work - Fatherland

**REPUBLIQUE DU CAMEROUN**

Paix – Travail - Patrie





**AFRICAN INSTITUTE OF COMPUTER SCIENCE**

**CAMEROON OFFICE, PAUL-BIYA TECHNOLOGICAL CENTER OF EXCELLENCE**

P.O Box: 13719 Yaoundé

Tel: +237.242.729.957 Fax: 22729958

E-mail: [contact@iaicaameroun.com](mailto:contact@iaicaameroun.com)

Web site : www.iaicameroun.com

**One Love, One Mind, One Legacy!**

Tel 678 358 509 / 692 199 722

Site : www.shaderl.com

Email: info@shaderl.com

**Professional Supervisor**

**MR. EPOUPA PELAYAH**

**Software Engineer at SHADERL**

**Academic Year 2024 – 2025**

**Academic Supervisor.**

**MR. ASANE DERICK**

**Lecturer at AICS**

**INTERNSHIP REPORT**

**THEME :**

**DESIGN AND IMPLEMENTATION OF A SMART SERVICE FINDER APPLICATION FOR CAMEROONIANS**

**:**

Internship carried out from the 1st of July to the 30th September 2025 in. In view of obtaining **an Engineering Diploma** in Computer Science Option: **Software Engineering**

Written by:

**WIRBA JORDAN WIRBA**

Level III Software Engineering student

**Supervisors.**

# **DEDICATION**

**THIS BOOK IS DEDICATED**

**TO THE ALMIGHTY GOD**

# **ACKNOWLEDGEMENTS**

This work would not have been accomplished without the help, encouragements, advices, teachings and precious time of the following people:

* We give thanks to the almighty God who made it possible without whom nothing would have been done.
* The Resident Representative of the African Institute of Computer Sciences, Mr. Armand Claude ABANDA.
* The chief executive officer of SHADERL, Mr. ASANE DERICK and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Mr. EPOUPA PELAYAH our Professional supervisor for his availability, advices and moral support throughout the internship period which lead to the realization of this project.
* Our Academic supervisor Mr. ASANE DERICK for his availability, advices and moral support throughout the internship period which lead to the realization of this project.
* Mr. AGBOR Anderson lecturer at AICS-Cameroon for his Inspirational pieces of advices, encouragements and moral support.
* To all our teachers of the African Institute of Computer Sciences for their support throughout our training.
* To Mr. NDENGE, Mr. EKITI, for their advices and assistance in realizing this document.
* The amazing family at SHADERL for their endless love and support during this internship period.
* To my family most especially to my parents for their guidance and support without which we could not attain the end of this work.
* To all our classmates and friends at AICS, for their knowledge, ideas, collaboration and most importantly the unforgettable moments spent together.

# **CONTEN**

# **LIST OF FIGURES**

[Figure 1 : functional organization of SHADERL 9](#_Toc209464858)

[Figure 2: Location of SHADERL 10](#_Toc209464859)

[Figure 3: Gantt project 27](#_Toc209464860)

[*Figure 4:UML diagrams* 39](#_Toc209464861)

[*Figure 5:Branches of 2TUP* 41](#_Toc209464862)

[Figure 6:Use case diagram formalism 45](#_Toc209464863)

[Figure 7: General use case Diagram 47](#_Toc209464864)

[Figure 8: Book service specific use case Diagram 48](#_Toc209464865)

[Figure 9:Manage Booking specific use case diagram 48](#_Toc209464866)

[Figure 10: Formalism of communication diagram 51](#_Toc209464867)

[Figure 11:Communication Diagram for Authenticate 53](#_Toc209464868)

[Figure 12: Communication Diagram to Verify Service provider 53](#_Toc209464869)

[Figure 13:Formalism of sequence diagram 54](#_Toc209464870)

[Figure 14:Sequence diagram for Authentication 56](#_Toc209464871)

[Figure 15: sequence Diagram for Book service 57](#_Toc209464872)

[Figure 16: Sequence Diagram for update service status 58](#_Toc209464873)

[Figure 17 : Sequence Diagram for verify service provider documentation 59](#_Toc209464874)

[Figure 18:Formalism of activity diagram 60](#_Toc209464875)

[Figure 19:Activity diagram for book service 63](#_Toc209464876)

[Figure 20:Activity Diagram for update service status by service provider 64](#_Toc209464877)

[Figure 21 : Admin verify Service provider activity Diagram 65](#_Toc209464878)

[Figure 22:Formalism of class diagram 69](#_Toc209464879)

[*Figure 23:Components of a class diagram* 70](#_Toc209464880)

[*Figure 24:Class diagram of our system* 71](#_Toc209464881)

[Figure 25:State machine Diagram for booking 73](#_Toc209464882)

[Figure 26:State machine Diagram for payment 73](#_Toc209464883)

[*Figure 27:Formalism of Package diagram* 74](#_Toc209464884)

[*Figure 28:Components of a package diagram* 74](#_Toc209464885)

[Figure 29:Package Diagram 75](#_Toc209464886)

[Figure 30:System deployment Diagram 80](#_Toc209464887)

[*Figure 31:Formalism of Component diagram* 81](#_Toc209464888)

[Figure 32: Web component Diagram 83](#_Toc209464889)

[*Figure 33:The N-tier architecture* 84](#_Toc209464890)

[*Figure 34:MVC design pattern* 85](#_Toc209464891)

# **LIST OF TABLES**

[Table 1 : Limits, consequences and solutions of existing system 17](#_Toc209464892)

[Table 2: Time Schedule 26](#_Toc209464893)

[Table 3: Actors of the project 28](#_Toc209464894)

[Table 4 : Hardware resources 29](#_Toc209464895)

[Table 5: Software resources 30](#_Toc209464896)

[Table 6 : Human resources 31](#_Toc209464897)

[Table 7: Total estimation of the project 31](#_Toc209464898)

[Table 8 : *UML diagrams* 38](#_Toc209464899)

[Table 9:Comparative between UML and MERISE 40](#_Toc209464900)

[Table 10: Components of a use case diagram 46](#_Toc209464901)

[Table 11: Table 11: Formalism of textual description 49](#_Toc209464902)

[Table 12: Table 12: Textual description of Authenticate 50](#_Toc209464903)

[Table 13: Components of Communication diagram 52](#_Toc209464904)

[Table 14: Components of a sequence diagram 55](#_Toc209464905)

[Table 15:: Components of activity diagram 61](#_Toc209464906)

[Table 16: Components of a state machine diagram 72](#_Toc209464907)

[Table 17: Table 16: Components of a deployment diagram 79](#_Toc209464908)

[Table 19 : Components of a component diagram 82](#_Toc209464909)

# **LIST OF ABREVIATIONS**

➢ 2TUP: Two Track Unified Process.

➢ AICS: African Institute of Computer Sciences.

➢ DBMS: Database Management System

➢ ERD: Entity Relational Diagram.

➢ GUI: Graphical User Interface.

➢ IDE: Integrated Development environment

➢ JSON: JavaScript Object Notation

➢ MVC: Model View Controller.

➢ PDM: Physical Data Model.

➢ UML: Unified Modelling Language.

# **ABSTRACT**

Humanity has not stopped evolving since its creation from agricultural revolution to industrial revolution and now towards an information and communication technology revolution. This is due to its exponential evolution in various domains such as transport, agriculture, communication, education, health, and many leading to innovations in these domains. Thus, we cannot see the future without technology. Based on this new revolution, the African Institute of Computer sciences Paul Biya Technological center of excellence turns to put Cameroon as one of the key players in this revolution by making its students practically efficient through an academic internship. This academic internship has as objective for students to obtain practical knowledge coupled to theoretical knowledge acquired academically. It is according to this perspective that we requested SHADERL Enterprise an IT cabinet, to carry out our internship. After two weeks of insertion at the latter, we were attributed the theme **“DESIGN AND IMPLEMENTATION OF A SMART SERVICE FINDER APPLICATION FOR CAMEROONIANS”.** Following our theme, we did the feasibility studies, followed by the analysis of the system using the UML methodology associated to 2TUP process, from the analysis passing through conception and realization phase. The application was developed using html, java script and PHP, some open-source APs in order to obtain a flexible, light and secure solution for the best user experience.

**Keywords:**

• API (Application Programming Interface)

• UML (Unified Modeling language)

2TUP (2 Track Unified Process)

# **RESUME**

L'humanité n'a cessé d'évoluer depuis sa création de la révolution agricole à la révolution industrielle et maintenant vers une révolution des technologies de l'information et de la communication. Cela est dû à son évolution exponentielle dans divers domaines tels que les transports, l'agriculture, la communication, l'éducation, la santé, et de nombreux conduisant à des innovations dans ces domaines. Ainsi, nous ne pouvons pas voir l'avenir sans technologie. S'appuyant sur cette nouvelle révolution, le pôle d'excellence technologique de l'Institut Africain d'Informatique Paul Biya se tourne pour placer le Cameroun comme l'un des acteurs clés de cette révolution en rendant ses étudiants pratiquement efficaces à travers un stage académique. Ce stage académique a pour objectif pour les étudiants d'acquériez connaissances pratiques couplées à des connaissances théoriques acquises académiquement. C'est dans cette optique que nous avons demandé à SHADERL un cabinet informatique, pour réaliser notre stage. Après deux semaines d'insertion au sein de cette dernière, on nous a attribué le thème **« "CONCEPTION ET MISE EN ŒUVRE D'UNE APPLICATION INTELLIGENTE DE RECHERCHE DE SERVICES POUR LES CAMEROUNAIS"** ». Suivant notre thème, nous avons réalisé les études de faisabilité, suivies de l'analyse du système en utilisant la méthodologie UML associée au procédé 2TUP, de l'analyse en passant par la phase de conception et de réalisation. L'application a été développée en utilisant html, java script et php certaines API open source afin d'obtenir une solution flexible, légère et sécurisée pour la meilleure expérience utilisateur.

**Mots clés :**

• API

• UML

• 2TUP

# **GENERAL INTRODUCTION**

Technology is advancing at an incredible rate, spanning almost every sector from business to medicine, and even higher. Due to the huge technological advancement, we now live in fast paced society where everything is gradually being digitized making life easier. As a developing country, most businesses in Cameroon strives to take advantage of the digital economy to grow their businesses there by boosting the economy as well. To achieve such goals, an enterprise needs qualified personnel having skills in computer sciences and related fields. It is in this regard that institutions like AICS Cameroon are the place to be nowadays in order to acquire such skills. Students in third year in AICS Cameroon, are required to carry out a 3 months academic internship at an enterprise in order to put in practice the theoretical knowledge they have in the professional milieu. For this reason, we applied for an internship at SHADERL, which specializes in providing IT solutions some of which include development of web and mobile application, IT consulting, IT support and innovation, graphic designing. While there, we were attributed the theme “**DESIGN AND IMPLEMENTATION OF A SMART SERVICE FINDER APPLICATION FOR CAMEROONIANS**”. To carry out this project, we will present briefly our host company in a first part entitled integration phase. Subsequently, we will enter the technical phase where we will establish the specifications, analysis, design and implementation of our project and a guide to using the implemented application and prototype will close this technical part.

* **Insertion phase:** This phase consists of presenting the enterprise in which we performed our internship as well as how we were welcomed.
* **Existing system:** This phase consists of presenting the actual system.
* **Specification book:** The specification book is the part of the document that describes exactly what is expected from us taking into consideration the time and the cost of the project. In other to proposed a suitable solution for the user.
* **Analysis phase:** It presents the analysis method chosen with presentation of all diagrams use for the analysis of the project.
* **Conception phase:** This phase enables us to model the solution based on the criticism done at the analysis phase.
* **Realization phase:** It permits to visualize the implementation of the process in an organized step by step process.
* **The installation and user manual:** It demonstrates how the system is being used**.**

# **PART 1: INSERTION PHASE**

Preamble

This section of the report will cover details of how we were welcomed in the host company, presentation, organization and brief introduction to our project

Content

INTRODUCTION

1. WELCOME AND INSERTION
2. GENERAL PRESENTATION OF THE COMPANY
3. Background
4. Mission
5. Vision
6. ACTIVITIES
7. ORGANIZATION OF THE COMPANY
8. **Executive Leadership**
9. **Software Engineering Department**
10. **Human Resource Department**
11. **Communication Department**
12. **Department of Financial Affairs**
13. GEOGRAPHICAL LOCATION
14. BRIEF PRESENTATION OF PROJECT THEME

CONCLUSION

# **INTRODUCTION**

The insertion phase in a company is a period during which we discover our working environment, the staff of the company and other interns. Here, we will begin by discussing our first two weeks 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 realizations. 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 elaborate on It.

## **WELCOME AND INSERTION**

We arrived at SHADERL on Tuesday 01st July 2025, at 08:00 a.m. we were received by the SHADERL team who introduced us to our workspace, gave us an official welcome to the enterprise, its activities and its different rules and regulations. Also, we discussed on the enterprise’s preferred languages and frameworks, our professional supervisor encouraged us through his past working experiences in different enterprises.

The first two week we started with our internship report by configuring our MS WORD processor, reviewing key concepts like page numbering, styles, section breaks, page breaks etc. There was a talk on project ideas, we were advised to propose project ideas and we were assigned the tasks to carry out research on these ideas. The objective was that the enterprise did not want to impose some themes on us. For those who could not find a theme, the enterprise could propose an idea for them to think about.

## **GENERAL REPRESENTATION OF SHADERL**

### **Background:**

SHADERL is a dynamic, non-governmental tech startup founded in 2023 by **ASANE DERICK** with a bold vision: to bridge the gap between traditional industries and the rapidly evolving digital economy. Recognizing the urgent need for digital transformation, SHADERL is committed to providing **cutting-edge IT solutions** while empowering the next generation of innovators.

### **Mission**

At SHADERL, we harness the power of **AI and cutting-edge technology** to democratize digital transformation. Our mission is to **empower individuals and businesses** with intelligent tools, accessible education, and personalized mentorship bridging the gap between today’s potential and tomorrow’s opportunities. We believe AI should **augment human potential, not replace it**. By making AI-driven solutions practical and ethical, we equip the next generation to lead, innovate, and thrive in an era where **human creativity meets machine intelligence**.

### **Vision**

At SHADERL, we envision a world where **technology unites humanity**, where innovation is driven by collective purpose, not just profit. By putting **“People First”**, we strive to build a future where:

1. **One Love**→ Technology fosters **inclusion, empathy, and global collaboration**, breaking down barriers.
2. **One Mind**→ AI and digital tools **amplify human potential**, creating shared knowledge and opportunity.
3. **One Legacy**→ Every individual we empower leaves a lasting impact, shaping a **smarter, kinder, and more connected world**.

We don’t just adapt to the digital age; we **redefine it with humanity at the core**.

### **ACTIVITIES**

The activities of SHADERL range from computer sciences, engineering, and training. We can outline the following:

* Conception realization, and hosting of websites.
* Software development and maintenance.
* Training in Software related fields.
* Conception and realization of multimedia.
* IT consulting and innovation.
* IT support.

## **III-ORGANISATION OF SHADERL**

### **ADMINISTRATIVE ORGANISATION**

SHADERL is administratively organized as follows:

#### **Executive Leadership**

This department is responsible for**:**

* Setting the company's vision, mission, and strategic direction
* Making high-level decisions about investments, partnerships, and growth
* Representing SHADERL in key meetings with investors and government agencies
* Ensuring all departments align with the company's core values and objectives
* Overseeing the overall performance and sustainability of the organization

#### **Software Engineering Department**

**This department is responsible for:**

* Designing, developing, and maintaining all of SHADERL's software products
* Implementing AI and machine learning solutions for company offerings
* Ensuring software security, scalability, and optimal performance
* Collaborating with other departments to understand technical requirements
* Staying updated with emerging technologies and industry best practices

#### **. Human Resource Department**

**This department is responsible for:**

* Recruiting and onboarding top talent that aligns with Shaderl's values
* Managing employee relations, welfare, and performance evaluations
* Developing training programs to enhance staff skills and capabilities
* Maintaining company culture and ensuring a positive work environment
* Handling compensation, benefits, and conflict resolution

#### **Communication Department**

**This department is responsible for:**

* Developing and executing marketing and PR strategies
* Handling all internal and external communications
* Managing social media platforms and digital content
* Organizing corporate events and press engagements

1. **Department of Financial Affairs**

**This department is responsible for:**

* Managing all financial operations and accounting
* Preparing budgets and financial forecasts
* Handling payroll, taxes, and financial reporting
* Ensuring compliance with financial regulations
* Managing investments and financial partnerships

1. **Functional organization**

The functional branch of SHADERLS is organized as follows

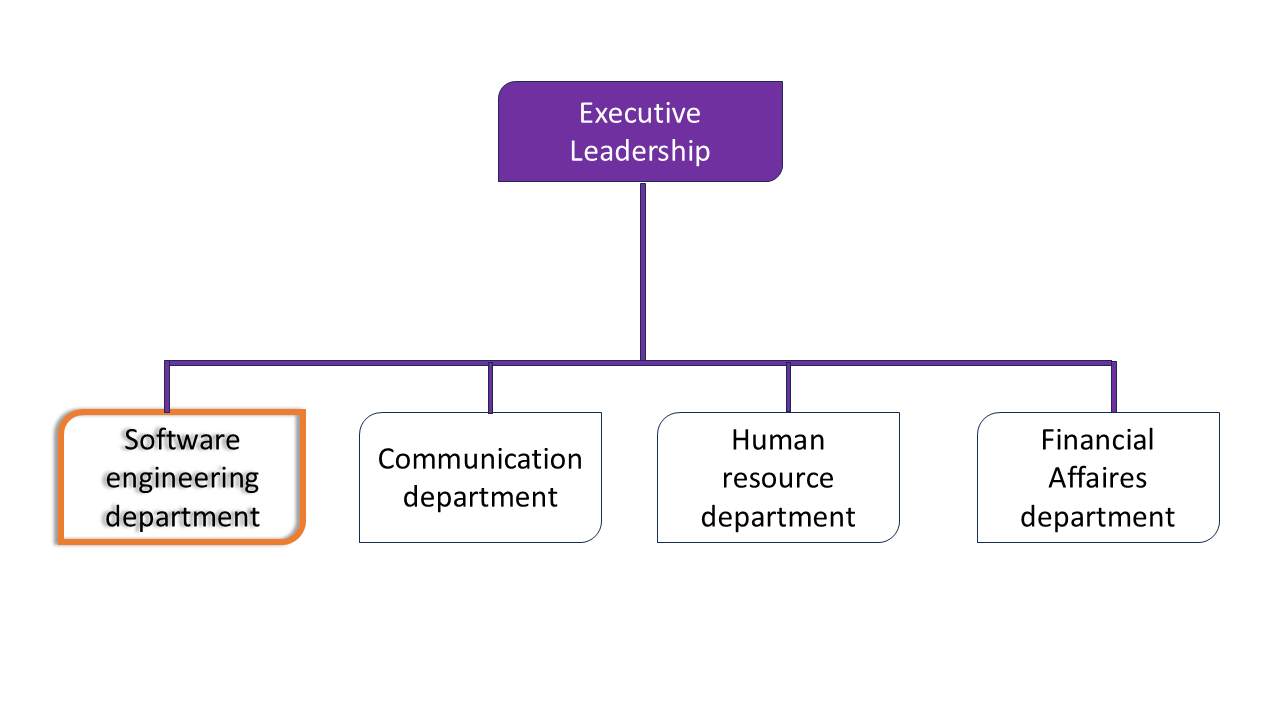


Figure 1 : functional organization of SHADERL

## **GEOGRAPHICAL LOCATION**

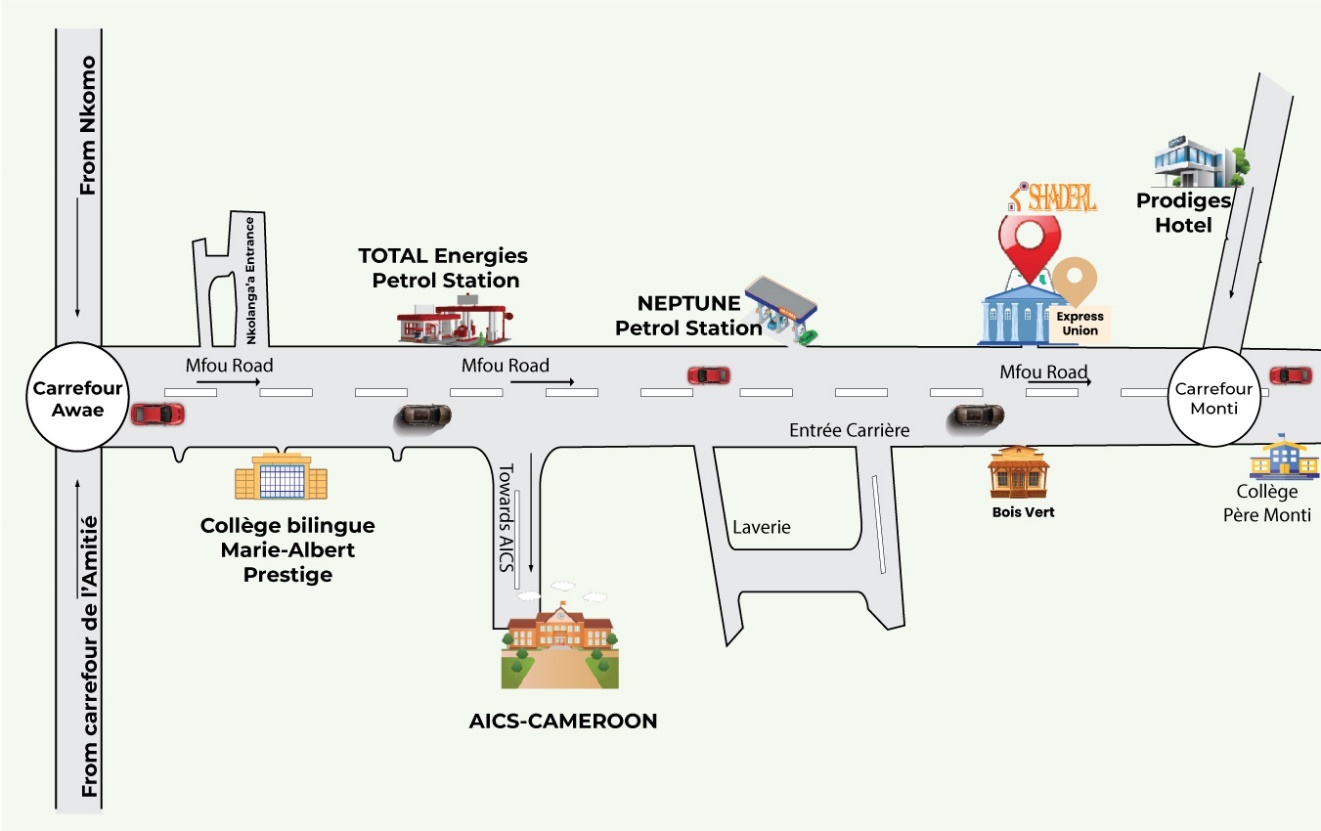


Figure 2: Location of SHADERL

## **BRIEF PRESENTATION OF THE THEME**

During our insertion phase at **SHADERL**, we were tasked with identifying innovative themes that address real-world challenges faced by Cameroonians. After thorough research and consultations with our professional supervisor, we proposed several ideas, with the most compelling being the: **"Design and Implementation of a Smart Service Finder Application for Cameroonians."**

This application is designed to bridge the gap between service providers and users by offering a seamless, location-based platform where individuals can easily find and access essential services such as plumbing, electrical work, tutoring, healthcare, and more. Key futures of the application include :

* **User-Friendly Interface:** Customers can search for nearby service providers based on their needs, location, and ratings.
* **Service Provider Profiles:** Professionals can create profiles showcasing their skills, experience, and customer reviews.
* **Real-Time Tracking:** Users can track the arrival of a service provider, ensuring transparency and security.
* **Secure Payments:** The app will integrate secure payment options, allowing for cashless transactions.
* **Admin Dashboard:** Administrators can verify service providers, monitor platform activity, and resolve disputes to maintain trust and reliability.

By implementing this solution, **SHADERL** aims to enhance service accessibility, reduce unemployment by empowering local entrepreneurs, and improve the overall efficiency of service delivery in Cameroon. This project aligns with our commitment to leveraging technology for socio-economic development

## **CONCLUSION**

To conclude, our insertion into the company was a serene, warm and convivial experience with the personnel and our professional supervisor. Their availability, and the integration to the function of the company. One of the most important things I learnt from this phase was the spirit of collaboration and discipline which is very important for every successful career. The next phase of the epic journey is the existing system which consist of a description of the system existing.

# **PART 2: TECHNICAL PHASE**

**Preamble**

The existing system is a part of the internship report where we shall give a detailed explanation of our theme. That is, what the theme is all about and also to give more precision concerning the application we are to produce. Also, it provides a deep understanding of the system currently in placed associated to the various limitations, the problems that result from these and the solution we propose.

**Content**

**INTRODUTION**

**CHAPTER 1: THE EXISTING SYSTEM**

**CHAPTER 2: SPECIFICATION BOOK**

**CHAPTER 3: ANALYSIS PHASE**

**CHAPTER 4: CONCEPTION PHASE**

**CHAPTER 5: REALIZATION PHASE**

**CHAPTER 6: FUNCTIONALITY TESTING**

**CHAPTER 7: INSTALLATION AND USER GUIDE**

**CONCLUSION**

## **CHAPTER 1: THE EXISTING SYSTEM**

**Preamble**

The study of the existing system is a crucial step in the field of IT. It provides an in-depth understanding of the systems, processes and technologies already in place, in order to make informed decisions for the improvement or creation of new IT systems. A well-conducted study of the existing situation contributes to the efficiency, performance and success of IT projects.

**Content**

**INTRODUTION**

1. **DESCRIPTION OF THE EXISTING SYSTEM**
2. **CRITICISMS OF THE EXIXTING SYSTEM**
3. **PROBLEMATIC**
4. **PROPOSED SOLUTION**

**CONCLUSION**

### **INTRODUTION**

Successful achievements of a project need clear instructions to avoid failures. Hence, realizing the document phase called the specification book will help avoid the failure of the project. The specification book permits to specify the expectations of the customer as well as the standards that will govern the project for a good, smooth, precise and concrete project. It establishes a certain agreement between the client of the project and the person who is supposed to realize the project. It presents the objectives and needs of the system to be developed. The present specification book refers to the context and justification, the needs of the project, the plan and deliverables.

### **DESCRIPTION OF THE EXISTING SYSTEM**

#### **Objectives**

#### This study aims to gather relevant information about the current service-finding methods in Cameroon before proposing a digital solution. The research is conducted through interviews, surveys, and analysis of existing service platforms (if any). Effective communication between end-users (Cameroonians seeking services) and developers is crucial to ensure the new system meets their needs.

#### **Description of the Existing System**

Based on our research, the current process of finding local services (plumbers, electricians, tutors, mechanics, etc.) in Cameroon is inefficient, time-consuming, and lacks a centralized digital platform. The existing system operates as follows:

➢ **Word-of-Mouth Referrals** – Most Cameroonians rely on recommendations from friends, family, or neighbors when searching for service providers. This method is unreliable, as it depends on personal networks and does not guarantee quality.

➢ **Manual Search (Physical Visits & Calls)** – Individuals often move from one location to another or make multiple phone calls to find available service providers. This process is slow, costly (transportation), and frustrating.

➢ **Use of social media & Informal Groups** – Some people post service requests on Facebook, WhatsApp groups, or community forums. However, these platforms are disorganized, lack verification mechanisms, and often lead to scams or unprofessional service delivery.

➢ **No Standardized Pricing or Reviews** – There is no unified system where users can compare service prices or check provider ratings, leading to inconsistent service quality and potential exploitation.

➢ **No Real-Time Availability Tracking** – Users cannot check if a service provider is available immediately or booked for another job, leading to delays.

### **CRITICISMS OF THE EXISTING SYSTEM**

Table 1 : Limits, consequences and solutions of existing system

|  |  |  |
| --- | --- | --- |
| LIMIT | CONSEQUENCES | SOLUTION |
|  |  |  |
| Service requests and provider registrations are handled manually or through fragmented channels | Time-consuming, error-prone, and discourages new users from joining the platform | Implement a centralized digital platform for seamless registration and service requests |
| Users have limited or no access to their service request history | Leads to confusion, lack of trust, and difficulty in resolving disputes | Provide users with real-time access to their complete service request and transaction history |
| No secure method for users to verify the authenticity of service providers | Users are vulnerable to fraud, scams, and poor-quality service | Enable users to view verified provider profiles and use QR code or digital verification for added security |
| Administrators lack real time information on service provider locations and activities | Providers may act unethically, and it is difficult to resolve complaint sort rack performance | Developing a platform that permits to the administrator to track his personnel when on field of work. |
| Reports, archiving, and data management are done manually | Risk of data loss, lack of backup, and inefficient reporting. | Digitize data archiving and automate report generation for secure and efficient data management |

### **PROBLEMATIC**

In today’s fast-evolving digital landscape, **security and trust** are fundamental to any service transaction, especially when personal well-being and livelihoods are involved. Many Cameroonians face challenges in finding reliable service providers, verifying their authenticity, and ensuring safe, efficient transactions. This situation raises a critical question: **How can we facilitate the process of connecting service seekers and providers in a secure, efficient, and transparent manner?** Through these observations, we recognize the need for a solution that simplifies and secures the process of finding and engaging with trusted service providers. Our response to this challenge is the development of the **SERVIGO** application.

### **PROPOSED SOLUTION**

Based on our analysis and critique of the current system, we propose the design and implementation of a **web application** **SERVIGO**—to address the identified issues by enabling the following features:

#### **For Service Providers :**

* Register and manage their service offerings via the application
* View and respond to service requests from users
* Access transaction and service history for transparency
* Generate digital receipts for completed services

#### **For Customer :**

* Search for and connect with verified service providers in their area
* View their service request and transaction history
* Request and Schedule services easily
* Contact Customer support for assistance
* Create and manage their account securely

### **CONCLUSION**

The Existing system file allows you to understand the current landscape, identify good practices, examine lacunes, make informed decisions, define objectives and requirements. This leads to solutions being taken to overcome the problems found.

## **CHAPTER 2: SPECIFICATION BOOK**

**Preamble**

The primary goal of the specification book is to outline with great precision the requirements or need of the users and the description of the resources necessary to realize the project. It is considered as a communication and description tool which permits us to avoid inadequate results.

**Content**

INTRODUCTION

1. CONTEXT AND PROBLEM DEFINITION
2. OBJECTIVES OF THE PROJECT
3. EXPRESSION OF NEEDS
4. PLANNING OF THE PROJECT
5. ESTIMATION OF RESOURCES NEEDED
6. CONSTRAINTS
7. DELIVERABLES

CONCLUSION

### **INTRODUTION**

In order to successfully achieve a project, we need to have clear instructions so as to avoid failure. Hence, realizing the document phase called the specification book will help avoid the failure of our project. The specification book permits to specify the expectations of the customer as well as the standards that will govern the project for a good, smooth, precise and concrete project. It establishes a certain agreement between the client of the project and the person who is supposed to realize the project. It presents the objectives and needs of the system to be developed. The present specification book relates to the context and justification, the needs of the project, the plan and the deliverables.

### **CONTEXT AND JUSTIFICATION**

#### **Context**

Cameroon, with a population of nearly 28.7 million inhabitants, faces significant challenges in access to reliable and secure services across various sectors. Many Cameroonians, especially those in informal or underserved communities, struggle to find trustworthy service providers for their daily needs. The current landscape is characterized by a lack of centralized platforms, reliance on word-of-mouth or informal networks, and limited means to verify the authenticity of service providers. This often results in wasted time, exposure to fraud, and a general lack of trust in service transactions. Additionally, there is little to no traceability or transparency in the interactions between service seekers and providers, making dispute resolution and accountability difficult.

#### **Justification**

The **SERVIGO** platform aims to address these challenges by providing a secure, efficient, and user-friendly digital solution for connecting Cameroonians with verified service providers. By digitizing the process of finding, verifying, and engaging with service providers, the platform will save users time, enhance security, and ensure transparency in all transactions. Real-time access to service histories and provider profiles will empower users to make informed decisions, while features such as provider localization and digital verification will build trust and accountability within the ecosystem.

### **OBJECTIVES OF THE PROJECT**

#### **General Objective**

The general objective is to develop an application that simplifies and secures the process of finding and engaging with service providers in Cameroon, while enabling users to access and manage their service histories efficiently.

#### **Specific Objectives**

The specific objectives of this project are to:

* Facilitate the process of searching for and connecting with reliable service providers.
* Maintain comprehensive records of service transactions to support informed decision-making.
* Enhance security by allowing users to verify the identity and authenticity of service providers.
* Provide real-time localization and activity tracking of service providers for greater transparency.
* Foster improved relationships and trust between service seekers and providers.
* Enable users to easily view and manage their past and present service requests and transactions.

### **EXPRESSION OF NEEDS**

#### **Functional needs**

##### **ADMINISTRATOR**

* Authenticate
* Manage service providers
* Manage users (service seekers)
* View the real-time location and activity status of service providers
* Assign service requests to providers
* Oversee and moderate platform activities
* Generate reports and analytics

##### **SERVICE SEEKER (USER)**

* Authenticate
* Create and manage account
* Update personal profile
* Search for and filter service providers
* View service request and transaction history
* Select and rate service providers
* Receive notifications (e.g., service updates, confirmations)
* Contact customer support

##### **SERVICE PROVIDER**

* Authenticate
* Create and update service provider profile
* Register and manage offered services
* Update service status (e.g., accepted, in progress, completed)
* View transaction and service history

#### **Non-Functional needs**

Specifies the quality attribute of a software system. The judge the software system or application based on Performance, Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to its success. Failing to meet nonfunctional requirements can result in systems that fail to satisfy user needs.

* + **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:** it 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 information.
      * Some of the security principles include reassuring confidentiality, accountability, Integrity, Authentication.

### **PROJECT PLANNING**

#### **Time scheduling**

This section presents how the work phases was scheduled throughout the internship period. We will present it on a table and on a Gantt diagram*:*

Table 2: Time Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| PHASE | OBJECTIVE | OUTGOING | DURATION |
| INSERTION | Collection of information on the enterprise | Insertion book | 1 week |
| EXISTING SYSTEM | Study of the existing system | Existing System | 1 week |
| SPECIFICATION BOOK | Specification of the user needs | Specification Book | 1 week |
| ANALYSIS | Capture of needs Use case and textual description  Modelling | Analysis Book | 2 weeks |
| CONCEPTION | Preliminary conception and  Detailed conception | Conception book | 1 week |
| REALIZATION | Implementation Unitary test, Integration Test  Development,  Deployment, Component diagrams | Realization book | 2 weeks |
| TEST OF FUNCTIONALITIES | Testing of the software and debugging | Test of functionalities | 1 week |
| INSTALLATION AND USER GUIDE | Documenting software | User Guide | 1 week |

The GANTT project software is one of the efficient tools to present the various projects tasks, their dates and the order in which they should be

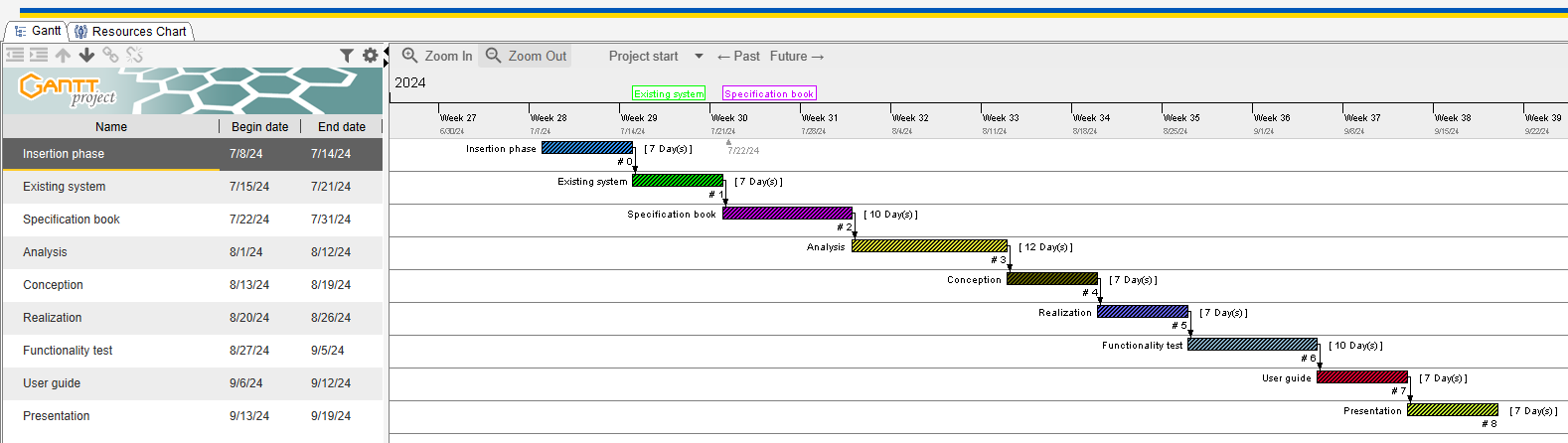


Figure 3: Gantt project

#### **Actors of the project**

Table 3: Actors of the project

|  |  |  |  |
| --- | --- | --- | --- |
| NAME |  | FUNCTION | ROLE |
| Mr. EPOUPA PELAYAH |  | Software Engineer at SHADERL | Professional supervisor |
| Mr. ASANE DERICK |  | Lecturer at AICS-Cameroon | Academic Supervisor |
| WIRBA JORDAN WIRBA |  | Software engineering student at AICS-Cameroon  Level III | Analyst and developer |

### **ESTIMATION OF RESOURCES NEEDED**

#### **Hardware resources**

Table 4 : Hardware resources

|  |  |  |
| --- | --- | --- |
| Materials | Quantity | Price (FCFA) |
| LAPTOP HP  Intel Core i3  500 GO, RAM 4 GO, | **1** | **250,000** |
| Printers | **1** | **500,820** |
| Local network installation | **1** | **300, 000** |
| 8GB USB Key | **1** | **6325** |
| Smart phone | **2** | **150,000** |
| Total2 | **6** | **1,207,145** |

#### **Software resource**

Table 5: Software resources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | DESIGNATION | USAGE | QUANTITY | UNIT COST  (FCFA) |
| Text Editor | Microsoft Office  2019 | Used for the creation of our report and  PowerPoint | 1 | **500,500** |
| Web browser | Google Chrome | View web pages | 1 | **Free** |
| Code Editor | Visual Studio  Code | For writing the code of the application | 1 | **Free** |
| Project planning | Gantt Project | For building a  Gantt chart | 1 | **Free** |
| Geo-location plan  designing tool | Ichogram | To draw the location plan | 1 | **Free** |
| UML  Analysis | Visual paradigm  (Community edition) | Modeling tool | 1 | **Free Trial** |
| TOTAL 1 |  | | **8** | **500,500** |

#### **Human resources**

Table 6 : Human resources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ROLE | NUMBER OF DAYS | QUANTITY | COST PER DAY | TOTAL PRICE (FCFA) |
| Project  Manager | **90** | **01** | **30 000** | **2 700 000** |
| Analyst | **21** | **01** | **25 000** | **525 000** |
| UI/UX Designer | **07** | **01** | **20 000** | **140 000** |
| Programmer | **30** | **01** | **15 000** | **450 000** |
| Tester | **14** | **02** | **10 000** | **140 000** |
| Margin error | **/** | **/** | **/** | **1,475,000** |
| TOTAL3 |  |  |  | **4 840 000** |

#### **Total estimation of the project**

Table 7: Total estimation of the project

|  |  |
| --- | --- |
| Item | Amount (FCFA) |
| Hardware Resources | **1,207,145** |
| Software Resources | **500,500** |
| Human Resources | **4 840 000** |
| Total | **6,574,645** |

### **CONSTRAINS**

With any project, some limitations and risks need to be taken into account and addressed to ensure the project’s ultimate success. So, we have as constraints:

#### **Criterions of Acceptability**

The delivered product is judged acceptable if it respects all the need of the user and all the different functionalities that have been presented.

#### **Time constrains**

One of the important aspects of the project is the limited time we had when carrying out the project. This project will be realized in three (3) months, starting from the 8th July to the 30th September. Due to the limited time, it will not be easy for us to finish all the document on time and to fulfil all the expectations given us for the full realization of the project.

### **DERIVABLES**

A deliverable is an expected result of a task or project, measurable or verifiable which results from the completion of a delivery service. During delivery, it will be a question for us to present an application with all the functional modules. In addition to this, we must present the elements below:

* A complete report including:
* The existing file
* The specification book,
* The analysis file,
* The design file,
* The production file,
* The user guide,
* A CD-ROM in which will be recorded:
* The source code of the application
* The database script,

### **CONCLUSION**

Reaching the end of this part. The list of objectives to be achieved has been enumerated and made clear. The specification book permitted us to present the different actors associated with the project as well as the requirement and the provisional planning needed for the achievement of our project. We will move directly to the next part which is the analysis phase. In the analysis phase, we will model our system with a modelling language and a unified process, do a comparative study between UML and Merise.

## **CHAPTER 3: ANLYSIS PHASE**

**Preamble**

The development of any system can be put into two major phases: Analysis and Design. The analysis document appears following the specification book, allows us to present a detailed analysis of the problem and solution, The chosen analysis method and the reasons that motivated this choice. To achieve this, we will use UML2.5 (Unified Modeling Language) with 2TUP (2 Track Unified Process) as method applied to UML to analyze the system.

**Content**

**INTRODUTION**

1. **PRESENTATION OF THE MODELING APPROACH**
2. **PRESENTATION OF UML AND 2TUP**
3. **CHOICE OF THE ANALYSIS METHOD**
4. **MODELING OF THE PROPOSED SOLUTION**

**CONCLUSION**

### **INTRODUTION**

The analysis phase is a systematic study that enables to differentiate between the parts of a problem and find solutions following a well-defined methodology. Thus, after the specifications of our needs, we are now going to do our analysis using the Unified Modeling Language (UML) Following the 2TUP process. Here, we are going to determine diagrams that intervene in the analysis phase, priority functions and take into considerations all the elements that can help in the solutions due to the fact that any change in the existing policies in an organization may require the existing information system to be restructured to have a complete development of a new information system. In case of an organization functioning manually and planning to computerize its functionality, the development of a new information system would be required.

### **PRESENTATION OF THE MODELING APPROACH**

#### **SOME ANALYSIS METHODS/APPROACH**

##### **MERISE**

MERISE stands for “Méthode d’Etude et de Réalisation Informatique pour des Systèmes d’Entreprise”. This is a method to build an automated (computerized) information system which is efficient, flexible and adapted to the organization. Like any other method of analysis and design of the Information system, MERISE includes: a **language** (vocabulary, rules of syntax...), a **process** (by level and in stages), t**ools** (software such as POWER AMC, Designer or Win Design …) and **models.** Its models listed as follows:

* CMC (Conceptual model of communication)
* CMT (Conceptual model of treatment)
* OMT (Organizational model of treatment)
* CDM (Conceptual data model)
* LDM (Logical data model)
* PDM (Physical data model)

##### **SCRUM**

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. Agile scrum methodology is a project management system that relies on incremental development. Each iteration consists of two- to four-week sprints, where each sprint's goal is to build the most important features first and come out with a potentially deliverable product. More features are built into the product in subsequent sprints and are adjusted based on stakeholder and customer feedback between sprints. Whereas other project management methods emphasize building an entire product in one iteration from start to finish, agile scrum methodology focuses on delivering several iterations of a product to provide stakeholders with the highest business value in the least amount of time.

##### **UNIFIED PROCESS (UP)**

The UP is an iterative and incremental software development methodology. The Unified Process is an iterative, architecture-centric software development process driven by use cases and geared towards reducing risk. It is a process pattern that can be adapted to a wide class of software systems, to different areas of application, to different types of businesses, to different skill levels and to different sizes of the business and different data. It qualifies a process or a procedure that performs a group of operations repeatedly until a well-defined condition is met.

##### **DYNAMIC SYSTEM DEVELOPMENT METHOD (DSDM)**

It is an organized, commonsense process focused on delivering business solutions quickly and efficiently rather than just team creativity. It is similar in ways to SCRUM and XP, but it has its best uses where the time requirement is fixed.

### **PRESENTATION OF UML AND 2TUP**

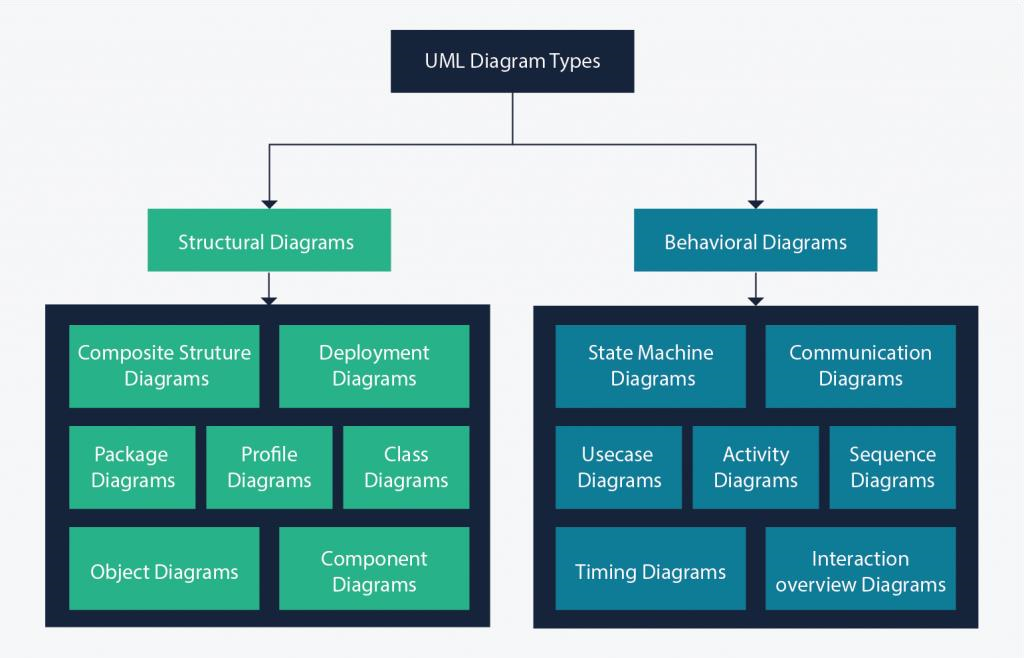
#### **UNIFIED MODELING LANGUAGE (UML)**

The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct and document the artifacts of a software system. It captures decisions and understandings about systems that must be constructed. It is used to understand, design, browse, configure, maintain and control information about a system. It is intended for use with all development methods, life cycle stages, application domains and media. UML captures information about the static structure and the dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. UML 2.5 comprise 14 diagrams which represents the different views of a system. The 14 diagrams can be subdivided into Static or structural and Dynamic or Behavioral diagrams. These diagrams are represented in the table below:

##### Table 8 : *UML diagrams*

|  |  |
| --- | --- |
| STATIC DIAGRAMS | DYNAMIC DIAGRAMS |
| 1 – Class Diagram | 8 – Use case Diagram |
| 2 – Component Diagram | 9 – Activity Diagram |
| 3 – Deployment Diagram | 10 – State Machine Diagram |
| 4 – Object Diagram | 11 – Sequence Diagram |
| 5 – Package Diagram | 12 – Communication Diagram |
| 6 – Profile Diagram | 13 – Interaction Overview Diagram |
| 7 – Composite Diagram | 14 – Timing Diagram |

The most important point to note here is that UML is not a method but a modeling language. As such, 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. There exist several Ups but our modeling approach will be the 2TUP (Two-track unified process) which we will use in the course of our project.



##### *Figure 4:UML diagrams*

#### **COMPERATIVE BETWEEN UML AND MERISE**

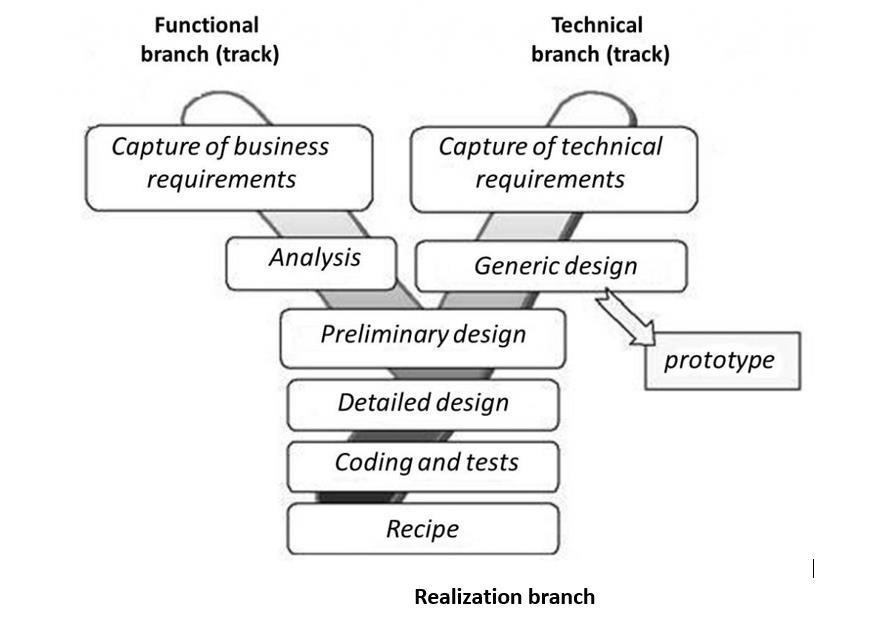
Table 9:Comparative between UML and MERISE

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

#### **TWO TRACK UNIFIED PROCESS (2TUP)**

In 2TUP, system analysis and design activities are generalized following 5 workflows namely; the **capture of needs**, **analysis**, **conception**, **implementation**, and **test**. 2TUP proposes life cycle in **Y**, which separates the **technical** and the **functional** aspects. It begins with preliminary studies which essentially consists of identifying the different actors of a system. This is followed by the terms of reference and the context of the project. This is articulated around 3 essential phases:

* The functional branch
* The technical branch
* The realization phase.

The following figure gives more details on how software development follows the three branches of the 2TUP.

##### *Figure 5:Branches of 2TUP*

##### **The functional branch**

This branch captures the functional needs which help in preventing the production of software that does not fit the needs 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. This branch consists of the following stages:

* **Capture of Business (functional) needs**: This part minimizes the risk of producing an inadequate system with the needs of users and also verifies its consistency.
* **Analysis**: This is the study of specifications to find out what the system will be made of in terms of business rules. We have the following diagrams: **Use case**, **Activity**, **State Machine** and **Communication**.

##### **The Technical branch**

This branch captures all the constraints and choices related to the conception of the system, the tools and equipment as well as the integration constraint with the existing system condition. This branch consists of the following stages:

* **Capture of Technical Needs**: This stage consists of the identification of tools, materials and technologies to use in developing the system. This technical architecture will be presented in this stage.
* **Generic Design**: The technical architecture will be presented in this stage. We have the following diagrams: **Mockup design**.

##### **The Implementation branch**

The preliminary conception, the detailed conception and the documentation of the system are studied here. This branch consists of the following stages:

* **Preliminary Design**: This is the stage where the analysis model is integrated into the technical architecture. The goal here is to know what technical component will be used depending on the features from the analysis. We have the following

diagrams: **Interaction** **Overview**, **Component**, **Deployment**, **Package** and **Composite structure diagrams**.

* **Detailed Design**: This is the detailed design of each feature of the system. We
* have the following diagrams: **Class,** **Object**, **Sequence** and **Timing Diagrams**.
* **Coding and Tests**: This is the programming phase of the designed features, alongside testing of the coded features.
* **Recipe (Results)**: This is the validation phase of the functions of the system developed.

##### **The capture of technical requirements**

Identification of Tools, materials and technologies to use constraints (maximum response time, integration with the existing constraints) all these will lead to a first design of the technical architecture.

##### **The middle branch (Implementation phase)**

* **Preliminary Design**

This is a delicate stage in which the analysis model is integrated into technical architecture. The goal here is to know what technical component we put our feature from the analysis. We have the following diagrams:

* Component Diagram
* Deployment Diagram
* Package Diagram
* Composite Structure Diagram
* **Detailed design**

This is the detailed design of each feature of the system. We have the following diagrams:

* Class Diagram
* Object Diagram
* Sequence Diagram
* **Coding and tests**

This is the programming phase of the designed features alongside with the coded features.

### **CHOICE OF THE ANALYSIS APPROACH**

#### **JUSTIFICATION AND MOTIVATION FOR THE CHOSEN APPROACH**

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

* UML is a language which is centered on the user’s needs
* UML is based on the object-oriented approach
* UML produces good standards for software development
* UML has large visual elements to conduct and easy to follow
* 2TUP is a process based on object approach and is constructed in UML
* 2TUP facilitates the modeling of complex systems which will undergo many evolutions with time
* 2TUP offers a deployment cycle (the Y shape development cycle) which dissociates the technical aspects from the functional aspects.
* 2TUP is a unified process, so it processes all the characteristics of this process (Incremental, Iterative, User-oriented and component-oriented).

### **MODELING OF THE PROPOSED SOLUTION**

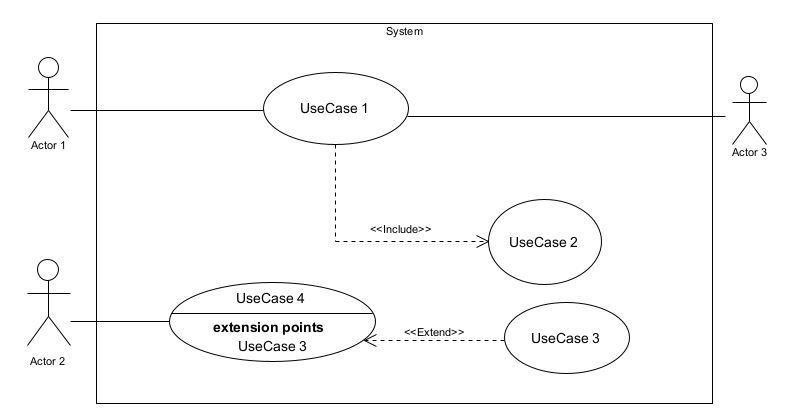
#### **USE CASE DIAGRAM**

##### **Definition**

The use case diagram identifies the functionality provided by the system, the users who interact with the system, and the interactions between them. Use cases are used in the analysis phase to define the high-level actors of the system. The main purposes of a use case diagram are:

* Provide a high-level view of what the system does,
* Identifies the users (actors) of the system
* Determine sectors which requires human-machine interfaces.

##### **Formalism**



###### Figure 6:Use case diagram formalism

##### **Components of Use case diagram**

Table 10: Components of a use case diagram

|  |  |  |
| --- | --- | --- |
| Element | Description of main properties | Notation |
| Actor | Represents any entity that directly interacts with the system. The actor is what performs the different possible actions of the system. |  |
| Use case | A use case represents a functionality of the system. It is an action that can be performed by an actor. | Name of use case 1 |
| Association | It indicates that an actor takes part in a use case. | B |
| Inclusion | This is a relation that denotes that an included action **must** be performed before another action is can be performed. A includes B signifies that B is a compulsory part of A. | <<Include>>  A  B |
| Exclusion | This is a relation that denotes that an action **may** be performed while another one is being performed. A use case B extends A means that B is an optional part of A. | <<Extend>>  A  B |
| Inheritance | It is the only possible relationship between actors | Acteur\_3  Acteur\_4 |
| Generalization | This shows that the use case is **kind of another**. This relation also permits to decompose a complex case into smaller and simple cases. | A  B  Actor\_1  Actor\_2 |
| System | It is a container of use case which interacts with the external actors |  |

* **General Use case Diagram**

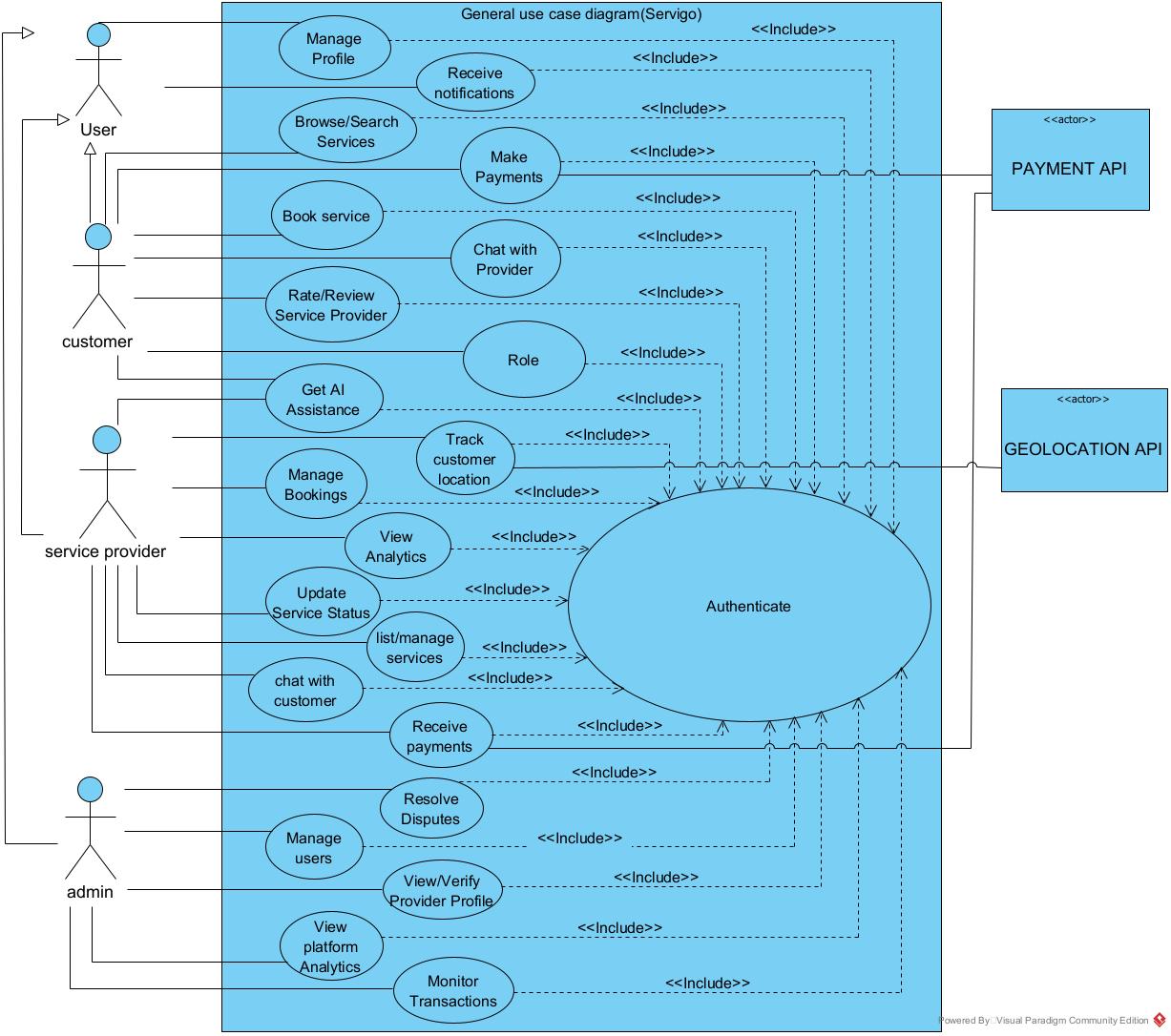
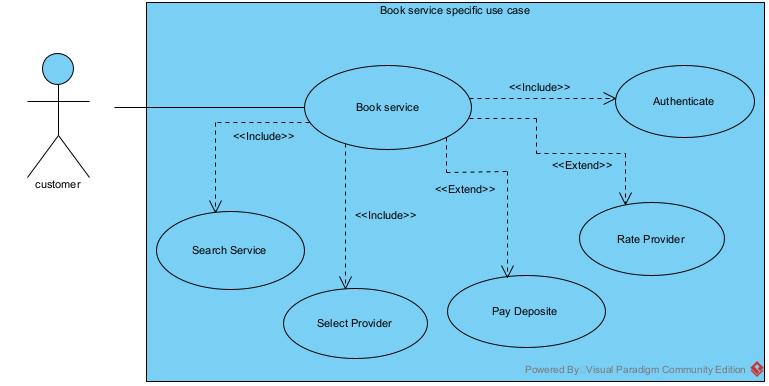


Figure 7: General use case Diagram

* **Detail use case diagram for book service**

Figure 8: Book service specific use case Diagram



* **Detail use case diagram for manage Bookings**

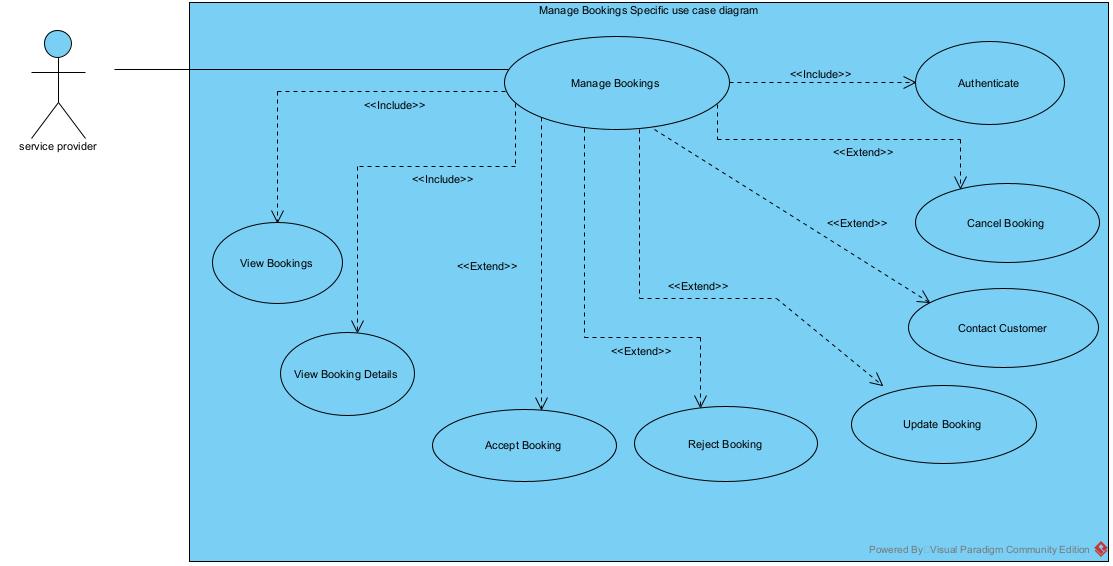


Figure 9:Manage Booking specific use case diagram

##### **TEXTUAL DESCRIPTION OF USE CASE DIAGRAM**

###### **Definition**

UML allows the execution of a use case to be described in a textual way, in a form called nominal scenario. A nominal scenario describes in more detail the execution of a use case by an actor until it is successfully completed.

###### **Formalism**

A textual description of a use case is represented in the following form

Table 11: Table 11: Formalism of textual description

|  |  |
| --- | --- |
| Name | EXAMPLE |
| Objective | A user should login into the application |
| Actors | All users |
| Trigger | Users clicks on trigger button |
| Pre-condition | The user should be in a particular page |
| Nominal scenario | The interaction steps between the user and the system |
| Alternative scenario | What happens when an action in the nominal scenario goes wrong |
| Post condition of success | What happens when the request is successful |
| Post condition of failure | What happens when request fails |

Table 12: Table 12: Textual description of Authenticate

|  |  |
| --- | --- |
| Name | Authenticate |
| Objective | A user should login into the application |
| Actors | Visitor/Student/Mentor/Preparation center / Administrator |
| Trigger | The user clicks on login button |
| Pre-condition | The user should be in the home page |
| Nominal scenario | 1. User clicks on login button 2. System Display login form 3. User Fills and submit form 4. System Check conformity 5. System sends query to database 6. System receives results from database 7. System checks results from the database 8. System displays home page |
| Alternative scenario | 4.a. In case the user enters wrong credentials  4.b. The system displays an error message and returns to step 2  7.a. In case an error occurs during the process  7.b. The system displays an error message and returns to step 2 |
| Post condition of success | The user’s session is displayed |
| Post condition of failure | An error message is displayed for incorrect credentials, system error or account inexistence |

#### **COMMUNICATION DIAGRAM**

##### **Definition**

Communication Diagrams model the interactions between objects in a sequence. They describe both the static structure and the dynamic behavior of a system. It is a simplified version of a Collaboration Diagram introduced in UML 2.0. A communication diagram is more focused on showing the collaboration of objects rather than the time sequence.

##### **Formalism**

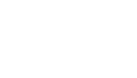


Objet\_1



1:

Message\_1



2:

Message\_2



3:

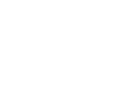
Message\_3



Objet\_4



Objet\_3



Objet\_2

###### Figure 10: Formalism of communication diagram

##### **Components of Communication diagram**:

Table 13: Components of Communication diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION | NOTATION |
| Call message | A Call message define a particular communication between lifeline of the interaction that represent an innovation of the target  lifeline |  |
| Dependency | A dependency is a relationship that signifies a single or a set of model elements for their specification |  |
| Lifeline | An object represents an individual participant in the interaction conversation |  |
| Generalization | A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. |  |

* **Authentication communication diagram.**

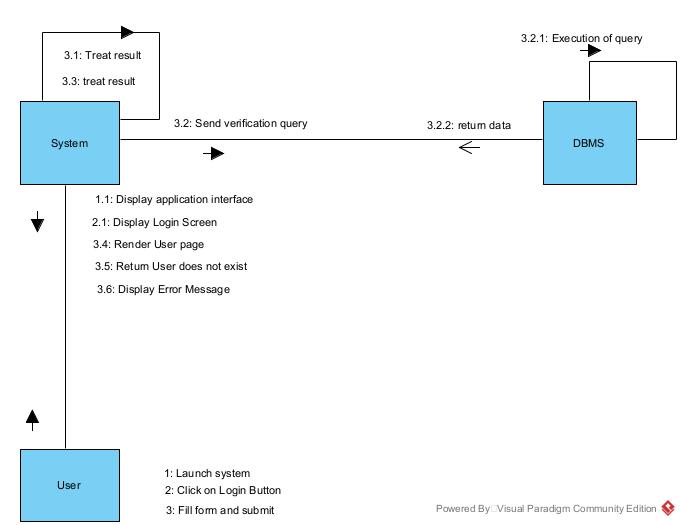


Figure 11:Communication Diagram for Authenticate

* **Admin verifies service provider communication diagram**



Figure 12: Communication Diagram to Verify Service provider

#### **SEQUENCE DIAGRAM**

##### **Definition**

A sequence diagram is a form of interaction diagram which shows objects as life lines running down the page with their interactions over time represented as messages drawn as arrows from the source life line to the target life line. Sequence diagrams are good at showing which objects communicate with which other object and what messages trigger this communication. Sequence diagrams are not intended to show complex procedural logic.

##### **Formalism**



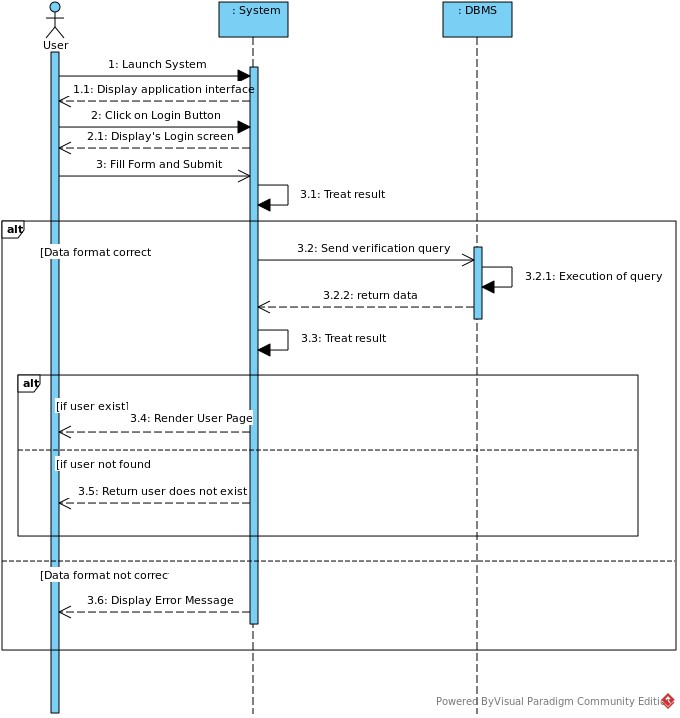
###### Figure 13:Formalism of sequence diagram

##### **Components of Sequence diagram**

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION | NOTATION |
| Lifeline | A lifeline represents and individual participant in a sequence diagram. A lifeline will usually have a rectangle containing its object name. |  |
| Asynchronous message | It is a message which does not require a response for the  interaction to continue. |  |
| Synchronous message | It is a message which requires a respond for the interaction to continue. |  |
| Self-message | A self-message can represent a recursive call of an operation, or one method calling another method belonging to the same object. |  |
| Return message | Return message as results of asynchronous messages. |  |
| Actor | The entities which perform actions in the system. |  |
| Fragment | It’s one or more processing sequence enclosed in a frame and executed under named circumstances. |  |
| Objects | They send and receive messages. |  |

Table 14: Components of a sequence diagram

* **Authentication Sequence Diagram**



###### Figure 14:Sequence diagram for Authentication

* **book service sequence diagram**

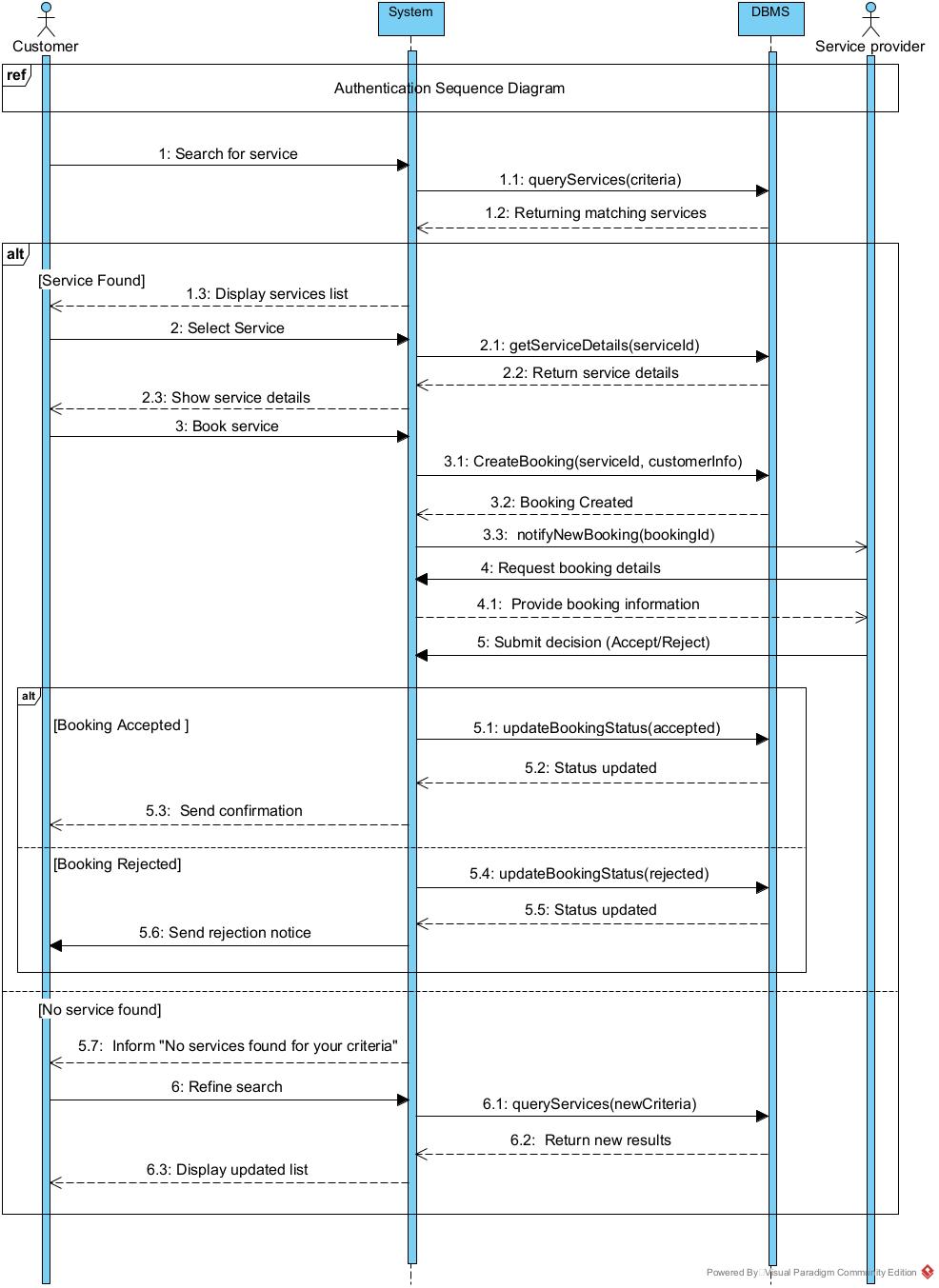


Figure 15: sequence Diagram for Book service

* **Update service status by provider sequence diagram**

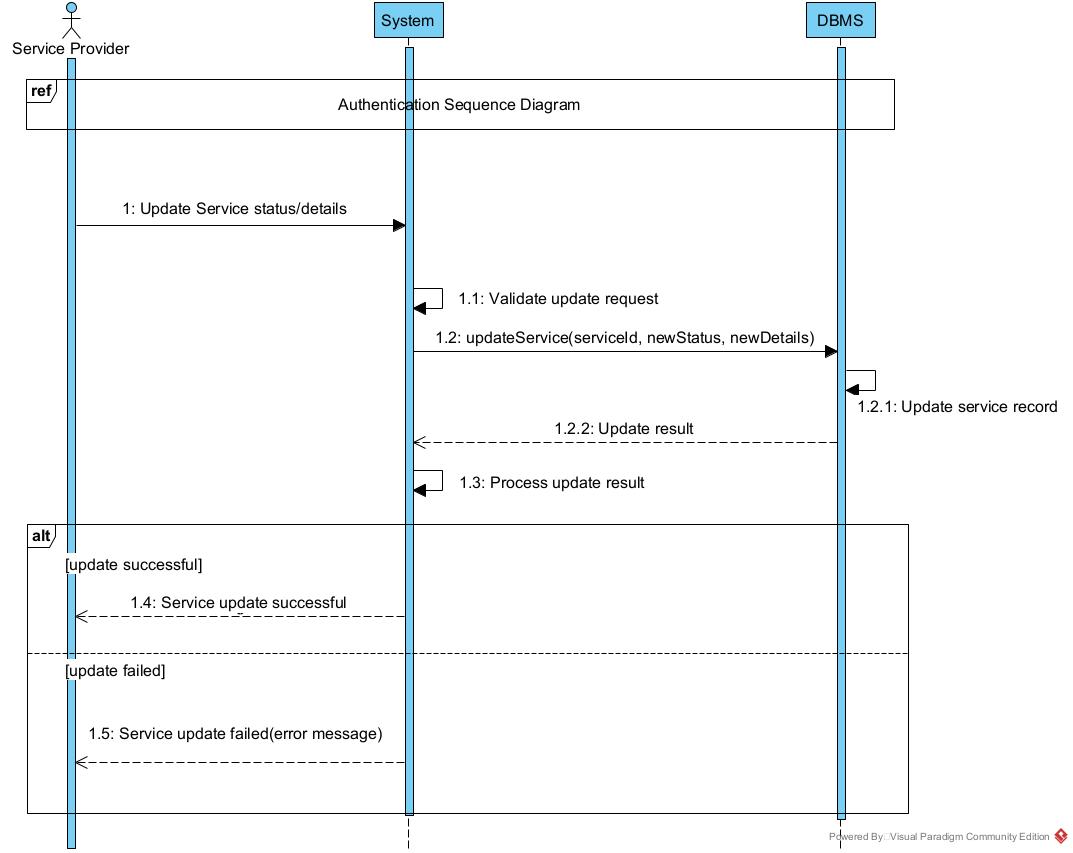


Figure 16: Sequence Diagram for update service status

* **Admin Verifies Service Provider Documentation Sequence Diagram**

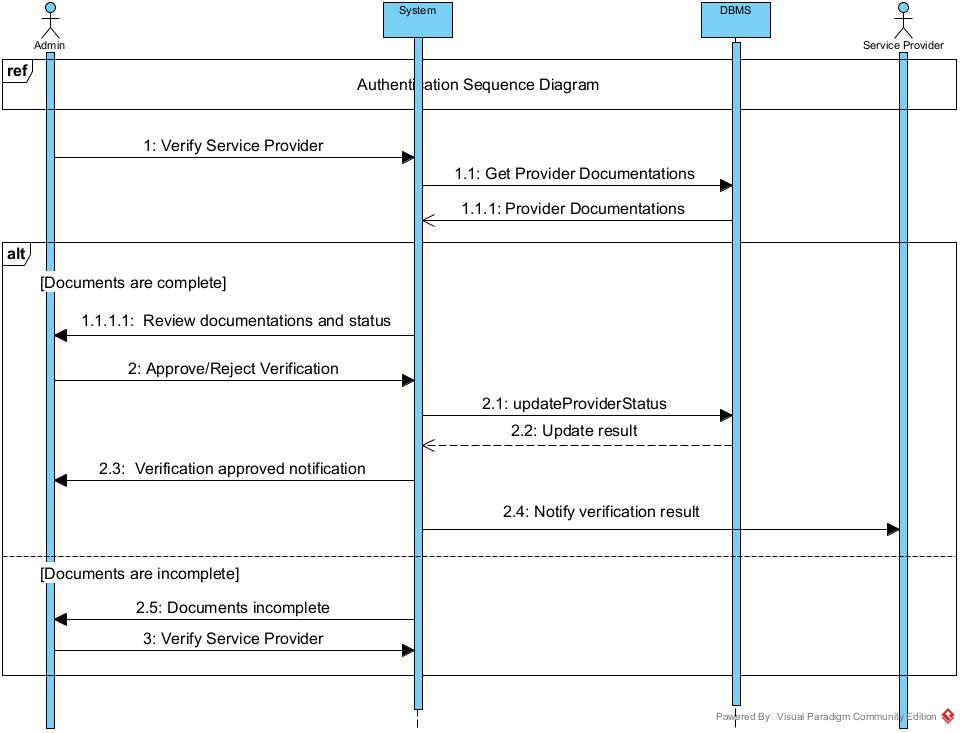


Figure 17 : Sequence Diagram for verify service provider documentation

#### **ACTIVITY DIAGRAM**

##### **Definition**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes. Activity diagrams show the overall flow of control. The basic purpose of activity diagrams is that it captures the dynamic behavior of the system.

##### **Formalism**

###### Figure 18:Formalism of activity diagram



Partition 1



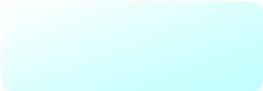
Partition 2



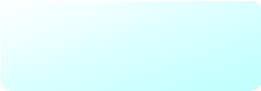
Partition 3



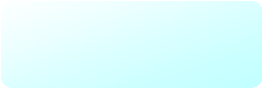
Activity\_1



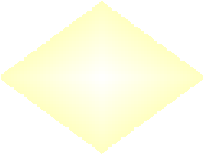
Activity\_3



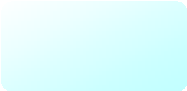
Activity\_2



Activity\_6



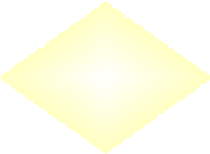
Decision\_



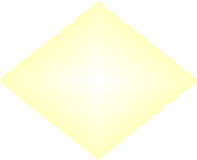
Activity\_4



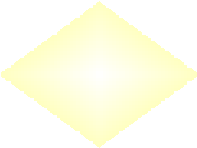
Activity\_5



Decision\_



Decision



Decision

Condition

Condition

1. **Components of Activity diagram**

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION AND MAIN PROPERTIES | NOTATION |
| Activity | An activity is a rounded corned rectangle enclosing all actions control flows and other elements that make up an activity. |  |
| Control Flow | It shows the flow of control form one end to the next |  |
| Initial Node | An initial node is a control node that splits a flow into multiple concurrent flows |  |
| Final Node | It denotes the end of all control flows |  |
| Decision and merge nodes | Indicates the check of a condition from a control flow |  |
| End nodes | Indicates the end of a single control flow |  |
| Swim Lanes | Used for partitioning the children in an activity diagram |  |

Table 15:: Components of activity diagram

* Authentication activity diagram.

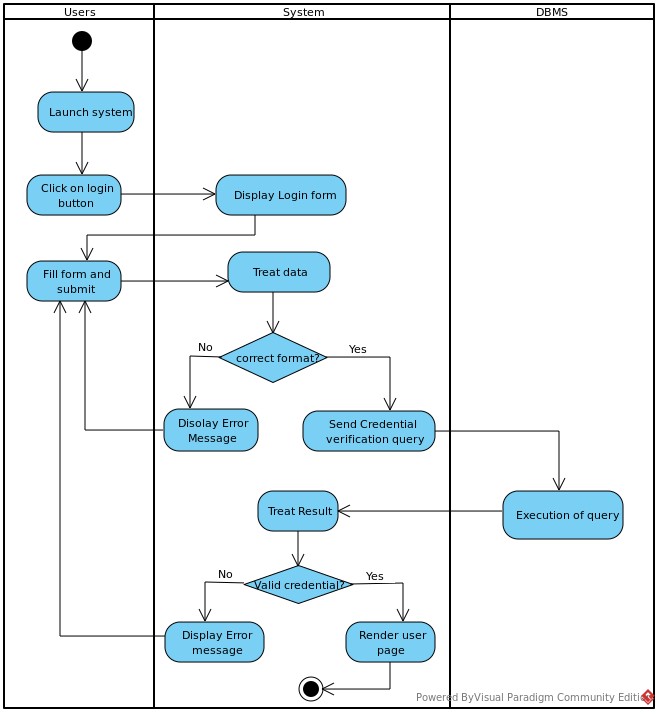


Figure 18: Activity diagram for authentication

* **Book service activity diagram**

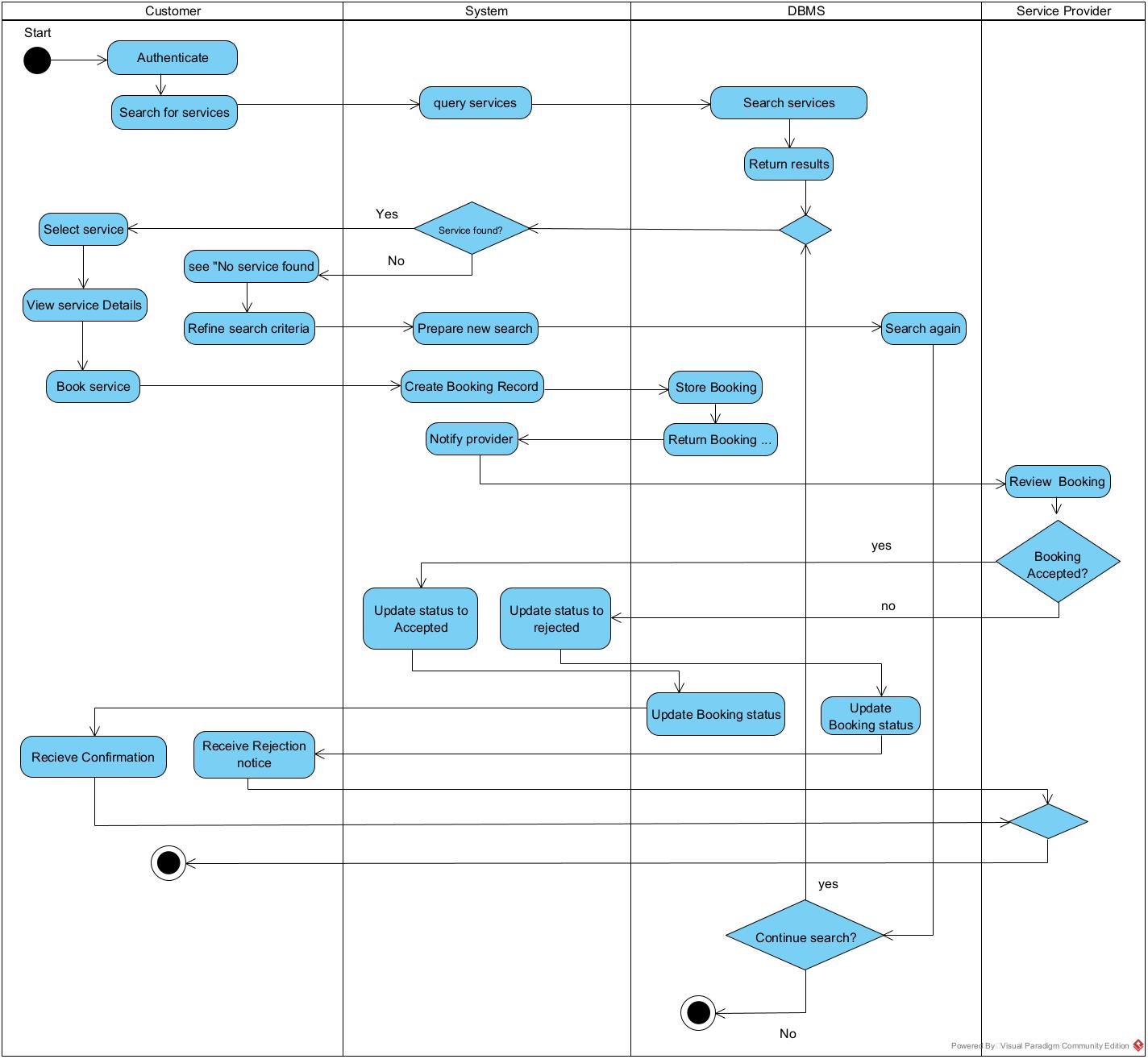


Figure 19:Activity diagram for book service

* **Update Service status by provider activity diagram**

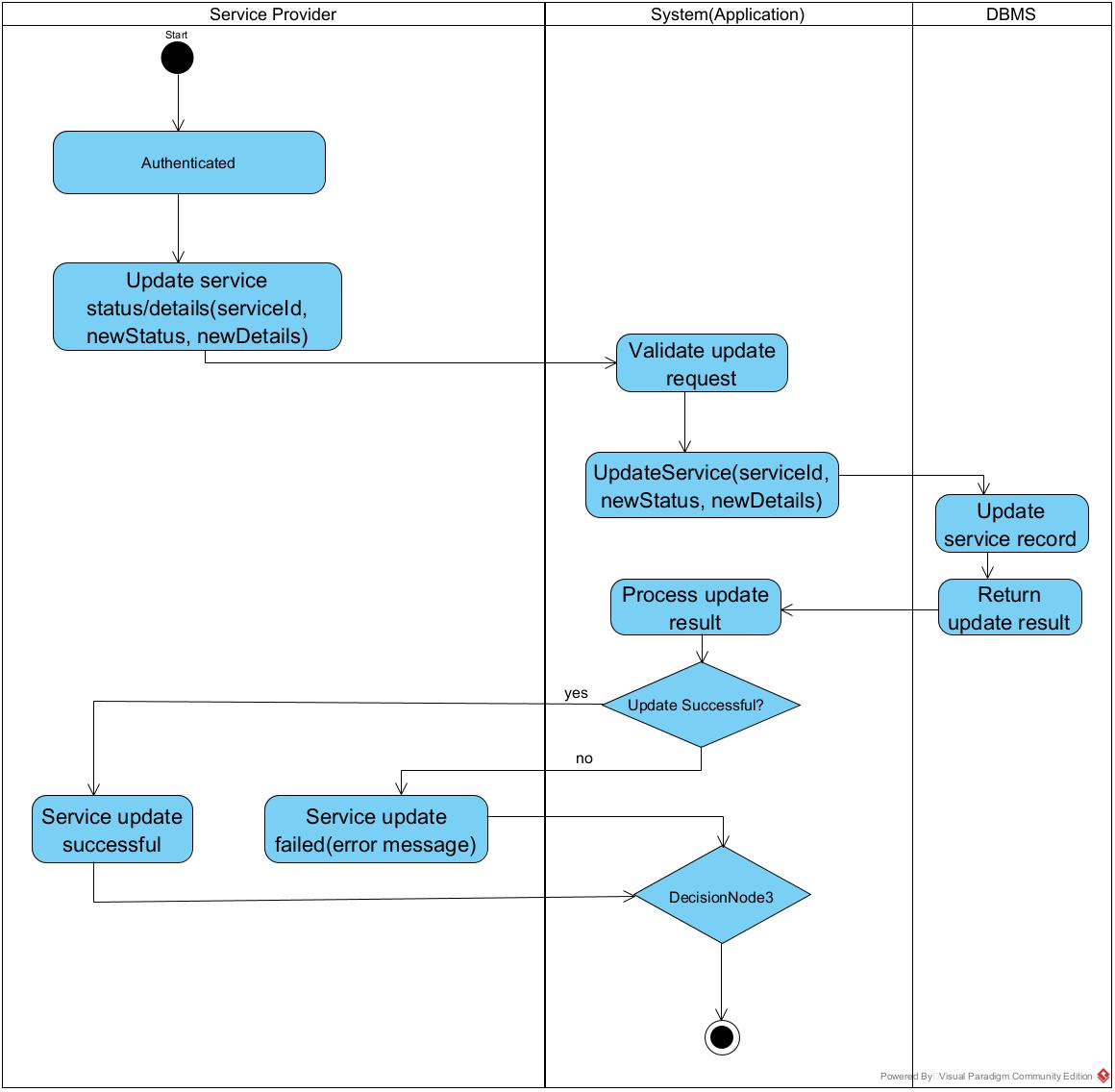


Figure 20:Activity Diagram for update service status by service provider

* **Admin Verifies Service Provider Activity Diagram**

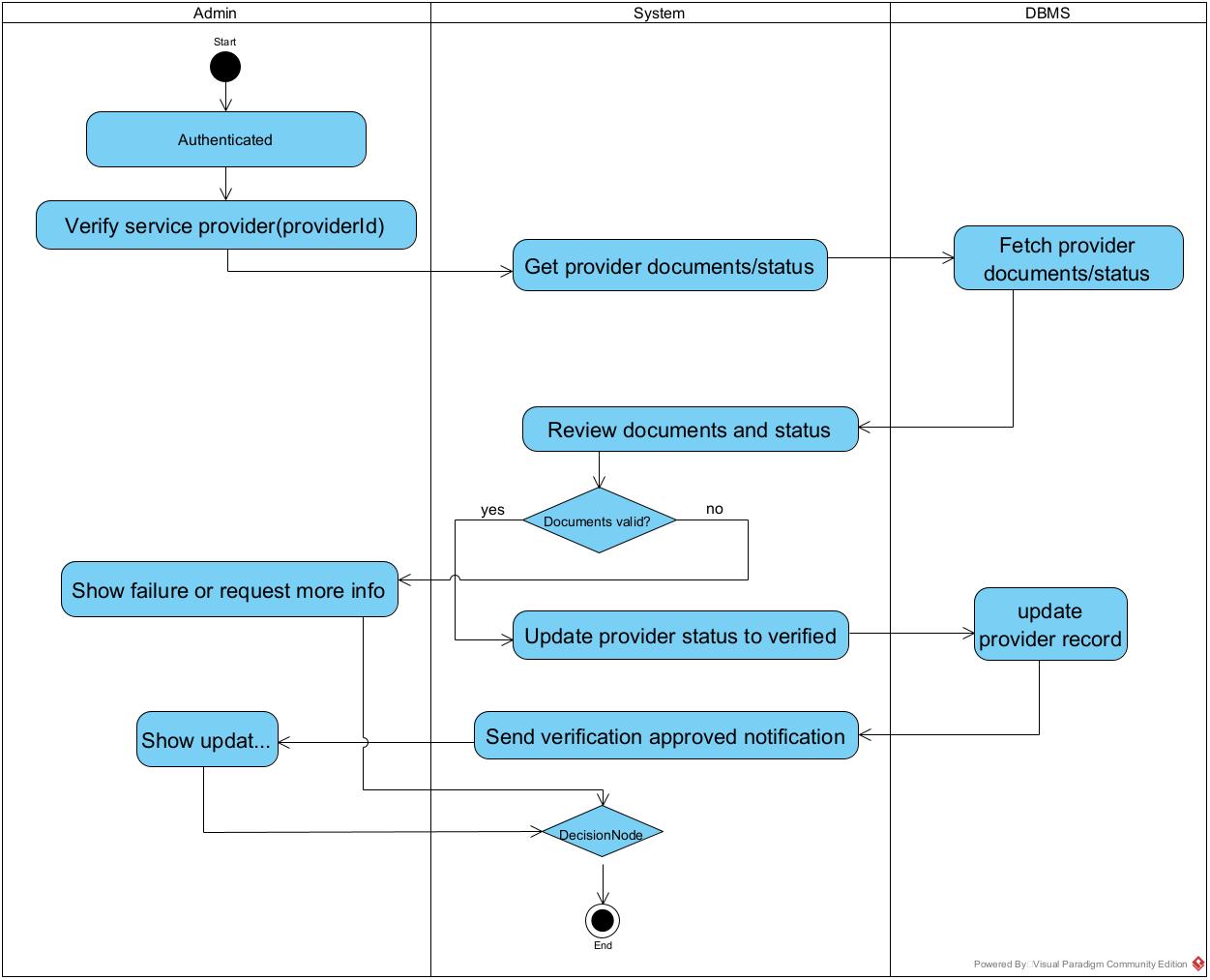


Figure 21 : Admin verify Service provider activity Diagram

### **CONCLUSION**

Having reached the end of our analysis, we were asked to present, then illustrate the modeling language and the method used and finally present the diagrams of the functional branch to carry out our analysis phase. We can say that this analysis file has allowed us to well delimit the functional needs of the web application that will be designed and to have a detailed overview of the new system to be set up. The chosen modeling language and associated process will allow us to subsequently start the conception phase.

## **CHAPTER 4: CONCEPTION PHASE**

**Preamble**

The conception phase which appears directly after the analysis phase, permits us to present in a precise manner the group of components necessary for the good functioning of the software and also the architecture used for the proposed solution. It bridges the analysis and the realization phase and is a continuation of the analysis phase, which represents the technical aspects used in modeling our system.

**Content**

INTRODUCTION

I.

CLASS DIAGRAM

II.

STATE MACHINE DIAGRAM

**III**.

PACKAGE DIAGRAM

CONCLUSION

### **INTRODUTION**

The conception phase describes, in detail, the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system as modelled in the analysis phase. This phase is meant to identify and consider the essential components (hardware and/or software), structure (networking capabilities), processes and procedures for the system to accomplish its objectives.

### **CLASS DIAGRAM**

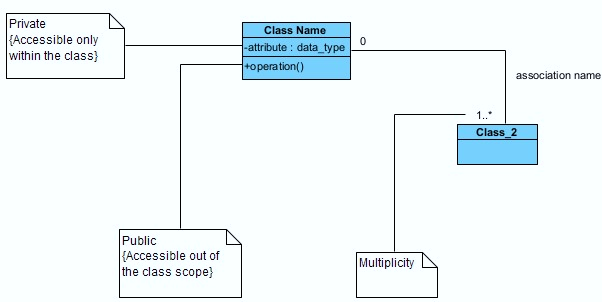
#### **Definition**

A class diagram shows the building block of any object-oriented system. Class diagrams portray a static view of the model, or part of the model, describing what attributes and behavior it has rather than detailing the method for achieving operations. Class diagrams are most useful in illustrating relationships between classes and interface, composition or usage and connections respectively.

A class is a set of functions and date (attributes) that are linked together by semantic field. Classes are used in object-oriented programming to model programs and to curt a complex risk into several small simple jobs.

#### **Formalism**

##### Figure 22:Formalism of class diagram

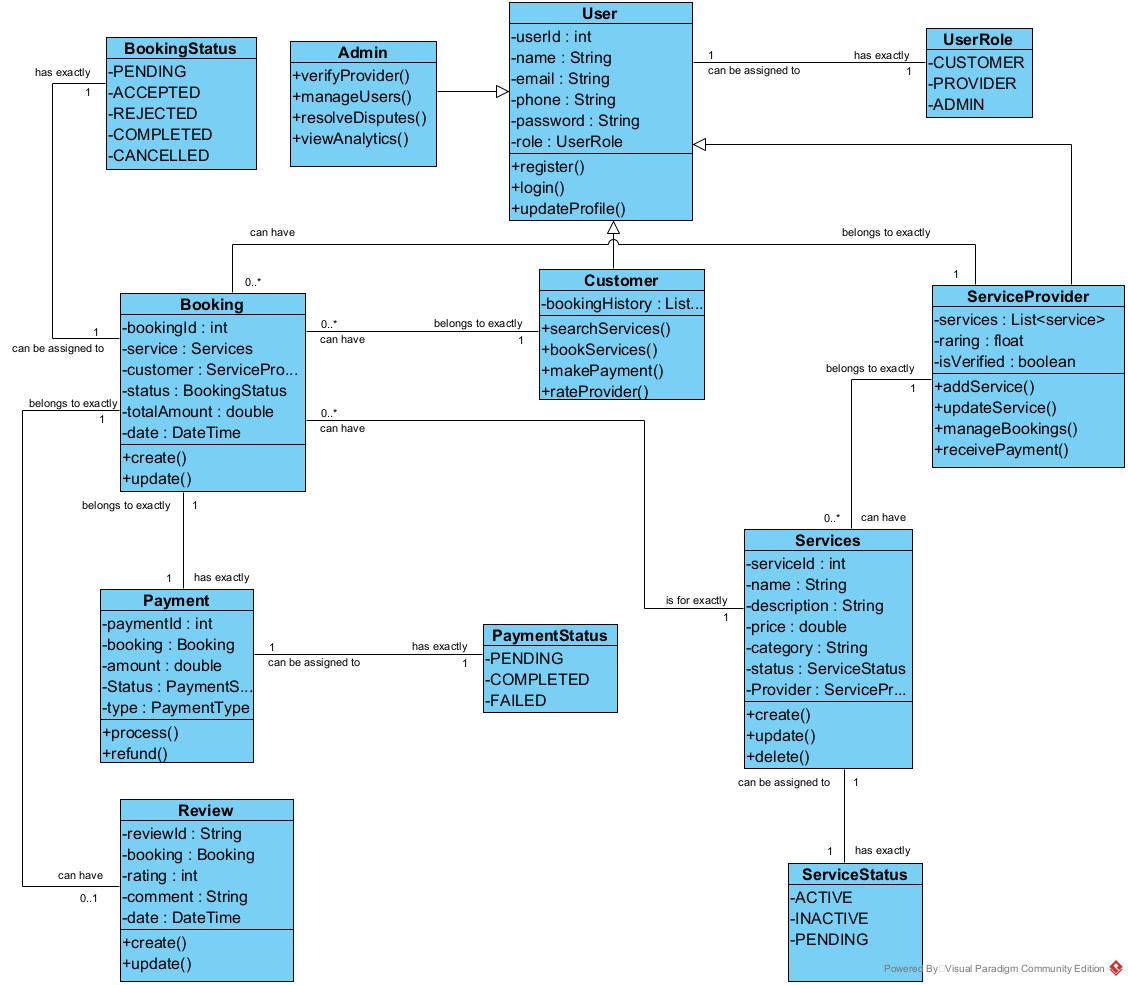


#### **Components of class diagram**

##### 

*Figure 23:Components of a class diagram*

* **Class Diagram**



##### *Figure 24:Class diagram of our system*

### **STATE MACHINE DIAGRAM**

#### **Definition**

State machine diagrams (or sometimes referred to as state diagram or state chart diagram) show the different states of an entity. State machine diagrams can also show how an entity responds to various events by changing from one state to another.

#### **Components of state machine diagram**

Table 16: Components of a state machine diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION AND MAIN PROPERTIES | NOTATION |
| State | A state represents situations during the life of an object |  |
| Transition | A solid arrow that represents the path between different states of an object. A state can have a transition that points back to itself. |  |
| Initial state | A filled circle followed by an arrow represents an object’s initial state. |  |
| Final state | An arrow pointed to a filled circle nested inside another circle represents an object’s final state. |  |
| Join | It is used to split transition or reduce two or more transitions into one |  |

* **Booking state machine diagram**

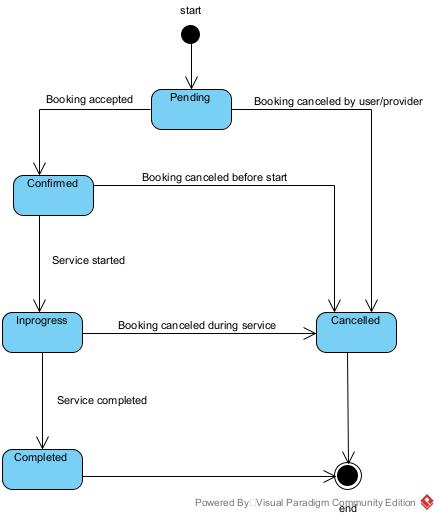


Figure 25:State machine Diagram for booking

* **Payment state machine diagram.**

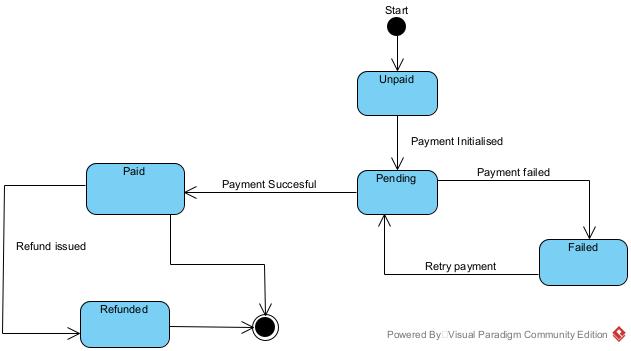


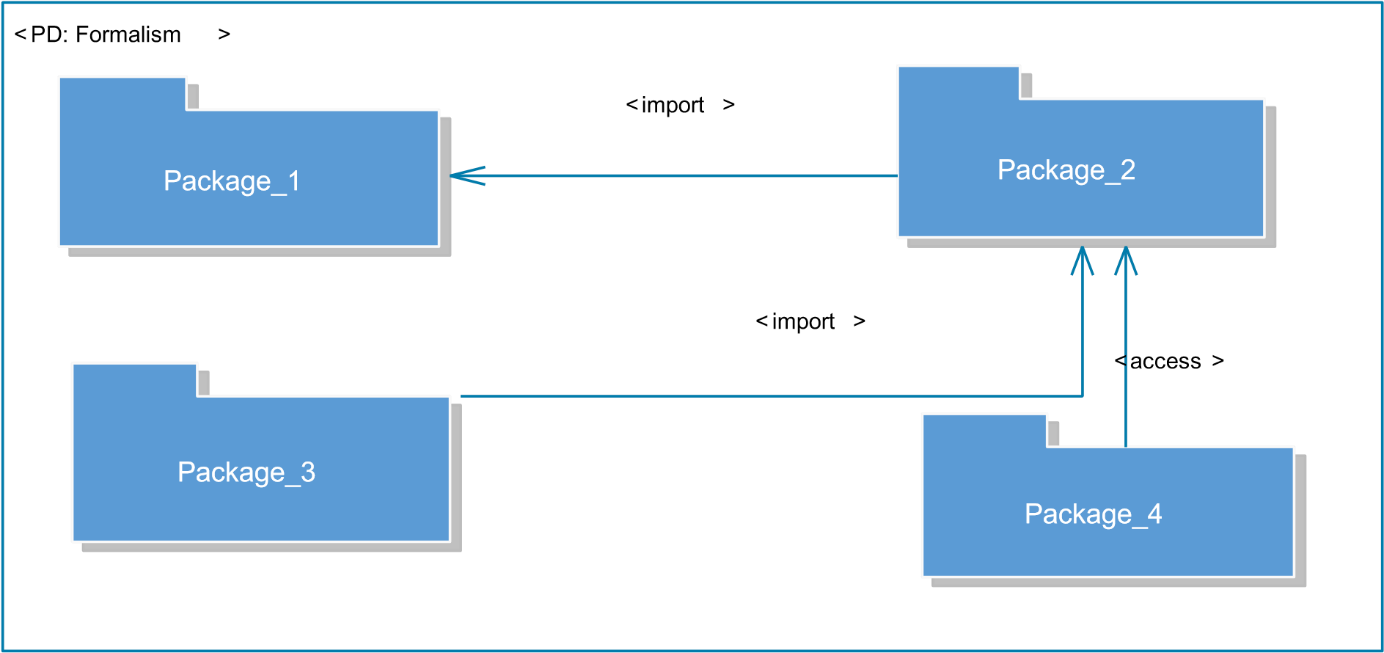
Figure 26:State machine Diagram for payment

### **PACKAGE DIAGRAM**

#### **Definition**

1. Package is a namespace used to group together elements that are semantically related and might change together. It is a general-purpose mechanism to organize elements into groups to provide better structure for system model. Owned members of a package should all be packageable elements. If a package is removed from a model, so are all the elements owned by the package. Package by itself is packageable element, so any package could also be a member of other packages.

#### **Formalism**



##### *Figure 27:Formalism of Package diagram*

#### **Components of a package diagram**

##### *Figure 28:Components of a package diagram*

* **Package diagram.**

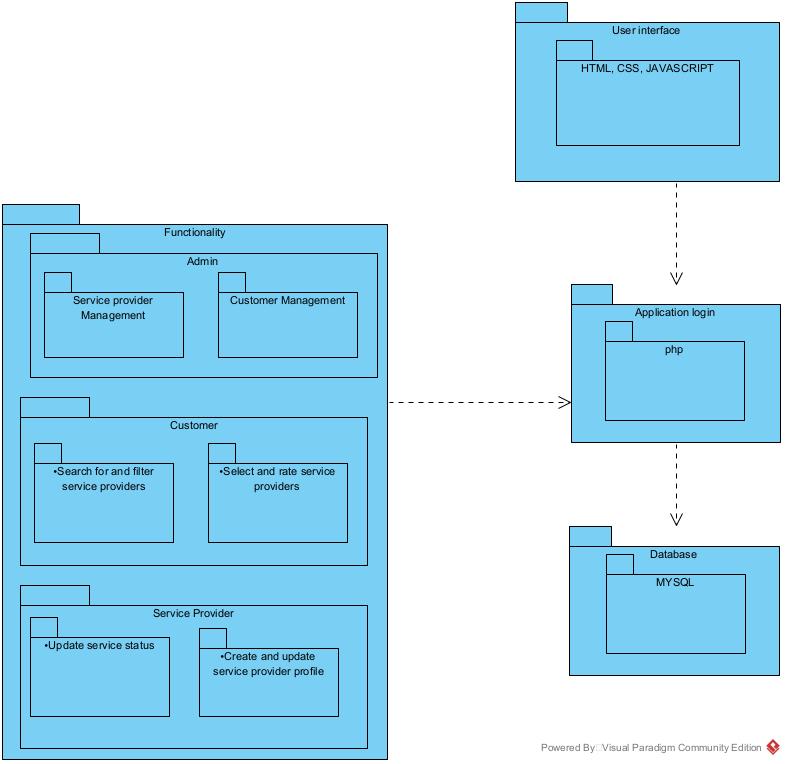


Figure 29:Package Diagram

### **CONCLUSION**

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

## **CHAPTER 5: REALIZATION PHASE**

**Preamble**

The realization document has as aim to present in details the significant points on how our application is developed. The aim here is to present the database, the tools used for the realization of our mobile and wen application, the architecture of our application, the programming language used and the result obtained.

**Content**

**INTRODUTION**

1. **DEPLOYMENT DIAGRAM**
2. **COMPONENT DIAGRAM**
3. **ARCHITECTURE OF THE APPLICATION**
4. **TECHNICAL CHOICE**

**CONCLUSION**

### **INTRODUTION**

The realization document is the one that contains all the information relating to the actual implementation of a project. The latter aim to present the elements involved in the development of our standalone application, this by presenting the working environment and the various software that will come into play. In this part, we will take a look at the development tools, the programming language we have used and the presentation of the physical architecture of our theme

#### **Components of deployment diagram**

Table 17: Table 16: Components of a deployment diagram

|  |  |  |
| --- | --- | --- |
| ELEMENTS | DESCRIPTION AND MAIN PROPERTIES | NOTATION |
| Node | A node is either a hardware or a software |  |
| Artefact | An artefact is a product of a software development process |  |
| Component | It represents a modular part of a system that encapsulates it content and whose manifestation is replaceable within its environment |  |
| Association | An association represents a communication path between nodes. |  |

* **Deployment Diagram.**

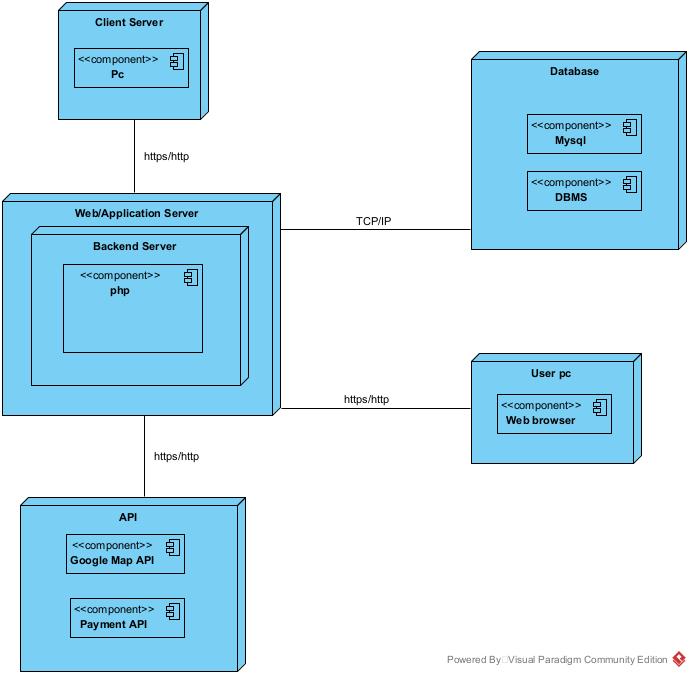


Figure 30:System deployment Diagram

### **COMPONENT DIAGRAM**

#### **Definition**

Component diagrams show the organization and dependencies involved in the implementation of a system. A Component diagram describes the organization and wiring of physical components in a system. Physical components are elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

#### **Formalism**



##### *Figure 31:Formalism of Component diagram*

#### **Components of component diagram**

Table 19 : Components of a component diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION AND MAIN PROPERTIES | NOTATION |
| Component | It represents a module part of a system that encapsulate it content and whose manifestation is replaceable within it environment |  |
| Interface | It is a small circle or semi-circle on a stick describing a group of operations used (required) or created (provided) by components. A full circle represents an interface created or provided by the component. A semi-circle represents an interface required, like a person’s input. |  |
| Port | These are square along the edge of a system or component. A port is often used to help expose the required and provided interfaces of a component. |  |

* **Web Component diagram.**



Figure 32: Web component Diagram

### **ARCHITECTURE OF THE APPLICATION**

#### **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 platforms. 3-tier architecture has the following layers:

##### **The user (presentation) tier**

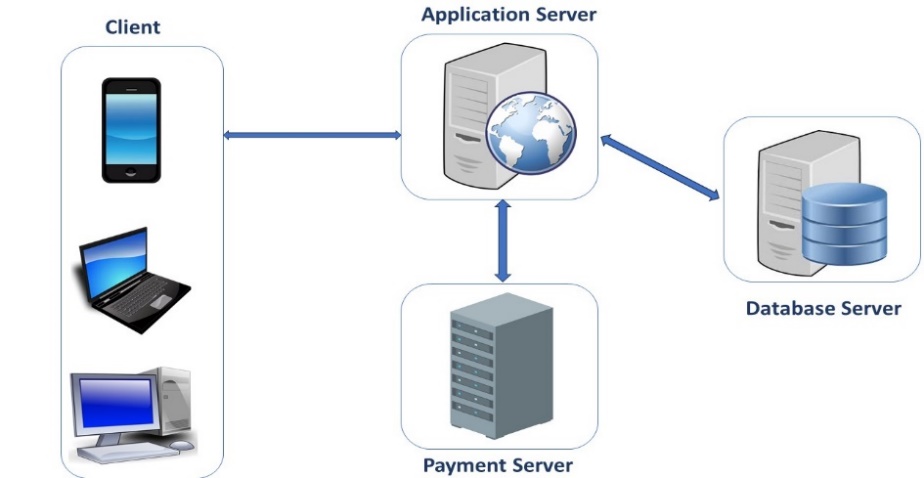
This is the tier which users can access directly. That is, this tier provides interfaces for the user and it is also responsible for transmitting users request to other tiers of the application and displaying the results of their actions back to them. It may be a web page or an operating system’s GUI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

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

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

##### **The database (data) tier**

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



##### *Figure 33:The N-tier architecture*

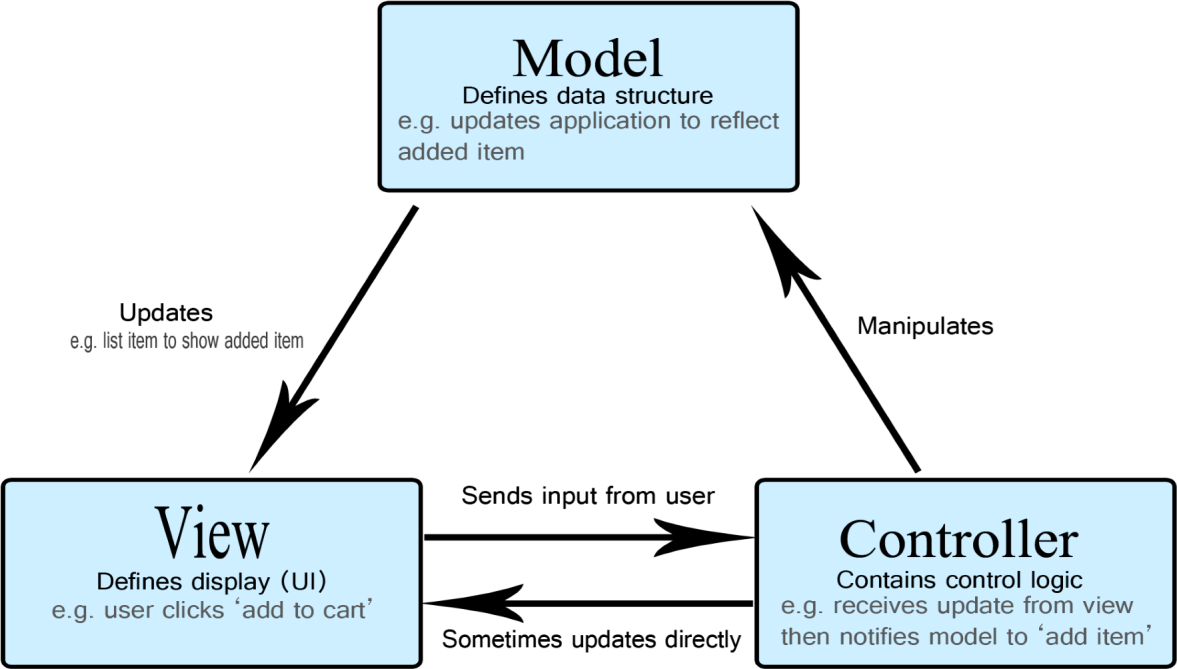
#### **Logical architecture**

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

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

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

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



##### *Figure 34:MVC design pattern*

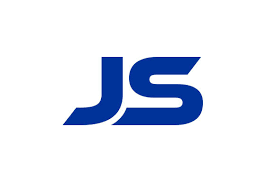
### **TECHNOLOGICAL CHOICE**

#### **Choice of languages used**

##### **Front end**

* **HTML**

HTML, or Hyper Text Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It structures web content and consists of a series of elements or tags that define the layout, formatting, and hyperlinks of a webpage. Key components of HTML include:

* **Elements:** Fundamental building blocks, usually represented by opening and closing tags (e.g., <p>...</p> for paragraphs).
* **Attributes:** Provide additional information about elements (e.g., <a href="https://example.com">Link</a>).
* **Headings:** Defined with tags like <h1> to <h6>, indicating the hierarchy of content.
* **Lists:** Ordered (<ol>) and unordered (<ul>) lists for organizing items.
* **Images and multimedia:** Embedding images (<img>), audio, and video using respective tags.
* **JavaScript**

JavaScript is a high-level, dynamic, and interpreted programming language primarily used for enhancing the interactivity and functionality of websites. It allows developers to create interactive elements, handle events, manipulate the Document Object Model (DOM), and communicate with servers asynchronously.

Key features of JavaScript include:

* **Client-Side Scripting:** JavaScript runs in the user's browser, enabling real-time updates without requiring a page reload.
* **Event Handling:** It can respond to user actions like clicks, form submissions, and keyboard events.
* **Asynchronous Programming:** JavaScript supports asynchronous operations, allowing tasks to run concurrently (e.g., using Promises and async/await).
* **Object-Oriented and Functional:** It supports both object-oriented and functional programming paradigms, enabling flexible coding styles.
* **Cross-Platform Compatibility:** JavaScript code can run on various devices and platforms, making it a versatile choice for web development.

##### **Back end**

**PHP (Hypertext Preprocessor)**

**PHP** (Hypertext Preprocessor) is a popular server-side scripting language designed primarily for web development. It can be embedded into HTML and is widely used to create dynamic web pages and applications. PHP is open-source, which means it is free to use and has a large community of developers contributing to its improvement.

**Advantages of PHP**

1. **Open Source**: PHP is free to use, which lowers the cost of development.
2. **Cross-Platform**: It runs on various platforms (Windows, Linux, macOS), making it versatile for different server environments.
3. **Easy to Learn**: PHP has a straightforward syntax that is easy for beginners to grasp, especially for those familiar with C or Java.
4. **Rich Library Support**: PHP has a wide range of built-in libraries and frameworks (like Laravel, Symfony) that expedite development.
5. **Database Integration**: It supports various databases (MySQL, PostgreSQL, SQLite), making it easy to work with data-driven applications.
6. **Community Support**: Being widely used, PHP has a large community that offers extensive documentation, tutorials, and forums for support.
7. **Performance**: PHP is optimized for web development and can handle a large number of requests efficiently.
8. **Security**: PHP provides several security features to protect against common web vulnerabilities, such as SQL injection and cross-site scripting (XSS).
9. **Flexible and Scalable**: PHP can be used for small-scale projects as well as large enterprise-level applications, providing flexibility in development.
10. **Integration Capabilities**: It easily integrates with various web technologies, including HTML, CSS, JavaScript, and APIs.

### **CONCLUSION**

In this phase, the implementation of the proposed solution was the main subject. It began with the presentation of the various architectures, which has permitted us to present in a general manner the components that are attached to the DBMS on which our software is based, the tools used in order to put the software in place and the choice of the programming language. In the next phase, we are going to provide a functionality test.

## **CHAPTER 6: FUNCTIONALITY TEST**

**Preamble**

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

**Content**

INTRODUCTION

1.

APPLICATION FUNCTIONALITIES

2.

TESTS SHOWCASES

CONCLUSION

### **INTRODUTION**

The test of functionalities phase helps us to know more about the solution we are building be it web or mobile. It provides the different functionalities or modules found in our application and how they are beneficial to the users. Hence, we are going to explore the different functionalities present in SERVIGO.

1. **APPLICATION FUNCTIONALITIES**

* **Authentication**

Enables users (service seekers, service providers, and administrators) to securely access their dashboards. New users can register for an account.

* **Service Request Creation**

Allows service seekers to create and submit requests for specific services they need.

* **View Service Providers List**

Enables administrators and users to view a comprehensive list of all registered service providers on the platform.

* **View Users List**

Allows administrators to view the list of all service seekers registered in the system.

* **Notification Management**

Administrators can send notifications (e.g., updates, alerts, promotions) to service providers and users, who receive them in real time.

* **Assign Service Requests :**

Administrators can assign specific service requests to appropriate service providers based on location, availability, or expertise.

* **Manage Reports**

Administrators can generate and view reports related to users, service requests, transactions, and platform activity.

* **View Transaction and Service History**

All users can view their past and current service requests, transactions, and interactions for transparency and record-keeping.

* **Rate and Review Service Providers**

Service seekers can rate and leave feedback for service providers after a service is completed, helping to build trust and improve quality.

* **Profile Management**

Users and service providers can update their personal or business profiles, including contact information and service details.

* **Log Out**

Allows users to securely end their session and log out of the application.

### **CONCLUSION**

In the test of functionalities phase, we thoroughly explored the diverse features integrated into SERVIGO App, demonstrating how each functionality enhances the experience for all users. By addressing the needs of different user roles, our application ensures accessibility, efficiency, and a user-friendly interface. This phase is vital for understanding the full potential and value that SERVIGO App brings to its users. As we transition to the final phase—installation and user guide—we will provide clear, step-by-step instructions to help users seamlessly install and navigate the app, ensuring everyone can make the most of its advanced capabilities.

## **CHAPTER 7: USER GUIDE**

**Preamble**

Once a solution is analyzed, conceived, and deployed, it is now necessary to produce a guide for its users. This part of our document concerns the installation of the necessary tools to set up an environment for the platform in local and the user guide.

**Content**

INTRODUCTION

I.

INSTALLATION GUIDE

II.

PERSPECTIVE

CONCLUSION

### **INTRODUTION**

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 its 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 show case our application by viewing the different screens and some brief explanation.

### **INSTALLATION GUIDE**

The installation of the tools for the functioning of our project will be in few steps, that is the installation of XAMPP server, and then the importation of our project.

1. Installation of XAMPP server

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP opensource package has been set up to be incredibly easy to install and to use.

* S**tep 1:** Download XAMPP from [www.apachefriends.org](http://www.apachefriends.org).
* **Step 2:** Run the .exe file.
* **Step 3:** Start the setup wizard.
* **Step 4:** Choose software components.
* **Step 5:** Choose the installation directory and start the installation process.
* **Step 6:** Complete installation.

Click next for the next two steps.





### **PERSPECTIVES**

Considering the fact that this project can be improved to a higher extend, as perspectives we can bring some ameliorations such as:

* Developing the mobile version of the app
* Implement a geo localization API
* Implement a complete payment API

### **CONCLUSION**

Having put in place the platform, it was not sufficient for we had to produce a manual that will help its various users. That is why we presented the different tools to be installed and how they are to be installed in order to run this application without any problem and how the users will use this platform once the environment is set up

# **GENERAL CONCLUSION**

Upon completion of my ENGINEERING DEPLOMA program, we were granted the opportunity to embark on an internship that served as a practical application of the knowledge we acquired throughout our three-year computer science training. This invaluable experience not only allowed us to put our skills into practice but also served as a gateway to the professional world, exposing us to the potential challenges we may encounter in our chosen field. The internship provided us with a comprehensive understanding of the issues faced by various companies within our society, equipping us with the foresight to prevent similar problems in our future endeavors. It is worth noting that this enriching internship experience played a crucial role in our ability to address the task at hand: **DESIGN AND IMPLEMENTATION OF A SMART SERVICE FINDER APPLICATION FOR CAMEROONIANS**. Looking ahead, we plan to implement updates and significant enhancements in the near future, with a focus on improving both the functionality and security of our platform.

# **ANEXES**

# **BIBLIOGRAPHY**

* The unified modelling Language User Guide by Graddy Booch et All.
* UML Distilled by Martin Fowler.
* Learn PHP in 24 Hours or Less by Dwight Robert.
* Object Oriented Modular Programming using HTML5, CSS by Steve Prettyman

# **WEBOGRAPHY**

* <https://www.google.com/search?q=uml+tutorial> ; for learning more about uml diagrams. 10th July 2025 11:AM
* <https://www.google.com/search?q=tuto+on+php&oq=tuto+on+php> ; to learn how to use php. 10th August 2025 2:PM
* <https://www.google.com/search?q=tuto+on+bostrap&oq=tuto+on+bostrap> ; to learn how to code the front end of the application with java scripts 25th  August 2025 8:AM.
* <https://www.google.com/search?q=tuto+on+2tup&oq=tuto+on+2tup> ; to know more about 2TUP process. 20th  September 2025 10:AM
* <https://www.w3schools.com/php/> ; To learn how to code in PHP 25th  August 2025 12:PM
* <https://www.youtube.com/results?search_query=php+html+crud+operations> ; to know more about CRUD (create, read, update, delete) operations. 1st September 2025 5:PM

# **TABLE OF CONTENT**

[**DEDICATION** i](#_Toc209466363)

[**ACKNOWLEDGEMENTS** ii](#_Toc209466364)

[**CONTEN** iii](#_Toc209466365)

[**LIST OF FIGURES** iv](#_Toc209466366)

[**LIST OF TABLES** v](#_Toc209466367)

[**LIST OF ABREVIATIONS** vi](#_Toc209466368)

[**ABSTRACT** vii](#_Toc209466369)

[**RESUME** viii](#_Toc209466370)

[**GENERAL INTRODUCTION** 1](#_Toc209466371)

[**PART 1: INSERTION PHASE** 3](#_Toc209466372)

[**INTRODUCTION** 4](#_Toc209466373)

[**I.** **WELCOME AND INSERTION** 5](#_Toc209466374)

[**II.** **GENERAL REPRESENTATION OF SHADERL** 5](#_Toc209466375)

[**1.** **Background:** 5](#_Toc209466376)

[**2.** **Mission** 5](#_Toc209466377)

[**3.** **Vision** 6](#_Toc209466378)

[**4.** **ACTIVITIES** 6](#_Toc209466379)

[**III-ORGANISATION OF SHADERL** 6](#_Toc209466380)

[**A.** **ADMINISTRATIVE ORGANISATION** 6](#_Toc209466381)

[**B.** **Functional organization** 9](#_Toc209466382)

[**IV.** **GEOGRAPHICAL LOCATION** 10](#_Toc209466383)

[**V.** **BRIEF PRESENTATION OF THE THEME** 11](#_Toc209466384)

[**CONCLUSION** 12](#_Toc209466385)

[**PART 2: TECHNICAL PHASE** 13](#_Toc209466386)

[**CHAPTER 1: THE EXISTING SYSTEM** 14](#_Toc209466387)

[**INTRODUTION** 15](#_Toc209466388)

[**I.** **DESCRIPTION OF THE EXISTING SYSTEM** 16](#_Toc209466389)

[II. **CRITICISMS OF THE EXISTING SYSTEM** 17](#_Toc209466390)

[**III.** **PROBLEMATIC** 18](#_Toc209466391)

[**IV.** **PROPOSED SOLUTION** 18](#_Toc209466392)

[**CONCLUSION** 19](#_Toc209466393)

[**CHAPTER 2: SPECIFICATION BOOK** 20](#_Toc209466394)

[**INTRODUTION** 21](#_Toc209466395)

[**I.** **CONTEXT AND JUSTIFICATION** 22](#_Toc209466396)

[**II.** **OBJECTIVES OF THE PROJECT** 22](#_Toc209466397)

[**III.** **EXPRESSION OF NEEDS** 24](#_Toc209466398)

[**IV.** **PROJECT PLANNING** 26](#_Toc209466399)

[**V.** **ESTIMATION OF RESOURCES NEEDED** 29](#_Toc209466400)

[**VI.** **CONSTRAINS** 32](#_Toc209466401)

[**VII.** **DERIVABLES** 32](#_Toc209466402)

[**CONCLUSION** 33](#_Toc209466403)

[**CHAPTER 3: ANLYSIS PHASE** 34](#_Toc209466404)

[**INTRODUTION** 35](#_Toc209466405)

[**I.** **PRESENTATION OF THE MODELING APPROACH** 36](#_Toc209466406)

[**II.** **PRESENTATION OF UML AND 2TUP** 37](#_Toc209466407)

[**III.** **CHOICE OF THE ANALYSIS APPROACH** 44](#_Toc209466408)

[**IV.** **MODELING OF THE PROPOSED SOLUTION** 45](#_Toc209466409)

[**CONCLUSION** 66](#_Toc209466410)

[**CHAPTER 4: CONCEPTION PHASE** 67](#_Toc209466411)

[**INTRODUTION** 68](#_Toc209466412)

[**1.** **CLASS DIAGRAM** 69](#_Toc209466413)

[**2.** **STATE MACHINE DIAGRAM** 72](#_Toc209466414)

[**3.** **PACKAGE DIAGRAM** 74](#_Toc209466415)

[**CONCLUSION** 76](#_Toc209466416)

[**CHAPTER 5: REALIZATION PHASE** 77](#_Toc209466417)

[**INTRODUTION** 78](#_Toc209466418)

[**1.** **COMPONENT DIAGRAM** 81](#_Toc209466419)

[**2.** **ARCHITECTURE OF THE APPLICATION** 84](#_Toc209466420)

[**3.** **TECHNOLOGICAL CHOICE** 86](#_Toc209466421)

[**CONCLUSION** 90](#_Toc209466422)

[**CHAPTER 6: FUNCTIONALITY TEST** 91](#_Toc209466423)

[**INTRODUTION** 92](#_Toc209466424)

[**CONCLUSION** 95](#_Toc209466425)

[**CHAPTER 7: USER GUIDE** 96](#_Toc209466426)

[**INTRODUTION** 97](#_Toc209466427)

[**1.** **INSTALLATION GUIDE** 98](#_Toc209466428)

[**2.** **PERSPECTIVES** 100](#_Toc209466429)

[**CONCLUSION** 101](#_Toc209466430)

[**GENERAL CONCLUSION** 102](#_Toc209466431)

[**ANEXES** ix](#_Toc209466432)

[**BIBLIOGRAPHY** x](#_Toc209466433)

[**WEBOGRAPHY** xi](#_Toc209466434)

[**TABLE OF CONTENT** xii](#_Toc209466435)