

REPUBLIC OF CAMEROON

Peace - Work - Fatherland



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

P.O. Box: 13719 Yaoundé

Tel: +237.242.729.957 Fax: 22729958

E-mail: contact@iaicameroun.com

Web site: www.iaicameroun.com

PERSONALISED SOFTWARE ENGINEERING

THEME: CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN
CAMEROON

Written by: **TSEMBON PERCY LINKWE**, student in SE3 at the African Institute of Computer
Sciences (AICS)-Cameroon Paul Biya

Academic Supervisor:

Mr. AGBOR Donald Anderson

Lecturer at AICS-Cameroon

Academic Year 2023 - 2024



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



DEDICATION

THIS BOOK IS DEDICATED TO MY ENTIRE
FAMILY



ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to everyone who has supported me throughout my internship period. Without your help, encouragements, advices, teachings, and precious time, I would not have been able to complete this project.

I would like to give special thanks to the following people:

- The almighty God who made it possible without whom nothing would have been done.
- **Mr. Armand Claude ABANDA**, the Resident Representative of the African Institute of Computer Sciences, for his unwavering support and guidance.
- **Mr. AGBOR Donald Anderson**, my academic supervisor, for her warm welcome, encouragements, availability, tolerance, advices and moral support throughout the internship period.
- All our teachers at the African Institute of Computer Sciences for their support throughout our training.
- My family most especially my parents for their guidance and support without which we could not attain the end of this work.
- All our classmates and friends at AICS for their knowledge, ideas, collaboration and most importantly the unforgettable moments spent together.



CONTENTS

DEDICATION	i
ACKNOWLEDGMENTS	ii
CONTENTS.....	iii
LIST OF FIGURES	iv
LIST OF TABLES	vi
ABBREVIATIONS.....	vii
ABSTRACT.....	viii
RESUME	ix
GENERAL INTRODUCTION.....	1
PART 1: EXISTING SYSTEM.....	2
PART 2: SPECIFICATION BOOK	8
PART 3: ANALYSIS PHASE	21
PART 4: CONCEPTION PHASE.....	55
PART 5: REALIZATION PHASE	72
PART 6: INSTALLATION GUIDE AND USER GUIDE	82
GENERAL CONCLUSION	95
BIBLIOGRAPHY	i
WEBOGRAPHY	ii
TABLE OF CONTENT	iii



LIST OF FIGURES

Figure 1: UML 2.5 diagrams overview (source: https://creately.com/blog/diagrams/uml-diagram-types-examples/)	25
Figure 2: Two Track Unified Process (source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing)	28
Figure 3: Use case diagram formalism	31
Figure 4: General use case diagram	34
Figure 5: Manage hotel use case diagram	35
Figure 6: Activity diagram formalism	39
Figure 7: Authenticate activity diagram	42
Figure 8: Create account activity diagram	43
Figure 9: Make payment activity diagram	44
Figure 10: Booking activity diagram	45
Figure 11: Communication diagram formalism	46
Figure 12: Authenticate Communication diagram	48
Figure 13: Create account communication diagram	48
Figure 14: Make Payment communication diagram	49
Figure 15: Sequence diagram formalism	50
Figure 16: Authenticate Sequence diagram	52
Figure 17: Make Payment sequence diagram	53
Figure 20: Hardware diagram	58
Figure 21: : n-tier Architecture (Source: https://www.pinterest.es/pin/752241943987565648/)	59
Figure 22: The MVC architecture (source: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/)	60
Figure 23: Class diagram formalism	61
Figure 24: Class diagram	63
Figure 25: Formalism of a state machine diagram	65
Figure 26: State machine diagram of authenticate	67
Figure 27: Package diagram formalism	68



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



Figure 28: CamerStay package diagram	70
Figure 29: Formalism of Deployment diagram	75
Figure 30: CamerStay deployment diagram	77
Figure 31: Component diagram formalism.....	78
Figure 32: iCivil component diagram	80
Figure 33: MongoDB logo.....	85
Figure 34: Downloading MongoDB community server	85
Figure 35: Launching the installation wizard	86
Figure 36: Click next	86
Figure 37: Agreement License.....	87
Figure 38: Select complete setup	87
Figure 39: Run service network user	88
Figure 40: Selecting MongoDB compass and click next.....	88
Figure 41: Click on install to start installation.....	89
Figure 42: Click on finish to complete installation.....	89
Figure 43: Home Page	90
Figure 44: Add hotel page.....	91
Figure 45: Search page.....	92
Figure 46:Bookings page	93
Figure 47:Create account page.....	94



LIST OF TABLES

Table 4: Software resources (source: MERCURIALE-2023).....	15
Table 5: Hardware resources (source: MERCURIALE-2023)	16
Table 6: Human resources (source: Mercurial 2022).....	17
Table 7: Global estimation	17
Table 8: List of participants	19
Table 9: Comparison between MERISE and UML	26
Table 10: Component of use case diagram	32
Table 11: Actors of our system.....	33
Table 12: Textual description formalism	36
Table 13: Textual description of Authenticate	37
Table 14: Textual description of Create account.....	38
Table 15: Component of an Activity diagram.....	40
Table 16: Component of a communication diagram.....	47
Table 17: Component of sequence diagram.....	51
Table 18: Components of class diagram	62
Table 19: Component of state machine diagram.....	66
Table 20: Component of package diagram	69
Table 21: Component of deployment diagram.....	76
Table 22: Component of deployment diagram.....	79



ABBREVIATIONS

- **2TUP:** Two Track Unified Process
- **AICS:** African Institute of Computer Sciences
- **API:** Application Programming Interface
- **CRVS:** Civil Registration And Vital Statistics
- **CR:** Civil Registration
- **CSS:** Cascading Style Sheet
- **DBMS:** Database Management System
- **GUI:** Graphical User Interface
- **HTML:** Hypertext Markup Language
- **HTTP:** Hypertext Transfer Protocol
- **IDE:** Integrated Development environment
- **JSON:** JavaScript Object Notation
- **OS:** Operating System
- **UML:** Unified Modelling Language



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



ABSTRACT

Humanity has been constantly evolving since the agricultural revolution, through the industrial revolution, and now towards the information and communication technology revolution. This exponential evolution has occurred in various domains, including transport, agriculture, communication, education, health, and many others, resulting in numerous innovations. Therefore, technology is crucial for envisioning the future. The African Institute of Computer Sciences Paul Biya Technological Centre of Excellence aims to position Cameroon as a major player in the revolution by providing students with practical skills through personalized projects. These projects have the objective of enabling students to gain practical knowledge in addition to their academic theoretical learning. Two weeks into my personalized project, I was assigned the theme of "CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON". I subsequently conducted feasibility studies and analyzed the system using the UML methodology associated with the 2TUP process, progressing through the conception and realization phases. The CamerStay application was created through the use of MongoDB, Express.js, React.js, and Node.js to achieve a versatile, lightweight, and secure solution for optimal user experience. I successfully launched an application named CamerStay.

Keywords:

- Hotel Booking
- Digital Transformation
- Tourism
- Technology



RESUME

L'humanité n'a cessé d'évoluer depuis la révolution agricole, en passant par la révolution industrielle, et maintenant vers la révolution des technologies de l'information et de la communication. Cette évolution exponentielle s'est produite dans divers domaines, y compris le transport, l'agriculture, la communication, l'éducation, la santé, et bien d'autres, entraînant de nombreuses innovations. Par conséquent, la technologie est cruciale pour envisager l'avenir. L'Institut Africain d'Informatique Centre Technologique d'Excellence Paul Biya vise à positionner le Cameroun comme un acteur majeur de cette révolution en fournissant aux étudiants des compétences pratiques grâce à des projets personnalisés. Ces projets ont pour objectif de permettre aux étudiants d'acquérir des connaissances pratiques en plus de leur apprentissage théorique académique. Deux semaines après le début de mon projet personnalisé, on m'a assigné le thème de "CONCEPTION ET REALISATION D'UNE APPLICATION DE RESERVATION D'HOTELS AU CAMEROUN ". J'ai ensuite mené des études de faisabilité et analysé le système en utilisant la méthodologie UML associée au processus 2TUP, progressant à travers les phases de conception et de réalisation. L'application CamerStay a été créée en utilisant MongoDB, Express.js, React.js, et Node.js pour obtenir une solution polyvalente, légère et sécurisée offrant une expérience utilisateur optimale. J'ai réussi à lancer une application nommée CamerStay.

Mots-clés

- Réservation d'Hôtels
- Transformation Digitale
- Tourisme
- Technologie



GENERAL INTRODUCTION

Over the years, computer science has progressed despite the challenges encountered. Nowadays, people worldwide are facing diverse obstacles. Therefore, a significant priority involves addressing these issues and meeting clients' requirements in different aspects of life. Companies require well-trained personnel in Information Technology and Communication. That's why the Africa Institute of Computer Sciences (AICS), a pioneering institution in the field, has established a personalized project programme for Level 2 students to apply the skills they've learnt in school to a practical setting, improve their abilities, and help companies overcome challenges or grow.

My project, titled "DIGITAL TRANSFORMATION OF HOTEL BOOKING IN CAMEROON: ENHANCING TOURISM THROUGH A SEAMLESS AND USER-FRIENDLY BOOKING PLATFORM," was divided into 7 main segments

1. **Existing System**, Here we present the current system in place.
2. **The specification book**, outlines the needs of future system users and identifies various constraints and project requirements.
3. **The analysis phase**, involves presenting the chosen analysis method and sketching the first model diagrams of the system to be realised.
4. **The conception phase**, focuses on studying analysed systems in detail to identify their real-world components.
5. **The realization phase**, involves implementing and making the analysed and conceived system available to users.
6. **The user guide**, which instructs the users on how to install and use the system



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



PART 1: EXISTING SYSTEM



PREAMBLE

This chapter will present and elucidate our central theme, which is the primary emphasis and objective of our project. We shall also expound on the predicament that prompted us to devise our proposed solution, and detail the characteristics and operations of the software we have planned to construct.

CONTENT

INTRODUCTION

I THEME PRESENTATION

II STUDY OF THE EXISTING SYSTEM

III LIMITATION OF THE EXISTING SYSTEM

IV PROBLEMATIC

V PROPOSED SOLUTION

CONCLUSION



INTRODUCTION

In this part of our report, we will explain our theme. We will also study the current system and identify its strengths and weaknesses. Based on this analysis, we will present the problem statement, which is the main challenge or gap that the current system faces. Finally, we will propose a solution that addresses this problem and improves the system.



I. THEME PRESENTATION

The theme of this project is the " CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON " with the primary objective of enhancing the tourism sector through a seamless and user-friendly booking platform. This project aims to bridge the gap between tourists and hotel services, providing an efficient and convenient solution to booking needs, ultimately boosting tourism and hospitality in Cameroon.

II. STUDY OF THE EXISTING SYSTEM

The existing hotel booking systems in Cameroon predominantly rely on traditional methods such as phone calls, emails, and walk-in reservations. Some hotels have basic online presence but lack integrated booking systems. These existing systems have several inefficiencies, including manual data handling, limited availability information, and poor customer engagement.

III. LIMITATIONS OF THE EXISTING SYSTEM

- **Inefficiency:** Manual booking processes are time-consuming and prone to errors.
- **Limited Accessibility:** Lack of a centralized online platform limits the reach to potential customers, especially international tourists.
- **Poor User Experience:** The absence of a user-friendly interface makes it difficult for customers to make reservations easily.
- **Data Management:** Inefficient data handling leads to issues in availability tracking and customer management.
- **Marketing and Promotion:** Limited online presence hinders effective marketing and promotional activities.



IV. PROBLEMATIC

Upon examining the current hotel booking systems in Cameroon, we have encountered significant challenges. This prompts the fundamental question: HOW CAN WE STREAMLINE AND ENHANCE THE PROCESS OF HOTEL BOOKING TO IMPROVE TOURISM THROUGH A SEAMLESS AND USER-FRIENDLY PLATFORM?

V. PROPOSED SOLUTION

The proposed solution is the development of a comprehensive online booking platform named CamerStay. This platform will:

- Integrate all hotels into a centralized system.
- Provide real-time availability and booking services.
- Offer a user-friendly interface for both hotel managers and customers.
- Implement secure data management and payment processing systems.
- Facilitate effective marketing and promotion through digital channels.



CONCLUSION

The project theme plays a crucial role in establishing the work scope and identifying the system. Conducting a thorough analysis of the present system is equally important to understand its limitations and existing issues, prior to implementing any improvements. Once the analysis is done, we can frame the problem and offer a well-formulated solution to tackle the identified issues.



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



PART 2: SPECIFICATION BOOK



PREAMBLE

The specification book serves as a contractual document before the application development process commences. It succinctly outlines the expectations of the project owner and the respective responsibilities of all parties involved. It provides a specific and detailed description of the topic and user requirements, as well as the necessary conditions for project execution. The project's success will depend on careful allocation and management of the various resources available, including physical, financial, and human assets. Therefore, this document outlines the criteria that the developer must adhere to in order to ensure the success of the project.

CONTENT

INTRODUCTION

I CONTEXT AND JUSTIFICATION

II OBJECTIVES OF THE PROJECT

III EXPRESSION OF NEEDS

IV PROJECT PLANNING

V ESTIMATED COST OF THE PROJECT

VI PROJECT CONSTRAINTS

VII LIST OF PARTICIPANTS AND DELIVERABLES

CONCLUSION



INTRODUCTION

The report's specification book enables us to provide comprehensive details about our topic, enhancing our understanding and improving its chances of success. To establish the project's scope, we will describe the context of our topic and identify the issues within it that we aim to solve. Once the solution is proposed, we will outline the objectives we have set for ourselves and for the project. Also, we will assess the functional and non-functional requirements of our system before designing a plan to ensure timely completion of the project and determine the financial resources needed. Subsequently, we will outline the expected project deliverables at the conclusion of the project.



I. CONTEXT AND JUSTIFICATION

1. CONTEXT:

The context of this project lies in the current state of hotel booking systems in Cameroon, which predominantly rely on traditional methods such as phone calls, emails, and walk-in reservations. These methods often result in inefficiencies and missed opportunities for both tourists and hotel operators. With the global shift towards digital solutions in various sectors, there is a pressing need to modernize Cameroon's hospitality industry by introducing a centralized, user-friendly online booking platform.

2. JUSTIFICATION

The justification for this project stems from the potential benefits it offers to both tourists and the local hospitality sector. By implementing a digital transformation of hotel booking, Cameroon can:

- Enhance accessibility and convenience for tourists seeking accommodations.
- Improve operational efficiency for hotels by automating booking processes and reducing administrative overhead.
- Stimulate economic growth in the tourism sector by attracting more visitors through improved service delivery and online visibility.



II. OBJECTIVES OF THE PROJECT

1. General Objective

The overarching aim of our endeavour is to digitize and streamline the process of producing and recording birth and death certificates and vital events.

2. Specific Objectives

- Enable tourists to easily search for and book accommodations online.
- Improve operational efficiency for hotels by automating booking processes.
- Enhance customer satisfaction through a user-friendly interface and real-time availability updates.
- Facilitate secure and efficient payment processing for bookings.
- Promote Cameroon's tourism sector by enhancing online visibility and accessibility for international tourists.
- Provide data analytics capabilities to hotels for better decision-making and planning.
- Implement robust data security measures to protect user information.
- Ensure scalability and adaptability of the platform to accommodate future growth in the hospitality industry.
- Reduce administrative costs and streamline management processes for hotels.
- Foster collaboration between hotels and tourism agencies to promote local attractions and cultural experiences.

Project Characteristics:

- Project Name: CamerStay
- Project Target: Hotels in Cameroon
- Technical Specification: Web application



III. EXPRESSION OF NEEDS

1. FUNCTIONAL NEEDS

In this context, functional requirements describe what the platform should do to enhance the hotel booking experience in Cameroon. The modules include:

Tourists:

- **Search and Book:** Tourists should be able to search for hotels based on various criteria (location, price, amenities) and easily book accommodations online.

Hotels:

- **Manage Listings:** Hotel managers should have the ability to manage their property listings, update availability, and set pricing.

Payment Gateway Integration:

- **Secure Transactions:** Integration with secure payment gateways to facilitate online payments for bookings.

Security Measures:

- **Data Protection:** Implement robust security measures to safeguard user data and transactions.



2. NON-FUNCTIONAL NEEDS

The quality attributes of a software system are specified to evaluate the software application's Performance, Responsiveness, Usability, Security, Portability, and other non-functional standards that determine its success. Failure to meet non-functional requirements can lead to an unsatisfactory system that fails to meet user needs.

Performance, Scalability and Security

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

Scalability accesses the highest workloads under which the system will still meet the performance requirements. In this project we will mainly leverage the power of cloud storage and third-party API's.

- ✚ The application should have a friendly user interface (UI) and should be easy to use.
- ✚ The code should be clear to facilitate future development and improvement.
- ✚ The web application should be resizable when opened on any device (android phone, computer, tablet web browsers)

Security is also one of the most important aspects of any system, especially the ones dealing with sensitive user's information.

- ✚ The application should provide a strong security mechanism to reassure user's that they can trust it for their sensitive information.
- ✚ Regular Data Backups: Perform regular backups of the system's data to ensure that critical information is not lost in case of hardware failures or other incidents.



IV. ESTIMATED COST OF THE PROJECT

a. Software Resources

Table 1: Software resources (source: MERCURIALE-2023)

RESOURCES	DESIGNATION	USAGE	QUANTITY	UNIT COST (FCFA)
Formatting	MICROSOFT OFFICE PROFESSIONAL 2016	Writing of the report	1	355465
Web browser	Google Chrome	Access and view web pages on the internet	1	Freeware
Code Editor	Visual Studio Code	For writing the code of the application	1	Freemium
Project Planning	Gannt Project	For building a Gantt chart	1	Freemium
Graphic Editor	Icogram	For building a geographical location of the enterprise	1	Freemium
UML Analysis	Visual Paradigm	For drawing UML diagrams	1	Community Version
Version Control	Git & GitHub	For tracking and managing changes to software code	1	Free Version
TOTAL 1			7	355465

b. Hardware Resources



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



Table 2: Hardware resources (source: MERCURIALE-2023)

RESOURCES	HARDWARE	QUANTITY	UNIT COST (FCFA)
Computer	DELL LATITUDE 7490 CORE i7 vPro 8 th Gen, HARD DISK 500 SSD, RAM 16GB	1	400 000
Printer	Printers	1	546 250
Network	Local network installation	1	300 000
Removable Disk	USB HP 2.2 FLASH DRIVE V225W - 16 GB	1	17250
Smartphone	Sony Xperia X2Z Compact	1	50 000
Network	Local network installation	1	300 000
Total 2		6	1 613 500



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



a. Human Resources

Table 3: Human resources (source: Mercurial 2022)

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	30	01	10 000	140 000
Programmer	30	01	15 000	450 000
Tester	14	02	10 000	140 000
Margin error	/	/	/	1,475, 000
TOTAL 3				4 840 000

b. Global Estimation

Table 4: Global estimation

TOTAL 1 (FCFA)	TOTAL 2 (FCFA)	TOTAL 3 (FCFA)	OVERALL TOTAL (FCFA)
355 465	1 613 500	4 840 000	6 808 965
SIX MILLION EIGHT HUNDRED AND EIGHT THOUSAND NINE HUNDRED AND SIXTY-FIVE			



V. PROJECT CONSTRAINTS

1. Technical constraint

We possess robust tools that ensure a minimum level of security, extensibility, and excellent scalability for the development of our system. Additionally, adherence to all technical standards during the programming phase will result in improved performance and reduced execution time, making the choice of development technologies crucial.

3. Cost constraint

The implementation of our project will necessitate expenses on manpower, materials, and software, amounting to a total of 6 808 965 FCFA.



VI. LIST OF PARTICIPANTS AND DELIVRABLES

1. LIST OF PARTICIPANTS

Table 5: List of participants

NAME	FUNCTION	ROLE
Mr. AGBOR Donald Anderson	Follows up student at the academic level	Academic Supervisor
TSEMBOM PERCY LINKWE	AICS Student Intern	Student at AICS

2. DELIVRABLES

1. In project management, any component materializing the result of a realization service is called a deliverable. In the case of our project, the deliverables are: A report composed of the following document
 - ❖ The Existing System
 - ❖ The specification book;
 - ❖ The analysis phase;
 - ❖ The conception phase;
 - ❖ The realization phase;
 - ❖ The user guide;
2. A CD-ROM
 - ❖ The source code of the software
 - ❖ Database
 - ❖ A soft copy of the report
 - ❖ A corresponding PowerPoint



CONCLUSION

In conclusion, this part has come to a close, wherein we have effectively listed and expounded on our goals. The specifications book provided us with the means to present the project's different stakeholders, as well as the prerequisites and tentative schedule essential for its realization. Moving forward, we will directly proceed with the analysis phase, where we will utilise a modelling language and a unified approach to exemplify our system. Furthermore, we will carry out a comparative evaluation of UML and Merise methodologies.



PART 3: ANALYSIS PHASE



PREAMBLE

The primary aim of the analysis phase is to comprehend the user's requirements, define the scope of study and gain a thorough understanding of the system under examination. To accomplish this goal, we will employ the Unified Modelling Language (UML) with the 2 Track Unified Process (2TUP) methodology to analyse the system. Our approach will involve a comparison between UML and MERISE and various other unified processes. Lastly, we will introduce a solution model for our proposed software.

CONTENT

INTRODUCTION

I COMPARATIVE STUDY OF UML AND MERISE

II CHOICE OF THE ANALYSIS METHOD

III JUSTIFICATION OF THE ANALYSIS METHOD

III MODELLING OF THE PROPOSED SOLUTION IV

CONCLUSION



INTRODUCTION

System development can be thought of as having two major components: System analysis and system design which both help in understanding the details of the existing system or the system to be designed. The analysis and design of information systems has most of the time vocation to allow the creation of databases, which must represent as closely as possible the reality of the field studied thus requiring the use of a design method. This is why our choice will be directed on the UML method as it offers much to developers seeking a user-centered approach and / or a wide scope in design. This part of the report consists of the comparative study of UML and MERISE, unified processes and finally the various diagrams that meet the functional need requirements.



I. COMPARATIVE STUDY OF UML AND MERISE

a. MERISE

MERISE stands for “Méthode d’Etude et de Réalisation Informatique pour des Systèmes d’Entreprise”. Although it is prescriptive to some extent, MERISE permits the participation of end users and senior management as well as data processing professionals in its decision cycle. MERISE is a method for designing, developing and carrying out IT projects. The goal of this method is to achieve the design of an information system. The MERISE method is based on the separation of data and processing to be carried out in several conceptual and physical models. The essentials of the approach lie in its three cycles: the decision cycle, the life cycle and the abstraction cycle, which cover data and process elements equally. The separation of data and processing ensures longevity in model. Indeed, the arrangement of data does not have to be often overhauled, while treatments are more frequently.

b. UML

UML (Unified Modelling Language) is a standard notation for the modelling of real world objects as a first step in developing an object-oriented design methodology. Its notation is derived from and unifies the notations of three object-oriented design and analysis methodologies: Grady Booch's methodology for describing a set of objects and their relationships, James Rumbaugh's Object-Modelling Technique (OMT), Ivar Jacobson's approach which includes a use case methodology. Other ideas also contributed to UML, which was the result of a work effort by Booch, Rumbaugh, Jacobson, and others to combine their ideas, working under the sponsorship of Rational Software. UML captures information about the static and dynamic view of a system. UML 2.0 comprises of 13 diagrams which represent the different views of a system. The 13 diagrams can be subdivided into two, Static or structural and Dynamic diagrams. These diagrams include;

1. STATIC OR STRUCTURAL DIAGRAMS

- ✚ Class diagram
- ✚ Object diagram
- ✚ Component diagram

- ✚ Deployment diagram
- ✚ Package diagram
- ✚ Profile diagram

2. BEHAVIORAL PR DYNAMIC DIAGRAMS

- ✚ Use case diagram
- ✚ Activity diagram
- ✚ State machine diagram
- ✚ Sequence diagram
- ✚ Communication diagram
- ✚ Global Interaction diagram
- ✚ Timing diagram

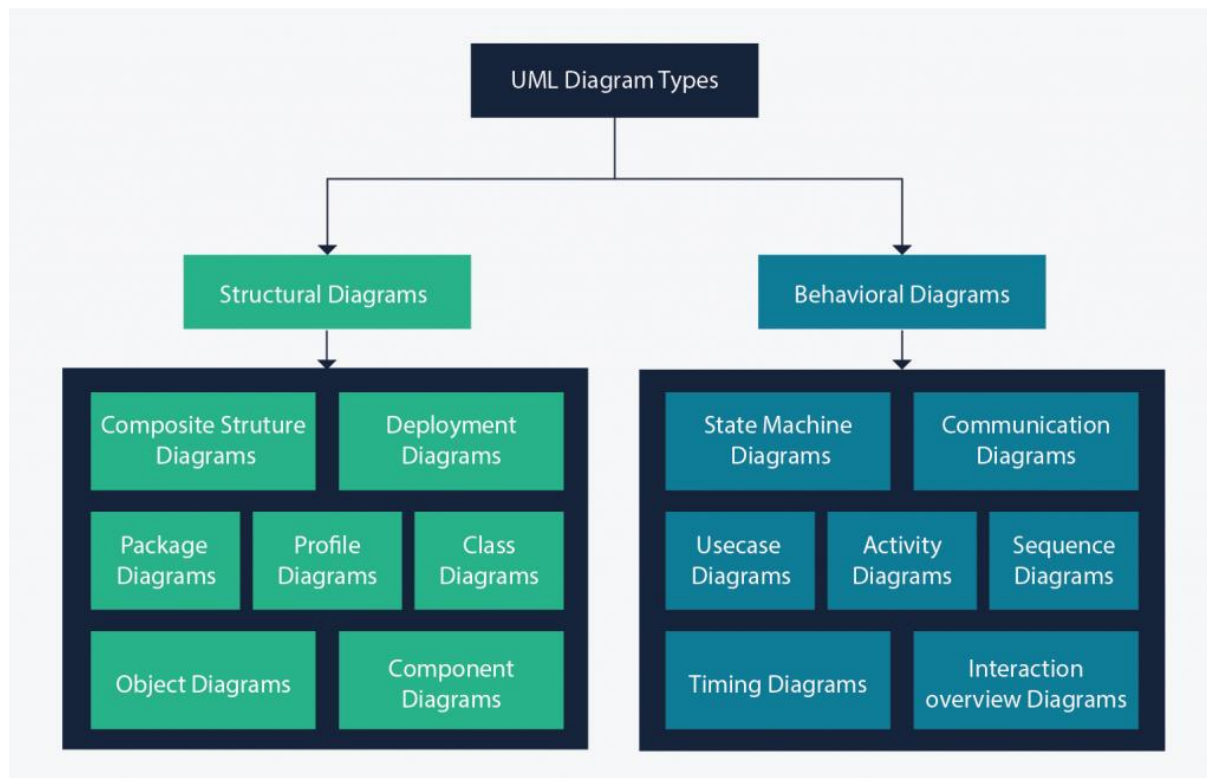


Figure 1: UML 2.5 diagrams overview (source: <https://createy.com/blog/diagrams/uml-diagram-types-examples/>)



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



On very important notice is that UML is not a method but a modelling language. As such to give it an approach we need to associate UML to a Unified Process (UP) in other to give our conception a methodology to follow. There exist several Unified Processes but our modelling approach will be the 2TUP (Two-track unified process) which we will use in the course of our project.

Table 6: Comparison between MERISE and UML

MERISE	UML
MERISE is a systemic method of analysis and design of information systems. That is, it uses a system approach	UML is however not a method but rather an object modelling 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.
Emphasizes data analysis and modelling using data flow diagrams (DFD), entity-relationship diagrams (ERD), and data dictionaries	Emphasizes system behaviour and structure using 14 diagram types, such as use case diagrams, class diagrams, and activity diagrams
Suitable for relational databases design and implementation	Suitable for object-oriented design and implementation



II. COMPARATIVE STUDY OF UNIFIED PROCESS

a. Unified Process

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

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

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

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

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

b. Two Track Unified Process (2TUP)

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

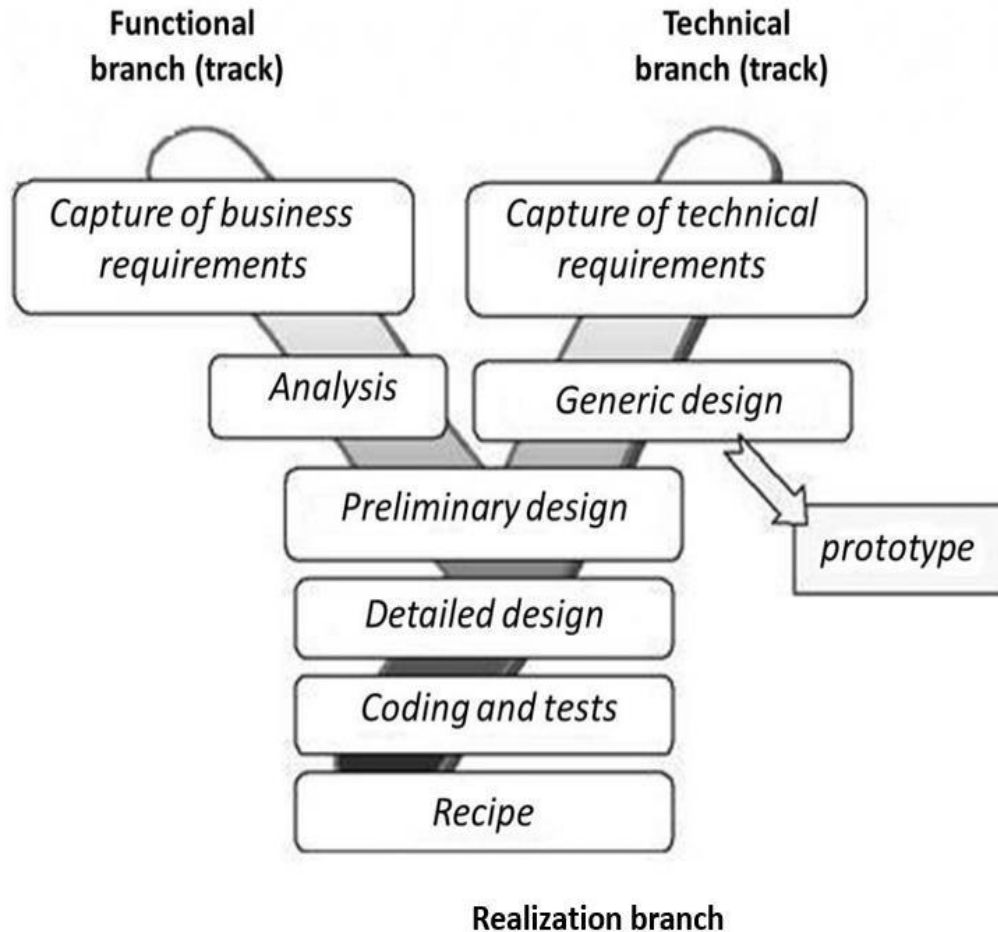


Figure 2: Two Track Unified Process (source: <https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing>)

a) The left branch (functional branch)

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

b) The right branch (Technical branch)

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



c) The middle branch (Realization or Implementation branch)

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

Preliminary conception: This is the most sensitive step of 2TUP as it is the confluence of the functional and technical branch. It is completed when the deployment model, the operating model, the logical model, interphases and the software configuration model are defined. We have the following diagrams:

- Component Diagram
- Deployment Diagram
- Package Diagram
- Composite Structure Diagram

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

- Class diagram
- Object diagram
- Sequence diagram
- Timing Diagram

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

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



III. JUSTIFICATION OF THE ANALYSIS METHOD

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

- UML is the current standard for programming in an object-oriented language. For this reason, it is widely understood and well known making it easy for a new programmer to join the project and be productive from the very first day.
- UML diagrams allow teams to virtualize how a project is or will be working, and they can be used in any field, not just software engineering. The diagrams will allow teams to virtualize together how a system, or a process will work or did work. It can provide new ideas for how teams have to collaborate to achieve the goal of the workflow process.
- 2TUP is centered around the creation and maintenance of a model, rather than the production of mountain documents.
- 2TUP is user oriented as it permits the development of software that responds to the needs of the users through the study of the user needs.
- 2TUP is iterative and incremental, hence it enables the project team to produce refined amelioration if necessary and easily integrate it in the already existing system.
- 2TUP by permitting the project team identify and test the key functionalities of the system limits the risk related to building the system.

IV. MODELLING OF THE PROPOSED SOLUTION

A. Capture of Functional Needs

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

1. USE CASE DIAGRAM

a. Definition

Use case diagram shows the functionalities of a system, their interdependencies and how they relate with actors of the system. A use case is a specification of behaviour. The main objectives of the use case diagram are:

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

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

b. Formalism

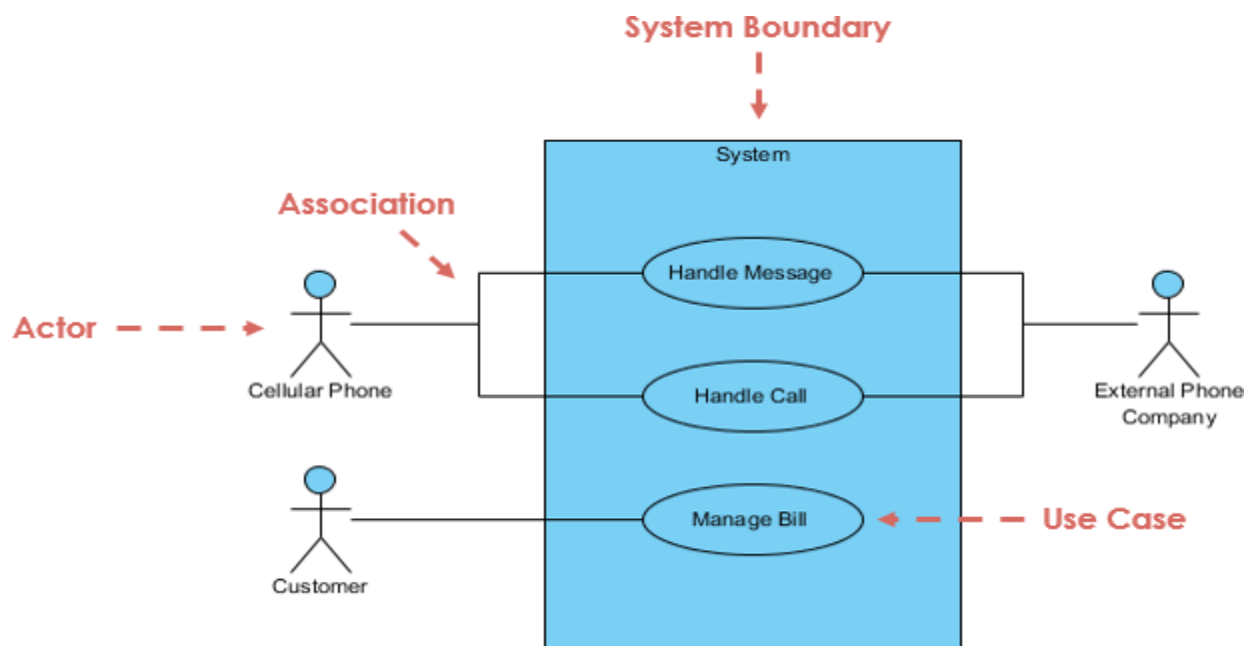


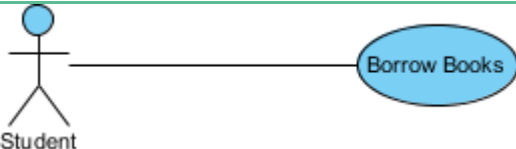
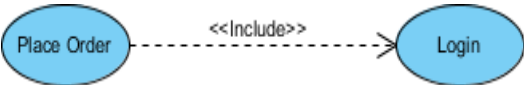
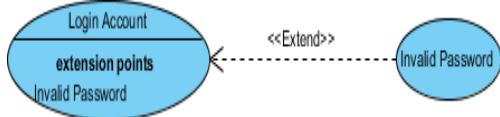
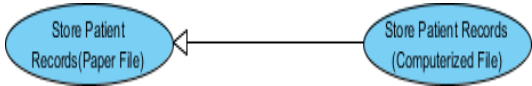



Figure 3: Use case diagram formalism

c. Components of use case diagram

Table 7: Component of use case diagram

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



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



d. Actors of our system

Table 8: Actors of our system

Actor	Role
Parent	Their responsibility is to fill a form this form could be a birth declaration or death declaration (for additional information), and a or apply for a declaratory judgement.
Head of hospital	He is in charge of creating birth declarations or death declaration from the hospital
Civil Registrar	He is charge of creating birth certificate or death certificate, reminding parents to complete their registration, and creating birth or death declarations.
Court personnel	He is charge of receiving applications for declaratory judgment and scheduling appointments for judgment.
Sectary	He is charge of creating a hospital accounts, civil registrar accounts, and court personnel account.
Super Admin	He is in charge of creating the sectary of each council, print vital statistics
SMS API	It is used to send message to parents to tell them when to come and collect their birth certificate

e. CamerStay use case diagram

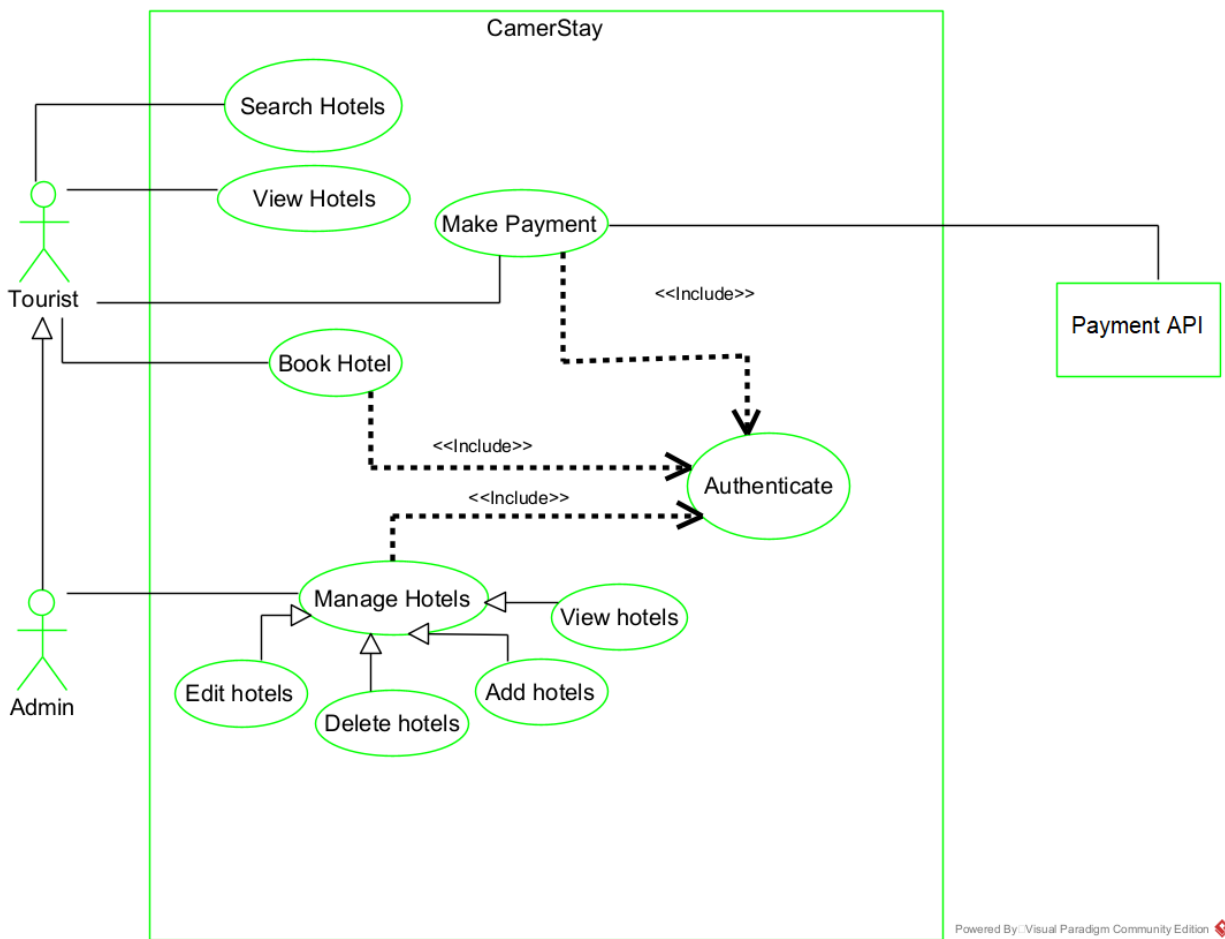


Figure 4: General use case diagram

e. Manage use case diagram

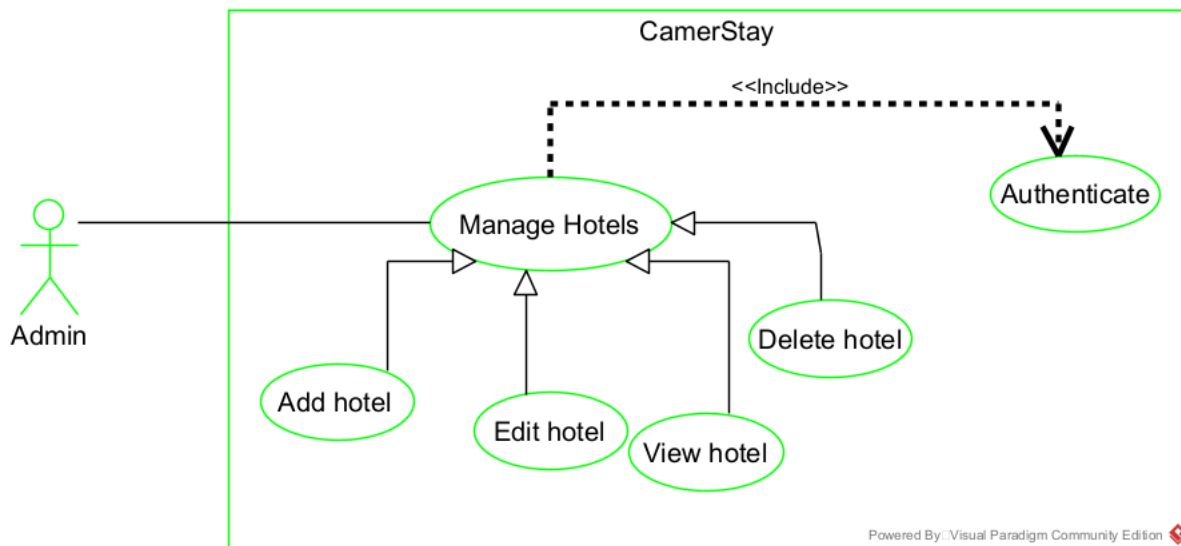


Figure 5: Manage hotel use case diagram



Textual Description

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 9: Textual description formalism

NAME OF THE USE CASE	
Summary	Aim of the use case
Actors	Users
Date	The date the app was done
Stakeholder	The owner of the application
Version	The version of the application
Pre-condition	Set of actions that must be completed before the launching of the use case
Trigger	The event that causes the use case to be initiated
Nominal scenario	It corresponds to a normal development of a use case
Alternative scenario	These are variants of the nominal scenario.
Post condition	Set of mechanism that can lead to the end of the use case

Table 10: Textual description of Authenticate

USE CASE AUTHENTICATE	
Summary	Permit client to access the system
Actors	Head of hospital, Civil Registrar, Court personnel, Sectary
Date	September 11, 2023
Stakeholder	CamerStay Directors
Version	1.0
Pre-condition(s)	<ol style="list-style-type: none"> 1. The actor already has an account; 2. The app is launched
Trigger	The user clicks on login button
Nominal scenario	<ol style="list-style-type: none"> 1. The user clicks on login button 2. The system displays the login page 3. The user fills form and submit 4. The system checks conformity 5. The system sends validation query 6. The system treats the result returned by the database 7. The user is directed to his page
	8.
Alternative scenario	<ol style="list-style-type: none"> 9. If step 4 of the nominal scenario the format is not correct, an error message is returned to the user and he returns to step 3 of the nominal scenario. 10. If step 6 of the nominal scenario no user was found, the system returns an error message and returns to step 3 of the nominal scenario
Post-condition of success	The user logs in successfully
Post condition of failure	An error message is display and user does not have access his space.

Table 11: Textual description of Create account

USE CASE CREATE ACCOUNT	
Summary	Grant the users access to the application
Actors	Sectary, Admin
Date	September 11, 2023
Stakeholder	CamerStay Directors
Version	1.0
Pre-condition(s)	<ol style="list-style-type: none"> 1. The actor already has an account; 2. The app is launched
Trigger	The user clicks on create button
Nominal scenario	<ol style="list-style-type: none"> 1. Click on create account 2. The system displays create account form 3. The user fills form and submit 4. The system checks for conformity of the form 5. The system sends validation query 6. The system treats the result returned by the database 7. The system sends a create account query 8. The system treats the result returned by the database 9. The system displays successful message
Alternative scenario	<ol style="list-style-type: none"> 10. If step 4 of the nominal scenario the format is not correct, an error message is returned to the user and he returns to step 3 of the nominal scenario. 11. If step 6 of the nominal scenario an email already exist, an error message is returned to the user and he returns to step 3 of the nominal scenario. 12. If step 8 of the nominal scenario an error is found, an error message is returned to the user and he returns to step 3 of the nominal scenario
Post-condition of success	The user's account is created
Post condition of failure	An error message is displayed

2. ACTIVITY DIAGRAM

a. Definition

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

b. Formalism

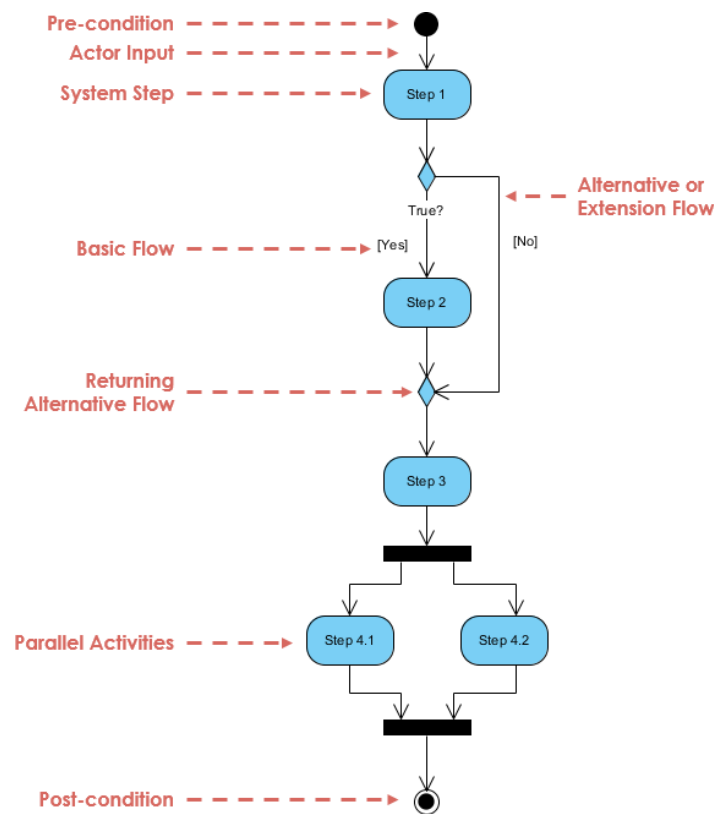






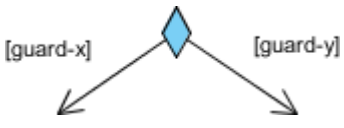

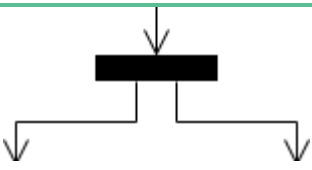


Figure 6: Activity diagram formalism

c. Component of an Activity diagram

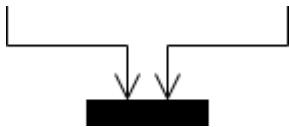
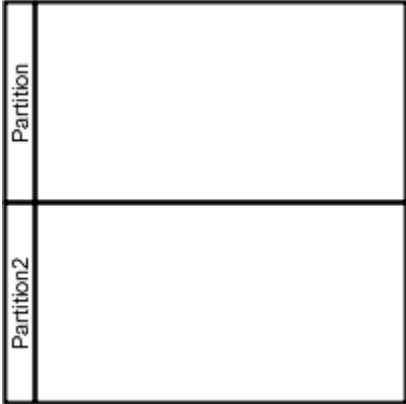
Table 12: Component of an Activity diagram

Element	Notation	Description
Activity		Is used to represent a set of actions
Action		A task to be performed
Activity Edge		
Initial node		Portrays the beginning of a set of actions or activities
Activity Final Node		Stop all control flows and object flows in an activity (or action)
Object node		Represent an object that is connected to a set of Object Flows
Decision node		Represent a test condition to ensure that the control flow or object flow only goes down one path
Merge node		Bring back together different decision paths that were created using a decision-node.
Fork node		Split behavior into a set of parallel or concurrent flows of activities (or actions)



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



Join node		Bring back together a set of parallel or concurrent flows of activities (or actions).
Swimlane and Partition		A way to group activities performed by the same actor on an activity diagram or to group activities in a single thread

d. Activity diagram of Authenticate

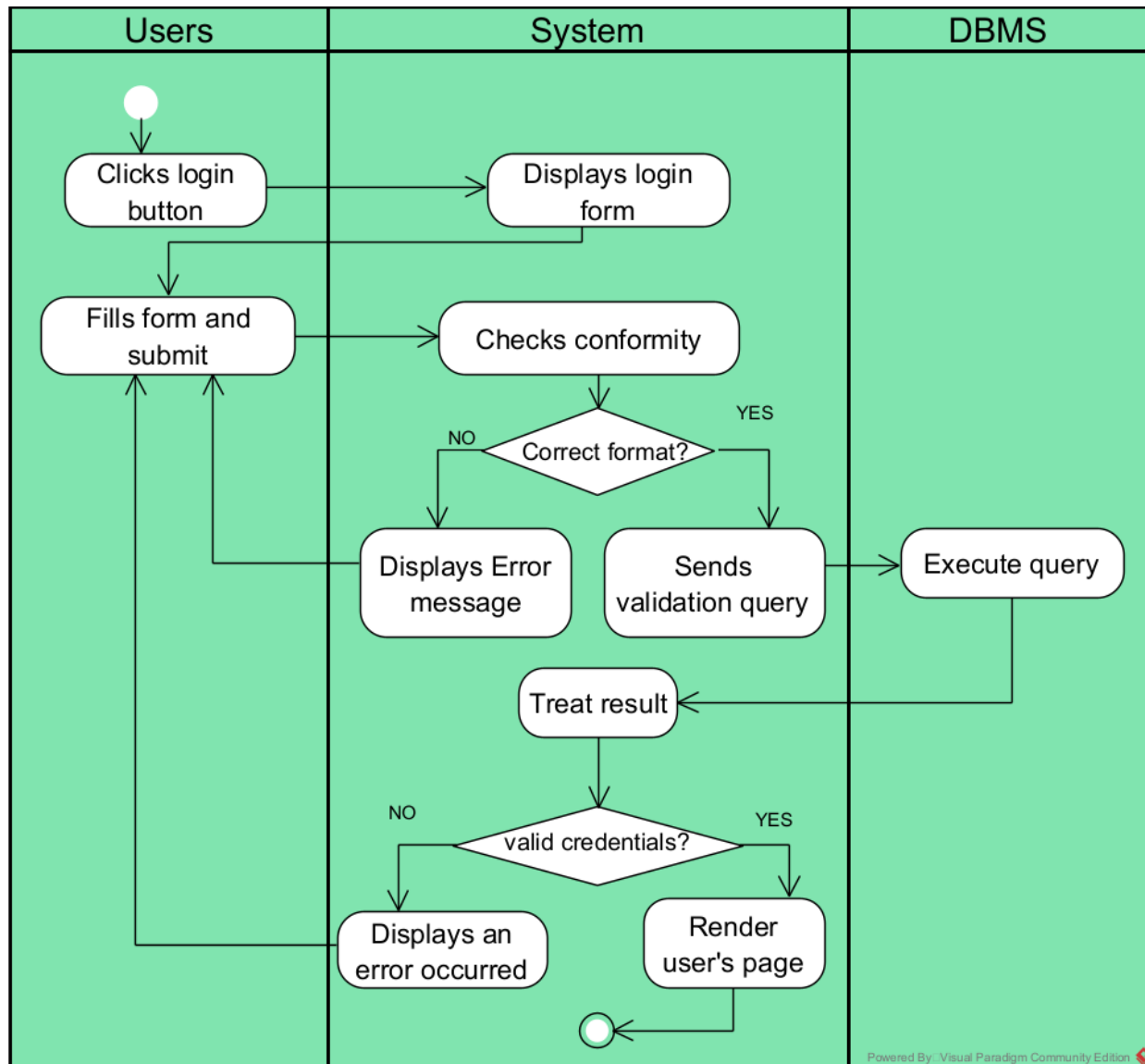


Figure 7: Authenticate activity diagram

e. Activity diagram of Create account

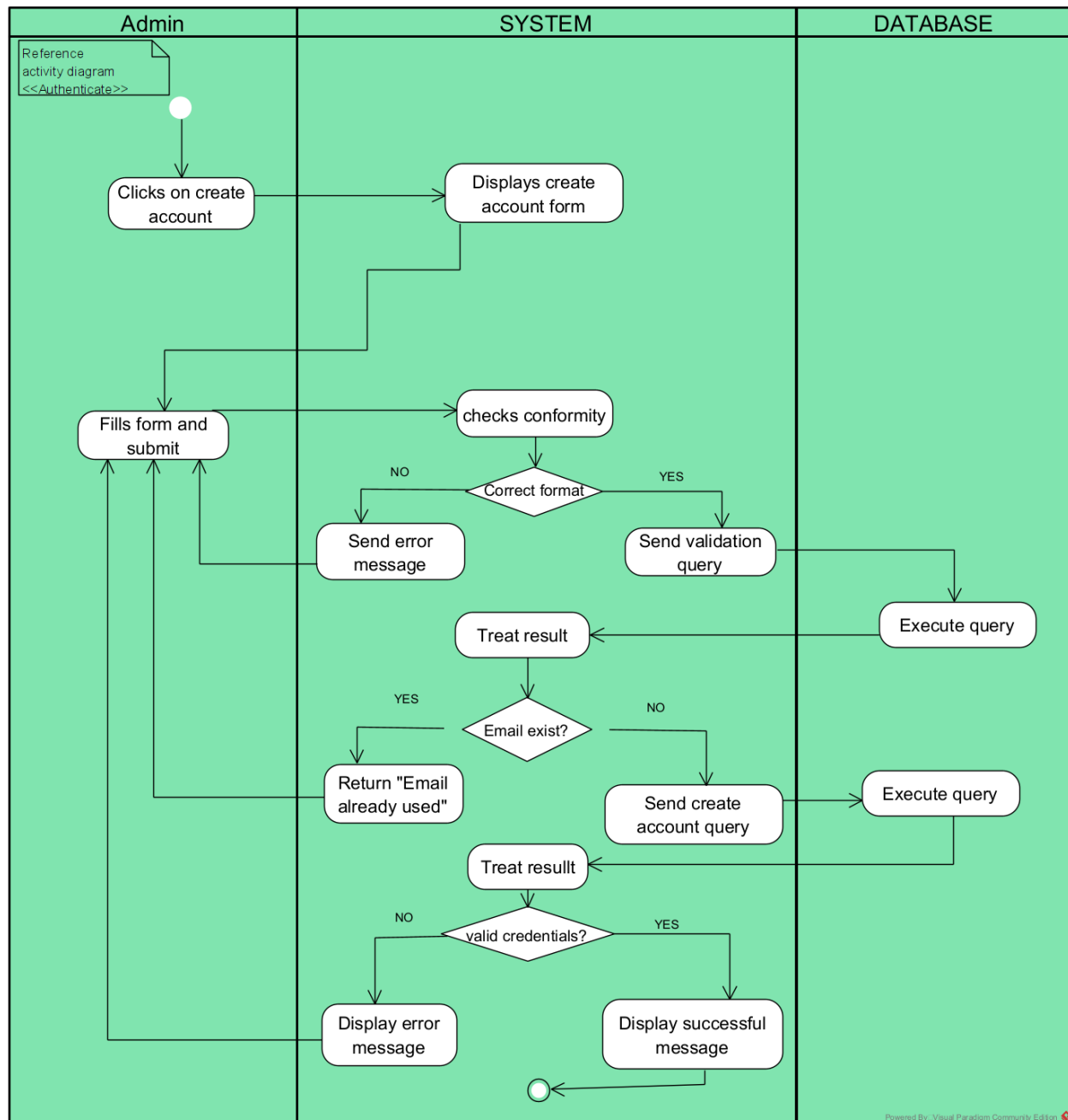


Figure 8: Create account activity diagram

f. Activity diagram of Make Payment

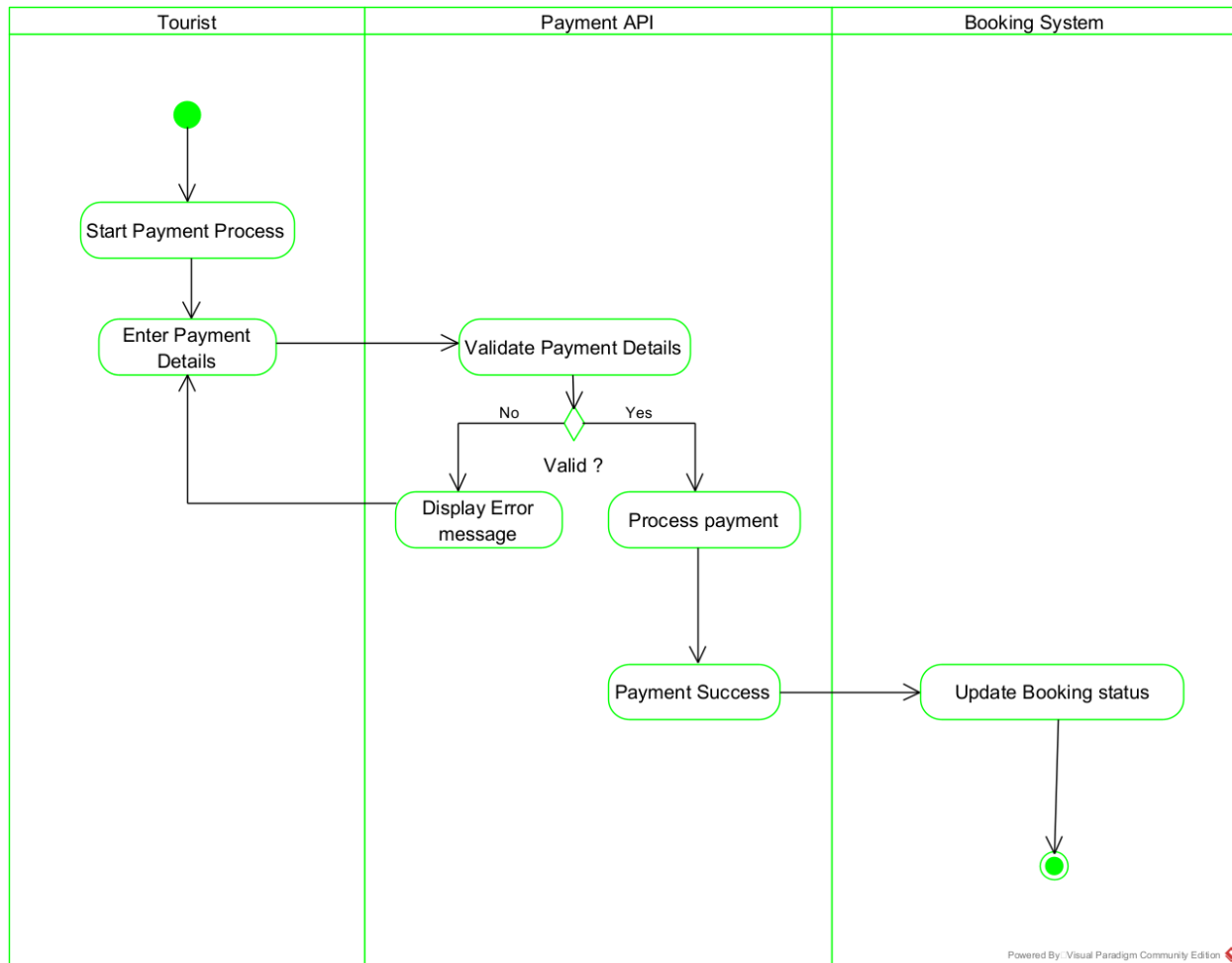


Figure 9: Make payment activity diagram

f. Activity diagram of Booking

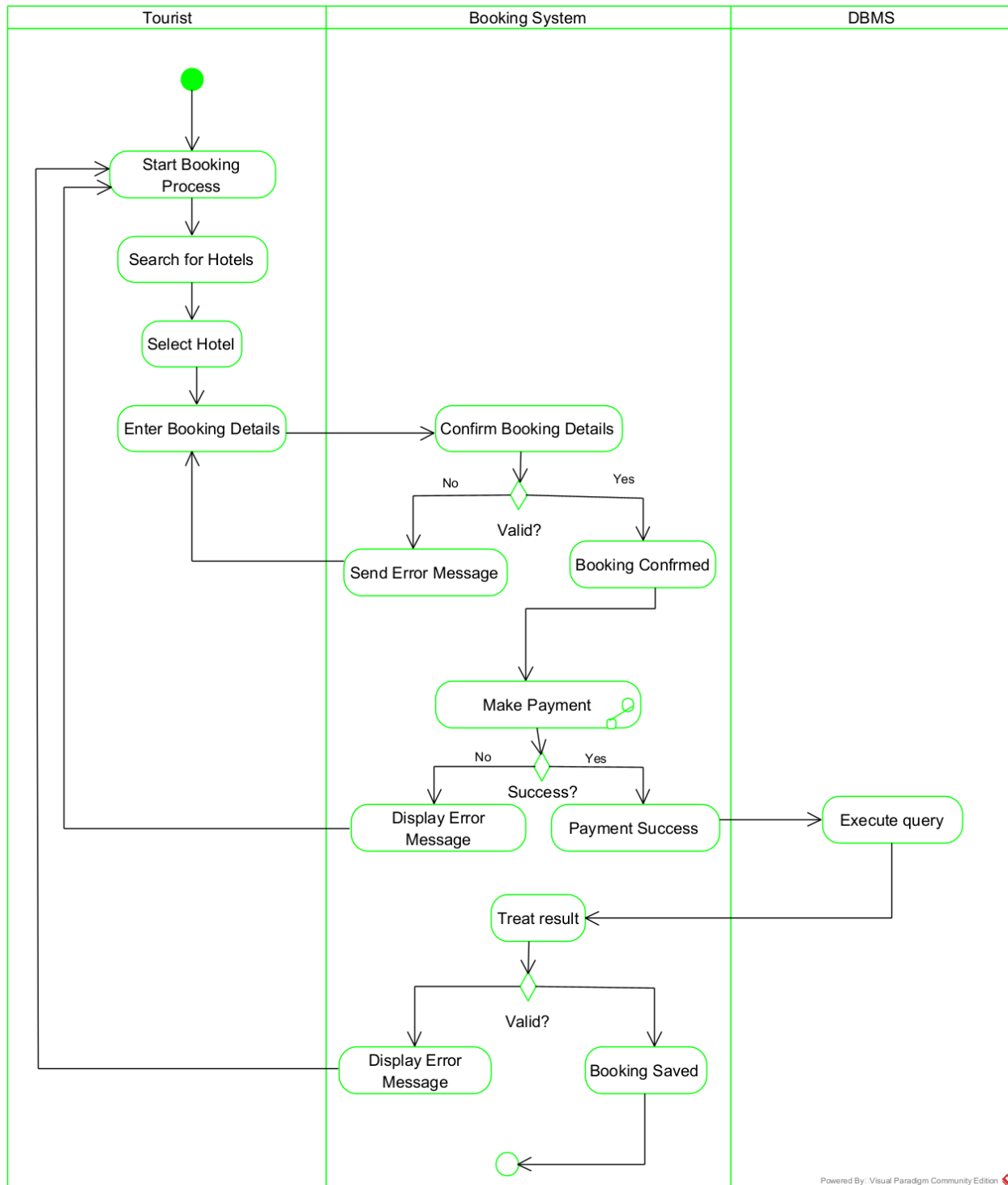


Figure 10: Booking activity diagram

3. COMMUNICATION DIAGRAM

a. 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.

b. Formalism

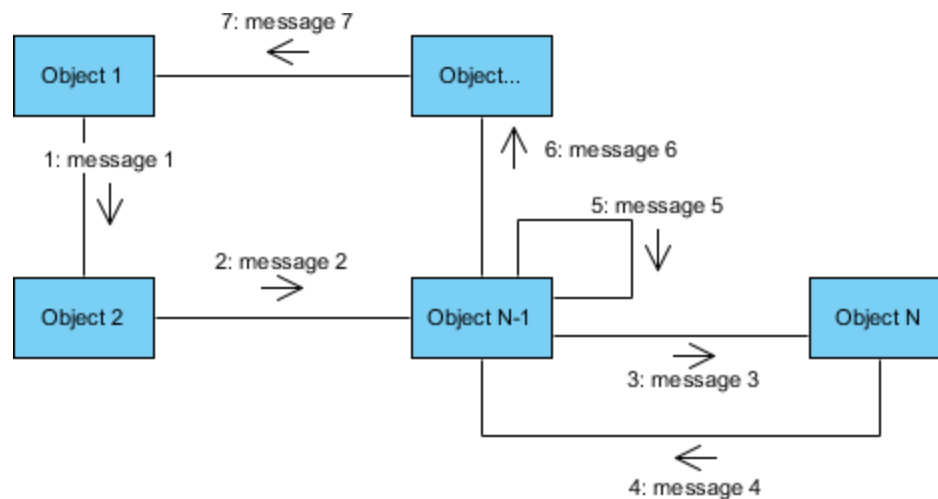






Figure 11: Communication diagram formalism

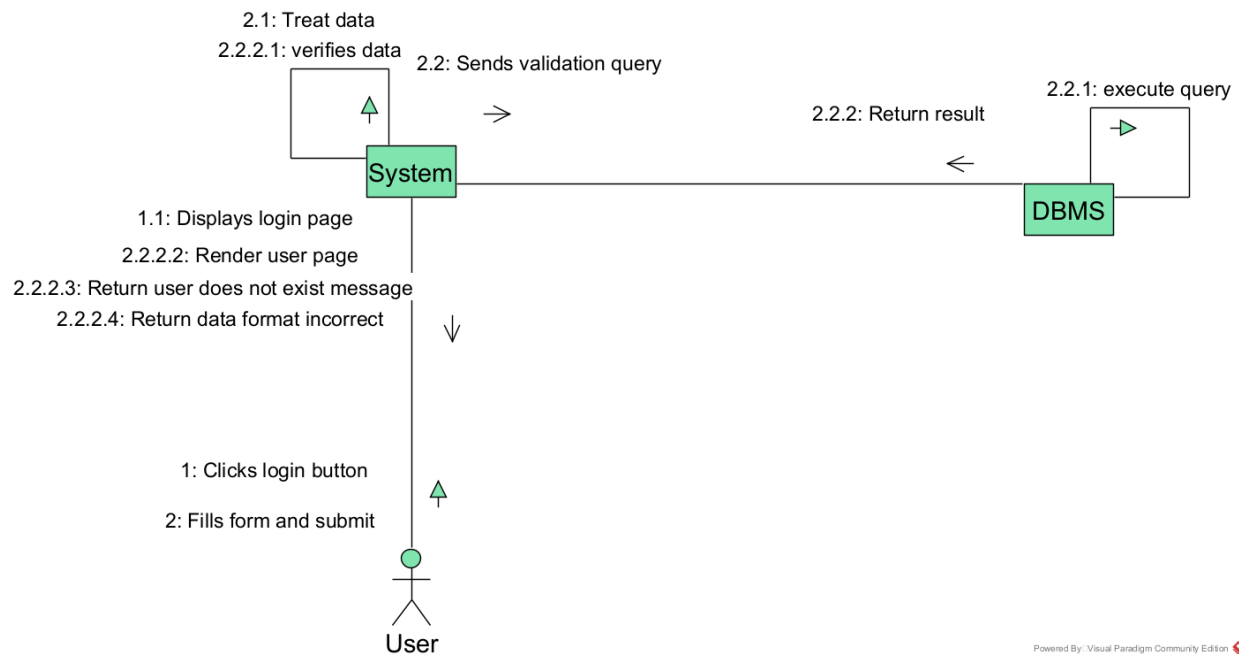


c. Components of a communication diagram

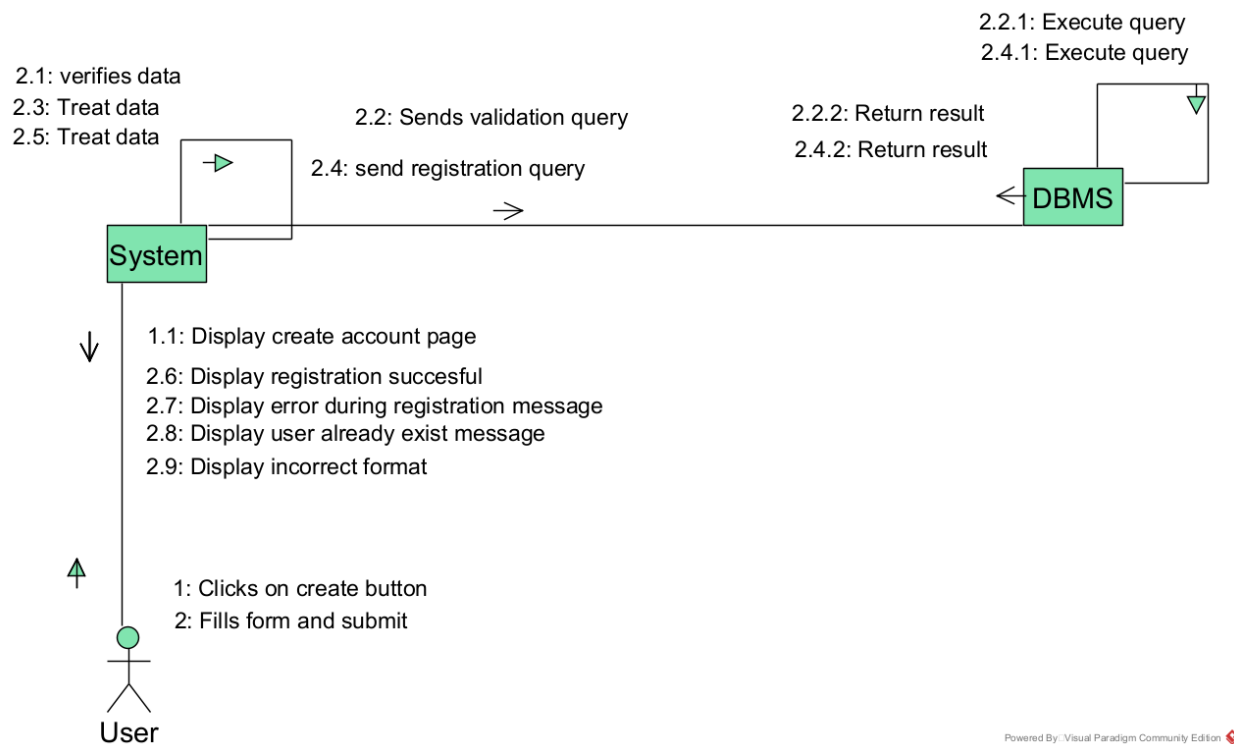
Table 13: Component of a communication diagram

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

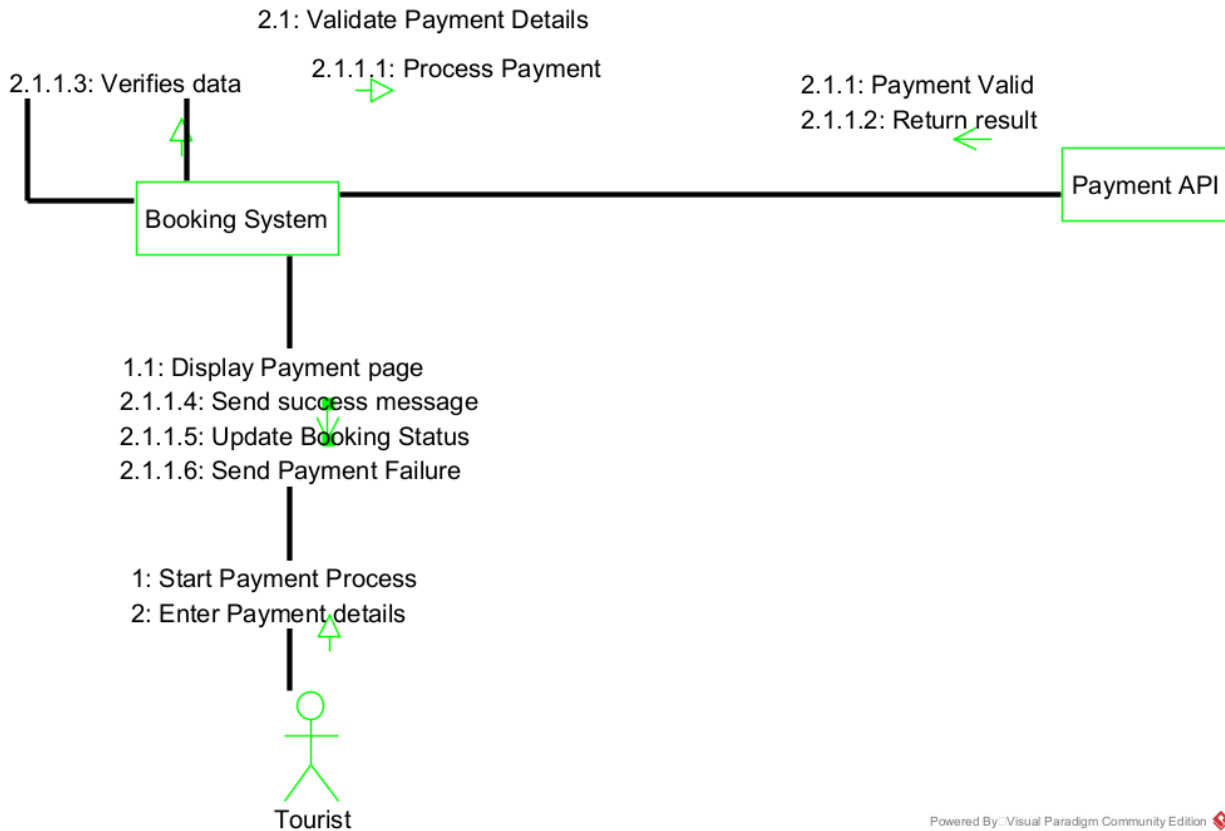
d. Communication diagram of Authenticate



e. Communication diagram of Create account



f. Communication diagram of Make Payment



Powered By: Visual Paradigm Community Edition

Figure 14: Make Payment communication diagram

4. SEQUENCE DIAGRAM

a. Definition

A Sequence diagram describes interactions among classes in terms of an exchange of messages over time. They are also called event diagrams. A Sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

b. Formalism

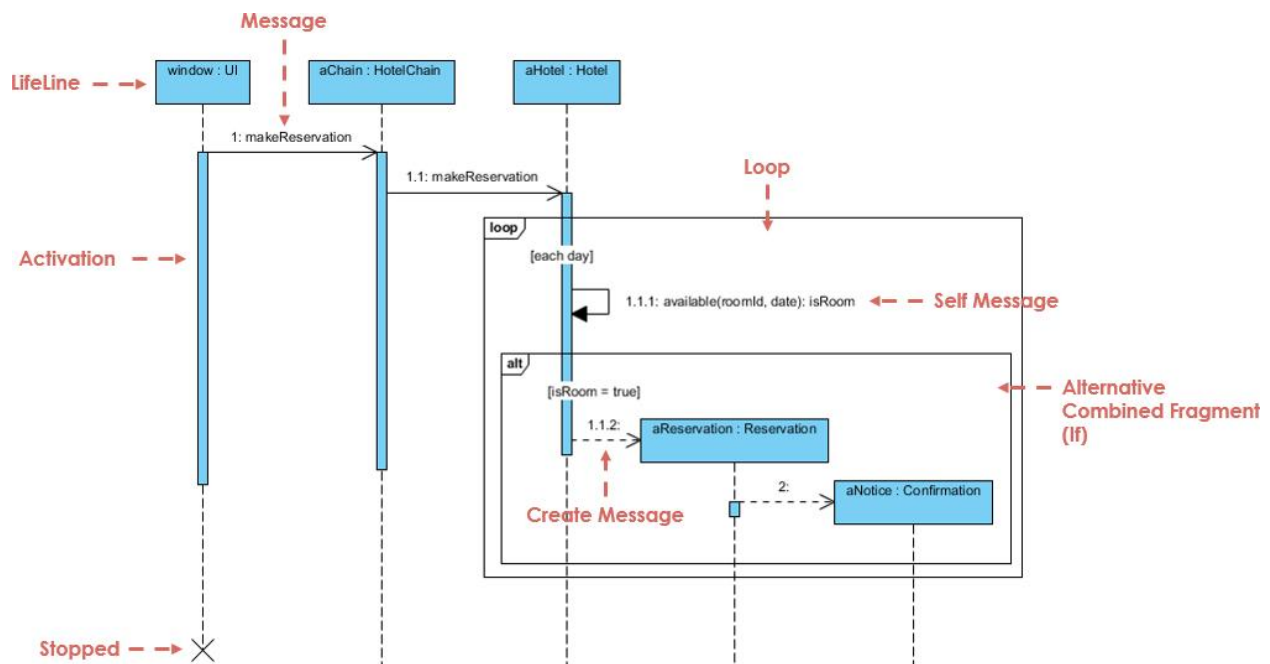
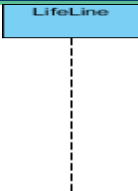
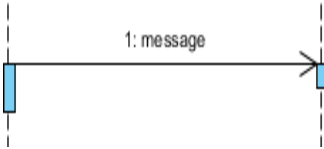
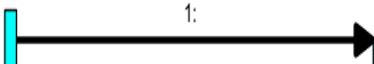

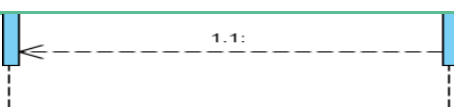


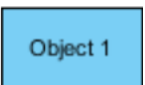


Figure 15: Sequence diagram formalism

c. Component of sequence diagram

Table 14: Component of sequence diagram

Element	Notation	Description
Lifelines		They represent rows or objects instances that participate in the sequence being modelled.
Asynchronous Message		It is a message that receives an indirect response.
Synchronous Message		It is a message that sends and want response before it continues a process
Self-Message		Self message is a kind of message that represents the invocation of message of the same lifeline.
Return Message		It represents the response of a message.
Actor		They send and receive message.
Combined Interaction Fragment		An articulation of interaction diagram, defined by an operator and operands.
Object		They send and receive messages

d. Sequence diagram of Authenticate

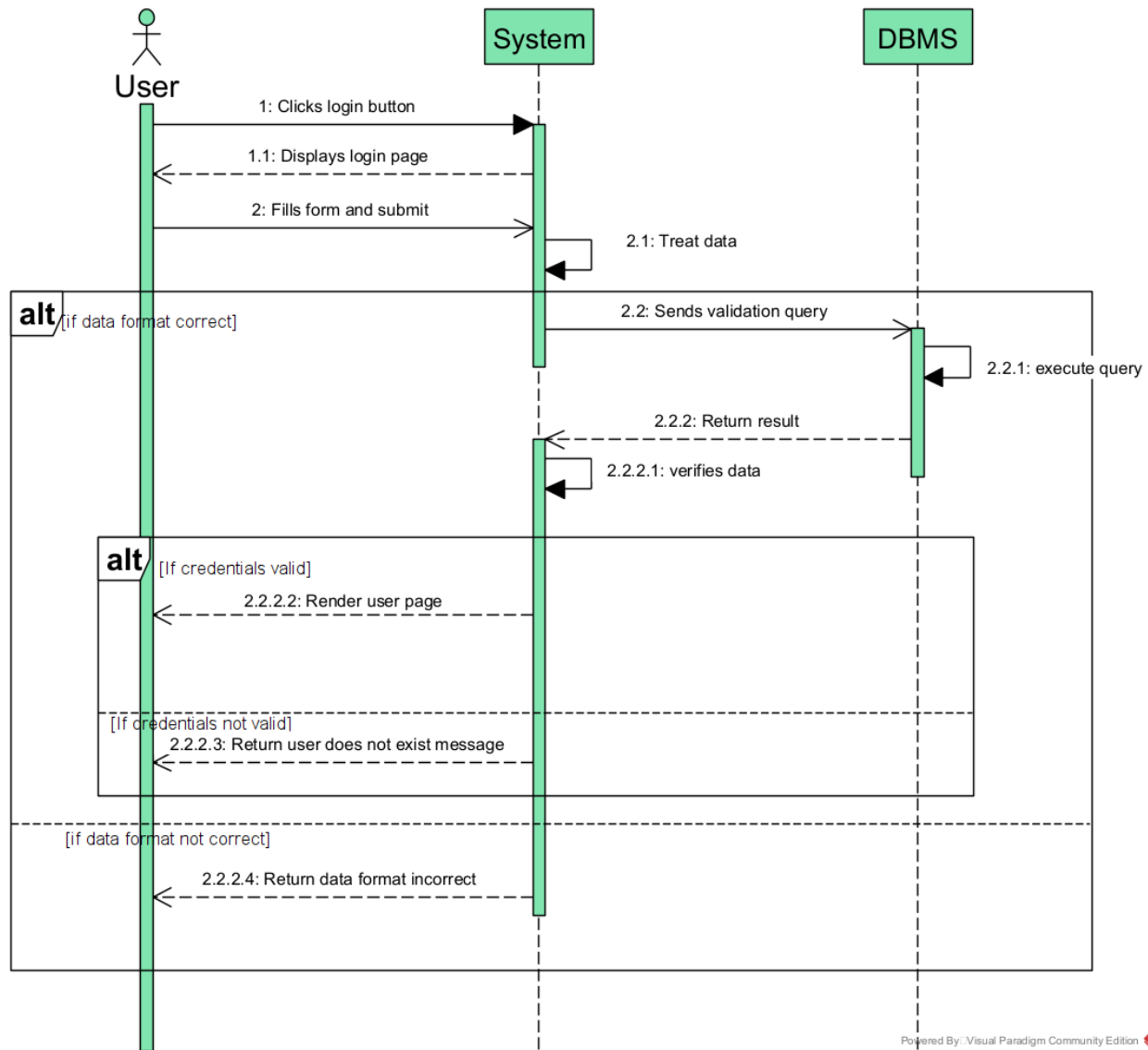


Figure 16: Authenticate Sequence diagram

e. Sequence diagram of Make Payment

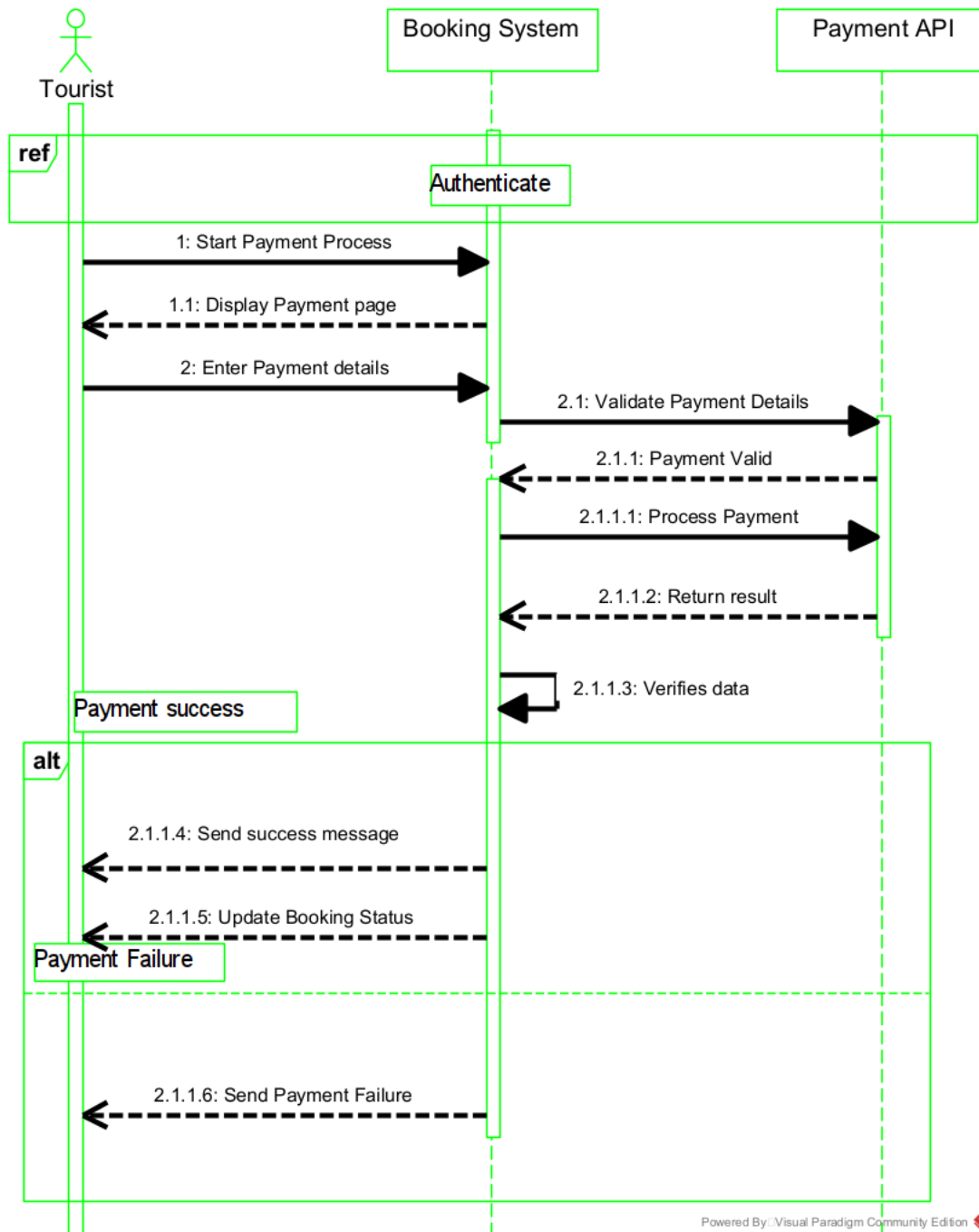


Figure 17: Make Payment sequence diagram

f. Sequence diagram of Create declaration



CONCLUSION

During the analysis phase, we selected a software development process and modelling language. Next, we elaborated on the functional requirements of our system. We reviewed the use case diagram, which outlines the relationship between actors and the actions they can perform in the system, as well as the communication diagram that illustrates the architecture of the system based on object-oriented programming. We also analysed the sequence diagram, which delineates the flow of messages between elements in the system. Lastly, we reviewed the activity diagram, which displays the workflow of our system. We will proceed to the conception phase, during which we shall showcase the Technical branch of our system, accompanied by relevant diagrams.



PART 4: CONCEPTION PHASE



PREAMBLE

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

CONTENT

INTRODUCTION

I TECHNICAL BRANCH

A. GENERIC BRANCH

B. CAPTURE OF TECHNICAL NEEDS

C. RELATED UML DIAGRAMS

i. CLASS DIAGRAM

ii. STATE MACHINE DIAGRAM

iii. PACKAGE DIAGRAM

CONCLUSION



INTRODUCTION

The concept stage will outline the precise requirements, characteristics and activities needed to fulfil the operational needs of the suggested system, as defined during the analysis stage. The objective during this phase is to recognise and examine key components (either hardware or software), configuration (network capacity), processes and procedures which are indispensable for the system to achieve its desired outcomes. We will scrutinise several diagrams such as the class diagram, state machine diagram and package diagram.

I. TECHNICAL BRANCH

A. GENERIC DESIGN

a. Hardware diagram of the system

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

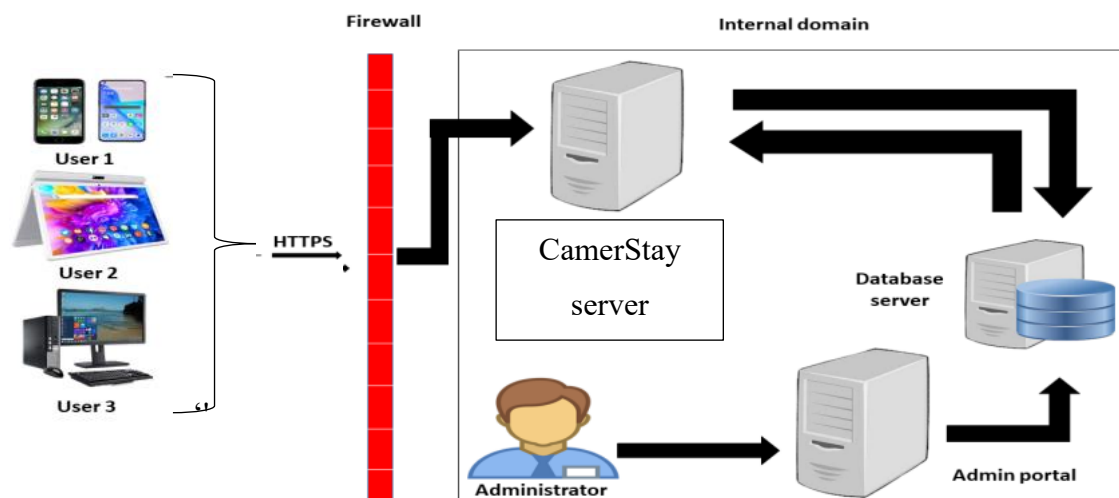


Figure 18: Hardware diagram

a. High Level Architecture of the System

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

B. CAPTURE OF TECHNICAL NEEDS

a. Physical Architecture

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

Within the scope of our project, we made use of the 3-tier architecture This architecture separate it tiers from each other based upon the user and the manipulated data in the database. Each layer has a well-defined communication interface, and the evolution of the layer is independent of the other. The 3-tier of our system is made up of:

- ✚ The client tier, which is also known as our presentation interphase.
- ✚ Application Tier, which represents our webserver.
- ✚ The data tier, which represents our DBMS server

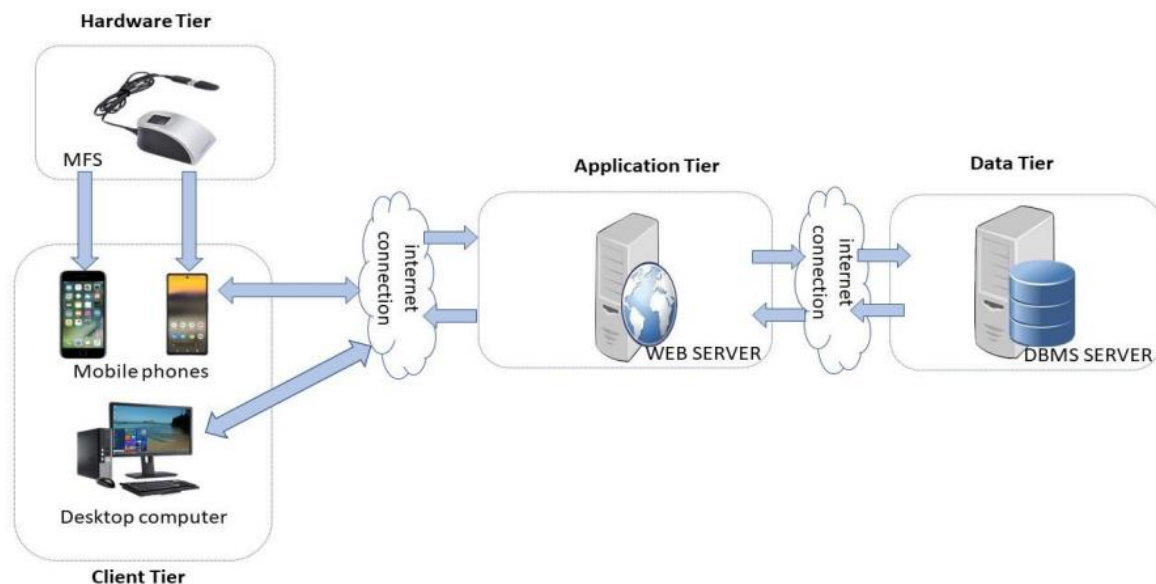


Figure 19: : n-tier Architecture (Source: <https://www.pinterest.es/pin/752241943987565648/>)

b. Logical Architecture

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

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

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

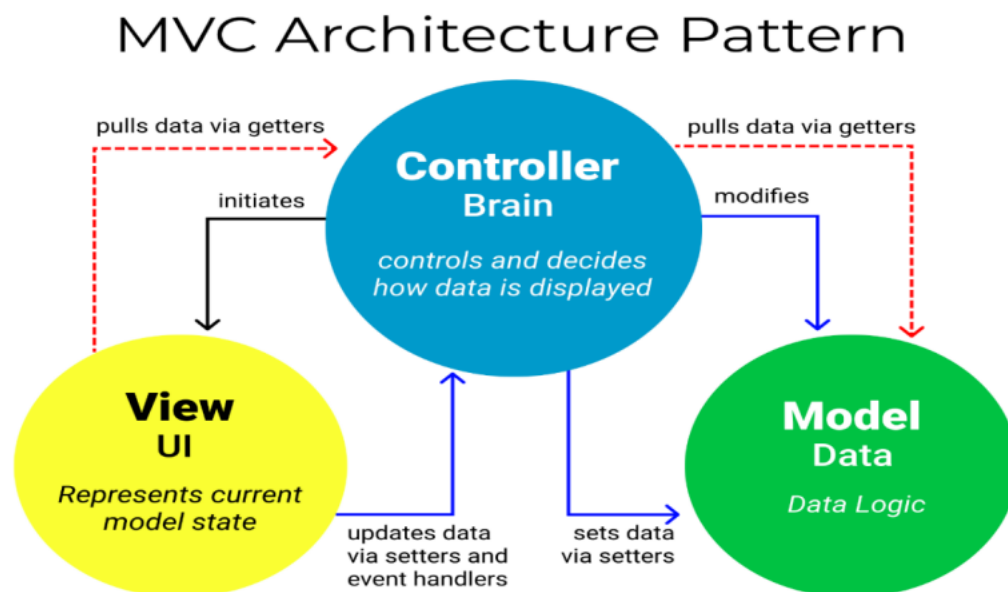


Figure 20: The MVC architecture (source: <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>)

C. RELATED UML DIAGRAMS

1. CLASS DIAGRAM

a. Definition

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

b. Formalism

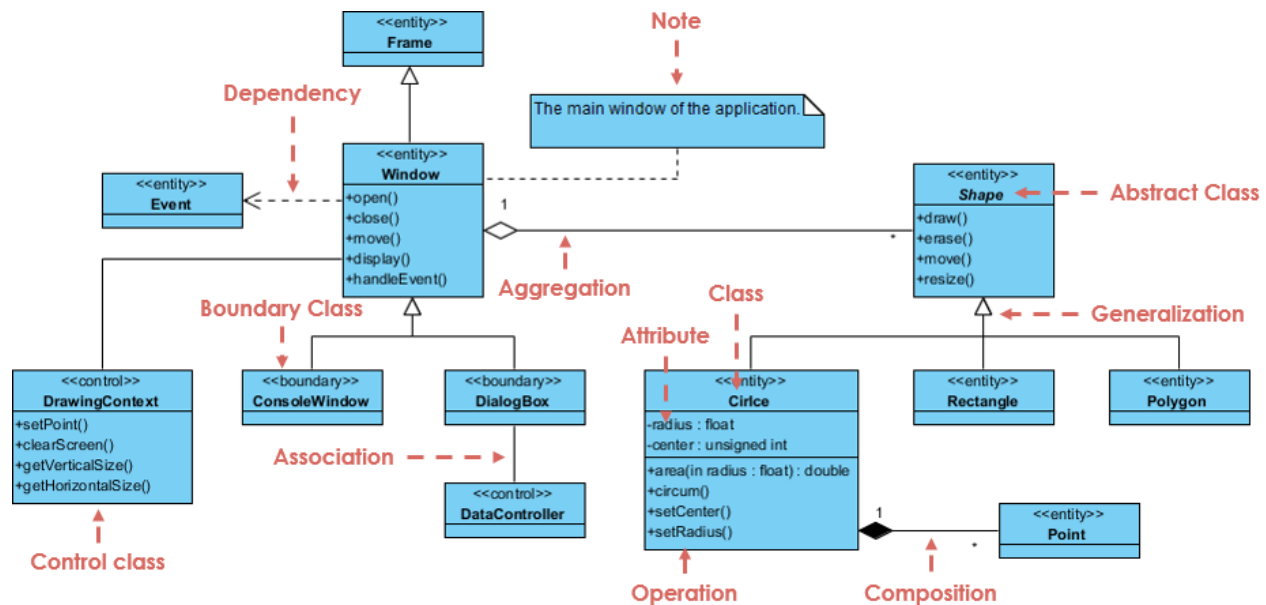
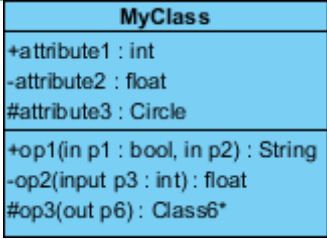
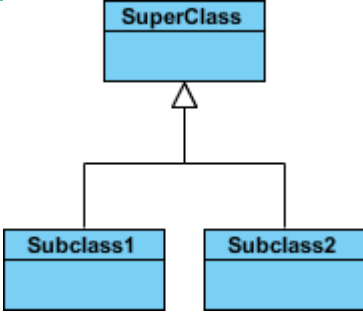

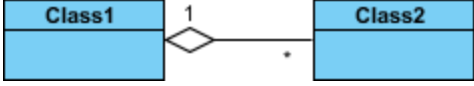

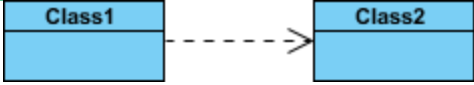


Figure 21: Class diagram formalism

c. Component of a Class diagram

Table 15: Components of class diagram

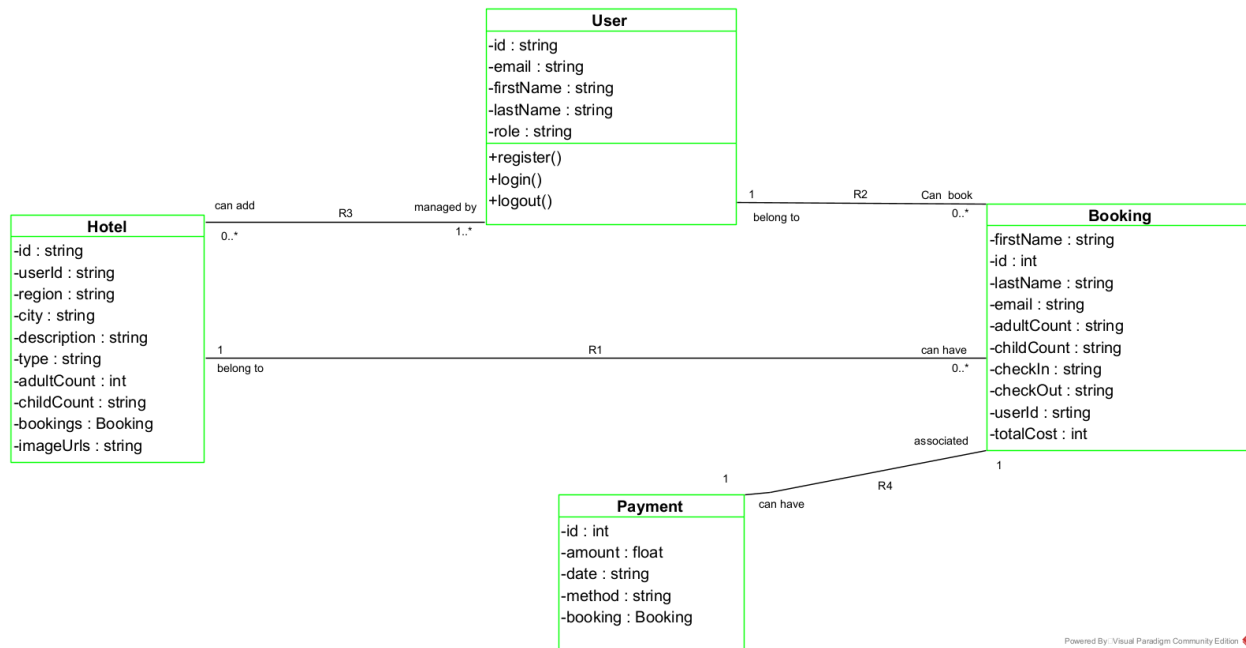
Element	Notation	Description
Class		A class is an element that defines the attributes and behaviours that an object can generate
Generalization		it a relationship between a whole thing (called superclass) and a more specific thing (called subclass)
Association		It is a general type of relationship between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



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



d. CamerStay Class diagram



Powered By: Visual Paradigm Community Edition

Figure 22: Class diagram



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



e. Business Rule

R1: A tourist can make multiple bookings, but each booking must belong to exactly one user.

R2: A hotel can have multiple bookings, but each booking must belong to exactly one hotel.

R3: An admin can manage multiple hotels, but each hotel is managed by one admin.

R4: One booking corresponds to a single payment, and each payment corresponds to a single booking.

2. STATE MACHINE DIAGRAM

a. Definition

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

b. Formalism

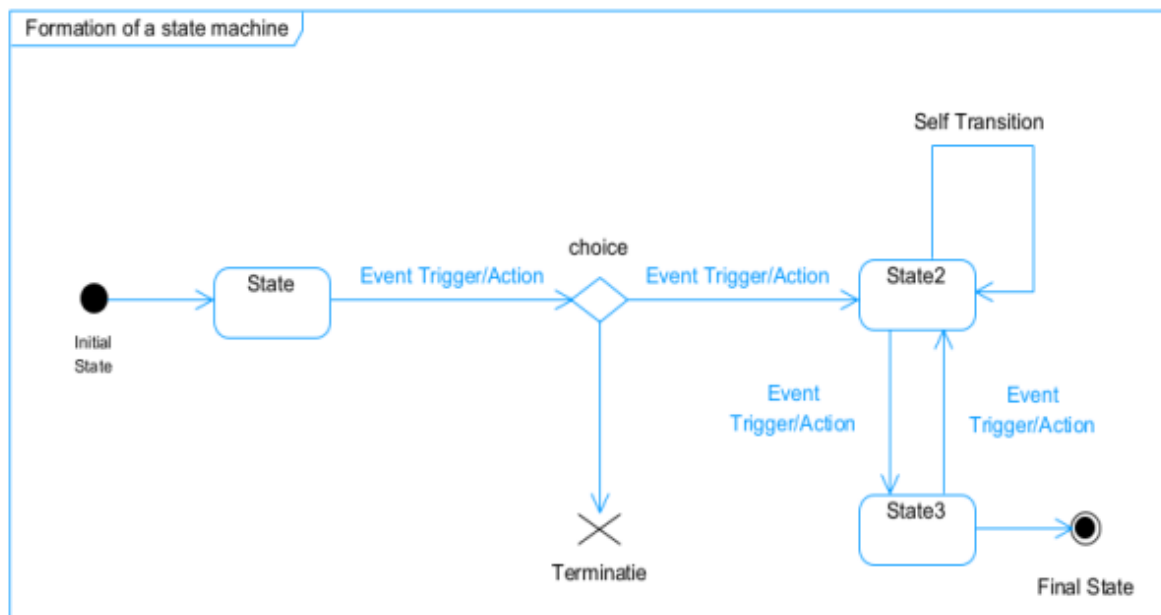
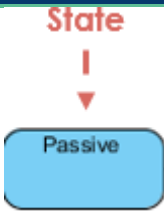


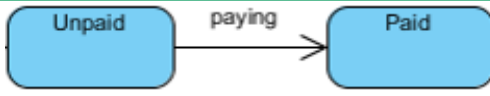


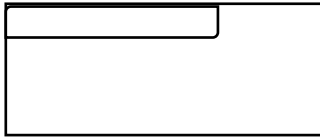


Figure 23: Formalism of a state machine diagram

c. Component of a state machine diagram

Table 16: Component of state machine diagram

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

d. State machine diagram of Authenticate

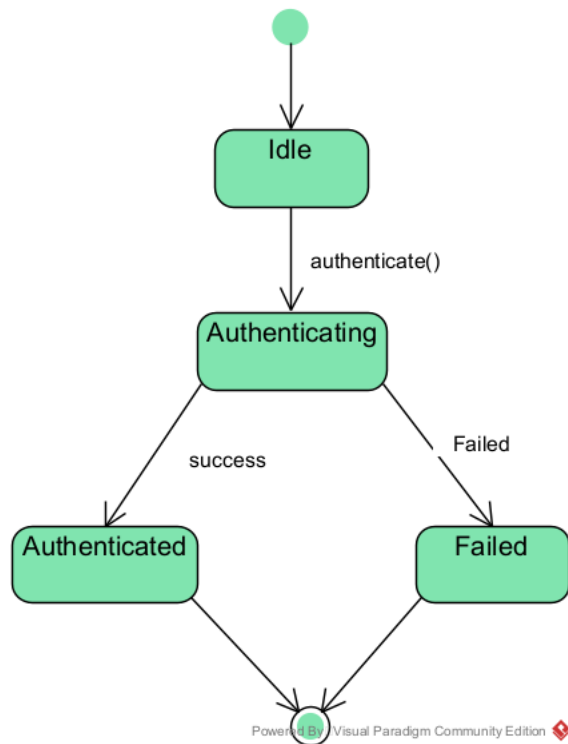


Figure 24: State machine diagram of authenticate

3. PACKAGE DIAGRAM

a. Definition

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

b. Formalism

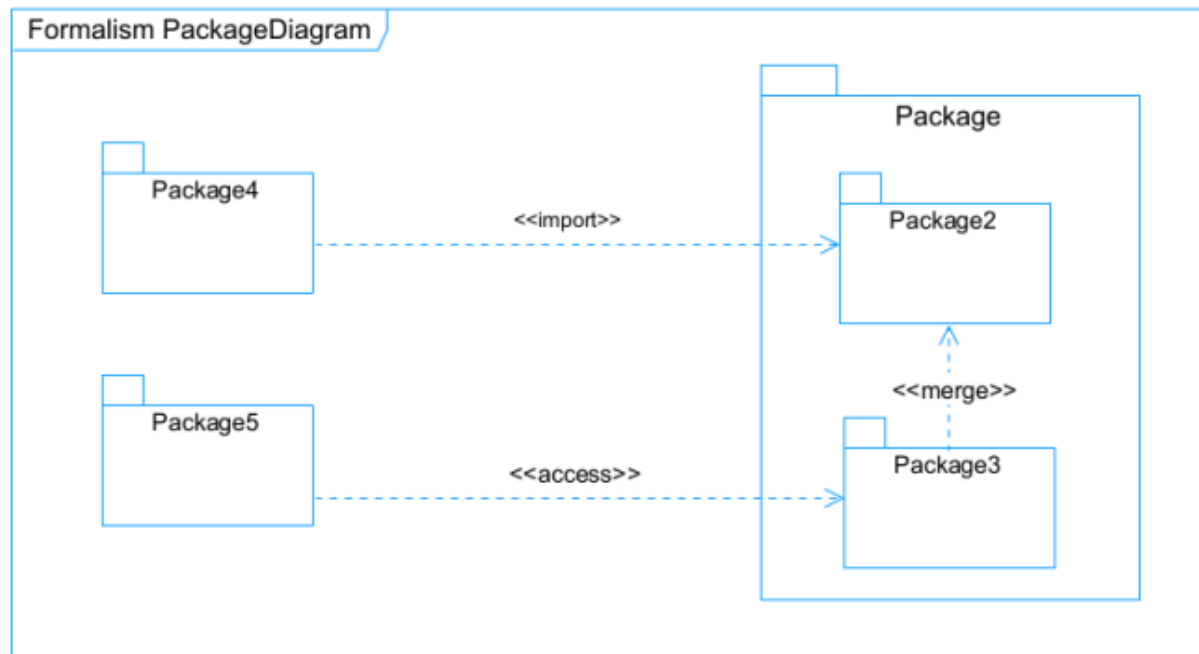
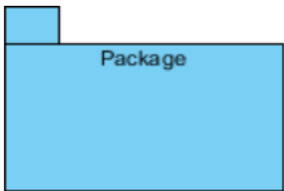
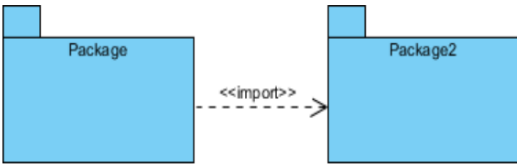
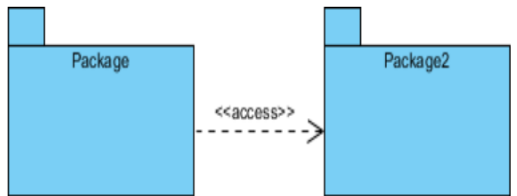



Figure 25: Package diagram formalism

c. Component of package diagram

Table 17: Component of package diagram

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

d. CamerStay Package diagram

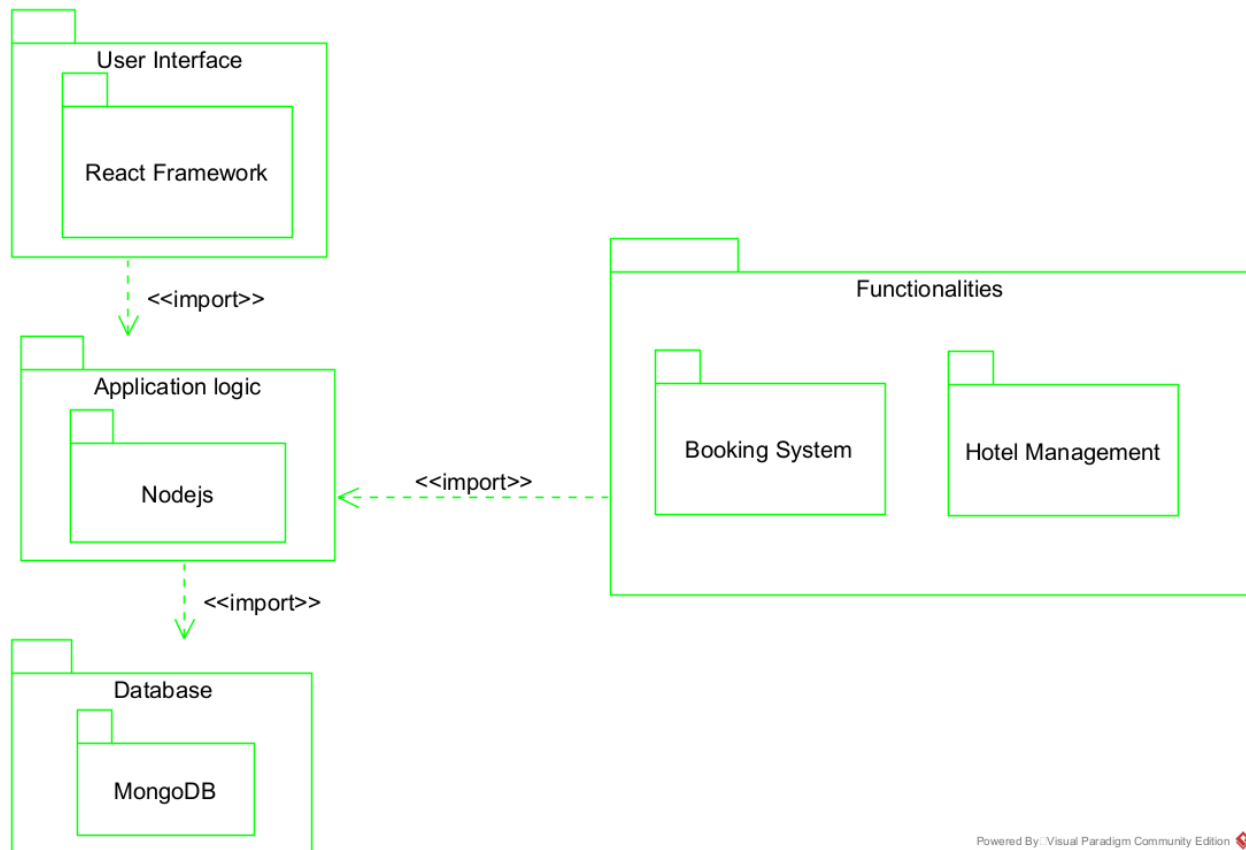


Figure 26: CamerStay package diagram



CONCLUSION

We presented the structure of our system and its integration into the existing technical architectures. In the initial stage, we aimed to investigate various aspects of our system. Primary actions included assessing the technical limitations, identifying the system's components with their classification and deployment onto relevant deployment targets. We then concluded this phase by evaluating the interactions between different components and participants of our system. Moving forward, our report will now progress to the implementation phase, which will entail concentrating on executing our system-related elements.



PART 5: REALIZATION PHASE



PREAMBLE

In this phase, we aim to implement our solution straightforwardly by relying on the analysis and conception phases. Additionally, we will showcase the component and deployment diagrams.

CONTENT

INTRODUCTION

- I. DEPLOYMENT DIAGRAM
- II. COMPONENT DIAGRAM

CONCLUSION



INTRODUCTION

Here in the realisation phase, we will review diagrams concerning the physical aspects of the system, such as libraries, documents and the physical topology of the system components when the software is deployed.

1. DEPLOYMENT DIAGRAM

a. Definition

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

b. Formalism

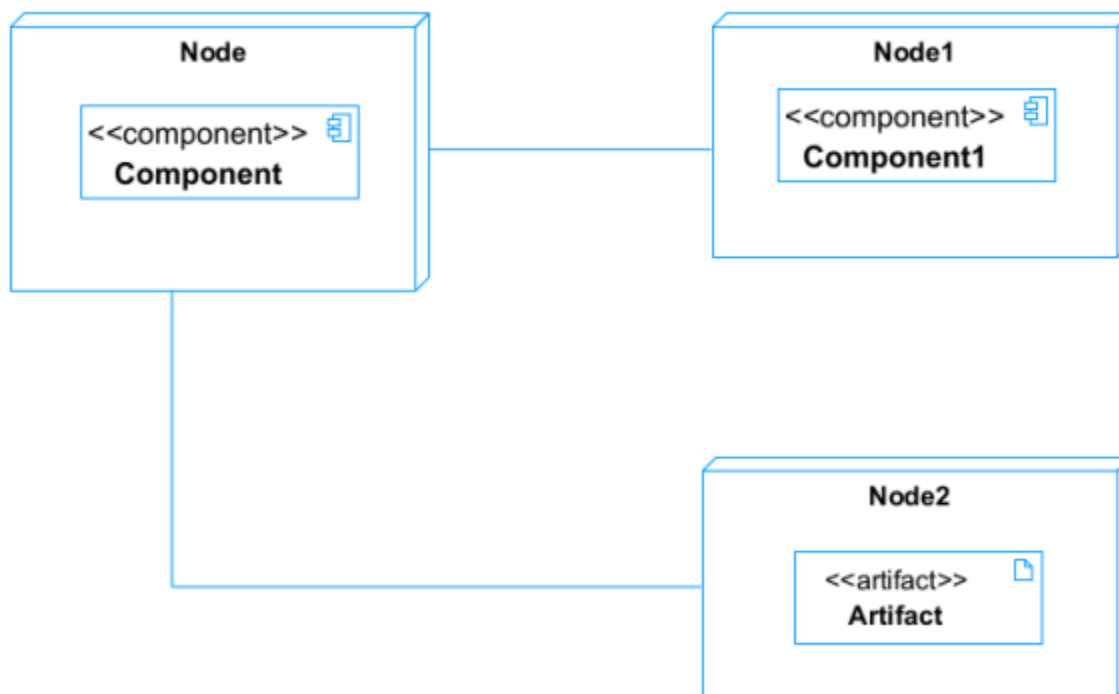


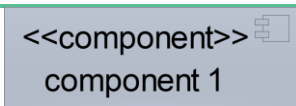



Figure 27: Formalism of Deployment diagram

c. Component of Deployment diagram

Table 18: Component of deployment diagram

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

d. CamerStay Deployment diagram

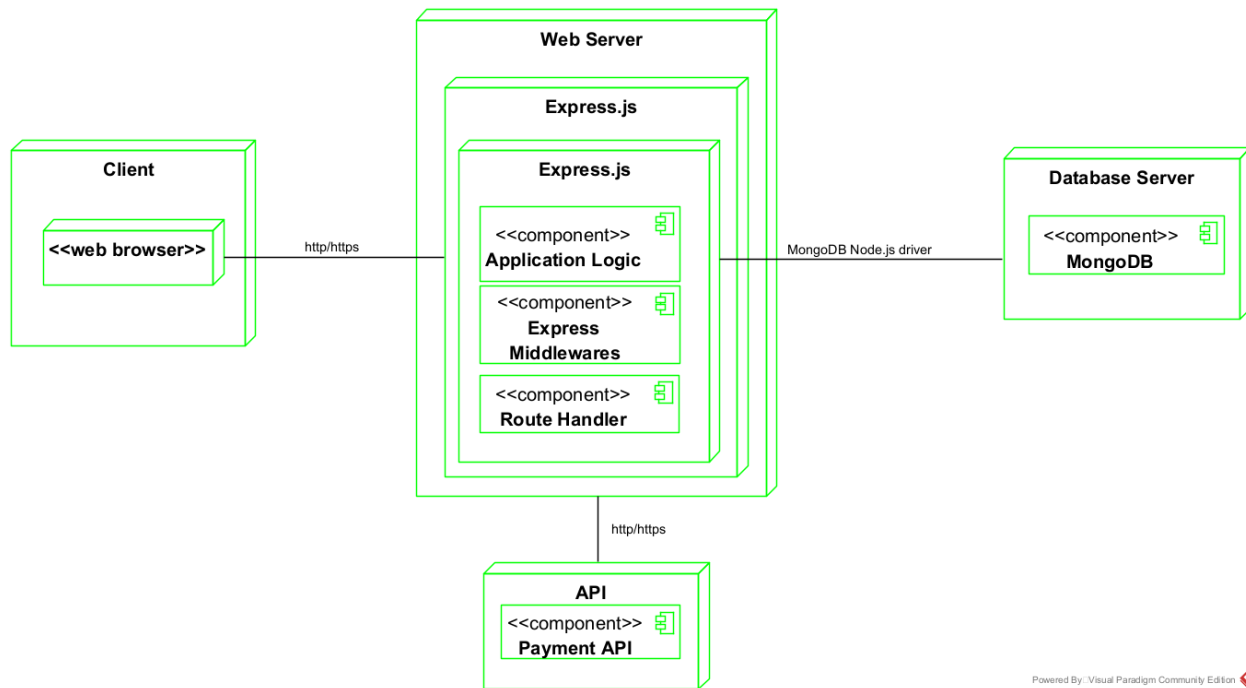


Figure 28: CamerStay deployment diagram

2. COMPONENT DIAGRAM

a. Definition

Component diagrams are used to model the physical aspect of a system. Now the question is what are this physical aspect? They are elements such as Executables, libraries, files, document etc. which resides in a node. The component diagram does not describe the functionality of the system, but it describes the components used to make those functionalities.

b. Formalism

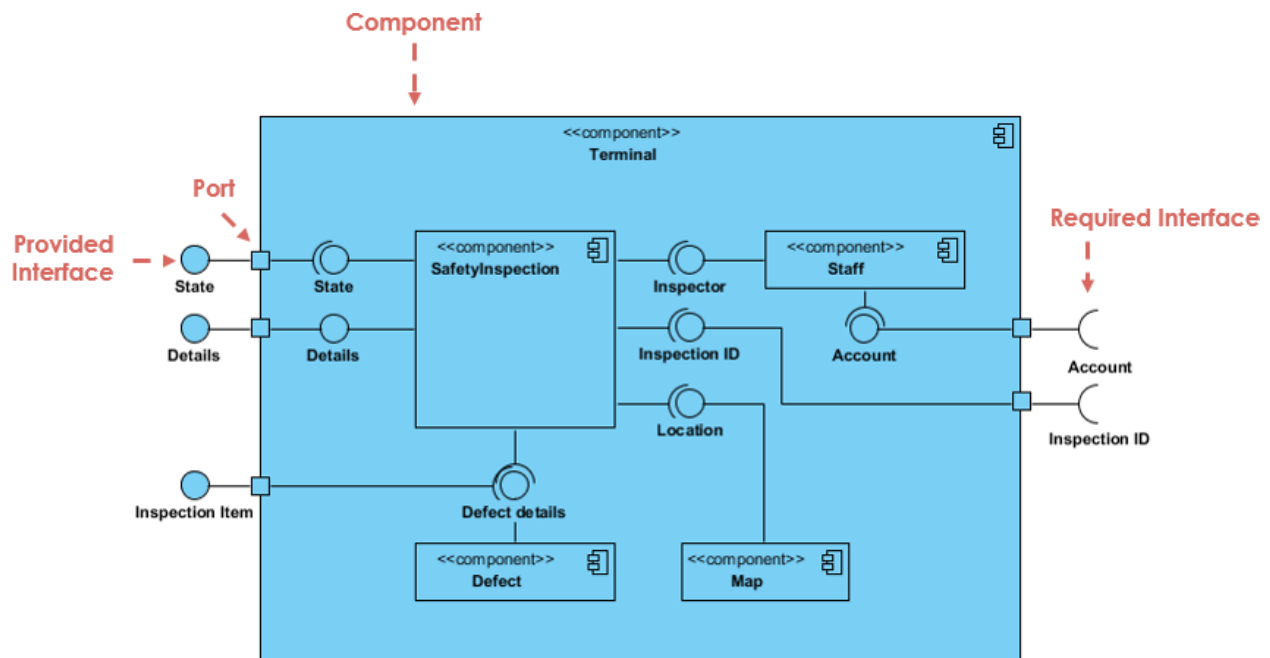
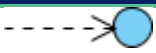





Figure 29: Component diagram formalism

c. Component of Component diagram

Table 19: Component of deployment diagram

Element	Notation	Description
Dependency		A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation.
Port		A port is often used to help expose required and provided interfaces of a component.
Interface		It is a point of interaction between a component and its environment or other components. It specifies the services that a component provides or requires from other components
Component		It represents a self-contained and replaceable unit that interacts with other components via provided and required interfaces

d. CamerStay Component diagram

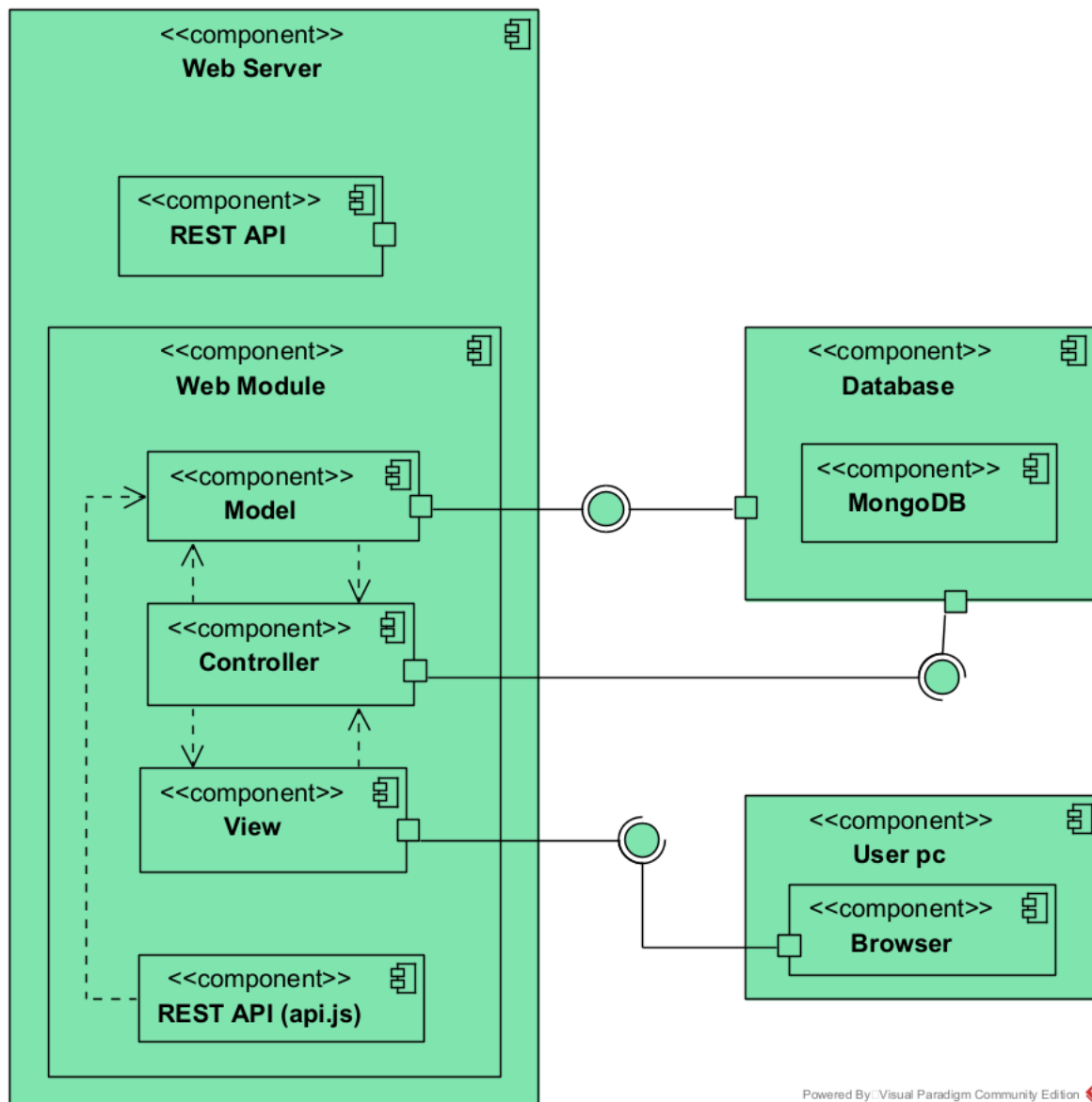


Figure 30: iCivil component diagram



CONCLUSION

During our realization phase, we successfully developed and implemented our application. To achieve this, we utilized our analysis and conception phase. We also created deployment and component diagrams which illustrate the modules, files, and assets, as well as how different elements interact with each other within our system. Our next step is to move onto the functionalities testing phase, where we will assess the efficacy of the various modules present in our app for different users.



PART 6: INSTALLATION GUIDE AND USER GUIDE



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



PREAMBLE

The aim of the user manual is to furnish our platform users with systematic guidance on installing and utilising the system.

CONTENT

INTRODUCTION

- I. INSTALLATION OF THE APPLICATION
- II. SHOWCASE

CONCLUSION



INTRODUCTION

This is the ultimate stage of our report. In this phase, we will dissect the requirements for our system, the essential installation process, system and feature accessibility in a step-by-step approach to aid first-time users with easy platform setup. Each process will be accompanied by relevant images. Subsequently, we shall present our application by showcasing the various screens.

I. INSTALLATION OF THE APPLICATION

A. DATABASE SERVER INSTALLATION

To deploy software, we need to install a database server. In our case, we will install MongoDB on our Windows system.



Figure 31: MongoDB logo

STEP 1: Downloading the MongoDB MSI installer Package

We go to [Download MongoDB Community Server | MongoDB](https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/) in order to download MongoDB. Note that we are downloading the community version since it is free and open source.

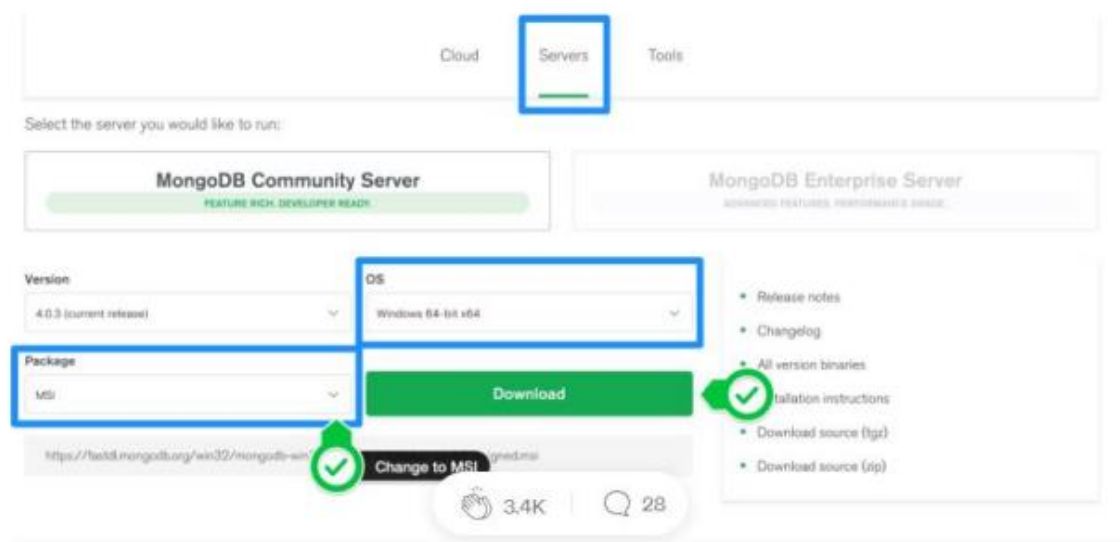


Figure 32: Downloading MongoDB community server

STEP 2: Install MongoDB with the installation Wizard

- A. Make sure you logged in as a user with admin privileges. Then navigate to your downloads folder and double click on the .msi package you just downloaded. This will launch the installation wizard.

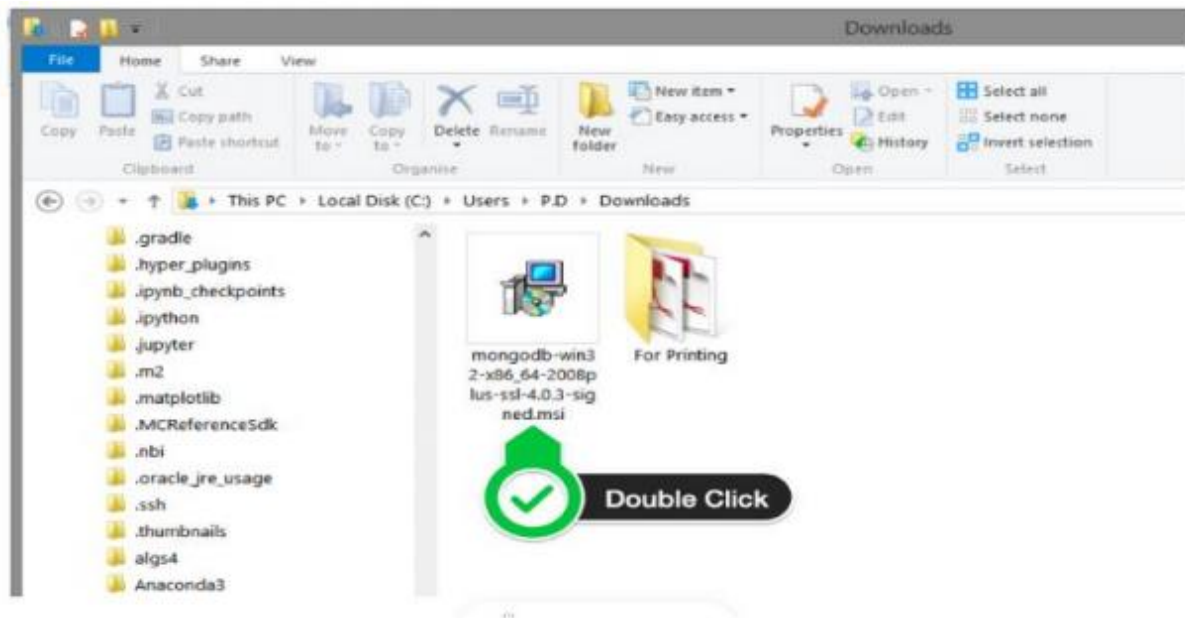


Figure 33: Launching the installation wizard

- B. Click on the next button



Figure 34: Click next

C. Accept the license agreement the click Next

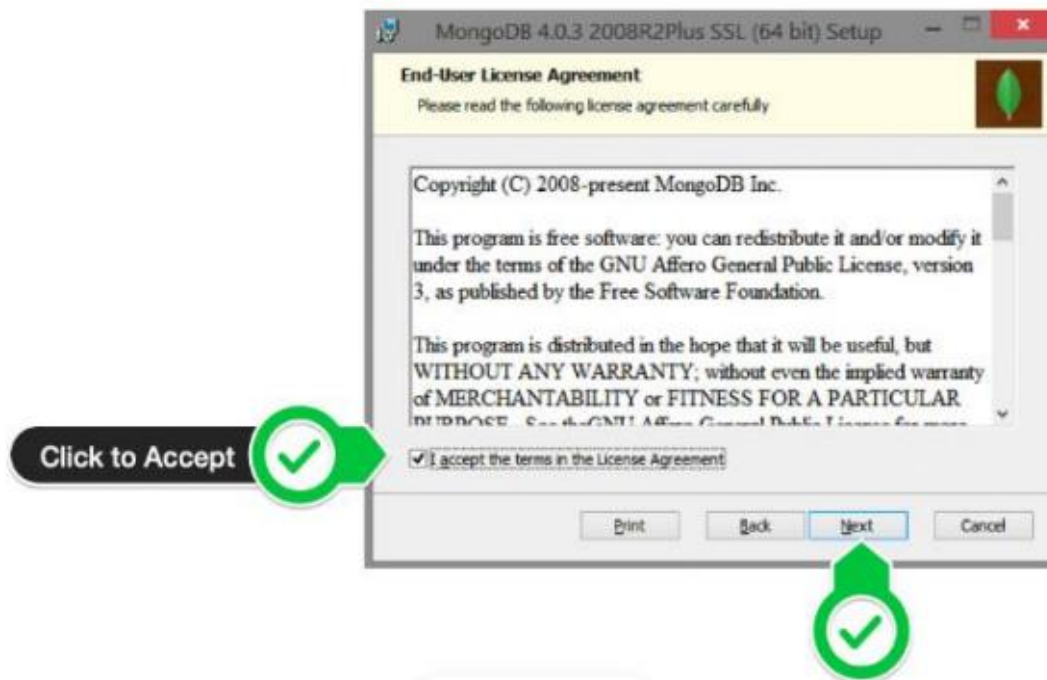


Figure 35: Agreement License

D. Select the complete setup

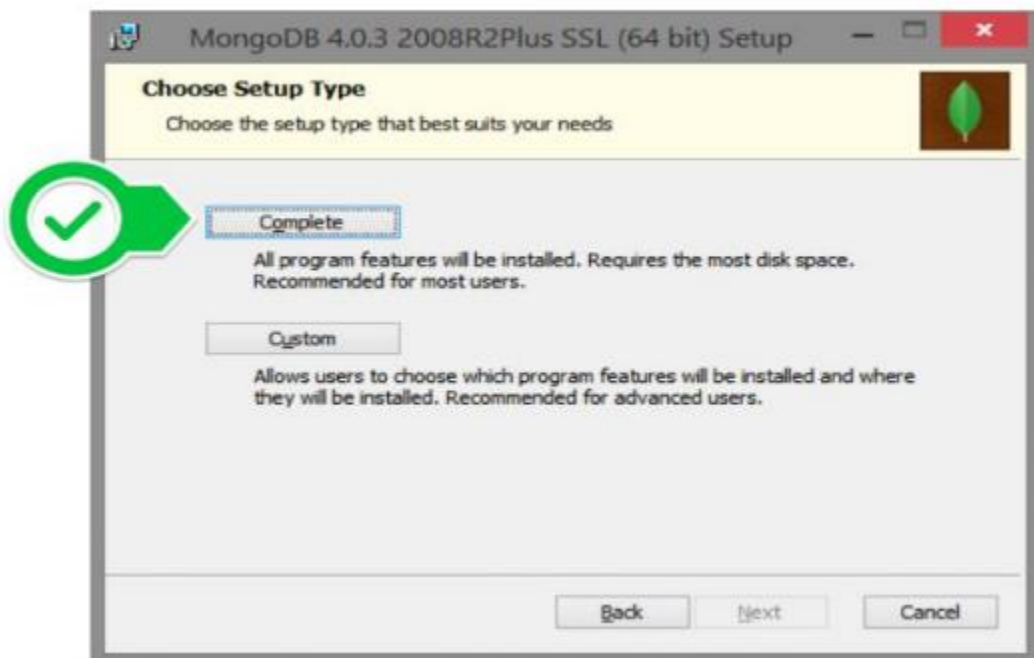


Figure 36: Select complete setup

- E. Select “Run service as Network Service” and make a note of the data direction, we’ll need this later.

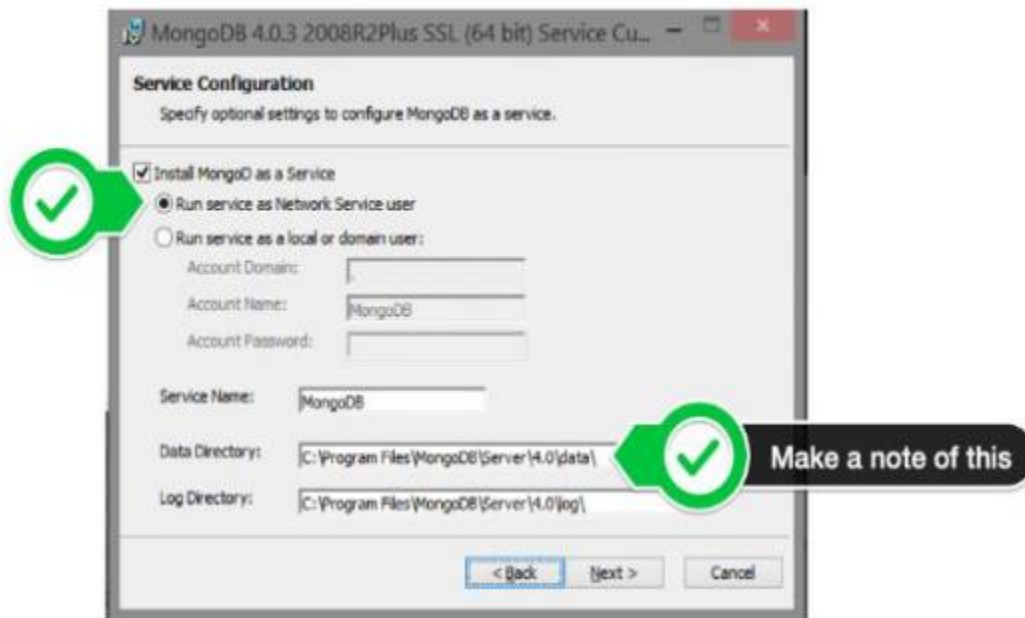


Figure 37: Run service network user

- F. Select mongo DB compass since you don’t have already

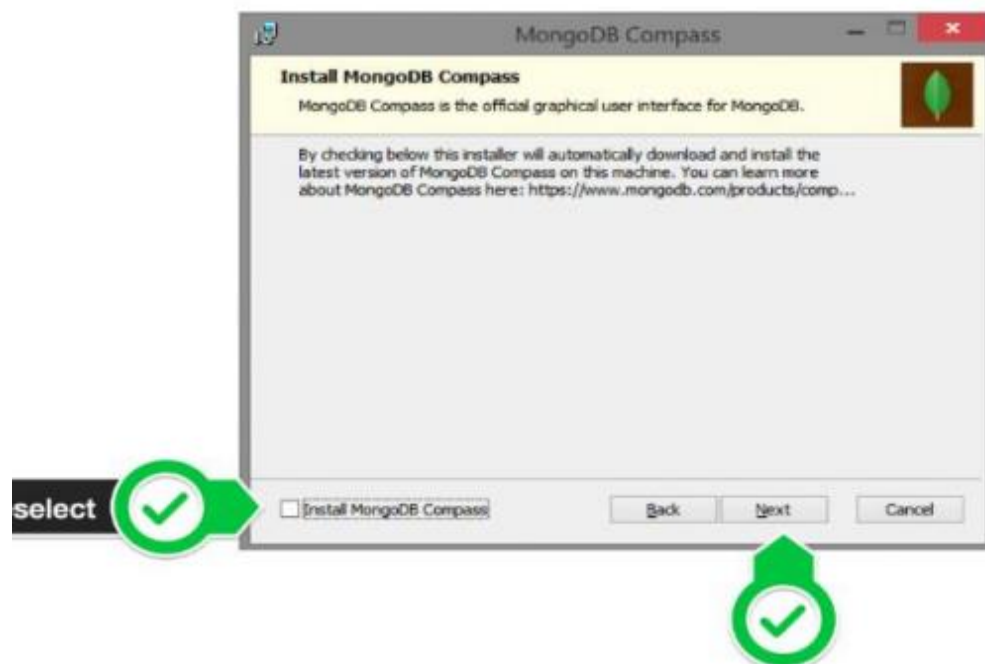


Figure 38: Selecting MongoDB compass and click next

G. Click install to begin installation

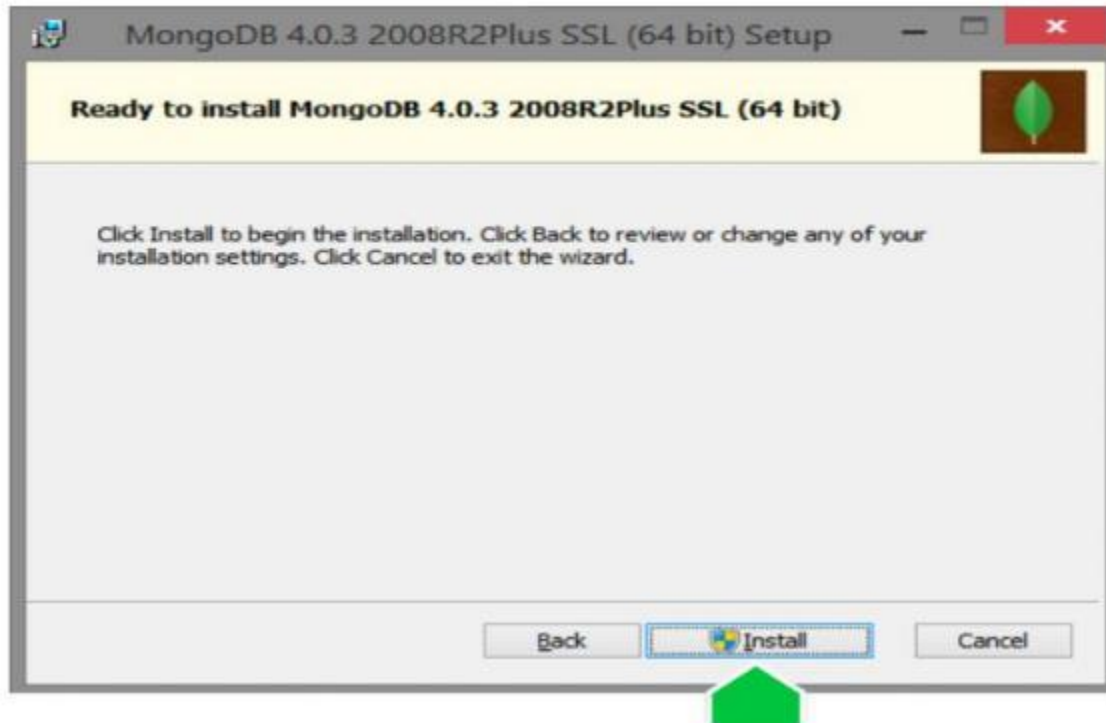


Figure 39: Click on install to start installation

H. Click on finish to complete installation



Figure 40: Click on finish to complete installation

II. SHOWCASE

In you web browser, enter the link of the application , no need to have an internet connection for it to run.

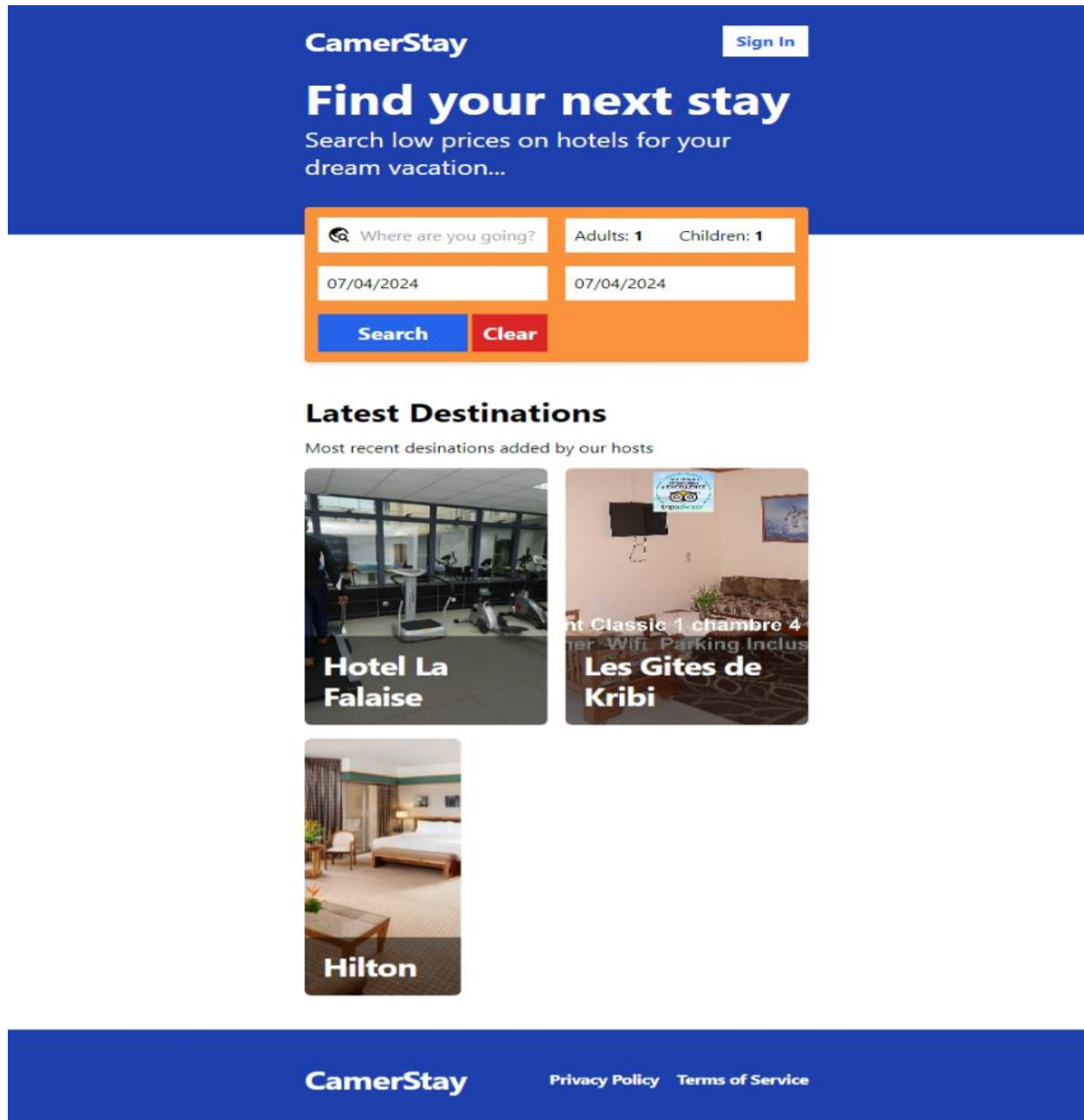


Figure 41: Home Page



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



CamerStay [My Bookings](#) [My Hotels](#) [Sign Out](#)

Find your next stay

Search low prices on hotels for your dream vacation...

Add Hotel

Type

Facilities

☐ Free WiFi ☐ Parking ☐ Airport Shuttle ☐ Family Rooms ☐ Non-Smoking Rooms
☐ Outdoor Pool ☐ Spa ☐ Fitness Center

Guests

Images

No file chosen

CamerStay [Privacy Policy](#) [Terms of Service](#)

Figure 42: Add hotel page



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



CamerStay[My Bookings](#)[My Hotels](#)[Sign Out](#)

Find your next stay

Search low prices on hotels for your dream vacation...

Where are you going?

Adults: **1** Children: **1**

07/04/2024

07/04/2024

Search

Clear

Filter by:

Property Rating

- ☐ 5 Stars
- ☐ 4 Stars
- ☐ 3 Stars
- ☐ 2 Stars
- ☐ 1 Stars

Hotel Type

- ☐ Budget
- ☐ Boutique
- ☐ Luxury
- ☐ Ski Resort

3 Hotels found

Sort By

Figure 43: Search page



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



CamerStay[My Bookings](#)[My Hotels](#)[Sign Out](#)

Find your next stay

Search low prices on hotels for your dream vacation...

Where are you going?

Adults: **1** Children: **1**

07/04/2024

07/04/2024

Search

Clear

No bookings found

CamerStay[Privacy Policy](#)[Terms of Service](#)

Figure 44:Bookings page



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



CamerStay[Sign In](#)

Find your next stay

Search low prices on hotels for your dream vacation...

Where are you going?

Adults: 1 Children: 1

07/04/2024

07/04/2024

Search

Clear

Create an Account

First Name

Last Name

Email

Password

Confirm Password

Create Account

Figure 45: Create account page



GENERAL CONCLUSION

In conclusion, CamerStay marks a pivotal advancement in the realm of hotel booking services tailored for Cameroon. By harnessing the power of MongoDB and the MERN stack, the platform not only delivers robust functionality but also ensures a seamless user experience from booking to management. This technological foundation supports administrators in efficiently overseeing multiple hotels while providing customers with an intuitive interface for browsing, booking, and managing reservations. As CamerStay continues to evolve, its commitment to scalability, security, and user-centric design positions it as a cornerstone in the digital transformation of the hospitality industry in Cameroon. With ongoing enhancements and a dedication to meeting user expectations, CamerStay promises to redefine convenience, reliability, and accessibility in hotel booking services across the region.



BIBLIOGRAPHY

- ♥ Mrs. TCHINGA Alice UML course AICS-Cameroon 2022-2023
- ♥ OMG, OMG Unified Modelling Language 2.5, Object Management Group, March 2015, 794 pages.
- ♥ Mercurial 2022
- ♥ Mercurial 2023



WEBOGRAPHY

- ♥ [OOAD - Object Oriented Analysis \(tutorialspoint.com\)](https://www.tutorialspoint.com/object-oriented-analysis/object-oriented-analysis-tutorial.htm) Object Oriented Analysis & Design Tutorial March 20th, 2024 1:00 PM
- ♥ [React Tutorial \(w3schools.com\)](https://www.w3schools.com/react/) React tutorial March 10th, 2024 8:25 AM
- ♥ [Installation - Tailwind CSS](https://tailwindcss.com/docs/installation) Tailwind installation March 10th 12: 11 PM
- ♥ [Tailwind CSS Navbar for React - Material Tailwind \(material-tailwind.com\)](https://material-tailwind.com/) How to create a tailwind CSS navbar March 11th 6: 23 PM
- ♥ https://primereact.org DataTable. March 18th, 2024, 3:30 PM.
- ♥ [React Table: A complete guide with updates for TanStack Table - LogRocket Blog](https://logrocket.blog/react-table-a-complete-guide-with-updates-for-tanstack-table/) How to use React data table April 18th, 2024, 3:30 PMs
- ♥ [Redux Essentials, Part 1: Redux Overview and Concepts | Redux](https://redux.js.org/overview) How to use Redux April 20th 2024, 7:51 AM
- ♥ [Redux Tutorial \(with Redux Toolkit\) - YouTube](https://www.youtube.com/watch?v=l8CS9AMBSIQ) Redux Toolkit tutorial April 19th, 2024
- ♥ [youtube.com/watch?v=l8CS9AMBSIQ](https://www.youtube.com/watch?v=l8CS9AMBSIQ) How to use React Router April 22nd, 2024
- ♥ [how to create layout in react router - YouTube](https://www.youtube.com/watch?v=l8CS9AMBSIQ) How to create a react layout April 22nd, 2024 9:00 PM
- ♥ [How to Validate Forms in React – A Step-By-Step Tutorial for Beginners \(freecodecamp.org\)](https://www.freecodecamp.org/news/how-to-validate-forms-in-react-a-step-by-step-tutorial-for-beginners/) How to validate forms May 24th, 2024 11:00 AM
- ♥ dev.to/mr_alin3n/folder-structure-for-nodejs-expressjs-project-4351 Nodejs Folder Structure May 25th, 2024 3: 3 PM
- ♥ [Getting Started with MongoDB & Mongoose | MongoDB](https://mongoosejs.com/docs/getting-started) Getting Started with MongoDB tutorial May 28th, 2024, 6:00 AM
- ♥ [The Axios Instance | Axios Docs \(axios-http.com\)](https://axios-http.com/docs/instance) How to use axios instance May 30th, 2024 10:21 PM



TABLE OF CONTENT

DEDICATION	i
ACKNOWLEDGMENTS	ii
CONTENTS.....	iii
LIST OF FIGURES	iv
LIST OF TABLES	vi
ABBREVIATIONS.....	vii
ABSTRACT.....	viii
RESUME	ix
GENERAL INTRODUCTION.....	1
PART 1: EXISTING SYSTEM.....	2
PREAMBLE	3
CONTENT	3
INTRODUCTION	4
I. THEME PRESENTATION.....	5
II. STUDY OF THE EXISTING SYSTEM.....	5
III. LIMITATIONS OF THE EXISTING SYSTEM	5
IV. PROBLEMATIC	6
V. PROPOSED SOLUTION	6
CONCLUSION.....	7
PART 2: SPECIFICATION BOOK	8
PREAMBLE	9
CONTENT	9
INTRODUCTION	10
I. CONTEXT AND JUSTIFICATION.....	11



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



1.	CONTEXT:.....	11
2.	JUSTIFICATION.....	11
II.	OBJECTIVES OF THE PROJECT.....	12
1.	General Objective	12
2.	Specific Objectives	12
III.	EXPRESSION OF NEEDS.....	13
1.	FUNCTIONAL NEEDS	13
2.	NON-FUNCTIONAL NEEDS	14
IV.	ESTIMATED COST OF THE PROJECT.....	15
a.	Software Resources	15
a.	Human Resources.....	17
b.	Global Estimation	17
V.	PROJECT CONSTRAINTS	18
1.	Technical constraint.....	18
3.	Cost constraint	18
VI.	LIST OF PARTICIPANTS AND DELIVRABLES	19
1.	LIST OF PARTICIPANTS	19
2.	DELIVRABLES.....	19
	ANALYSIS PHASE	19
	CONCLUSION.....	20
	PART 3: ANALYSIS PHASE	21
	PREAMBLE	22
	CONTENT.....	22
	INTRODUCTION	23
I.	COMPARATIVE STUDY OF UML AND MERISE	24



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



a. MERISE	24
b. UML	24
1. STATIC OR STRUCTURAL DIAGRAMS	24
2. BEHAVIORAL PR DYNAMIC DIAGRAMS	25
II. COMPARATIVE STUDY OF UNIFIED PROCESS	27
a. Unified Process	27
b. Two Track Unified Process (2TUP)	27
a) The left branch (functional branch)	28
b) The right branch (Technical branch).....	28
c) The middle branch (Realization or Implementation branch)	29
III. JUSTIFICATION OF THE ANALYSIS METHOD	30
IV. MODELLING OF THE PROPOSED SOLUTION.....	31
A. Capture of Functional Needs	31
1. USE CASE DIAGRAM	31
a. Definition.....	31
b. Formalism.....	31
c. Components of use case diagram	32
d. Actors of our system	33
e. CamerStay use case diagram.....	34
e. Manage use case diagram.....	35
Textual Description.....	36
Formalism.....	36
2. ACTIVITY DIAGRAM	39
a. Definition.....	39
b. Formalism.....	39
c. Component of an Activity diagram	40
d. Activity diagram of Authenticate	42
e. Activity diagram of Create account.....	43
f. Activity diagram of Make Payment	44
f. Activity diagram of Booking	45
3. COMMUNICATION DIAGRAM.....	46



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



a.	Definition.....	46
b.	Formalism.....	46
c.	Components of a communication diagram.....	47
d.	Communication diagram of Authenticate.....	48
e.	Communication diagram of Create account	48
f.	Communication diagram of Make Payment.....	49
4.	SEQUENCE DIAGRAM	50
a.	Definition.....	50
b.	Formalism.....	50
c.	Component of sequence diagram.....	51
d.	Sequence diagram of Authenticate.....	52
e.	Sequence diagram of Make Payment	53
f.	Sequence diagram of Create declaration	53
	CONCLUSION.....	54
	PART 4: CONCEPTION PHASE.....	55
	PREAMBLE	56
	CONTENT.....	56
	INTRODUCTION	57
I.	TECHNICAL BRANCH	58
A.	GENERIC DESIGN	58
a.	Hardware diagram of the system.....	58
a.	High Level Architecture of the System	58
B.	CAPTURE OF TECHNICAL NEEDS.....	59
a.	Physical Architecture.....	59
b.	Logical Architecture.....	60
C.	RELATED UML DIAGRAMS	61
1.	CLASS DIAGRAM.....	61
a.	Definition.....	61
b.	Formalism.....	61
c.	Component of a Class diagram	62



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



d. CamerStay Class diagram	63
e. Business Rule.....	64
2. STATE MACHINE DIAGRAM	65
a. Definition.....	65
b. Formalism.....	65
c. Component of a state machine diagram.....	66
d. State machine diagram of Authenticate	67
3. PACKAGE DIAGRAM.....	68
a. Definition.....	68
b. Formalism.....	68
c. Component of package diagram	69
d. CamerStay Package diagram	70
CONCLUSION.....	71
PART 5: REALIZATION PHASE	72
PREAMABLE	73
CONTENT	73
INTRODUCTION	74
1. DEPLOYMENT DIAGRAM	75
a. Definition.....	75
b. Formalism.....	75
c. Component of Deployment diagram	76
d. CamerStay Deployment diagram	77
2. COMPONENT DIAGRAM.....	78
a. Definition.....	78
b. Formalism.....	78
c. Component of Component diagram	79
d. CamerStay Component diagram	80
CONCLUSION.....	81
PART 6: INSTALLATION GUIDE AND USER GUIDE	82
PREAMABLE	83



CONCEPTION AND REALIZATION OF A HOTEL BOOKING APP IN CAMEROON



CONTENT	83
INTRODUCTION	84
I. INSTALLATION OF THE APPLICATION	85
A. DATABASE SERVER INSTALLATION	85
II. SHOWCASE	90
GENERAL CONCLUSION	95
BIBLIOGRAPHY	i
WEBOGRAPHY	ii
TABLE OF CONTENT	iii