REPUBLIC OF CAMEROON

Peace-Work-Fatherland



LOCALHOST ACADEMY

MINEFOP Approval No:000623/MONEFOP/SDGSF/CSACD/CBAC

Tel : (+237) 696 984 233/ 674 647 010

E-mail: contact@localhost-academy.com

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

**INTERNSHIP REPORT**

THEME:

CONCEPTION AND REALIZATION OF A TRAVEL PLANNING AND BOOKING APPLICATION

Case study: Cameroon

Internship carried out from the 08th July to the 30th September 2024.

In view of obtaining an **Engineering Diploma** in Computer Sciences level III Software Engineering.

Written by: **MENKAHE TEUM ELISE EMMANUELLA,** level III student at AICS Cameroon

Under the supervision of:

ACADEMIC SUPERVISOR

**Mr. MOHAMADOU Alhadji** Teacher at AICS Cameroon

PROFESSIONAL SUPERVISOR

**Mr. FOLONG EMERSON**

Teacher at LocalHost Academy

ACADEMIC YEAR 2023-2024

# DEDICATION

# ACKNOWLEDGEMENTS

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

* **Mr. Armand Claude ABANDA** resident representative, head of school of the African Institute of Computer Sciences, Paul BIYA Center of Excellence, without his supervision during the academic year this project would have not been possible to realize;
* **Mr. MOHAMADOU Alhadji** our academic supervisor for his guidance and follow up during this project;
* The director of LocalHost Academy, **Mr. ROSNY NGOUANET TSIDIE** for giving us the opportunity to serve as interns in the enterprise;
* **Mr. FOLONG EMERSON**, our professional supervisor for a cheerful welcome and a good working experience in the enterprise;
* **Mr. AGBOR Anderson,** and many others for the guidance during the academic year and during this project.;
* Our thanks also go to all the staff of AICs Cameroon who relentlessly have impacted knowledge for over two years to prepare us for this challenge;
* Thanks to all the staff of **LocalHost Academy** for the support they gave us throughout this internship period.;
* We also thank all our family members and tutors who gave us both their moral, and financial support during this project;
* We thank our friends and classmate;
* We warmly thank the members of the jury for accepting to evaluate our work, and who were kind enough to honor us with their presence;
* Finally, all our gratitude to all those who have contributed closely or at a distance by their support for the realization of this project;

# CONTENT

# LIST OF TABLES

# LIST OF FIGURES

# GLOSSARY

Admin: Administrator.

AICS: African Institute of Computer Sciences.

API: Application Programming Language.

CD-ROM: Compact Disk Read Only Memory.

CSS: Cascading Style Sheet.

DBMS: Data Base Management System.

HTML: HyperText Transfer Protocol

HTTP: Hyper Text Transfer Protocol.

MERISE:Méthode D’étude et de Réalisation Informatique pour les Systèmes D’Entreprise

MVC: Model View Controller.

UML: Unified Modeling Language.

URL: Uniform Resource Locator.

2TUP: Two Track Unified Process.

# ABSTRACT

# RESUME

# GENERAL INTRODUCTION

# PART ONE : INSERTION PHASE

Preamble

This section of our document will enable us to present how we were welcomed in the enterprise, how we discovered our working environment, partners, the structure, the localisation plan, the organisation of the enterprise and a brief introduction to our project.

CONTENT OVERVIEW

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF LOCALHOST ACADEMY
3. ORGANISATION OF LOCALHOST ACADEMY
4. HARDWARE AND SOFTWARE RESOURCES
5. BRIEF PRESENTATION OF THE PROJECT

CONCLUSION

CONTENT OVERVIEW

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF LIFE TECH SARL
3. ORGANISATION OF LIFE TECH SARL
4. HARDWARE AND SOFTWARE RESOURCES
5. BRIEF PRESENTATION OF THE PROJECT

CONCLUSION

CONTENT OVERVIEW

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF LIFE TECH SARL
3. ORGANISATION OF LIFE TECH SARL
4. HARDWARE AND SOFTWARE RESOURCES
5. BRIEF PRESENTATION OF THE PROJECT

CONCLUSION

CONTENT OVERVIEW

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF LIFE TECH SARL
3. ORGANISATION OF LIFE TECH SARL
4. HARDWARE AND SOFTWARE RESOURCES
5. BRIEF PRESENTATION OF THE PROJECT

CONCLUSION

## INTRODUCTION

The insertion phase is the integration of an individual in a foreign group in other to familiarise with the new environment where the language, the actions, the way of thinking must be purely professional. The objective being to put in practise what was taught in the African Institute of Computer Sciences (AICS) Cameroon, we did an internship for a duration of 3 months which is the reason why we integrated LOCALHOST ACADEMY. The goal been to convert our theoretical knowledge into practice that can help to provide a solution to a specific problem.

## Welcome and integration

On Monday July 15th, 2024, we went to the premises of LOCALHOST ACADEMY where we were received by **Mr. ROSNY NGOUANET TSIDIE** (director of LOCALHOST ACADEMY). He presented himself and then drilled on the history of LOCALHOST ACADEMY and how work was carried out in the enterprise. After this, we were asked to present our selves. This was to familiarize us with the new environment and give us a better room for adaptation. Afterward, we started with the analysis phase of the project.

## II. General presentation of LOCALHOST ACADEMY

A. History

LOCALHOST ACADEMY is company created by **Mr. DURRELL NGOUANET DONGMO** (Polytechnic Design Engineer) to meet the needs of Cameroonians in the digital domain. It was created in the year 2019.

### B. Mission

With the vision and perseverance of its CEO, this company has as mission;

* Provide training and certifications to improve qualified human resources in many fields of study.
* Take an active part in the sustainable development of the world through innovative solutions and virtual reality.

### C. Activities

The main activities of LOCALHOST ACADEMY are;

* Application design and development.
* Software maintenance.
* Specialised and practical training in;
  + Full stack PHP and JavaScript web development
  + Mobile programming with Flutter.
  + Full stack Python development.
  + Network, security system and cloud.
  + Hacking and cybersecurity.
  + Artificial intelligence and data analysis with python.
  + Computer science and office tools.
  + Digital marketing and web design.
  + Graphic design.
  + Video editing.
  + Creation of professional websites.

### D. Key realisations of LOCALHOST ACADEMY

LOCALHOST ACADEMY has many realisations, some of which are;

* E-commerce website (DURRELL Market).
* Point of sale management (Opti Sale).
* Website that brings together entrepreneurs (The Starterbook).
* Announcement website (Annonce Flash).
* Snapchoice social network.

### E. Juridical status

LOCALHOST ACADEMY is a limited liability company with a low financial social capital and currently directed by its founder Mr. DURRELL NGOUANET DONGMO.

### F. Geographical location

LOCALHOST ACADEMY is located at Carrefour Kaka in the town of Yaounde. It can easily be localised through the localisation plan below.

### F. Company’s identification form

Table 1 : Company’s identification form

|  |  |
| --- | --- |
| Identification form | |
| Company name | LOCALHOST ACADEMY |
| Type of enterprise | Institution |
| Chief executive | Mr. DURRELL NGOUANRT DONGMO |
| Phone number | +237 696 984 233 |
| Website | www.localhost-academy.com |
| Language | English and French |
| Logo |  |

## III. Organisation of LOCALHOST ACADEMY

### A. Administrative organisation

LOCALHOST ACADEMY is headed by the Chief Executive Officer (CEO) who manages the day-to-day activities and the implementation of the company’s policies. LOCALHOST ACADEMY is administratively organized as follows;

#### a) The General Manager

It is the biggest unit in the chain of decision making within the company. It has as mission:

* The good functioning of all the departments.
* Define project strategies.
* Motivate the employees of all services.
* To evaluate the company activities and gives orientations.

#### b) Academic Affairs Department

* Designing and producing the specification book.
* Respect and enforce the application of standards of the company.
* Design and validate the products created by the development teams.
* Write training content.
* Help desk assistance.
* Innovation.

#### c) Training and Certification Department

In a way, this is the heart of the company This department is in charge of:

* Elaboration of training programs and projects.
* Training and issuing certificates after training.

#### d) The Marketing and Communication Department

This department takes cares of the reputation, the images of the company and its products. It also handles sales and after sales services. Executes the business strategies and manages human capital. Its functions are;

* Executes marketing plans and strategies related to corporate image policy.
* Train and animate sales people to execute the business strategy.
* Communicate about the different services offers by the enterprise.

#### e) Financial Department

This department manages the accounting, the archives and the functioning of the

enterprise at the administrative level. Its missions are:

* Produce financial statement and appendices.
* Produce the tax document required by the tax authorities.
* Produce the cash statements.
* Monitor cash-flow and accounts.

### B. Functional organisation

The functional branch of LOCALHOST ACADEMY is organised as follows;

Chief Executive Officer

General Director

Academic Affairs Department

Financial Department

Marketing and Communication Department

Secretary

Cleaning Service

## IV. Hardware and software resources of LOCALHOST ACADEMY

### Hardware

|  |  |  |
| --- | --- | --- |
| Hardware resources | | |
| Name | Name | Quantity |
| Computer | HP, 2 laptops | 03 |
| Printer | EPSON | 01 |
| Flybox | Camtel | 02 |
| Television | LG | 03 |
| HDMI cables |  | 03 |

### Software

|  |  |
| --- | --- |
| Software resources | |
| Operating system | Windows, Linux |
| Antivirus | 360 Total Secuirity |
| Application | Visual studio code, Xampp, Google Chrome, Postman, Visual Paradigm, GitLab, Docker, Adobe Photoshop |

## V. Brief presentation of the project

Our project consists in developing a platform that will enable users to plan their trips in Cameroon and the user can also search for an accommodation in any town of Cameroon and reserve a travel ticket.

## CONCLUSION

Being at the end of the insertion phase, which covered not only the general presentation of the company, but also our familiarisation with the work environment. This step took place in a friendly atmosphere, with a spirit of trust due to the professional experience of the supervisor and co-workers. In addition, this phase was a benefit to us due to the perception of the need, importance of collaborative work and especially the most important aspect confronting the realities of the workplace.

# PART TWO : TECHNICAL PHASE

**Preamble**

It was a question for us in the insertion phase to familiarize ourselves with our reception structure. The focus in this phase will be on the different characteristics, specificities and expectations of the subject submitted to our study.

CONTENT OVERVIEW

CHAPTER ONE: THE EXISTING SYSTEM

CHAPTER TWO: THE SPECIFICATIO BOOK

CHAPTER THREE: ANALYSIS PHASE

CHAPTER FOUR: THE CONCEPTION PHASE

CHAPTER FIVE: THE REALIZATION PHASE

CHAPTER SIX: FUNCTIONAL TESTING

CHAPTER SEVEN: USER GUIDE

# CHAPTER ONE: THE EXISTING SYSTEM

Preamble

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

CONTENT OVERVIEW

INTRODUCTION

1. PROJECT PRESENTATION
2. STUDY OF THE EXISTING SYSTEM
3. CRITICISM OF THE EXISTING SYSTEM
4. PROBLEMATIC
5. PROPOSED SOLUTIONS

CONCLUSION

## INTRODUCTION

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

## Project presentation

This travel planning platform is designed to simplify the process of organizing trips. It offers an all-in-one solution that allows users to explore destinations, book accommodations and travel tickets, and create personalized itineraries. The app is targeted at travelers who seek a unified experience, eliminating the need to switch between multiple apps or websites to plan and manage their journeys.

## Study of the existing system

The existing travel planning systems, whether through traditional travel agencies, online platforms like Expedia or TripAdvisor, or relying on third-party advice, offer fragmented and often inefficient solutions for travelers. Users typically need to navigate multiple sources to gather information about required booking options, and destination details. With these systems, travelers are forced to switch between different platforms or rely on inconsistent advice from friends, family, or online communities.

## Criticism of the existing system

**Fragmented Information**: Current travel planning systems are highly disjointed, requiring users to navigate multiple platforms to gather all necessary details for a trip. This makes the process inefficient and time-consuming.

**Reliance on Third-Party Advice**: Users frequently depend on inconsistent information from third-party sources like friends, family, or travel forums, which can lead to inaccuracies, outdated advice, and confusion.

**Accessibility Issues**: Traditional travel agencies, which many users still rely on, require physical visits and offer limited digital support, which can be inconvenient for users looking for quick and easy access to travel services.

## Problematic

**HOW CAN WE CREATE A FULLY INTEGRATED AND PERSONALIZED TRAVEL PLANNING PLATFORM THAT ELIMINATES THE NEED FOR USERS TO SWITCH BETWEEN MULTIPLE APPLICATIONS AND WEBSITES FOR BOOKING ACCOMMODATIONS, MANAGING ITINERARIES, AND PLANNING ACTIVITIES?**

## Proposed solutions

To address the issues with existing travel platforms, the proposed solution is to develop a unified travel planning application that integrates all essential services such as travel ticket booking, accommodations, activities, and itinerary management into a single platform. This app would feature personalized recommendations based on user preferences, ensuring a tailored experience. It would also provide real-time updates on travel conditions and secure payment options. By offering a simplified user interface and integrated customer support, the app would minimize confusion, reduce reliance on multiple platforms, and ensure a seamless, efficient travel planning process.

## CONCLUSION

The topic is a very essential part of the project. It helps to delimit the scope of work and to identify the system. The study of the existing system is also very crucial. This is because in order to ameliorate a system we first need to study it, criticize it to find the existing problems and limitation. When the study is completed, we find the problematic and proposed a solution which will solve the problems discovered.

# CHAPTER TWO: SPECIFICATION BOOK

Preamble

The specification book is a project manual that details the different products, construction materials and methods to be used in the project development. In other words, the specification book is a document which gives all the requirements to a structural, analytic and durable development of the results awaited.

CONTENT OVERVIEW

INTRODUCTION

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

CONCLUSION

## INTRODUCTION

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

## Context and justification

### Context

In today's travel landscape, users often face the challenge of juggling multiple platforms to organize a single trip. For example, imagine a traveler planning a vacation to Limbe. They might book their flight on one platform, search for accommodations on another platform, and then use another app to plan activities. On top of this, they need to manage their itinerary manually, switching between emails, apps, and calendars to keep track of bookings, confirmations, and plans. This fragmented process can lead to confusion, missed reservations, and even misinformation about travel requirements.

### Justification

A unified solution is necessary to streamline the travel planning process, offering an integrated, personalized experience that eliminates the need to switch between various apps and websites. This platform will provide a user-friendly interface that consolidates all essential travel services and delivers accurate, real-time information.

## Objectives of the project

### General objective

The general objective of this project is to develop a unified and user-friendly travel planning platform that integrates all essential travel services, such as booking travel tickets, accommodations, and activities, into a single application.

### Specific objectives

* Offer real-time travel updates.
* Implement personalized travel recommendations.
* Provide absolute itinerary management.
* Ultimately enhance the overall travel experience by reducing inefficiencies.
* Ensure secure transactions.

## Expression of needs

### Functional needs

In this context and functional requirement describe what the system or application should do.

1. User Registration and Authentication:

* Enable users to create accounts, login, and manage their profiles securely.

1. Booking Services:

* Enable users to book travel tickets, hotels, and activities from multiple providers.

1. Personalized Search and Recommendations:

* Provide personalized search results and suggestions based on user preferences.

1. Schedule Management:

* Allow users to view, manage, and modify their travel plans in a unified itinerary.

1. Real-Time Updates:

* Provide real-time notifications for modifications, cancellations, and travel advisories.

1. Payment Processing:

* Integrate secure payment gateways for users to book services directly within the app.

1. User Notifications and Alerts:

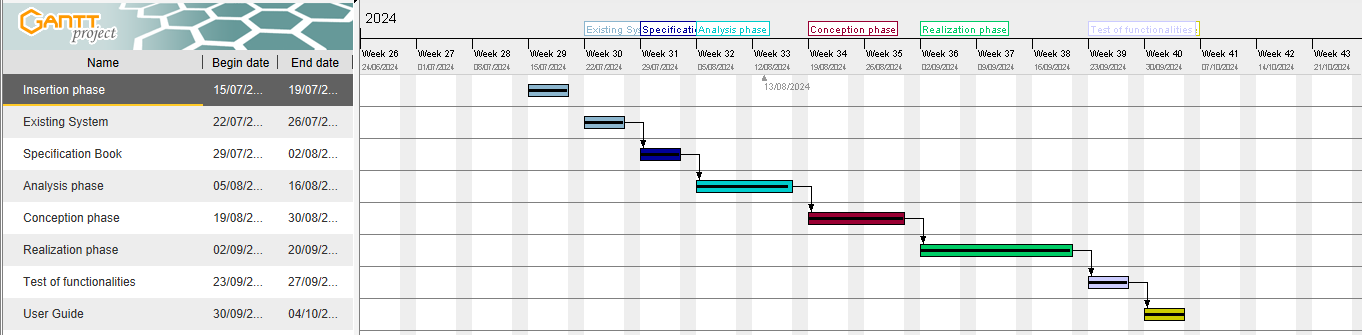
* Notify users of booking confirmations, reminders for upcoming trips, and special deals or promotions.

### Non-functional needs

* **Performance**: The system should load pages and respond to user actions within 2-3 seconds to ensure a smooth user experience.
* **Scalability**: The platform must be scalable to handle increasing numbers of users, bookings, and real-time updates as the user base grows.
* **Security**: Implement strong encryption for user data and transactions, ensuring the protection of personal information, payment details, and travel documents.
* **Maintainability**: The codebase should be modular and well-documented to allow for easy updates, debugging, and feature additions.
* **System Availability**: Ensure the system is available 24/7, with continuous monitoring to detect and resolve issues quickly without impacting user experience.

## Project planning

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.



## Estimated cost of the project

### Human resources

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Quantity | Cost Per Day | Total Price (FCFA) |
| Project Manager | 01 | 30 000 | 3 600 000 |
| Analyst | 01 | 25 000 | 600 000 |
| Designer | 01 | 20000 | 1 400 000 |
| Programmer | 01 | 15000 | 600 000 |
| Tester | 02 | 1000 | 140 000 |
| Total |  |  | 6 340 000 |

### Hardware resources

|  |  |  |
| --- | --- | --- |
| Material | Quantity | Price (FCFA) |
| Lenovo | 5 | 2 722 362.5 |
| Printer | 1 | 1 700 000 |
| Media installation | 2 | 2000 |
| Local network installation | 1 | 300000 |
| Removable disk | 2 | 253000 |
| Total |  | 3 208 182 |

### Software resources

|  |  |  |  |
| --- | --- | --- | --- |
| Designation | Description | Logo | Price (FCFA) |
| Microsoft Office 365 | Used for the creation of our report and PowerPoint |  | 70,701.340 |
| Google Cloud | Cloud computing platform provided by google to save our report |  | Free |
| Google Search | Free browser |  | Free |
| Visual Studio Code | Development environment |  | Free |
| SQLite | Lightweight, serverless, and self-contained database management system. |  | Free |
| Visual Paradigm | Modeling tool |  | Free |
|  | TOTAL |  | 70,701.340 |

### Total project estimated cost

|  |  |
| --- | --- |
| TOTAL PROJECT COST | |
| Resources | **Amount (FCFA)** |
| Human | **6 340 000** |
| Hardware | **3 208 182** |
| Software | **70,701.340** |
| Unforeseen 10% of sales | **407.680** |
| TOTAL | **80 657 202** |

## Project constraints

### Software constraints

* **Compatibility**: The platform must be compatible with different web browsers (Chrome, Firefox, Safari, etc.) and devices (computers, smartphones, tablets) to reach a wide audience.
* **Security**: Robust security measures must be put in place to protect users’ personal data and prevent any breach of confidentiality.
* **User-friendliness**: The user interface must be user-friendly to ensure a positive user experience, whatever the level of technological expertise.

### Material constraints

* **Internet availability**: The platform assumes that users have access to the Internet. However, efforts must be made to guarantee an acceptable experience even with limited Internet connections.
* **Hardware compatibility**: The platform must be designed to run on a variety of devices, from desktops to smartphones, taking into account differences in screen size and processing power.

### Quality constraints

* **Accuracy of information**: Guides and information provided must be accurate, up-to-date and verified, to avoid giving incorrect information to users.
* **Performance**: The platform must be fast and responsive to avoid excessive loading times and maintain user engagement.
* **Accessibility**: The platform must be designed to be accessible to people with disabilities, in accordance with web accessibility standards.

### Time constraints

* **Deadlines**: The project must be completed within the specified deadlines to meet user needs and stakeholder expectations.
* **Regular updates**: Future updates and enhancements to the platform must be planned to meet the changing needs of citizens and to maintain long-term relevance.
* **Thorough testing**: The time required to conduct rigorous tests to ensure that the platform functions correctly, is user-friendly and secure, should not be underestimated.

## Deliverables

A deliverable is any result, measurable (product, executable*)* or verifiable (Document), that results from the completion of a project phase or the entire project. At the end of our project we need to have:

* A full report including:
* The insertion file,
* Specifications,
* The analysis file,
* The design file,
* The project file,
* Installation and user guide.
* CD-ROM containing:
* Application source code,
* Our PowerPoint presentation,
* The database script,
* A link to the platform.

## CONCLUSION

The specifications are like a real compass that guides and supervises all the actions and activities of the project manager. Moreover, it allows him to surround himself with human, material, or even technical resources likely to satisfy the desire of the sponsor called to receive the product according to the criteria previously established.

# CHAPTER THREE: ANALYSIS PHASE

Preamble

The analysis is an important step in all computer science projects, no matter the size. In this section we shall present in details the analysis of the project to put in place, the modelling of the solution proposed and the functionalities to take into consideration with the different diagrams.

CONTENT OVERVIEW

INTRODUCTION

1. METHODOLOGY
2. MODELING

CONCLUSION

## INTRODUCTION

The analysis is a systematic study that enables to differentiate the different parts of a problem and to find solutions following a well-defined methodology. For our analysis, we are going to use the Unified Modeling Language (UML) which is coupled with the Two Tract Unified Process (2TUP). Here, we are going to compare the UML and MERISE and also compare the unified processes. Then we will dive directly into the modeling of the proposed solution consisting of diagrams that meets the requirements of the functional needs.

## Methodology

### Comparative study between UML and MERISE

1. **MERISE:**

MERISE (**Méthode d’Etude et de Réalisation Informatique pour des Systèmes d’Entreprise)**, is a method to build an automated information system which is efficient and adapted to the organisation. Like any other method of analysis and design, MERISE includes models, a language a process and tools. The MERISE method of analysis and design proposes an approach articulated simultaneously according to 3 axes to prioritize the concerns and question to be answered when conducting a project:

1. **UML:**

UML is a general-purpose visual [modeling language](https://en.wikipedia.org/wiki/Modeling_language) that is intended to provide a standard way to visualize the design of a system. The main aim of UML is to define a standard way to visualize the way a system has been designed. UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis

UML 2.5 consists of fourteen diagrams classified into structural and behavioral diagrams.

Table 2 : UML diagrams

|  |  |
| --- | --- |
| Structural diagrams | Behavioral diagrams |
| Component diagram | Use case diagram |
| Class diagram | Activity diagram |
| Package diagram | State machine diagram |
| Object diagram | Sequence diagram |
| Deployment diagram | Communication diagram |
| Composite diagram | Timing diagram |
| Profile diagram | Interaction overview diagram |

UML and MERISE are two modeling approaches used in information systems engineering. Although they have similar objectives, which are to design and model information systems, they differ in their different origin, scopes and approaches. Here’s a comparative study of the two.

Table 3. Differences between UML and MERISE

|  |  |  |
| --- | --- | --- |
|  | UML | MERISE |
| Orientation | It is object oriented which makes it ideally suited to modeling modern object-based IT systems. | It is based on a data processing approach and is better suited to traditional information systems. |
| Application domain | Widely used in software engineering for modeling and designing software systems, ranging from small applications to complex enterprise systems. | It is specifically tailored for information systems that handle large amounts of data such as databases, decision support systems. |
| Modelling constructs | It offers a rich set of modeling constructs that can be used to represent different perspectives of a system. | It provides a set of modeling concepts and techniques for capturing the structural and functional aspects of an information system. |
| Flexibly | It is more flexible and can be adapted to different situations, including object-oriented and non-object-oriented systems. | It is more structure and data-driven, which makes it less suitable for some modern systems. |

### Comparative study of the unified processes

1. **Unified Processes (UP):**

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.
* **An increment** is the difference between two released products at the end of two iterations.
* **Centered on architecture** the different models derived during the establishment of system must be reliable and coherent.
* **Driven by use case and requirements** enables the clear definition of a users’ needs and priorities respectively thereby minimizing the risk of project failure.

1. **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-sharped development life cycle that separates the functional aspect from the technical aspects, and the merging of these two forms the implementation aspect. 2TUP distinguishes therefore two branches: the functional and technical branches, the combination of the result of these two branches forms the third: the realization branch; where we realize our system.

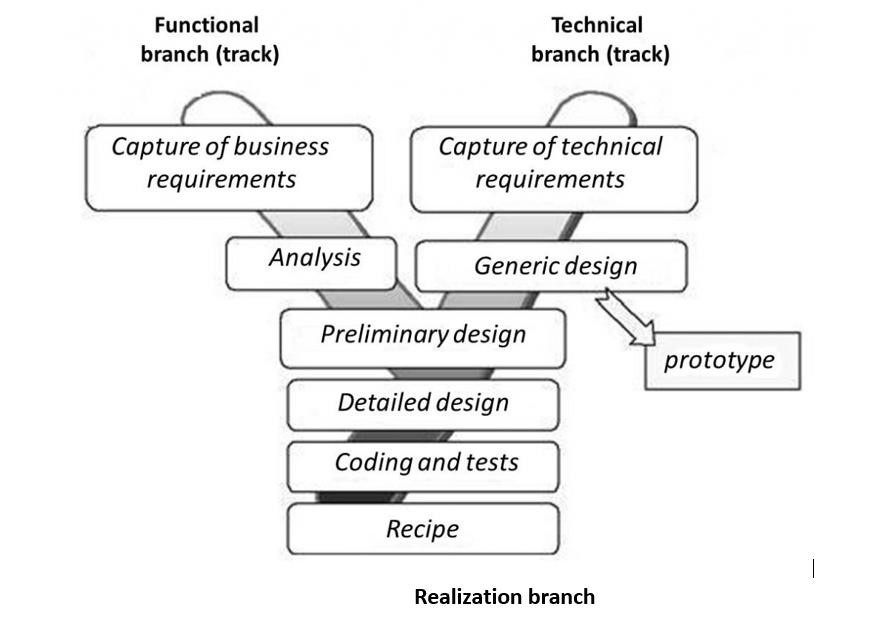


Figure 1 :2TUP diagram (Source: [https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing)](https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)

1. **Functional branch**

It captures the functional needs which help to prevent the production of software that does not fit the needs of the user. The analysis here consists of studying precisely the functional specification to obtain an idea of what the system is going to realize that its result is independent of the technology used.

1. **Technical branch**

The technical branch enumerates the technical needs and proposes a generic design valeted by a prototype. The technical needs here include the tools, materials, and technologies that will be used along with different constraints such as maximum response time and integration with the existing controls.

1. **Realization branch**

The realization branch helps to elaborate a detailed design of each feature of the system. The middle branch supports the following:

* Preliminary conception: Being the most sensitive step of the 2TUP, this is the confluence of the functional and technical branch. It ends when the deployment model, the operating model, the logical model, interfaces and software configuration model are defined.
* Detailed conception: This is the detailed design of each feature of the system.
* Coding and testing: This are the programming phase of the designed features, alongside testing of coded features.
* The recipe: Also known as the deliverables, is the validation phase of the functions of the developed system.

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

## Modeling

### Use case diagram

#### Definition

Use case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

#### Formalism

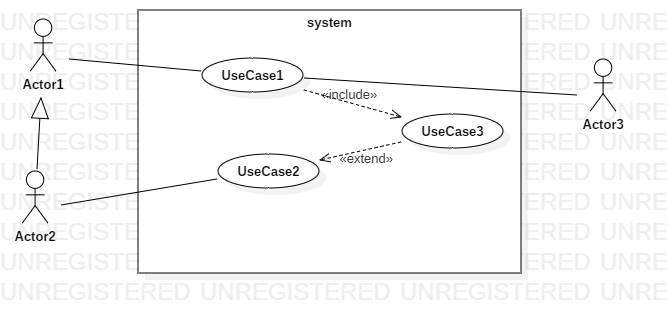


Figure 2 : Use case diagram formalism

Table 4 : Components of the use case diagram

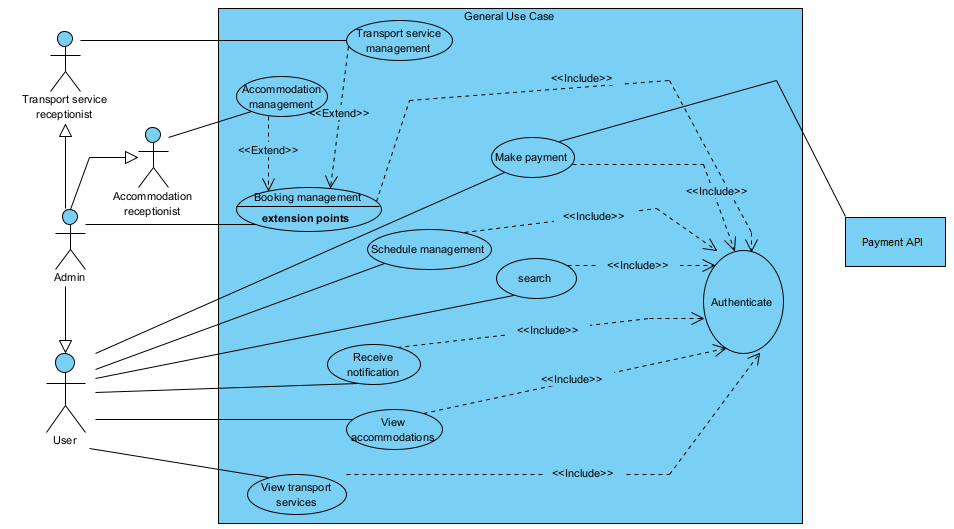
|  |  |  |
| --- | --- | --- |
| Element | Notation | Description |
| Use case |  | A use case corresponds to the objective of a system motivated by the needs of the actor. A use case is a verb in the infinitive form. |
| Actor |  | An actor is model element that interact with a system. |
| Association |  | The association expresses the relationship between the actor and the use case. |
| Extension |  | The extension relationship indicates that the source use case adds behaviour to the destination use case. |
| Inclusion |  | The inclusion relationship indicates that the use case is mandatory and part of the base use case. |
| Generalisation |  | The generalisation relationship shows that a use case is a kind of another. This relation also permits to decompose a complex case into smaller and simple cases. |
| System |  | The system represents the perimeter to which the use case relates. |

#### Actors of the system

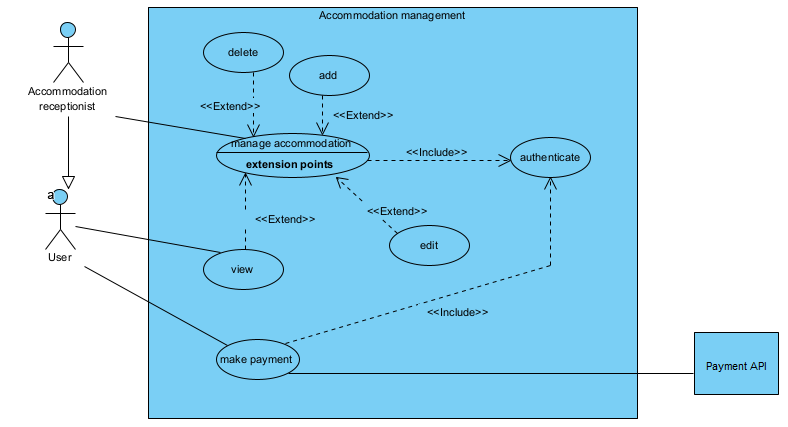
Table 5 : Actors of our system

|  |  |
| --- | --- |
| Actor | Role |
| Administrator | The administrator is in charge of the general functioning of the system. |
| Accommodation receptionist | Vacation rentals and other logging providers. |
| Transport service receptionist | Transport service providers. |
| User | Any person willing to use the application. |

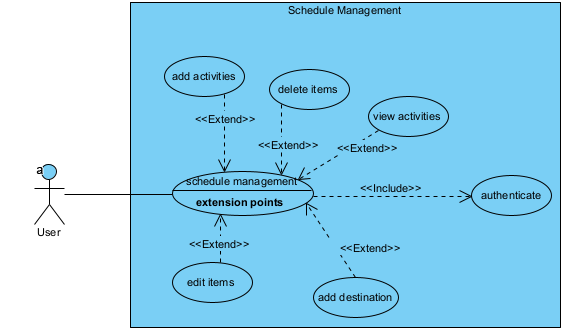
#### General use case diagram



#### Accommodation Management use case diagram



#### Schedule Management 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.

|  |  |
| --- | --- |
|  | USE CASE AUTHENTICATE |
| Summary | Permit users to access the system |
| Actors | User, hotel receptionist, agency receptionist, administrator |
| Pre-condition(s) | 1. The actor already has an account; 2. The app is launched |
| Trigger | The user clicks on login button |
| Nominal scenario | 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 |
| Alternative scenario | 1. 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. 2. 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. |

### Communication diagram

#### Definition

A communication diagram is a visual representation that illustrate the interactions and relationships between various components or objects within a system. It is a type of UML diagram that focuses on how objects or components communicate with each other to achieve a particular functionality or accomplish a specific task.

#### Formalism

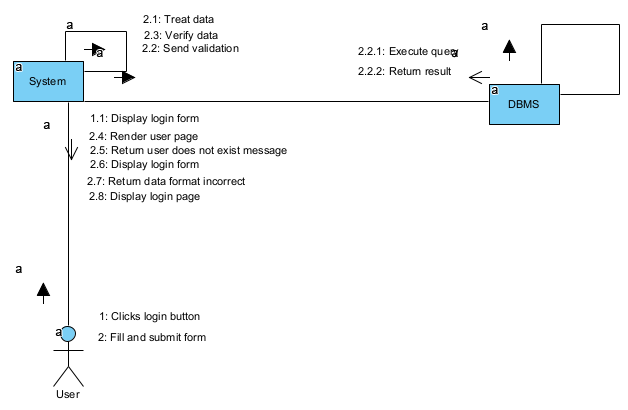


Figure 3 : Communication diagram formalism

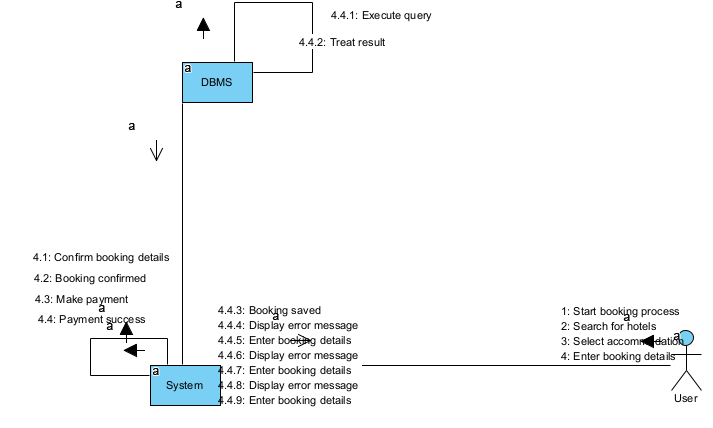
Table 6 : Communication diagram components

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Object |  | An object represents an individual participant in the interaction conversion. |
| Link |  | a straight line connecting two objects indicates a relationship between them |
| Call message |  | A call message defines a particular communication between lifeline of interaction that represent an innovation of target lifeline. |

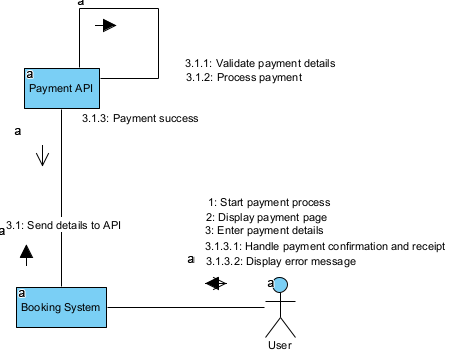
#### Authenticate communication diagram



#### Book accommodation communication diagram



#### Make payment communication diagram



### Sequence diagram

#### Definition

A sequence diagram is a type of UML diagram that illustrates the interactions and order of messages exchanged between objects or components within a system.it depicts the dynamic behaviour of a system by showing the sequence of events or actions that occur over time.

#### Formalism

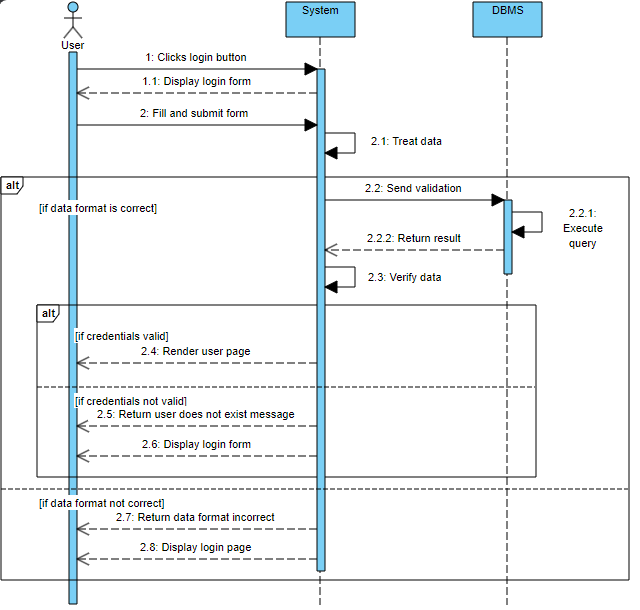


Figure 4 : Sequence diagram formalism

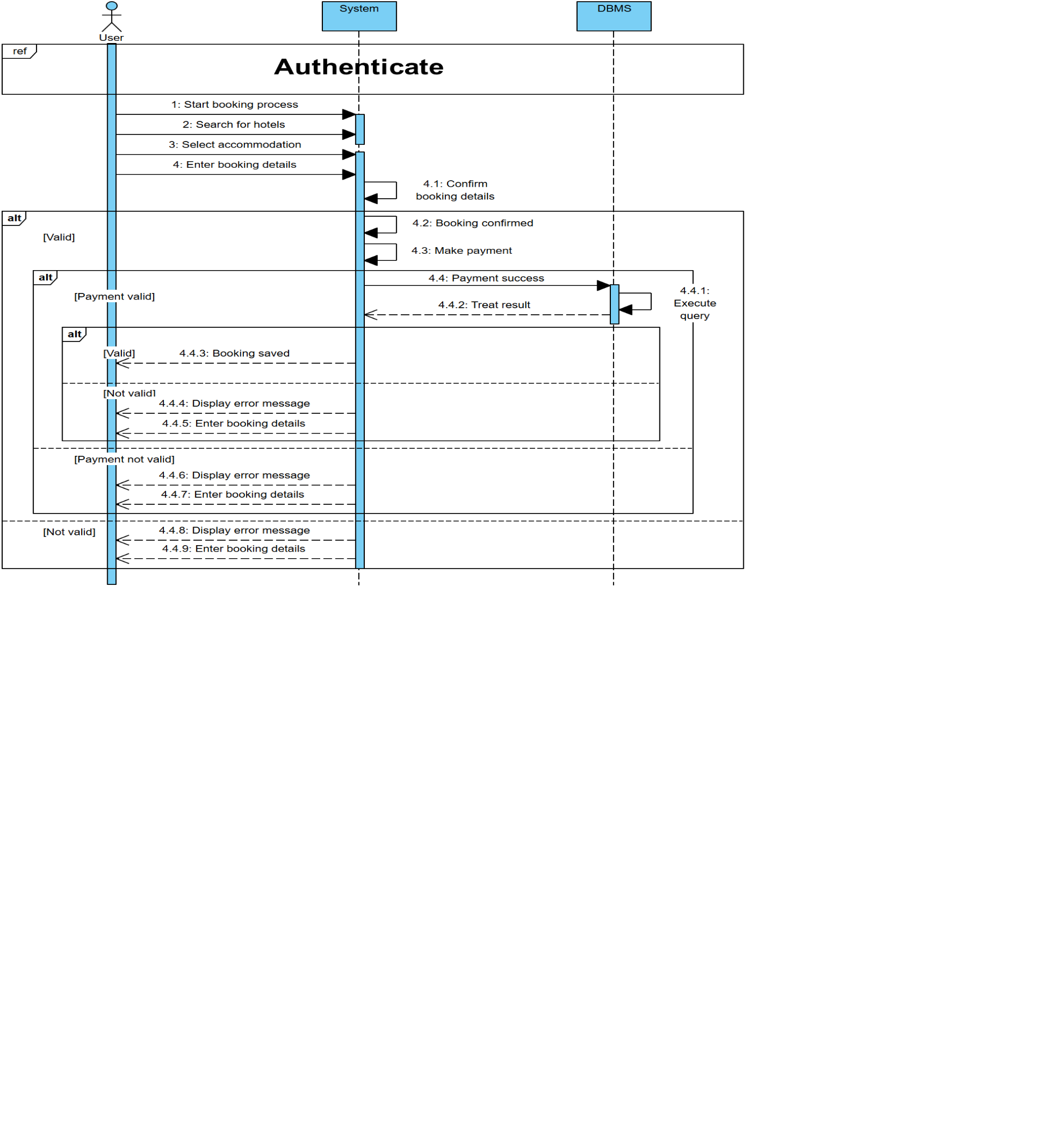
Table 7 : Sequence diagram components

|  |  |  |
| --- | --- | --- |
| Elements | Representation | Description |
| Lifeline |  | Represents the objects or components participating in the interaction. |
| Synchronous message |  | It is a message that sends wait response before it continue the process. |
| Asynchronous message |  | It does not need the response of the receiver before the sender continues. |
| Return message |  | It represents a response of a message. |
| Combined fragment |  | It represents a choice of behaviour in which at most one operand will be chosen. |
| Activation |  | It describes the time period in which an operation is performed by an element. |

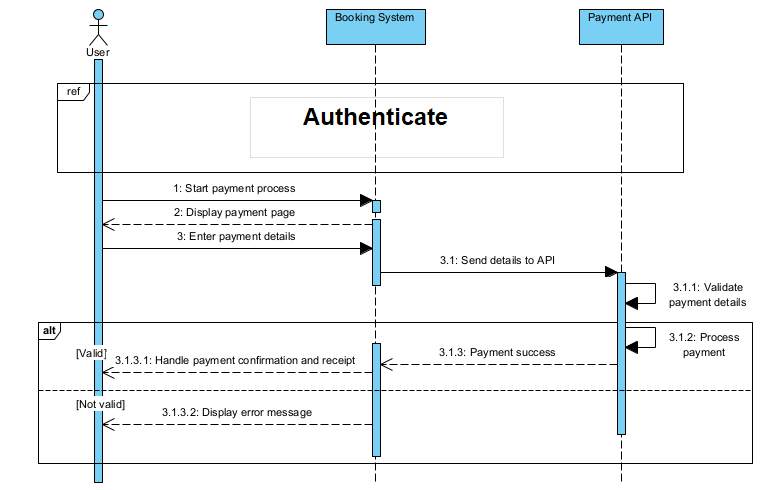
#### Authenticate sequence diagram



#### Book accommodation sequence diagram



#### Make payment sequence diagram



### Activity diagram

#### Definition

An activity diagram is a diagram that represents the flow of activities or actions within a system or process. It illustrates the behaviour of a system by depicting the sequential and parallel activities performed by objects, components or system actors.

#### Formalism

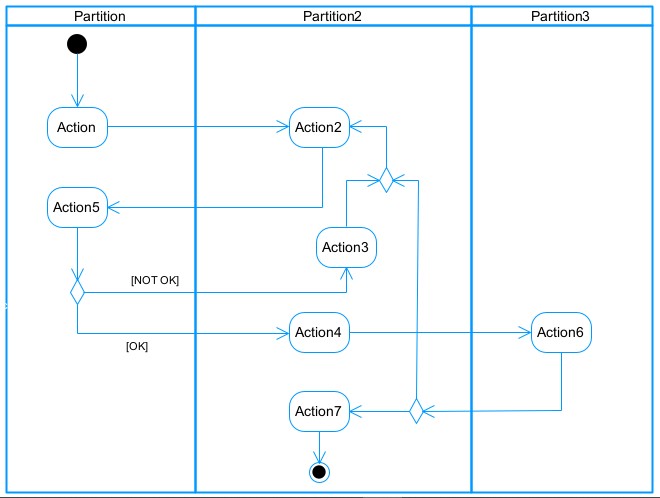
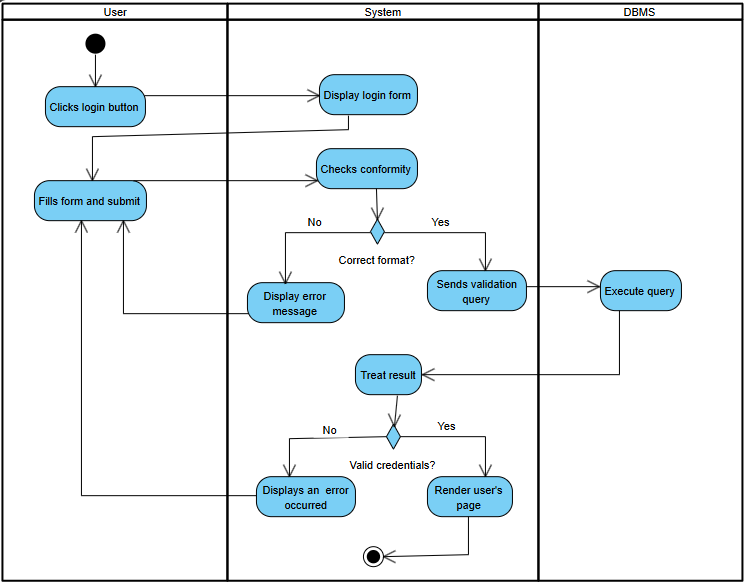


Figure 5 : Activity diagram formalism

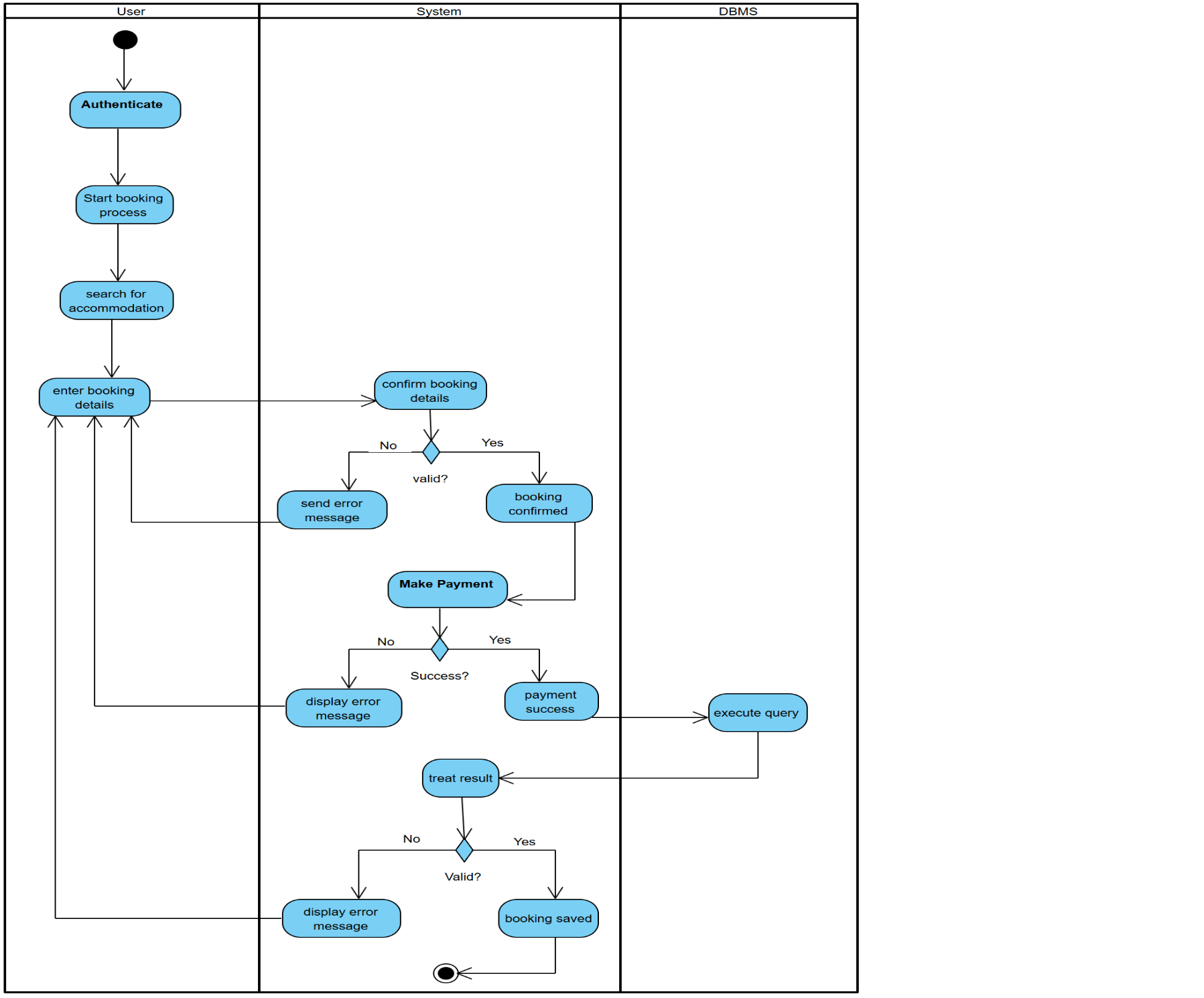
Table 8 : Activity diagram components

|  |  |  |
| --- | --- | --- |
| Elements Representation | | Description |
| Activity |  | Represents the actions or tasks performed within the system. |
| Action |  | Represents a task to be performed. |
| Control flow |  | Control flows show the flow of control from one activity to the next. |
| Initial node |  | Represents the beginning of a process or workflow in an activity. |
| Final node |  | It marks the end state of an activity and represents the completion of all flows of a process. |
| Decision node |  | Represents a decision and always has at least two paths branching out with condition text to allow users to view options. |
| Merge node |  | Reunite different decision paths created using a decision node. |
| Fork node |  | Slits behaviour into parallel or concurrent flows of activities  (or actions) |
| Join node |  | Unites a set of parallel or concurrent flows of activities or actions. | |
| Swimlane and partition |  | A way of grouping activities performed by the same actor in an activity diagram or to group actions in the same thread. | |

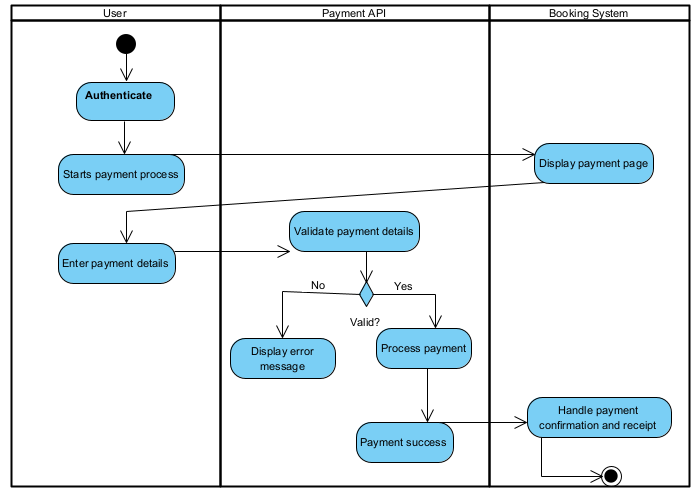
#### Authenticate activity diagram



#### Book accommodation activity diagram



#### Make payment activity diagram



## CONCLUSION

Having reached the end of our analysis, we were asked to differentiate between UML and MERISE and compare the unified processes and finally the use case, communication, sequence and activity diagrams of our analysis phase. We can say that this analysis file has allowed us to well delimit the functional needs of the 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 FOUR : CONCEPTION PHASE

Preamble

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

CONTENT OVERVIEW

INTRODUCTION

1. CLASS DIAGRAM
2. STATE MACHINE DIAGRAM
3. PACKAGE DIAGRAM

CONCLUSION

## INTRODUCTION

The design of a software requires a study, a good analysis of the system to be set up. A well-designed system is easy to understand, realize and maintain. The conception phase is therefore a crucial step in the software development process, as the implementation of the application depends a lot on the design. After an analysis of the system to be set up we move on to the design.

## Class diagram

#### Definition

A class diagram is a diagram the represents the structure of a system by illustrating the classes, their attributes, methods, relationships and associations. It provides a visual representation of a static aspects of the system, and can be essential for software design, development and documentation as they provide a layout for implementing and organising classes and their relationships within a system.

#### Formalism

multiplicity

private (accessible

only within the class

scope)

public

(

accessible out

of the class

scope)

0..1

0..\*



class name

-

Attribute

:

int



Classe\_2

Figure 6 : Class diagram formalism

Table 9 : Class diagram components

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Class |  | A class is an abstract description of a set of objects in the application domain. |
| Composition |  | A composition relationship shows that a class can not exist without the class that it is composed of. |
| Aggregation |  | An aggregation relationship shows that a class is part of another class. |
| Dependency |  | It represents a relationship where one class depends on or uses another class. |
| Inheritance/Generalization |  | It is a relationship where one class inherits the properties and behaviors of another class. |
| Association |  | It represents the relationship between two or more classes. |

#### System class diagram

## II. State machine diagram

#### a) Definition

A state machine diagram is use to model the dynamic nature of a system. It defines the different states of an object during its lifetime and the states are changed by the events. State machine diagram also help visualise the possible states and transitions within a system or object, making it easier to understand and analyse the system’s behavior.

#### b) Formalism

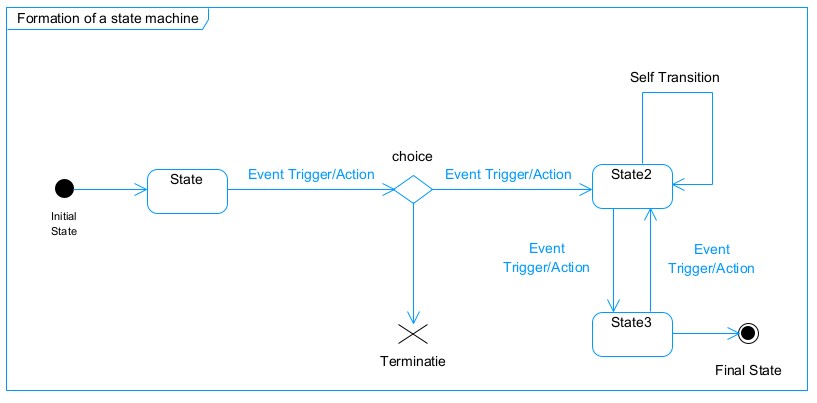


Figure 7 : State machine diagram formalism

Table 10 : State machine diagram components

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| State |  | It depicts the conditions or circumstances of a particular object of a class at a specific point of time. |
| Initial State |  | It defines the beginning of a system. |
| Final State |  | It represents the end of a system. |
| Transition |  | It defines the events or triggers that cause an object or system to move from one state to another. |
| Decision box |  | A diamond symbol that indicates a dynamic condition with branched potential results |
| Terminate |  | Implies that the execution of a state by means of its context is terminated. |
| Diagram Overview |  | A placeholder for the linked states in a state machine diagram. |

#### c) Account state machine diagram

## III. Package diagram

#### a) Definition

A package diagram is type of structural diagram in UML that represents the organisation and dependencies of different packages within a system or software application. Package diagrams are useful for visualising the overall structure and organisation of a system, as well as understanding the dependencies and relationships between the different components.

#### b) Formalism

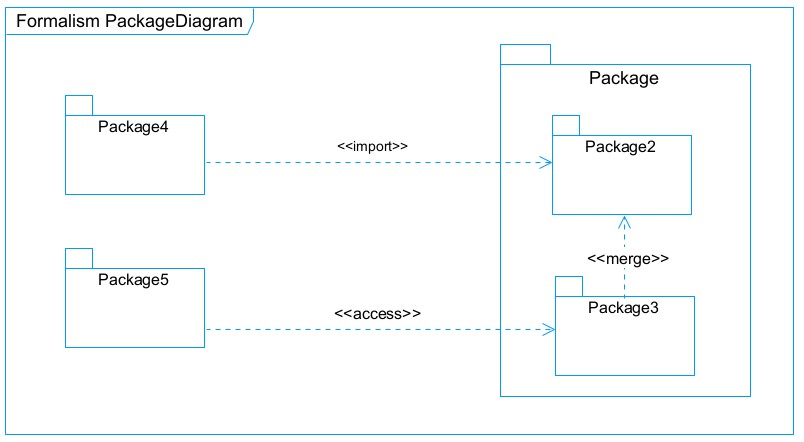
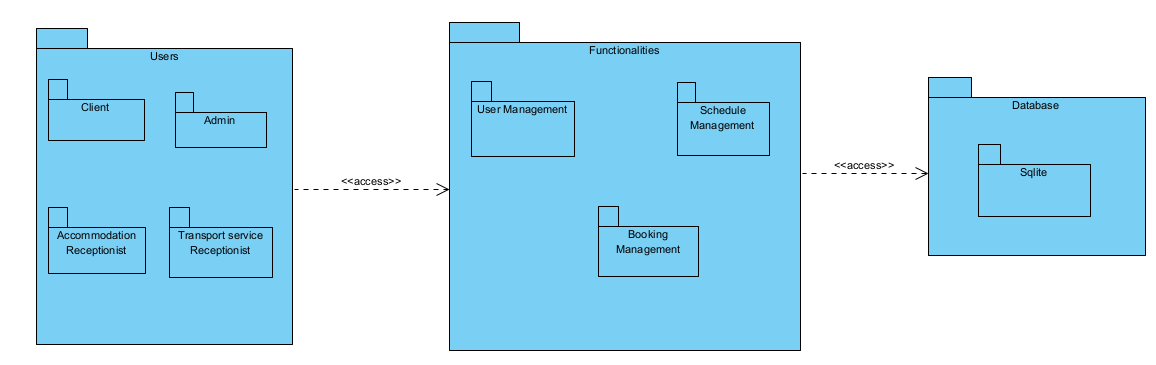


Figure 8 : Package diagram formalism

Table 11 : Package diagram components

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Package |  | A package is a namespace use to group together logically related elements within a system. |
| Import dependency |  | A relationship Indicate that, functionality has been imported from one package to another. |
| Access dependency |  | A relationship Indicates that one package requires assistance from the function of another package. |
| Merge dependency |  | It is a relationship which shows that, the functionality of two packages are combines to a single function. |

#### System package diagram



## CONCLUSION

In the conception phase, we set as objective to plane the different aspect of our system by showing how it will be structure and deployed within existing technical architectures. We began by 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 to 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 FIVE : REALISATION PHASE

Preamble

This part of the project constitutes the realization and implementation of the application studied in the conception phase.

CONTENT OVERVIEW

INTRODUCTION

1. CHOICE OF TECHNOLOGY
2. ARCHITECTURE OF OUR APPLICATION
3. DEPLOYMENT DIAGRAM
4. COMPONENT DIAGRAM

CONCLUSION

## INTRODUCTION

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

## Choice of technology

To set up our platform and according to the needs of the end-users, the programming languages chosen are HTML5, CSS3 for the front-end, Django for the back-end and PostgreSQL for interaction with the database. Each of these languages will be presented in detail in the following paragraphs.

### HTML

HTML (HyperText Markup Language): first introduced in 1991 when the web was launched. Its role is to manage and organize the content of web pages. Its importance lies in creating content for an application. It's worth noting that we've used version 5 of HTML to develop our application, as it's better suited to our requirements.



Figure 9 : HTML logo

### CSS

CSS or Cascading Style Sheet: the role of CSS is to manage the appearance of the web page (layout, positioning, decoration, color, text size...). This language complements HTML to define the style of our platform.



Figure 10 : CSS logo

### SQLite

SQLite is a lightweight, serverless, and self-contained database management system. It is widely used for applications where simplicity, reliability, and low resource requirements are important. Unlike traditional databases like MySQL or PostgreSQL, SQLite doesn’t require a separate server process, making it an ideal choice for embedded systems, mobile apps, small web applications, or testing environments.



Figure 11 : PostgreSQL logo

### Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It was created to simplify the process of building complex, database-driven websites and web applications. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of "don't repeat yourself" (DRY). It includes built-in features for handling common web development tasks such as authentication, content management, and form processing, allowing developers to focus more on the unique aspects of their project.

Figure 12 : Django logo

### Visual Paradigm

Visual Paradigm is Unified Modeling Language (UML) modeling software. Visual Paradigm is a popular tool used to create UML diagrams, including sequence and communication diagrams, which are commonly used for modeling software systems and representing interactions between objects or actors in a system.



Figure 13 : Visual Paradigm logo

### Gantt Project

Project Gant is a widely used project management tool for planning and tracking project tasks over a given period. A Gantt chart is particularly useful for visualizing dependencies between tasks, start and finish dates, and project progress.

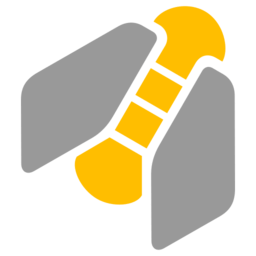


Figure 14 : Gantt project logo

### Postman

Postman is a popular API development tool that facilitates the creation, testing and documentation of APIs. It is widely used by developers, testers and API development teams to simplify the process of working with HTTP APIs.

Une image contenant Graphique, cercle, logo, conception

Description générée automatiquement

Figure 15 : Postman logo

## Architecture of our application

### Logical Architecture

The application logic of our project is based on the MVT design pattern which is a very practical way to organize its different classes in an application development.

The MVT (Model-View-Template) architecture is a design pattern used in Django, a high-level Python web framework designed for rapid development and clean, pragmatic design. MVT is a variation of the MVC (Model-View-Controller) pattern, adapted to fit the needs of web development in Django. This architecture promotes the separation of concerns, making applications more modular, maintainable, and scalable.

1. **Model**:

The Model is responsible for managing the data of the application. It defines the structure of the data, including the database schema and the relationships between different pieces of data. In Django, models are defined using Python classes, and Django's ORM (Object-Relational Mapping) handles the interaction with the database.

**Key Responsibilities**:

* + - Define data structure
    - Handle database operations (CRUD: Create, Read, Update, Delete)
    - Enforce data validation rules

1. **View:**

The View contains the business logic of the application. It processes user requests, interacts with the model to retrieve or update data, and returns a response. In Django, views can be implemented as functions or classes (class-based views).

**Key Responsibilities**:

* + - Handle HTTP requests
    - Interact with the model
    - Prepare data for the template
    - Return HTTP responses

1. **Template**:

The Template is responsible for the presentation layer of the application. It defines how the data from the view is presented to the user. Django templates are HTML files that can include template tags and filters to dynamically generate the final HTML content.

**Key Responsibilities**:

* + - Define the user interface
    - Render data into HTML
    - Provide a dynamic and interactive user experience

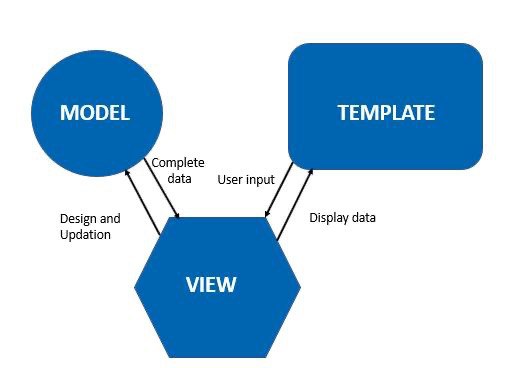


Figure 16 : The MVT design pattern

### Physical Architecture

We implement a three-tier architecture system.

The three-tier architecture is a client-server software architecture which consists of the user interface (presentation tier), business rule (application tier) and data access/storage tier (data tier).

1. **Presentation tier:**

It is the user interface and communication layer of the application, where the end user interacts with the application. Its main purpose is to display information to and collect information from the user.

1. **Application tier:**

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed – sometimes against other information in the data tier – using business logic, a specific set of business rules. The application tier can also add, delete or modify data in the data tier.

1. **Data tier:**

The data tier, sometimes called database tier, data access tier or backend, is where the information processed by the application is stored and managed. This can be a **relational database management system** such as **PostgreSQL**, **MySQL**, **MariaDB**, **Oracle**, **DB2**, **Informix or Microsoft SQL Server**, or in a **NoSQL** Database server such as **Cassandra**, **CouchDB** or **MongoDB**.

In a three-tier application, all communication goes through the application tier. The presentation tier and the data tier cannot communicate directly with one another.

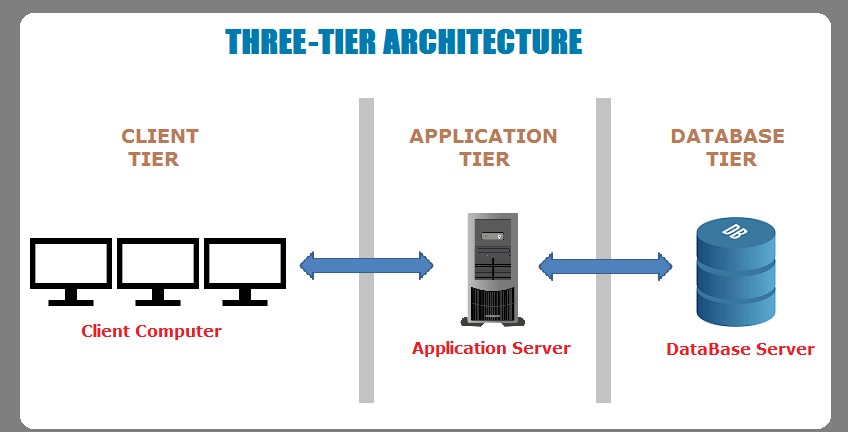


Figure 17 : The three-tier application representation

## Deployment diagram

### Definition

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

### Formalism

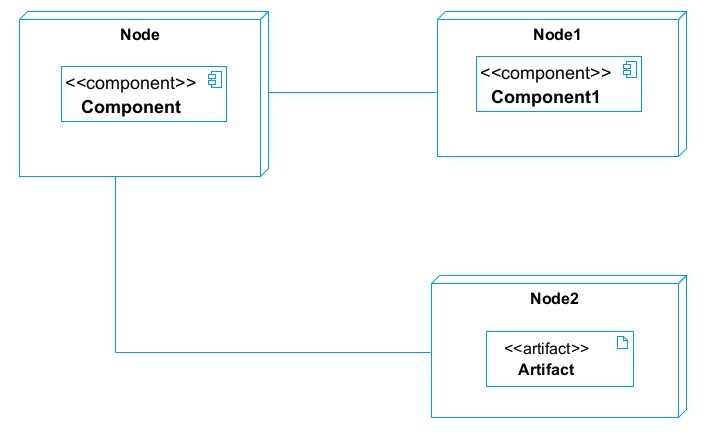
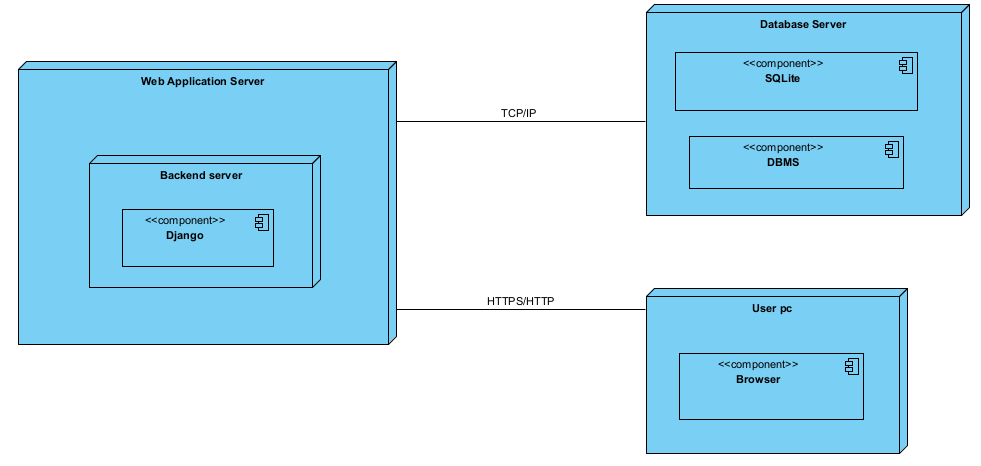


Figure 18 : Deployment diagram formalism

Table 12 : Deployment diagram components

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Node |  | It represents the physical hardware devices or computing resources on which the software components are deployed. |
| Artifact |  | It represents the physical files or data that are used or produced by components. |
| Component |  | It represents the software entity of the system. |
| Association |  | An association helps to connect two nodes together which permits them to communicate  together |

### System deployment diagram



## Component diagram

### Definition

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

### Formalism

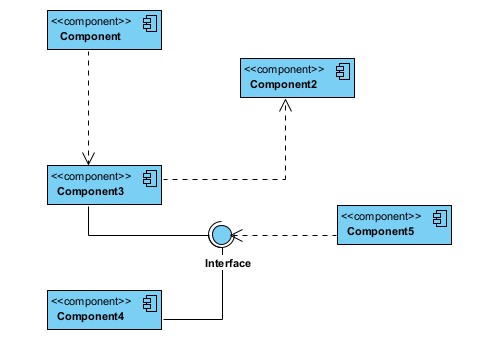
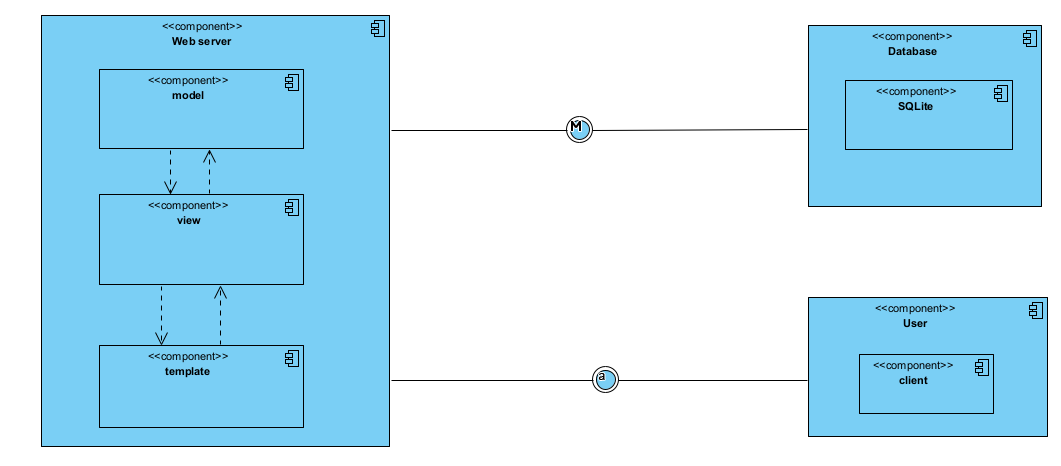


Figure 19 : Component diagram formalism

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Component |  | It represents the physical or logical units of a system that encapsulate functionality and provide interfaces to other components. |
| Dependency |  | It represents a relationship between two components where one component depends on or uses the functionalities of another component. |
| Interface |  | It represents the specifications that define how components interact with each other. |
| Port |  | Ports are connection points on a component that define the communication channels through which the components interact with the other components. |

Table 13 : Component diagram components

### System component diagram



## CONCLUSION

In this chapter dedicated to implementation, we have moved from theory to practice, which has enabled us to achieve our goals described in the previous steps. To use our application, it is necessary to know its functionalities. As a result, the last part of our project is devoted to its drafting.