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

REPUBLIC OF CAMEROON
Peace-Work-Fatherland

LOCALHOST DIGITAL
MINEFOP Approval
No:000623/MONEFOP/SDGSF/CSACD/CBAC
Tel : (+237) 696 984 233/ 674 647 010
E-mail: contact@localhost-academy.com

INTERNSHIP REPORT

THEME:

CONCEPTION AND REALIZATION OF A TRAVEL RESERVATION AND TRAVEL PLANNING SYSTEM

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
Full Stack Developer

ACADEMIC YEAR 2023-2024

DEDICATION

I DEDICATE THIS DOCUMENT TO MY ENTIRE FAMILY

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:

- To the Almighty GOD he who made all this.
- **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 Digital, **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 Digital** 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;
- **Mr. NDANDA JEAN MARIE** my grand-father for his support during these three years.
- **Mr. KAMGANG GILDAS** my uncle and his wife for their support during these three years.
- My parents **Mr/Mrs MENKAHE** for the support and motivations.

- 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

The issue of development through Information and Communication Technologies (ICT) continues to fuel controversy. In today's information-intensive world, ICTs are a decisive factor in Africa's development, and even a prerequisite for it. For this reason, most companies are embracing this trend to increase their business activity and ensure the sustainability of their various services. Our internship period, which was designed to help us make our contribution to this major development, took us to LOCALHOST DIGITAL, a local company specializing in innovative IT services, which enabled us to carry out a number of tasks, including the CONCEPTION AND REALIZATION OF A TRAVEL RESERVATION AND TRAVEL PLANNING SYSTEM, to simplify the planning and booking of trips for travelers while enhancing operational efficiency for travel agencies using just a Smartphone or any device capable of connecting to the Internet. Thanks to the knowledge acquired at IAI Cameroon, we were able to set up the said Application. The UML modeling language combined with the 2TUP method were used to model our diagrams, enabling us to gain a better understanding of the system under study. This report describes the various stages from conception to realization of the application.

Key words: Booking, travel, planning

RESUME

La question du développement à travers les Technologies de l'Information et de la Communication (TIC) continue de susciter des débats. Dans le monde actuel où l'information occupe une place prépondérante, les TIC constituent un facteur décisif pour le développement de l'Afrique, et même une condition préalable. Pour cette raison, la plupart des entreprises adoptent cette tendance afin d'accroître leurs activités commerciales et d'assurer la durabilité de leurs différents services. Notre période de stage, qui visait à nous permettre d'apporter notre contribution à ce grand développement, nous a conduits à LOCALHOST DIGITAL, une entreprise locale spécialisée dans les services informatiques innovants, ce qui nous a permis d'accomplir un certain nombre de tâches, notamment la CONCEPTION ET LA RÉALISATION D'UN SYSTÈME DE RÉSERVATION DE VOYAGES ET DE PLANIFICATION DE VOYAGES, afin de simplifier la planification et la réservation de voyages pour les voyageurs tout en améliorant l'efficacité opérationnelle des agences de voyages à l'aide d'un smartphone ou de tout autre appareil capable de se connecter à Internet. Grâce aux connaissances acquises à l'IAI Cameroun, nous avons pu mettre en place ladite Application. Le langage de modélisation UML combiné à la méthode 2TUP a été utilisé pour modéliser nos diagrammes, ce qui nous a permis d'avoir une meilleure compréhension du système étudié. Ce rapport décrit les différentes étapes de la conception à la réalisation de l'application.

Mots clés : Réservation, voyage, planification

GENERAL INTRODUCTION

The travel management system is an innovative solution designed to address the complexities of trip planning and booking in today's fast-paced travel environment. This comprehensive web-based platform serves as a centralized hub for travelers, enabling them to effortlessly plan, book, and manage their entire travel experience from a single interface. With functionalities that encompass travel plan creation and activity scheduling, the system not only simplifies the user experience but also enhances operational efficiency for travel agencies. By integrating secure payment processing and automated notifications, the system ensures real-time updates, ultimately aiming to provide a seamless and enjoyable travel experience while optimizing agency operations.

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

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 DIGITAL. The goal been to convert our theoretical knowledge into practice that can help to provide a solution to a specific problem.

I. Welcome and integration

On Monday July 15th, 2024, we went to the premises of LOCALHOST DIGITAL where we were received by **Mr. ROSNY NGOUANET TSIDIE** (director of LOCALHOST DIGITAL). He presented himself and then drilled on the history of LOCALHOST DIGITAL 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 DIGITAL

A. History

LOCALHOST DIGITAL 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. Missions

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 DIGITAL 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 DIGITAL

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

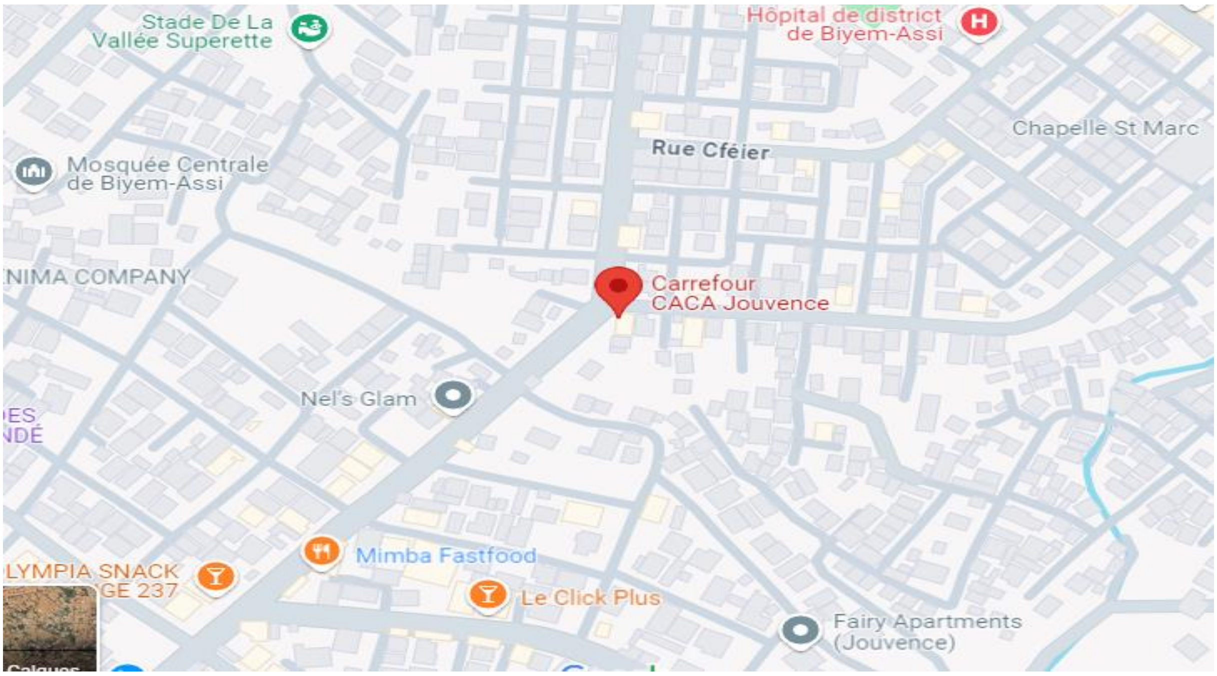

Stade De La Vallée Superette
Hôpital de district de Biyem-Assi
Chapelle St Marc
Rue Cféier
Mosquée Centrale de Biyem-Assi
NIMA COMPANY
Carrefour CACA Jouvence
Nel's Glam
Mimba Fastfood
LYMPIA SNACK GE 237
Le Click Plus
Fairy Apartments (Jouvence)

Figure 1 : Geolocation of LOCALHOST DIGITAL

F. Company’s identification form

Table 1 : Company’s identification form

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

III. Organisation of LOCALHOST DIGITAL

A. Administrative organisation

LOCALHOST DIGITAL is headed by the Chief Executive Officer (CEO) who manages the day-to-day activities and the implementation of the company's policies. LOCALHOST DIGITAL 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 DIGITAL is organised as follows;

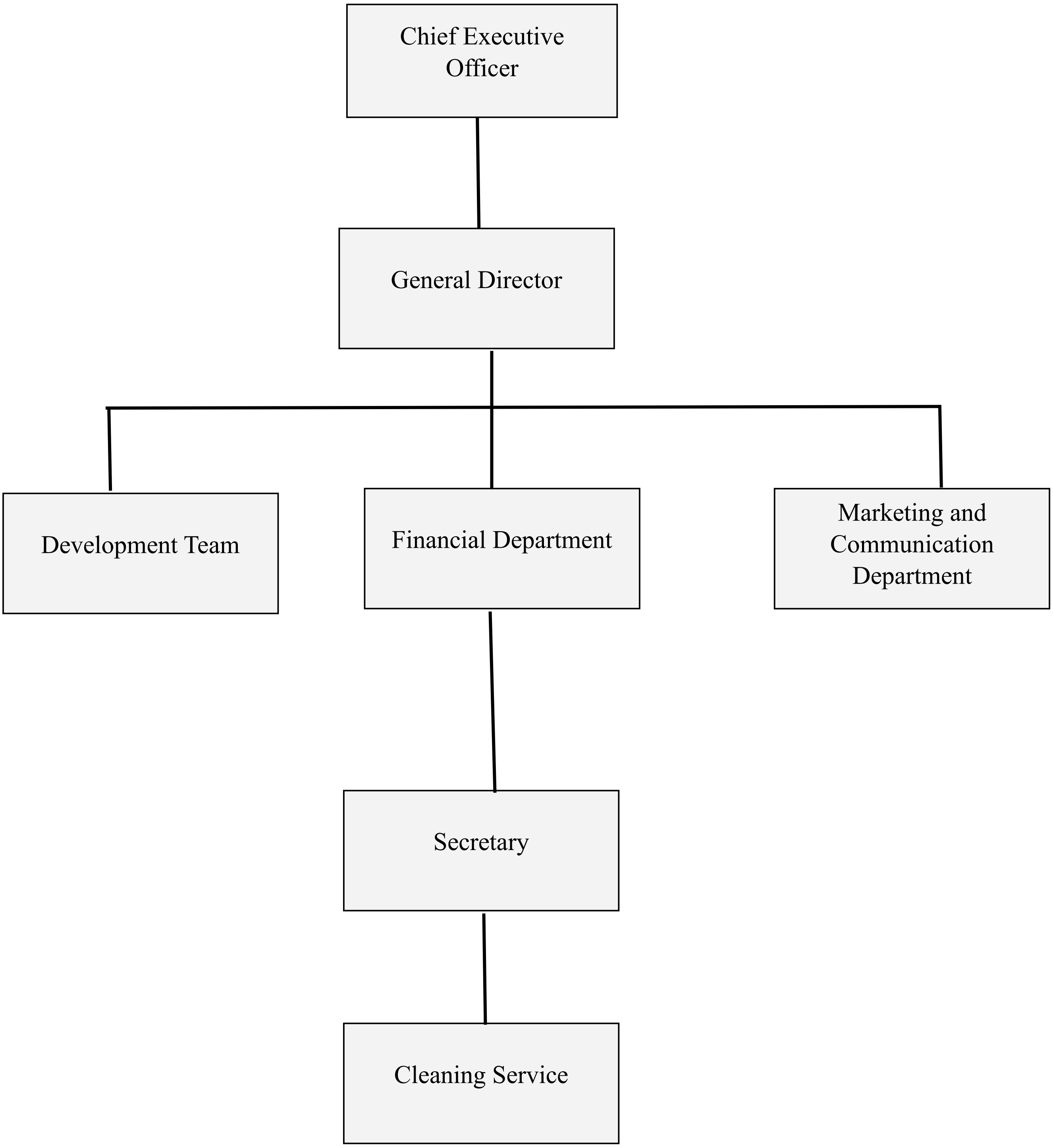


Figure 2 : Organisational chart of LOCALHOST DIGITAL

IV. Hardware and software resources of LOCALHOST DIGITAL

A. Hardware

Table 2 : Hardware components of LOCALHOST DIGITAL

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

B. Software

Table 3 : Software components of LOCALHOST DIGITAL

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 book agencies for any destination in Cameroon

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

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

I. PROJECT PRESENTATION
II. STUDY OF THE EXISTING SYSTEM
III. CRITICISM OF THE EXISTING SYSTEM
IV. PROBLEMATIC
V. 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.

I. 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 efficiently plan their trips and easily have their travel tickets. 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.

II. Study of the existing system

The existing travel planning systems, whether through traditional travel agencies, online platforms, 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. With these systems, travelers are forced to switch between different platforms or rely on inconsistent advice from friends, family, or online communities.

III. Criticism of the existing system

Limited Online Presence: While some agencies have basic websites, they often provide limited functionality for online reservations. Clients may still need to contact the agency directly to confirm bookings, making the process less convenient.

Manual Reservation and Booking: Most travel agencies use manual methods for booking, such as phone calls, emails, or in-person visits. This process is time-consuming and lacks real-time updates on availability.

Fragmented Information: Current travel planning systems are highly disjointed, requiring users to navigate multiple platforms to gather all necessary information. 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.

IV. Problematic

HOW CAN WE CREATE A FULLY INTEGRATED AND PERSONALIZED TRAVEL MANAGEMENT PLATFORM THAT ELIMINATES THE NEED FOR USERS TO SWITCH BETWEEN MULTIPLE APPLICATIONS AND WEBSITES FOR THIER BOOKINGS?

V.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

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.

I. Context and justification

A. Context

In today's travel landscape, users often face the challenge of juggling multiple platforms to organize a single trip. For instance, imagine a family planning a summer vacation to Douala. They might book thier agency 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.

B. Justification

A unified solution is necessary to ease 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.

II. Objectives of the project

A. General objective

The general objective of our travel management system is to streamline the travel planning and booking process by providing a centralized, user-friendly platform that integrates various services and functionalities.

B. 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.

III. Expression of needs

A. 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.

2. Agency Management:

- Permit agency receptionist to manage its assigned agencies.

3. Reservation Management:

- Enable clients to book and manage their travel tickets.

4. Personalized Search and Recommendations:

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

5. Schedule Management:

- Allow users to view, manage, and modify their schedule.

6. Real-Time Updates:

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

7. Payment Processing:

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

8. User Notifications and Alerts:

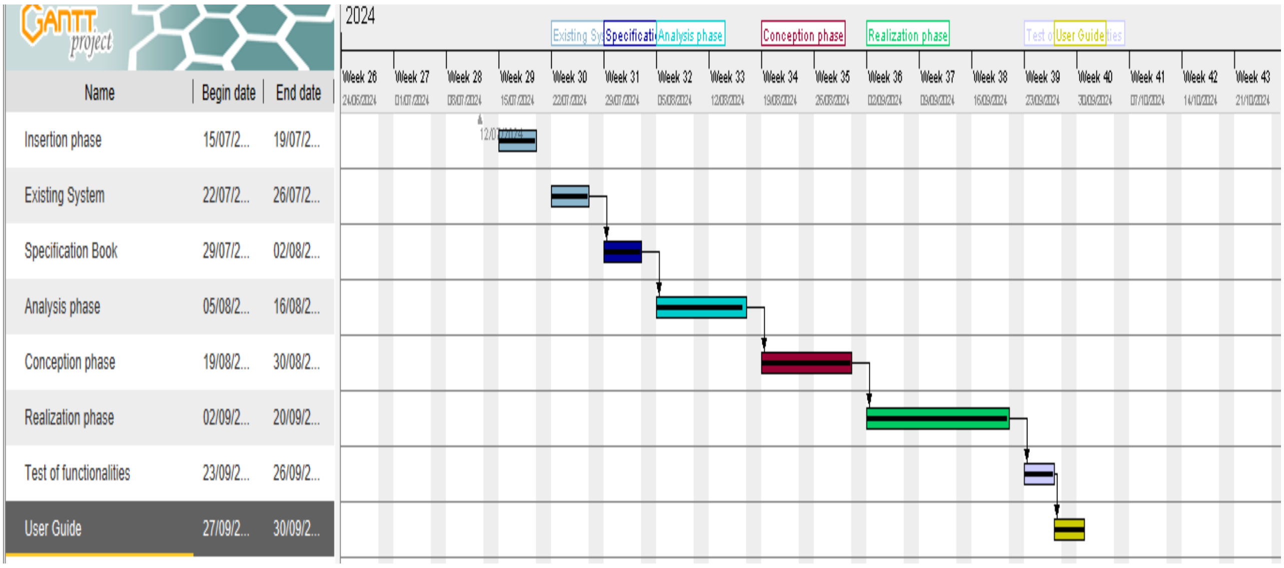
- Notify clients for updates and deletion of a travel plan.

B. 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.

IV. 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.



Name	Begin date	End date
Insertion phase	15/07/2...	19/07/2...
Existing System	22/07/2...	26/07/2...
Specification Book	29/07/2...	02/08/2...
Analysis phase	05/08/2...	16/08/2...
Conception phase	19/08/2...	30/08/2...
Realization phase	02/09/2...	20/09/2...
Test of functionalities	23/09/2...	26/09/2...
User Guide	27/09/2...	30/09/2...

Figure 3 : Project gantt chart

V.Estimated cost of the project

A. Human resources

Table 4 : Human resources of the project (source: Mercurial 2024)

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

B. Hardware resources

Table 5 : Hardware resources of the project (source: Mercurial 2024)

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

: Human resources of the project (source: Mercurial 2023)

C. Software resources

Table 6 : Software resources of the project (source: Mercurial 2024)

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

	contained database management system.		
Visual Paradigm	Modeling tool		Free
	TOTAL		70,701.340

D. Total project estimated cost

Table 7 : Total project estimation 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**

VI. 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.

VII. 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

I. METHODOLOGY

II. 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.

I. Methodology

A. Comparative study between UML and MERISE

a. 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:

b. UML:

UML is a general-purpose visual 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 8 : 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 9. 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.

B. Comparative study of the unified processes

a. 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.

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-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.

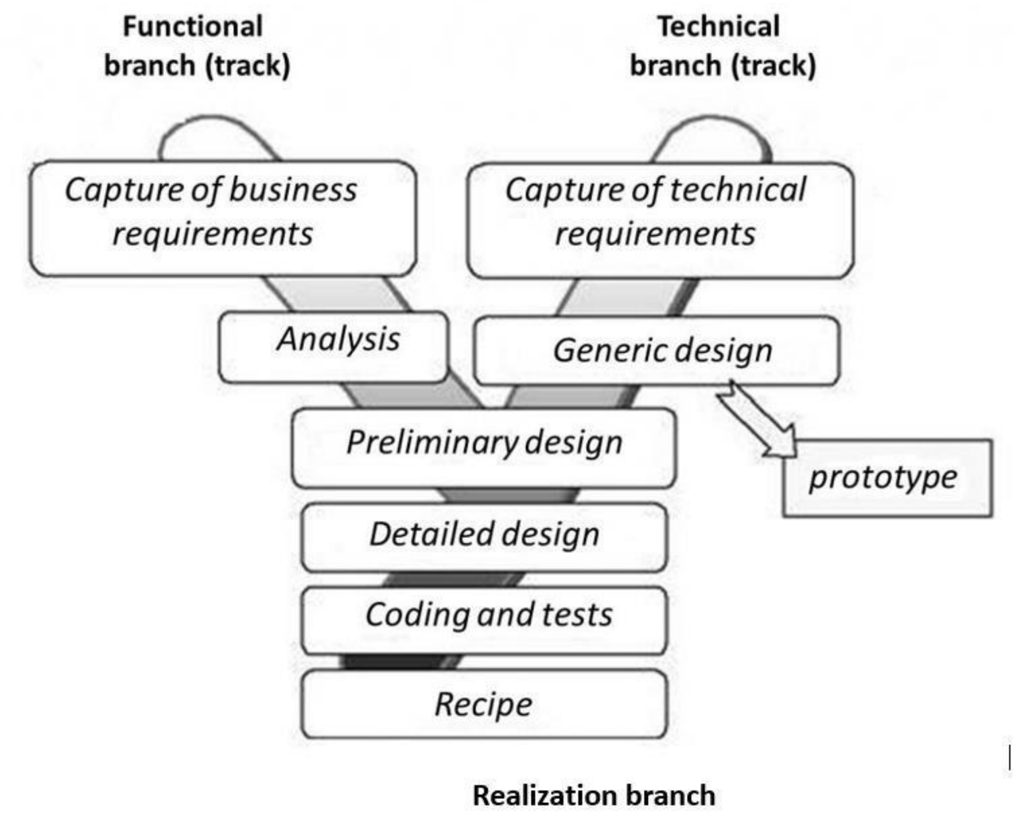


Figure 4 :2TUP diagram (Source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing)

i. 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.

ii. 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.

ii. 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.

C. 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.

II. Modeling

A. Use case diagram

a) 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.

b) Formalism

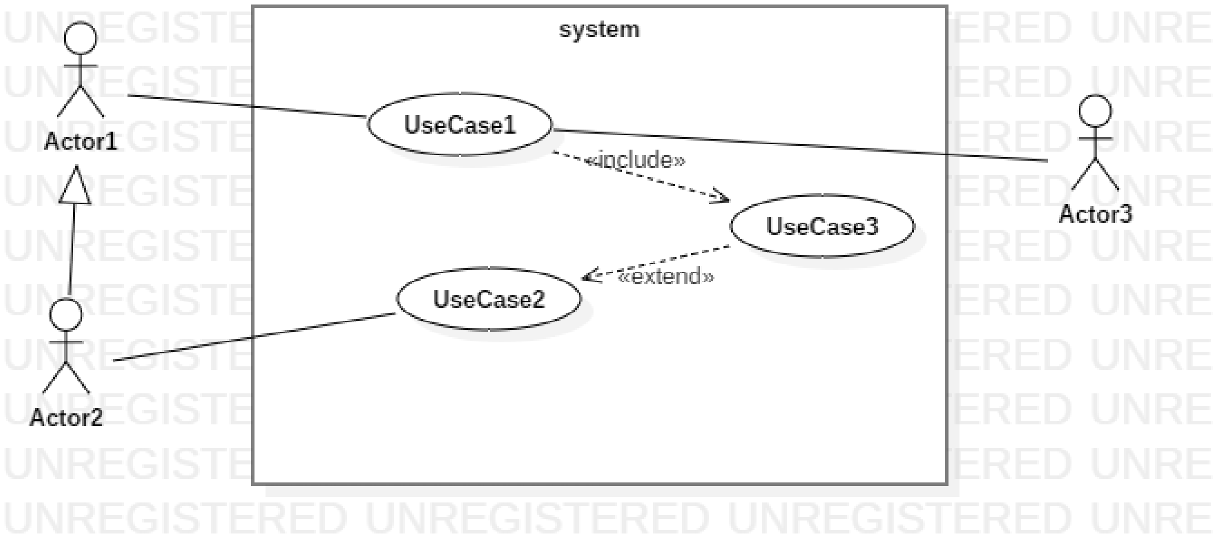
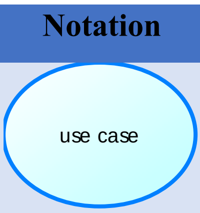
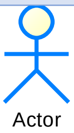
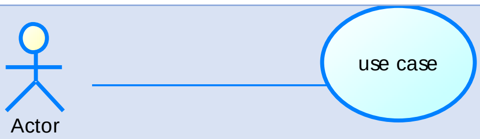
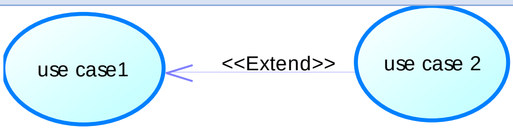
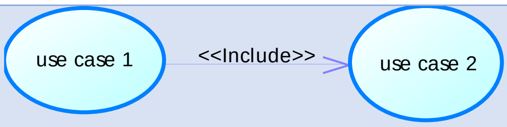
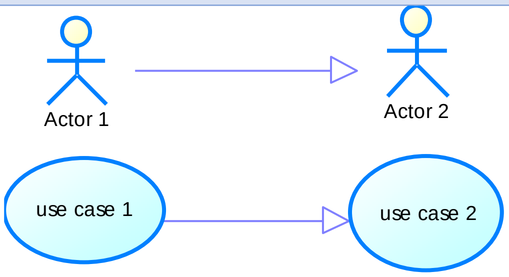
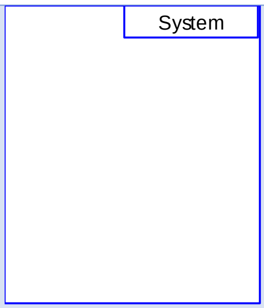


Figure 5 : Use case diagram formalism

Table 10 : 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.

c) Actors of the system

Table 11 : Actors of our system

Actor	Role
Administrator	The administrator is in charge of the general functioning of the system.
Agency receptionist	Agency providers.
Client	Any person willing to use the application.

d) General use case diagram

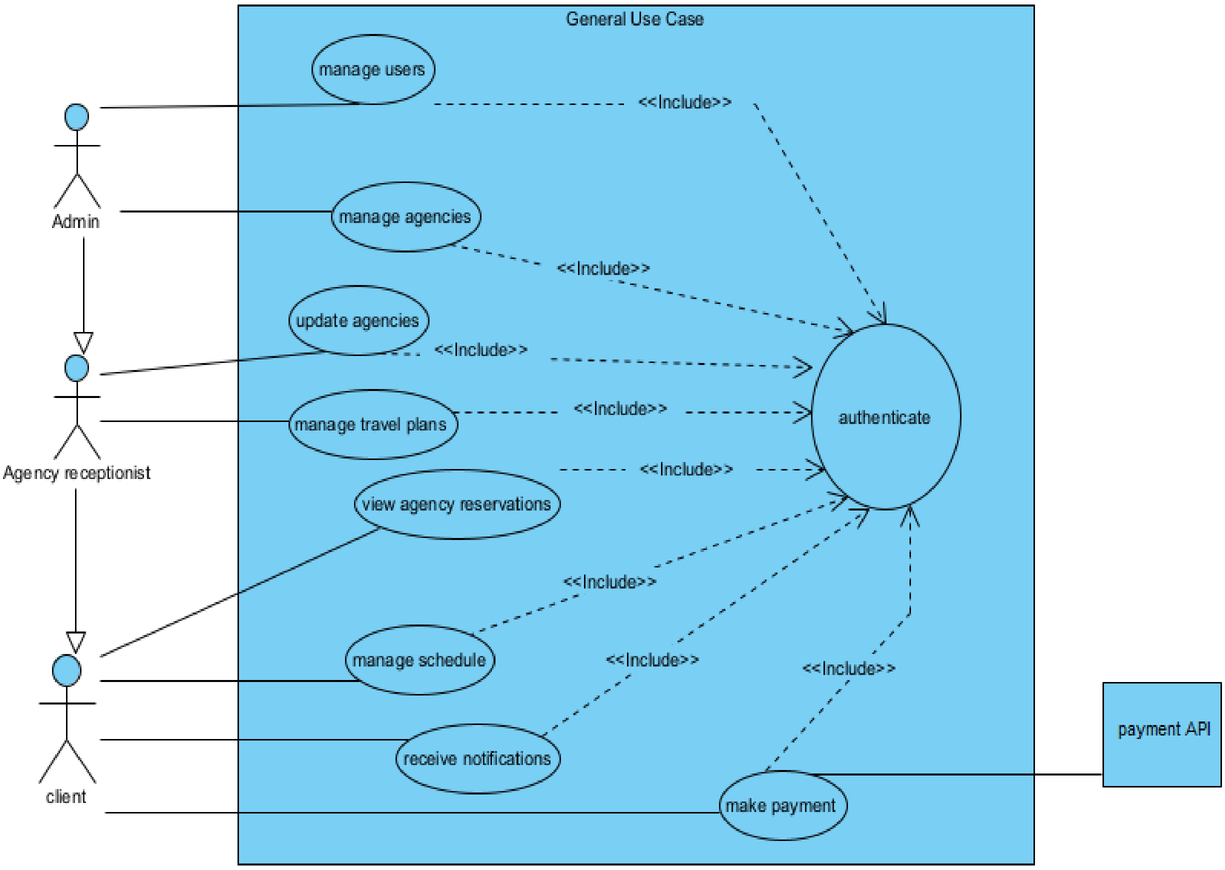


Figure 6 : General Use case diagram

e) Travel Plan Management use case diagram

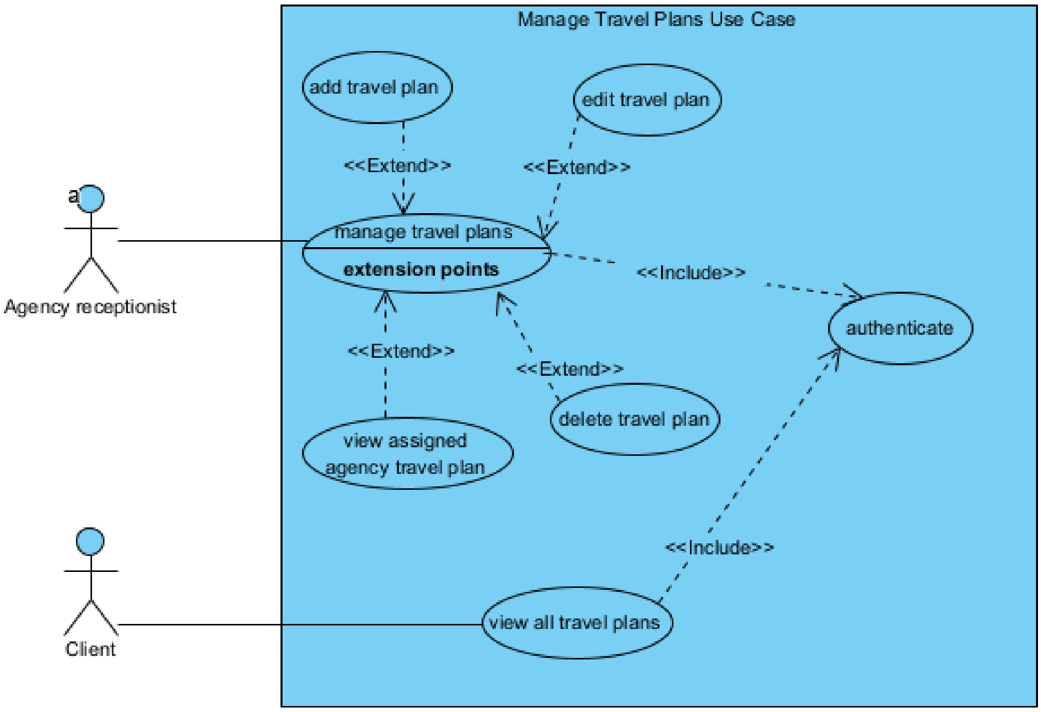


Figure 7 : Travel plan management use case diagram

f) Schedule Management use case diagram

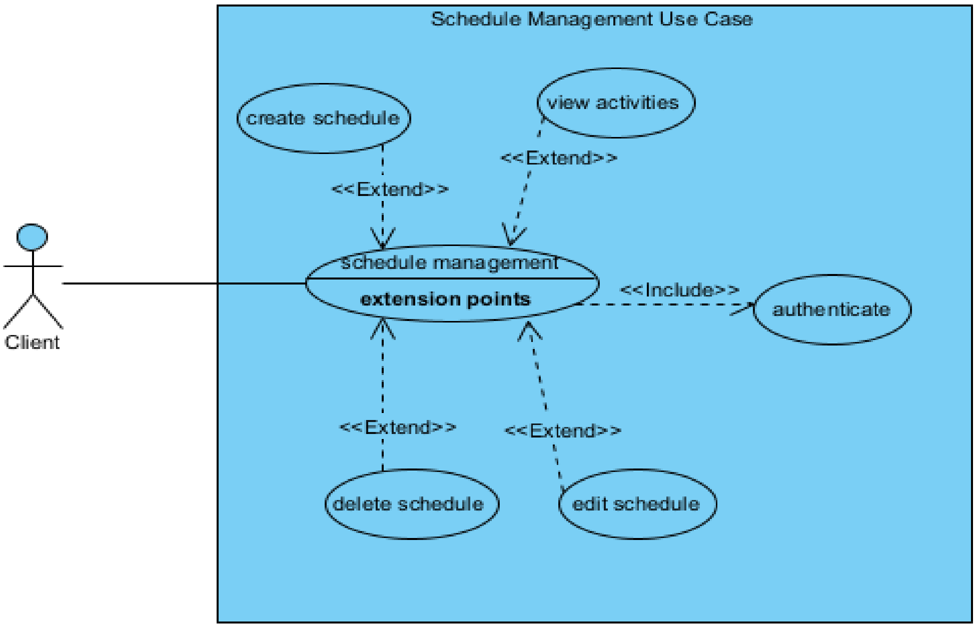


Figure 8 : Schedule management use case diagram

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.

Table 12 : Authenticate textual description

USE CASE AUTHENTICATE	
Summary	Permit users to access the system
Actors	Administrator, Agency Receptionist, Client
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	- 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. - 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 13 : Make reservation textual description

MAKE RESERVATION	
Summary	Permit clients to book a travel plan.
Actors	Client
Pre-condition(s)	1. The client is authenticated and logged into the system. 2. There are travel plans available for booking.
Trigger	The client clicks on the make reservation button
Nominal scenario	1. The client clicks on the make reservation button. 2. The system displays the reservation page. 3. The client fills the form. 4. The system checks the form conformity and the availability of the requested number of places. 5. The system sends the query to the database. 6. The database saves the reservation in the and reduces the number of available places for the travel plan. 7. The database sends a success message 8. The client receives a confirmation message.
Alternative scenario	- At step 4 of the nominal scenario, if there are insufficient number of places the system notifies the client that the requested number of places is unavailable.and he returns to step 3 of the nominal scenario.
Post-condition of success	The reservation is successfully created and stored.
Post condition of failure	The reservation is not created.

Table 14 : update travel plan textual description

Update Travel Plan	
Summary	Permit clients to book a travel plan.
Actors	Agency Receptionist
Pre-condition(s)	1. The receptionist is authenticated and assigned to the relevant agency.
Trigger	The agency receptionist clicks on the edit button
Nominal scenario	2. The receptionist clicks the edit button to update the travel plan. 3. The system allows the receptionist to modify the fields. 4. The system checks the form conformity. 5. The system sends the query to the database. 6. The database saves the changes of the travel plan. 7. The database sends a confirmation message.
Alternative scenario	- At step 4 of the nominal scenario, the system prompts the receptionist to correctly fill the form for invalid inputs
Post-condition of success	The travel plan is updated, and notifications are sent to clients.
Post condition of failure	The trave plan does not change.

B. Communication diagram

a) 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.

b) Formalism

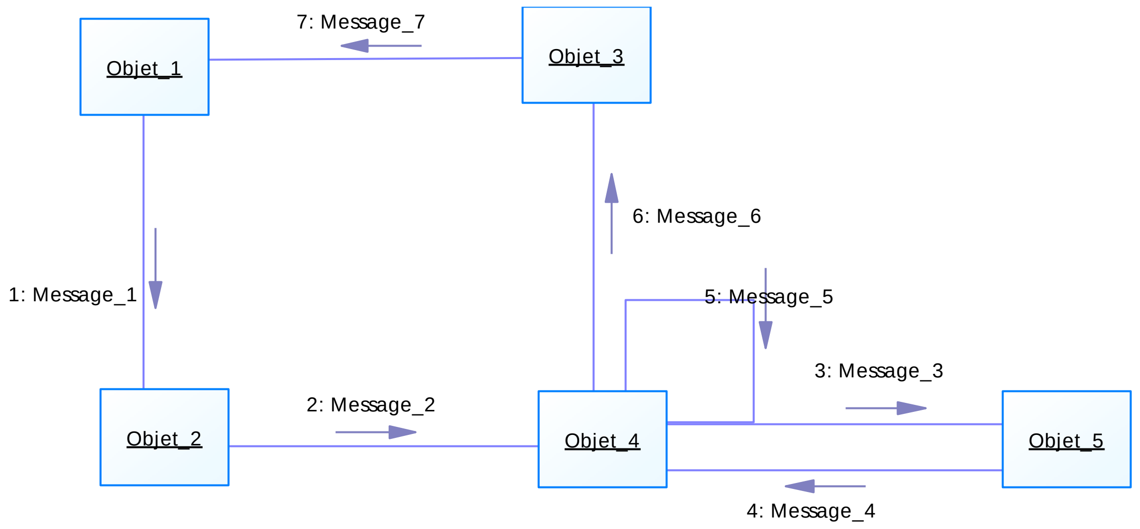


Figure 9 : Communication diagram formalism

Table 15 : Communication diagram components

Element	Representation	Description
Object	Objet_1	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.

c) Authenticate communication diagram

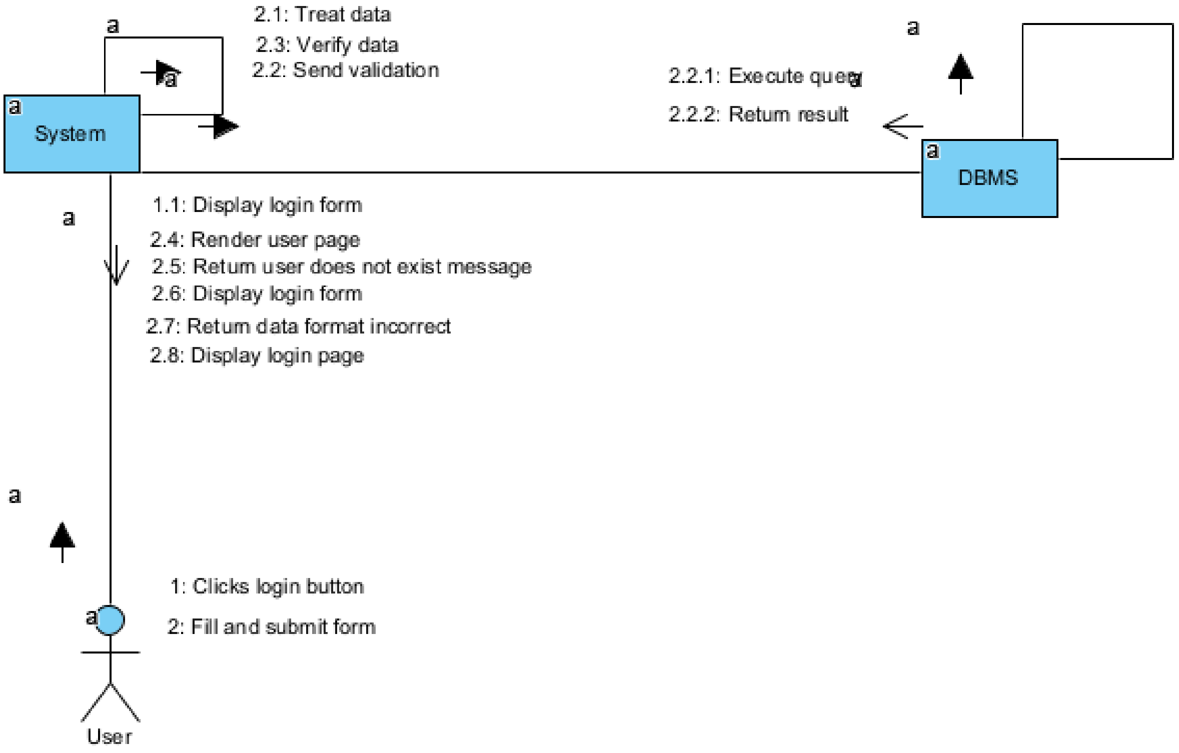


Figure 10 : Authenticate communication diagram

d) Make reservation communication diagram

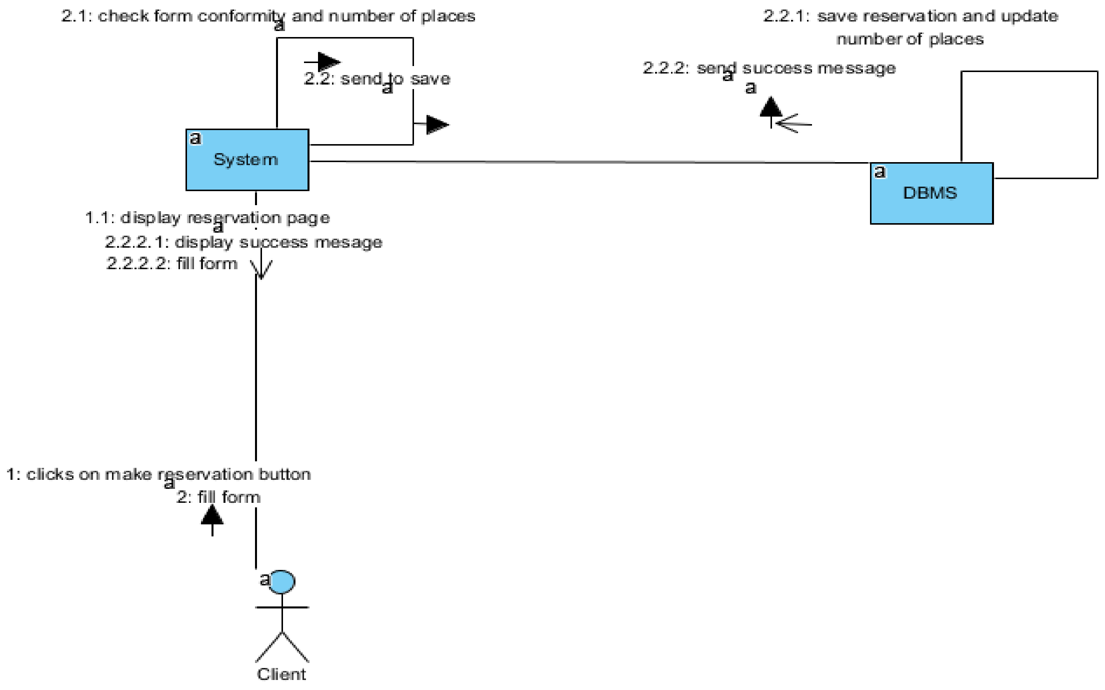

2.1: check form conformity and number of places
2.2: send to save
2.2.1: save reservation and update number of places
2.2.2: send success message
System
DBMS
1.1: display reservation page
2.2.2.1: display success mesage
2.2.2.2: fill form
1: clicks on make reservation button
2: fill form
Client

Figure 11 : Make reservation communication diagram

e) Update travel plan communication diagram

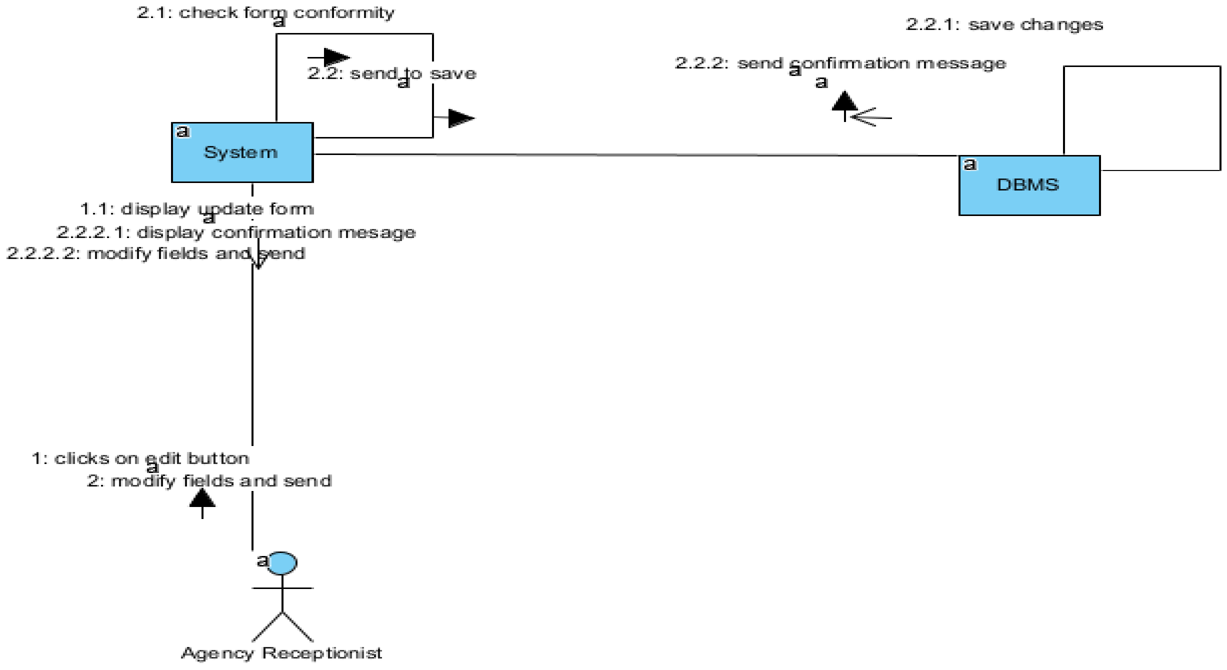

2.1: check form conformity
2.2: send to save
2.2.1: save changes
2.2.2: send confirmation message
System
DBMS
1.1: display update form
2.2.2.1: display confirmation mesage
2.2.2.2: modify fields and send
1: clicks on edit button
2: modify fields and send
Agency Receptionist

Figure 12 : Update travel plan communication diagram

C. Sequence diagram

a) 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.

b) Formalism

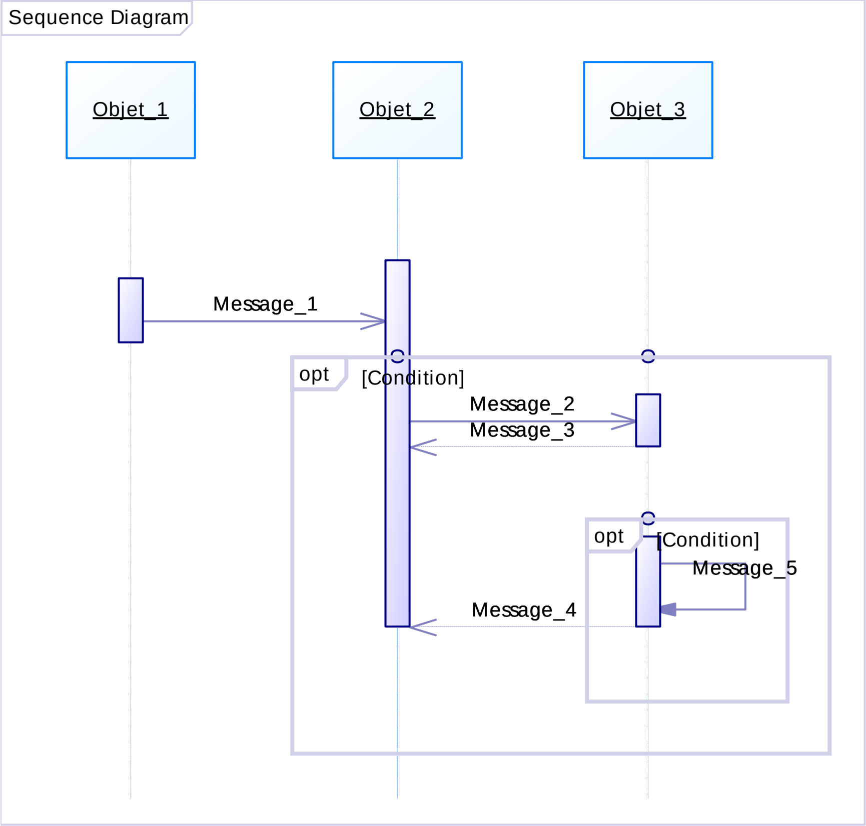
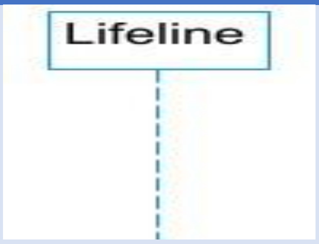
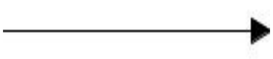
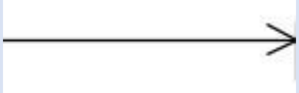
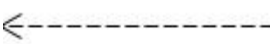
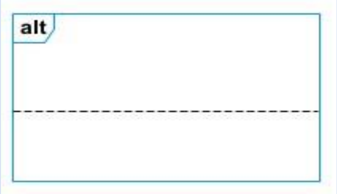
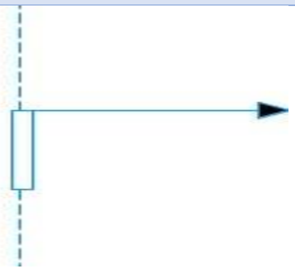


Figure 13 : Sequence diagram formalism

Table 16 : 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.

c) Authenticate sequence diagram

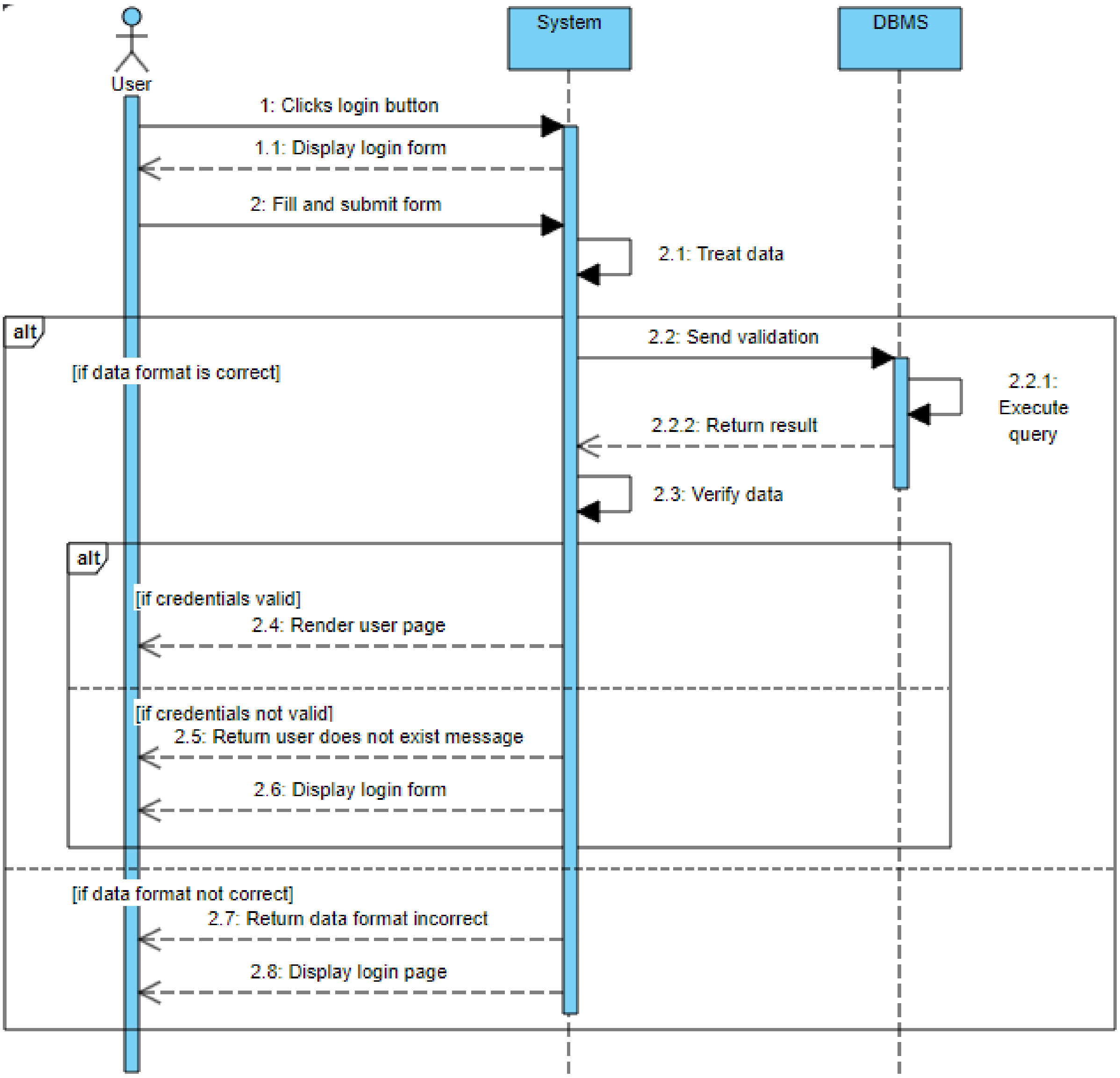


Figure 14Authenticate sequence diagram

d) Make reservation sequence diagram

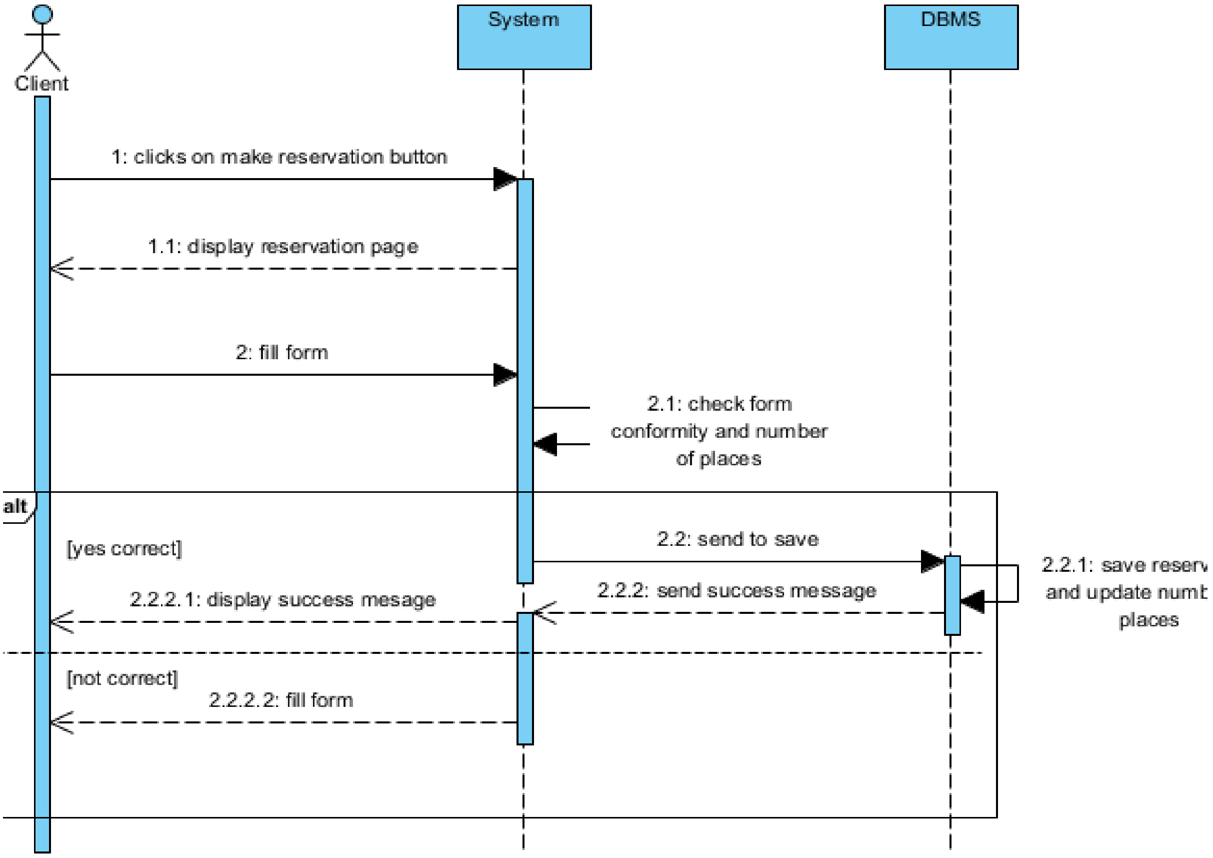

Client
System
DBMS
1: clicks on make reservation button
1.1: display reservation page
2: fill form
2.1: check form conformity and number of places
alt
[yes correct]
2.2: send to save
2.2.1: save reserv and update numb places
2.2.2: send success message
2.2.2.1: display success mesage
[not correct]
2.2.2.2: fill form

Figure 15 : Make reservation seque,ce diagram

e) Update Travel plan sequence diagram

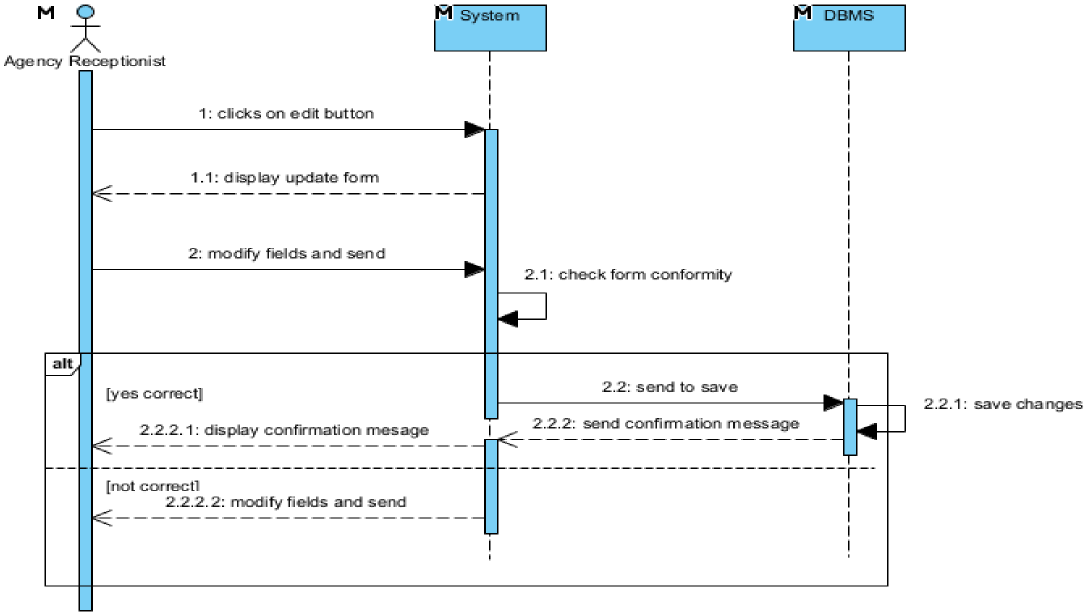

Agency Receptionist
System
DBMS
1: clicks on edit button
1.1: display update form
2: modify fields and send
2.1: check form conformity
alt
[yes correct]
2.2: send to save
2.2.1: save changes
2.2.2: send confirmation message
2.2.2.1: display confirmation mesage
[not correct]
2.2.2.2: modify fields and send

Figure 16 : Update payment sequence diagram

D. Activity diagram

a) 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.

b) Formalism

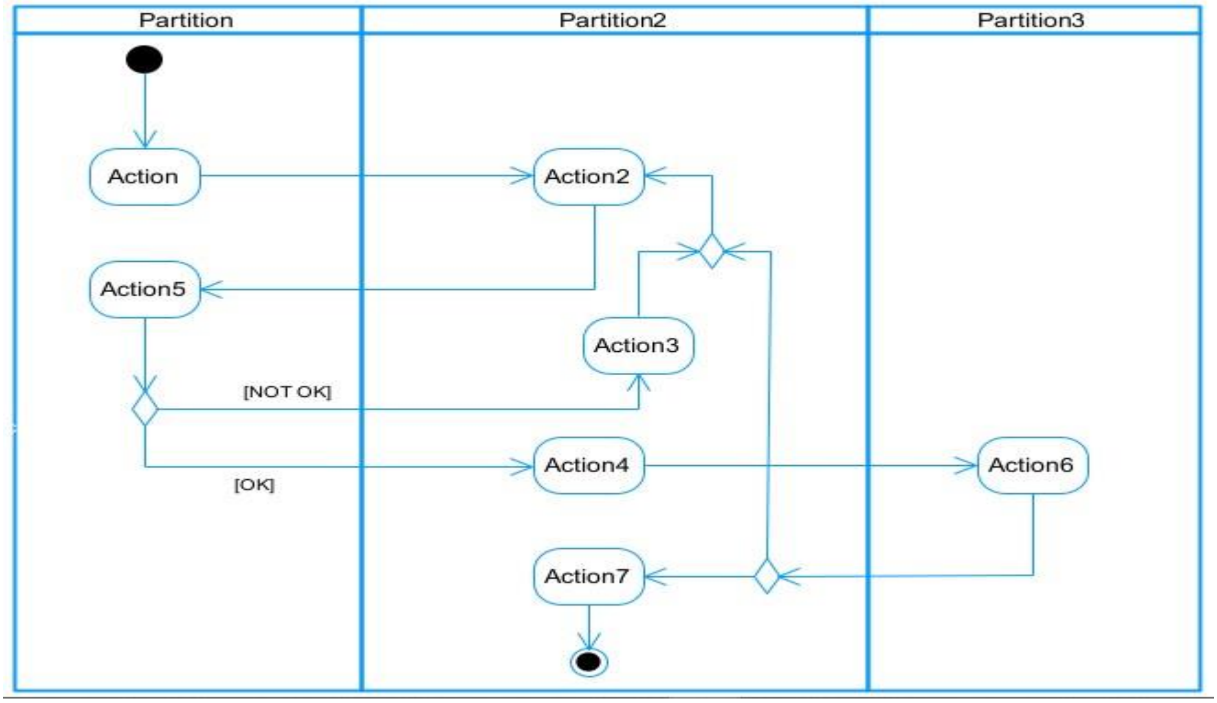
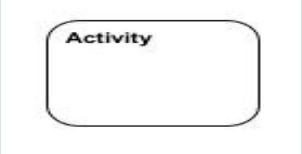
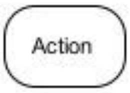
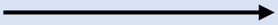
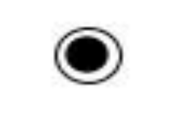
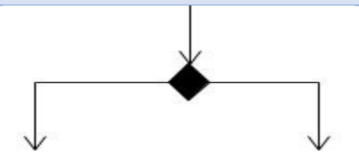
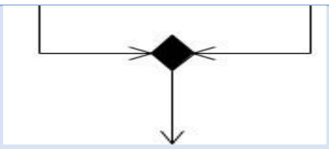
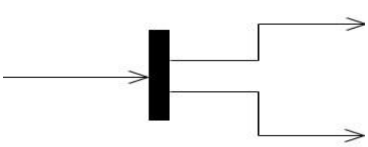
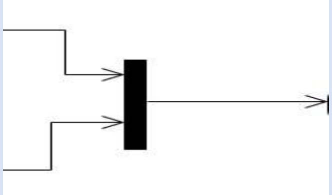
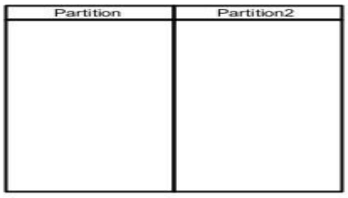


Figure 17 : Activity diagram formalism

Table 17 : 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.

c) Authenticate activity diagram

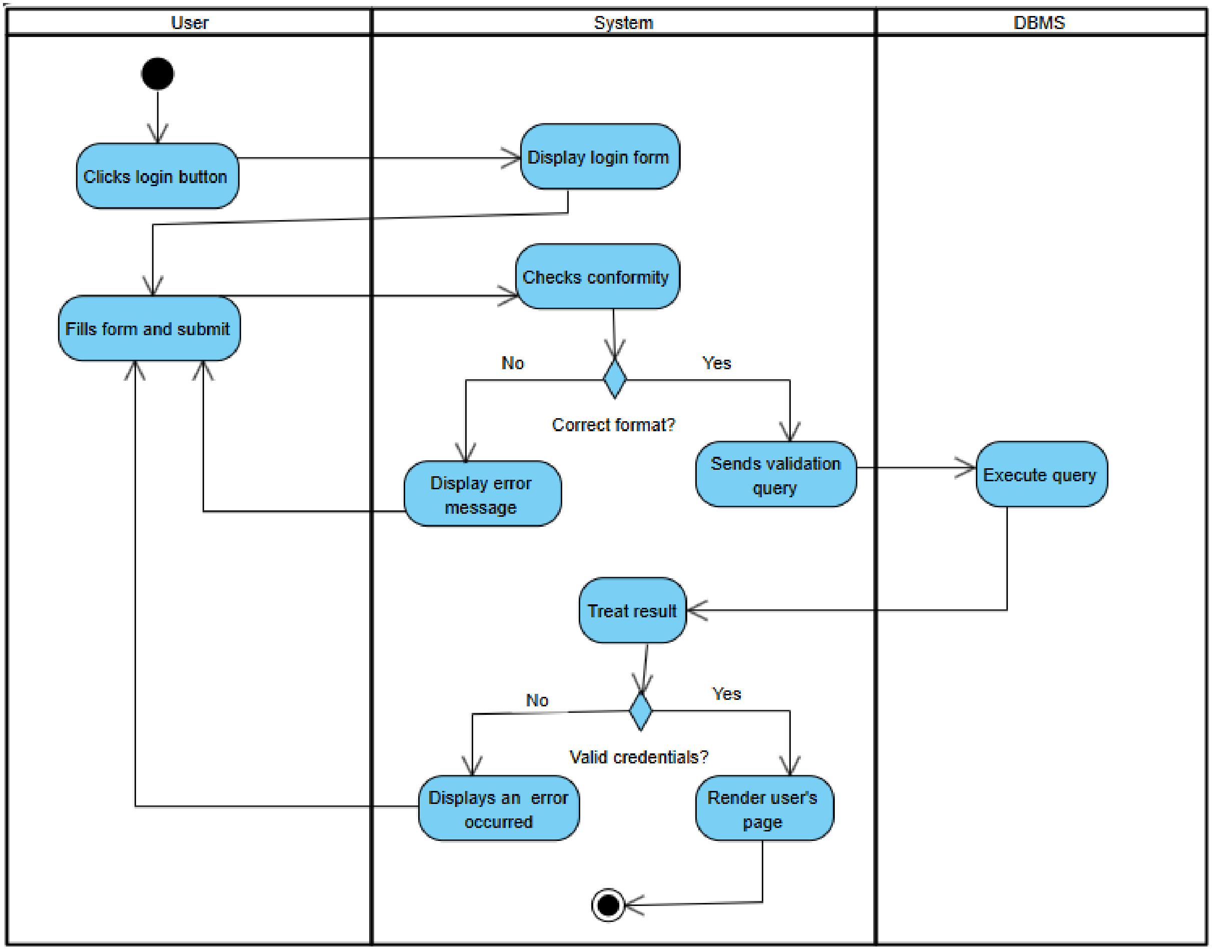


Figure 18 : Authenticate activity diagram

d) Make reservation activity diagram

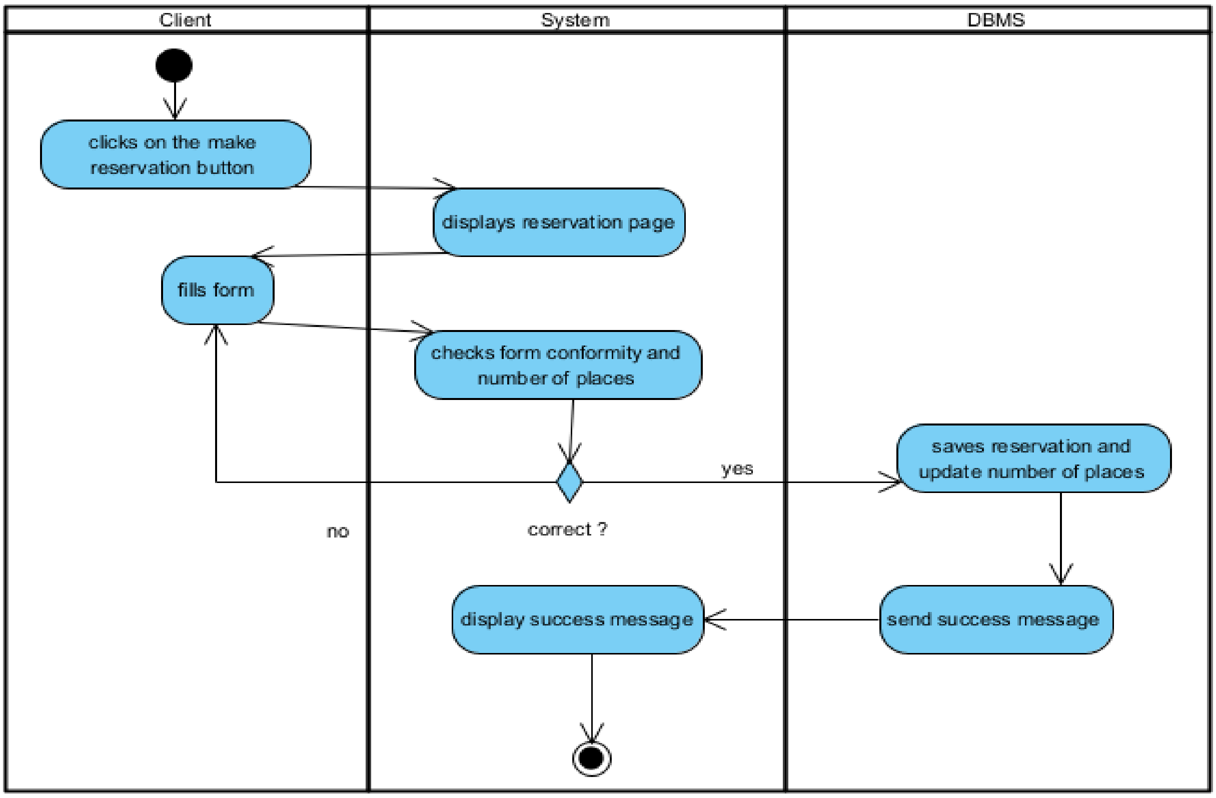

Client
System
DBMS
clicks on the make reservation button
displays reservation page
fills form
checks form conformity and number of places
yes
saves reservation and update number of places
no
correct ?
display success message
send success message

Figure 19 : Make reservation activity diagram

e) Update Travel plan activity diagram

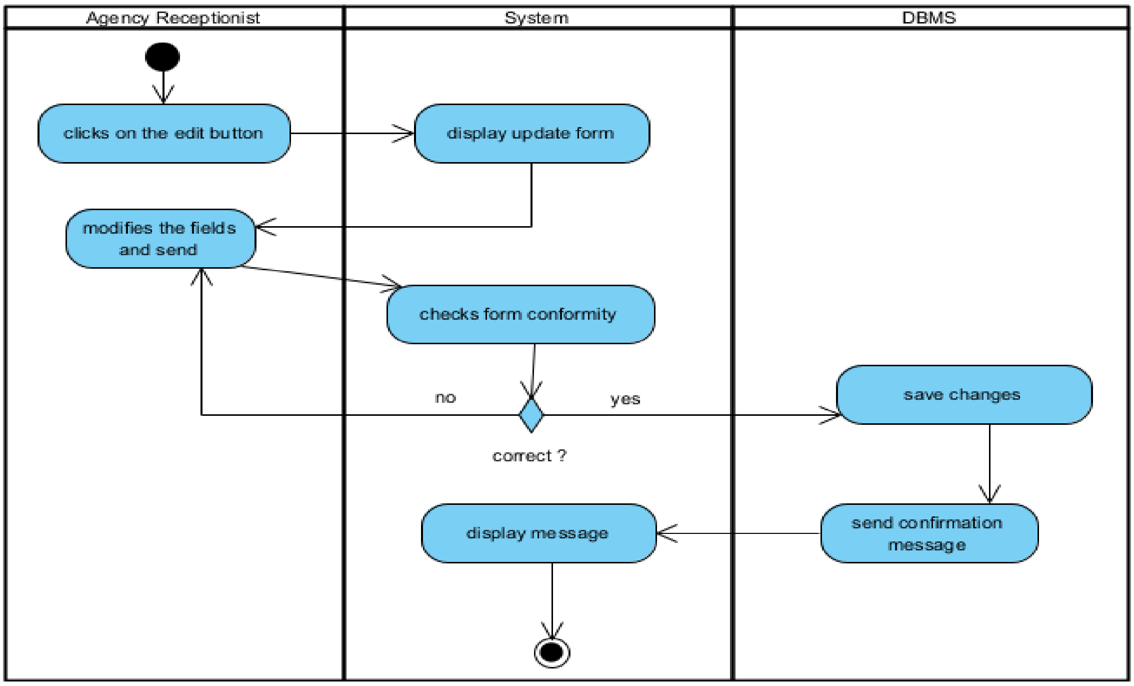


Figure 20 : Update travel plan activity diagram

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

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.

II. Class diagram

a) 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.

b) Formalism

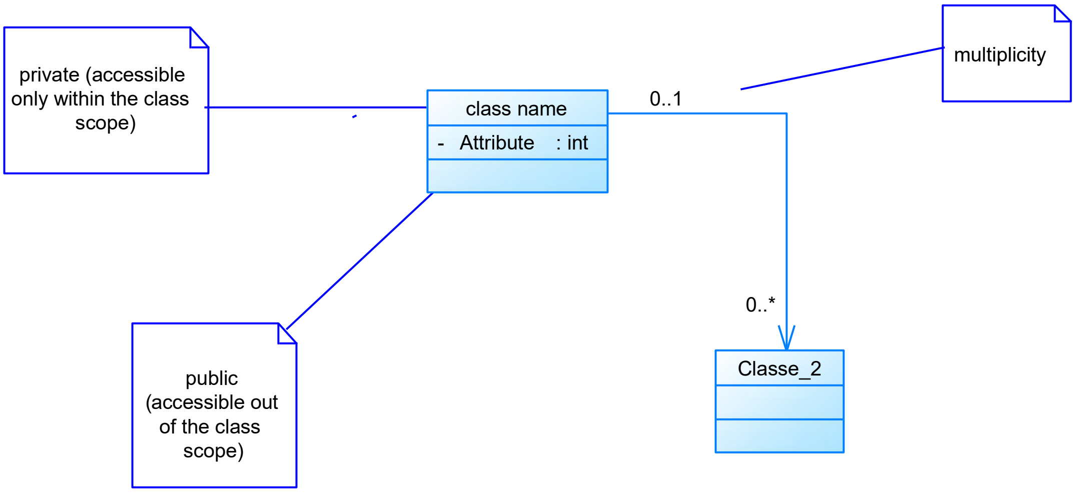
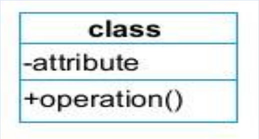
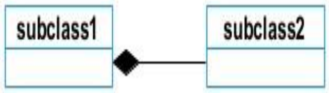
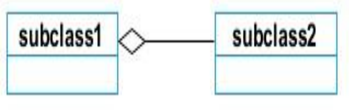
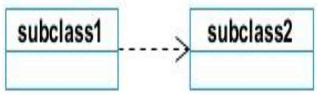
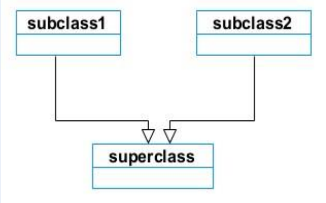
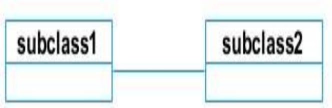

private (accessible only within the class scope)
multiplicity
class name
- Attribute : int
0..1
0..*
public (accessible out of the class scope)
Classe_2

Figure 21 : Class diagram formalism

Table 18 : 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.

c) System class diagram

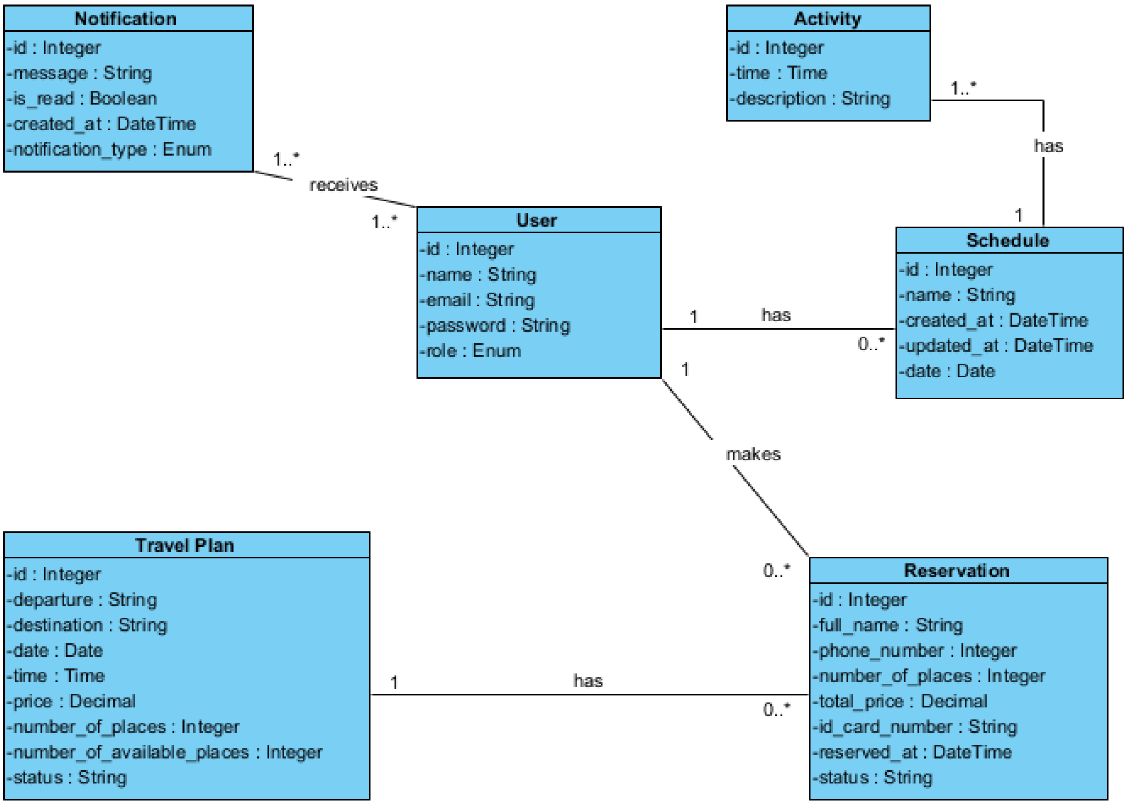


Figure 22 : System class diagram

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

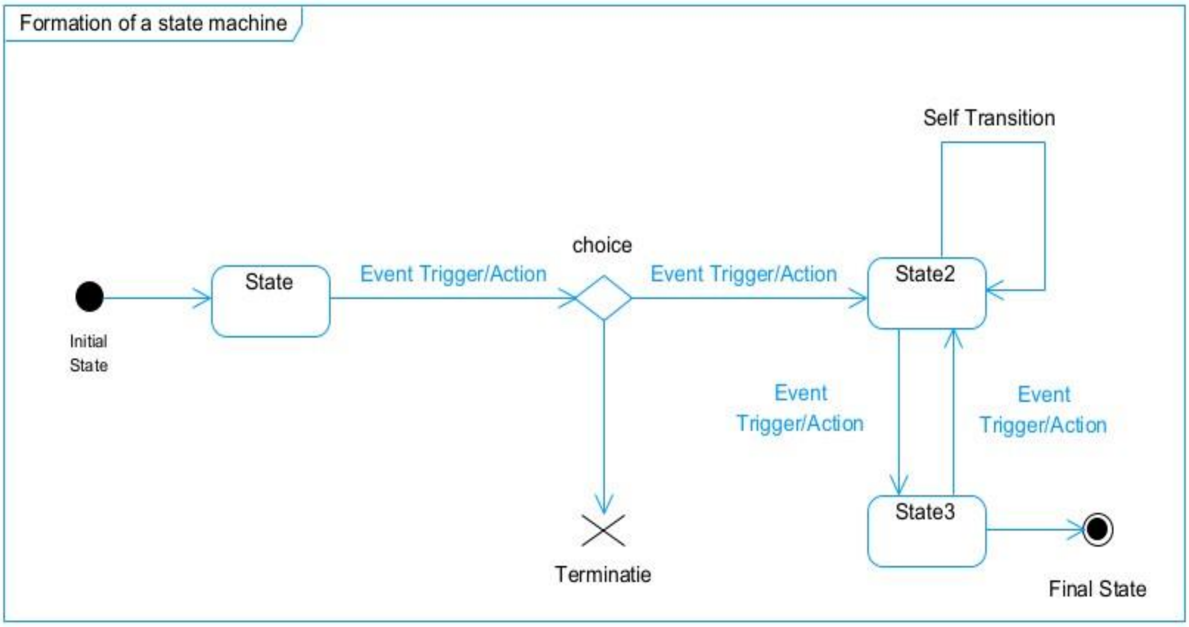
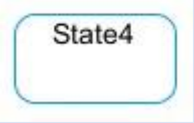
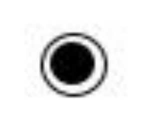
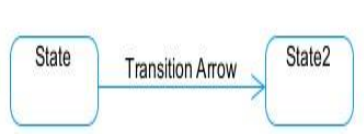
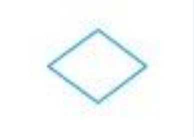
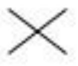
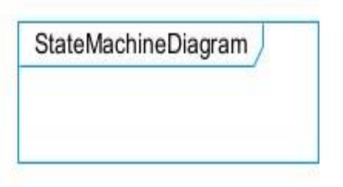


Figure 23 : State machine diagram formalism

Table 19 : 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) Reservation state machine diagram

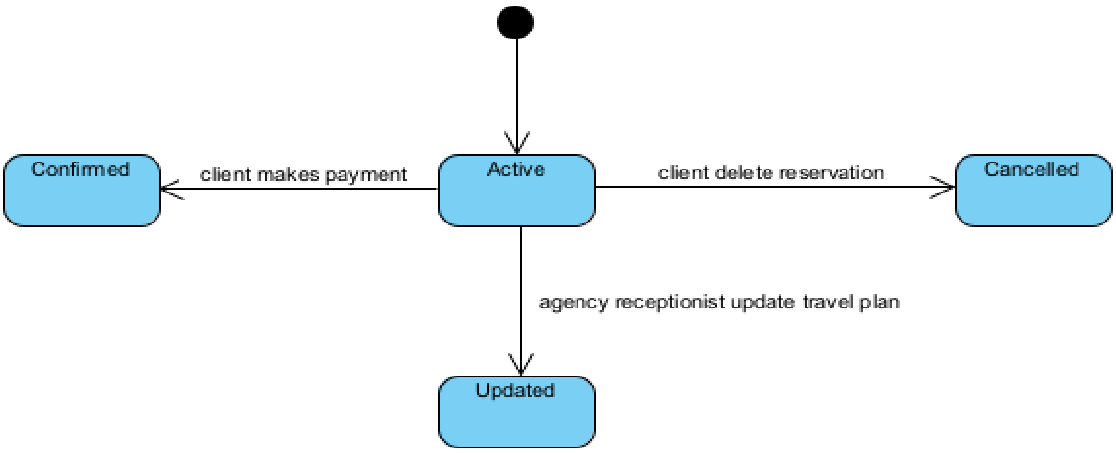

Confirmed
client makes payment
Active
client delete reservation
Cancelled
agency receptionist update travel plan
Updated

Figure 24 : Reservation 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

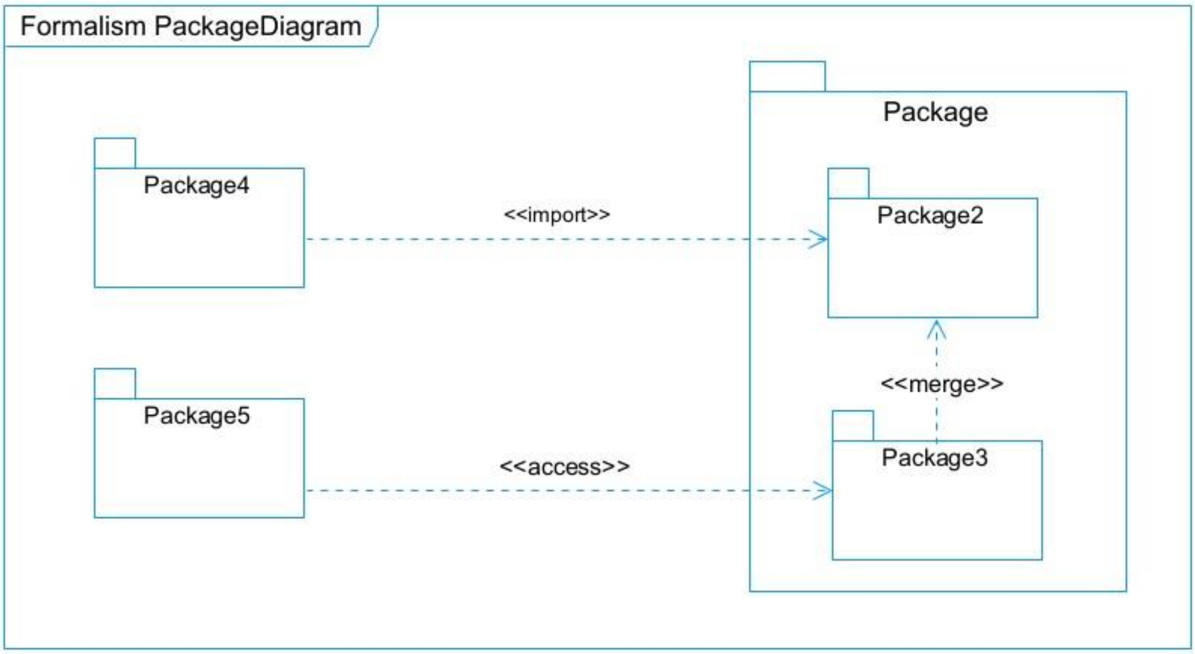
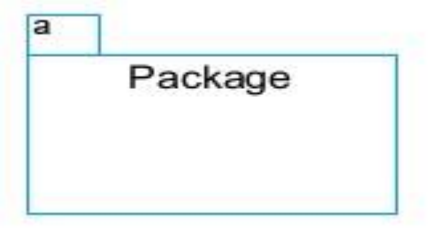
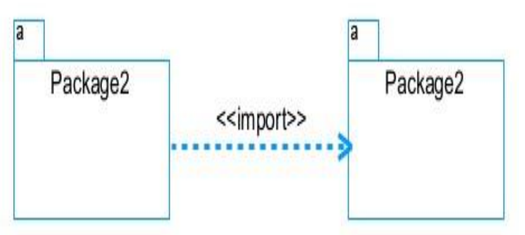
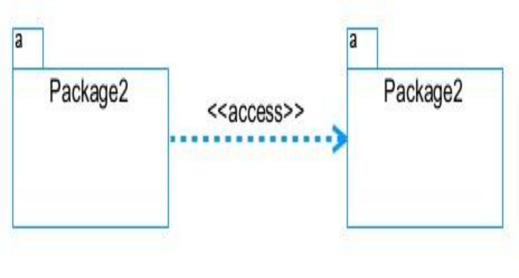
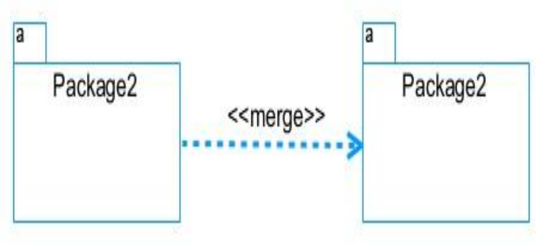


Figure 25 : Package diagram formalism

Table 20 : 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.

c) System package diagram

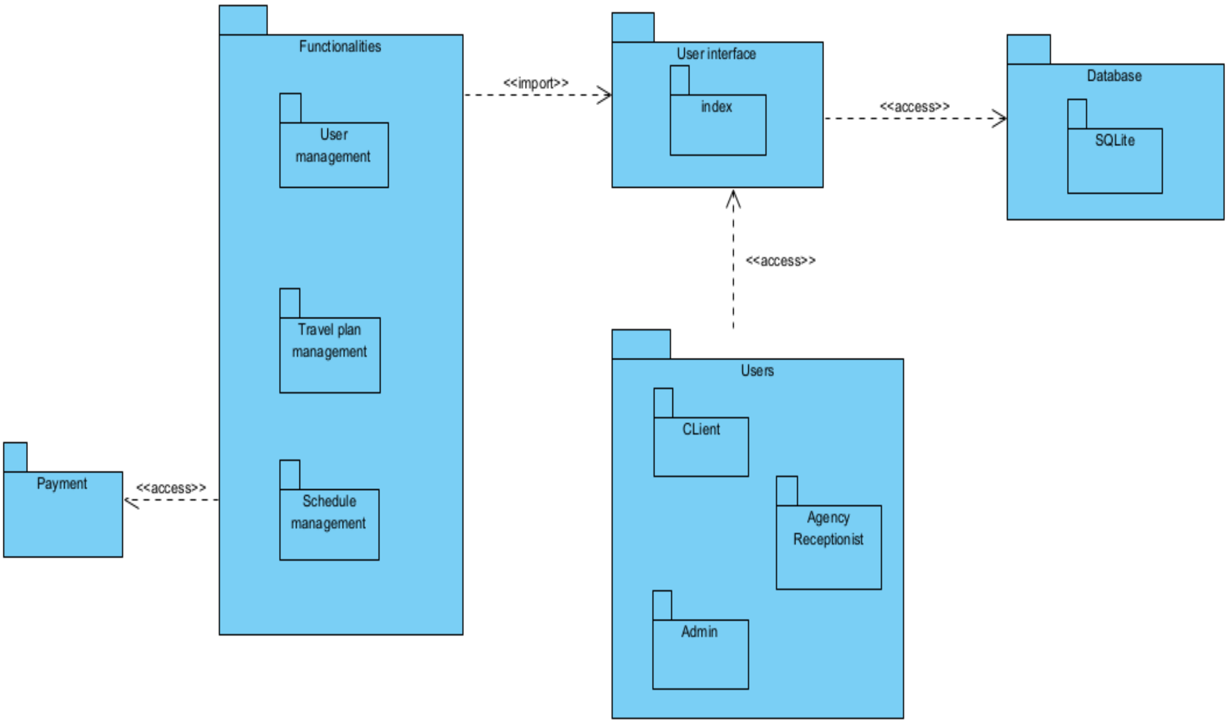


Figure 26 : System package diagram

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

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.

I. 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.

1. 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 27 : HTML logo

2. 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 28 : CSS logo

3. 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 29 : SQLite logo

4. 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 30 : Django logo

5. 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 31 : Visual Paradigm logo

6. 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 32 : Gantt project logo

7. Github

GitHub is a powerful platform for version control and collaboration, primarily used for software development.

Figure 33 : Github logo

II. Architecture of our application

A. 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

2) **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

3) **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

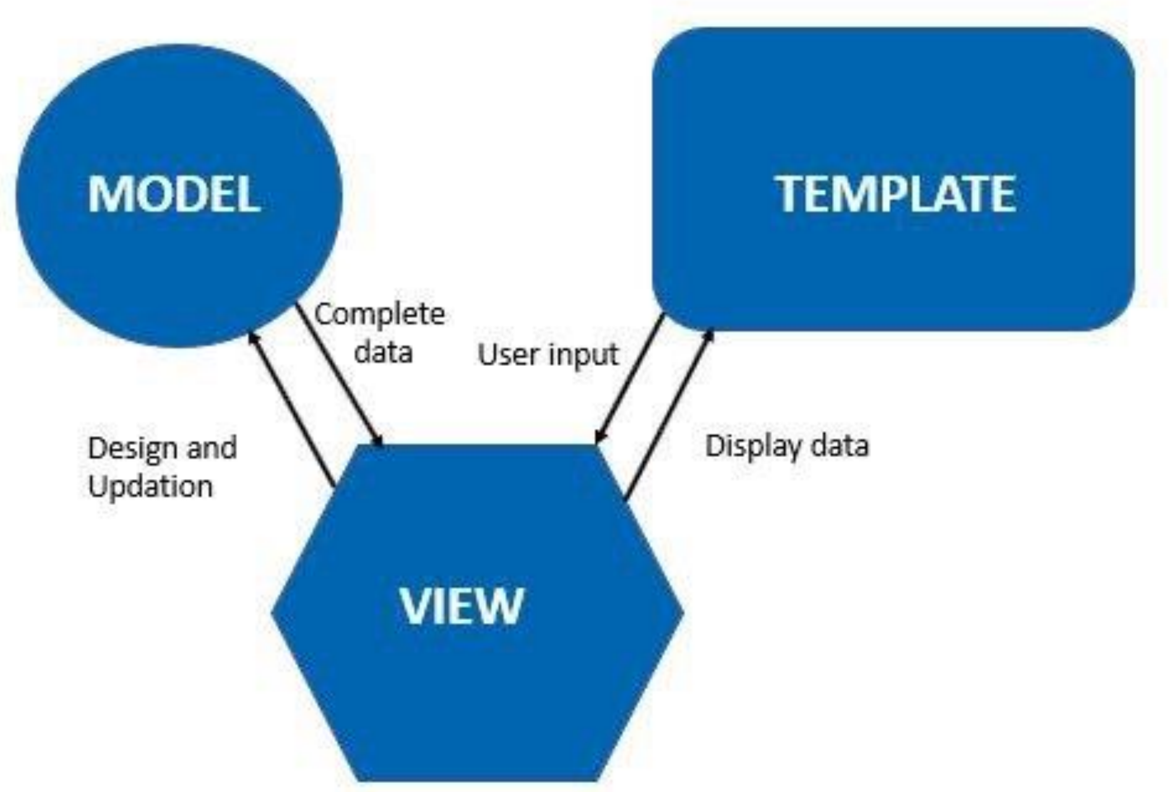


Figure 34 : The MVT design pattern

B. 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).

i. 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.

ii. 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.

iii. 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.

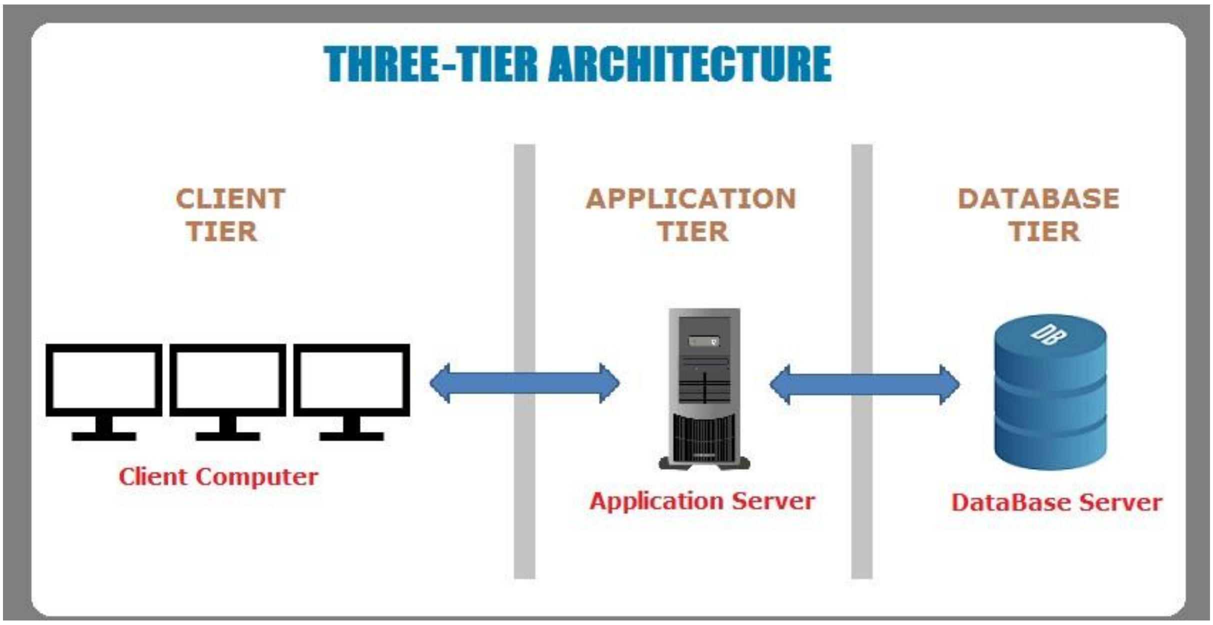


Figure 35 : The three-tier application representation

III. Deployment diagram

a) 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.

b) Formalism

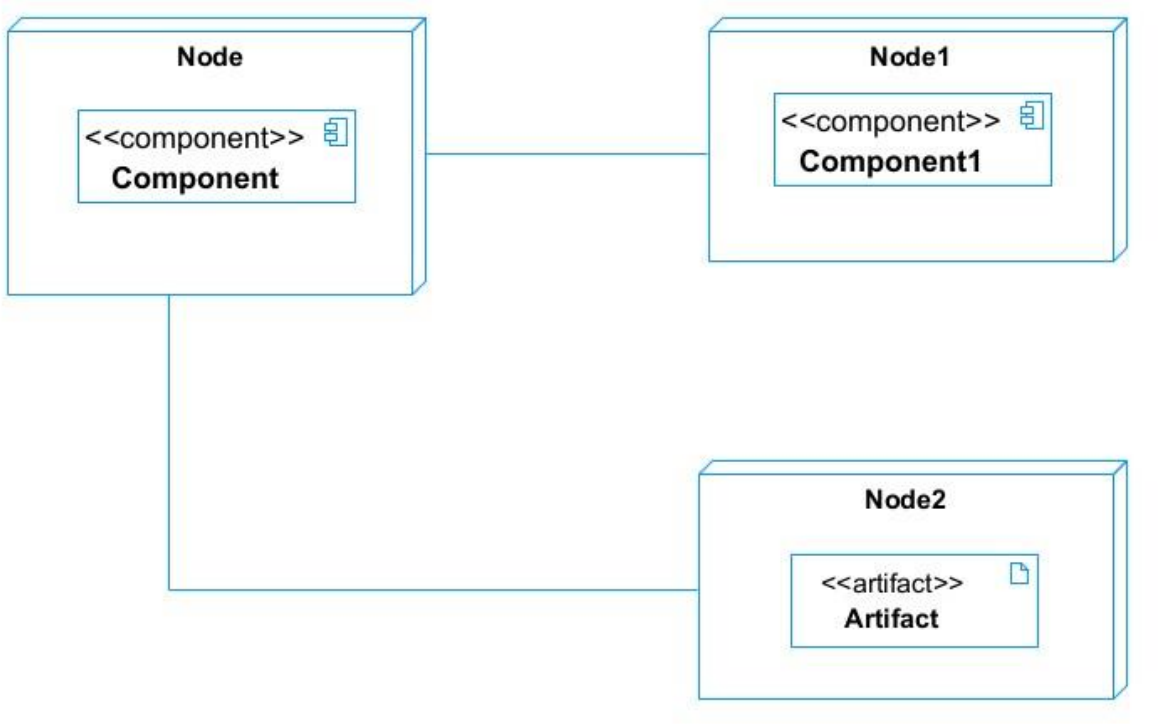
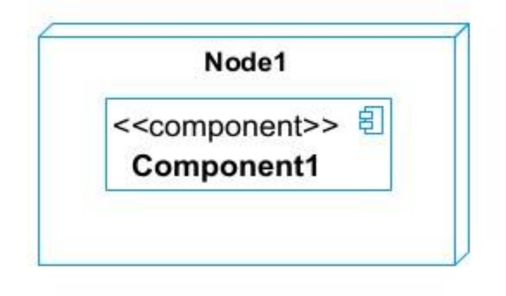
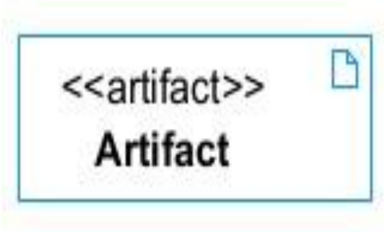
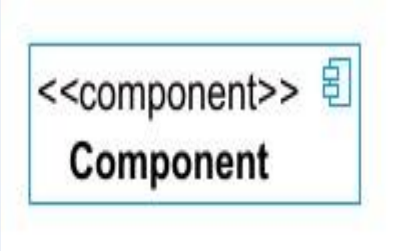
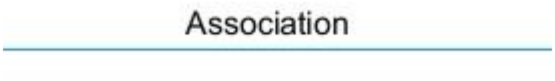


Figure 36 : Deployment diagram formalism

Table 21 : 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

c) System deployment diagram

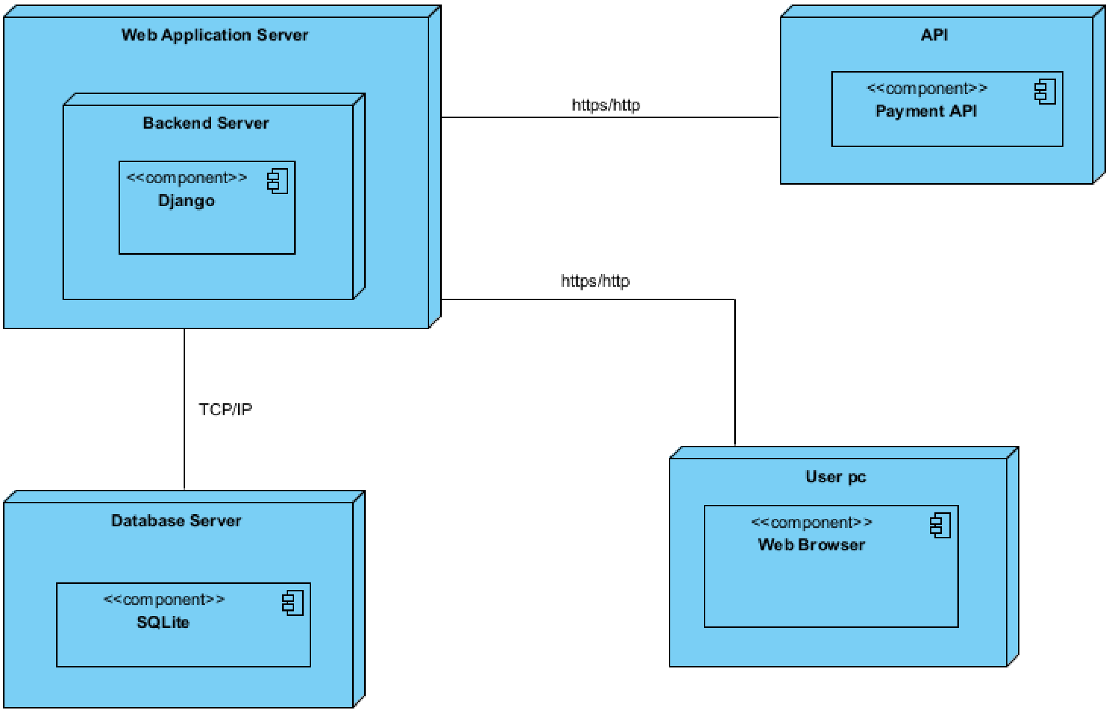


Figure 37 : System deployment diagram

IV. Component diagram

a) 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.

b) Formalism

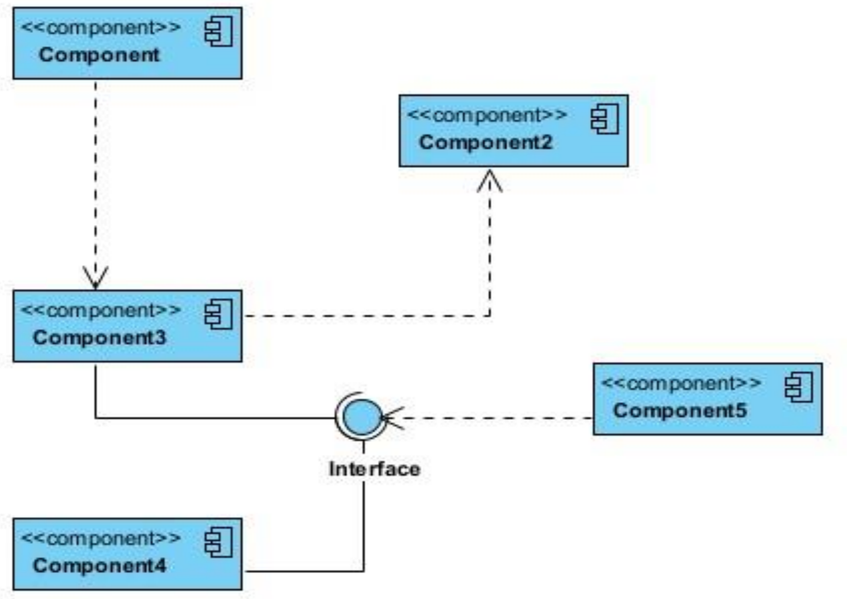
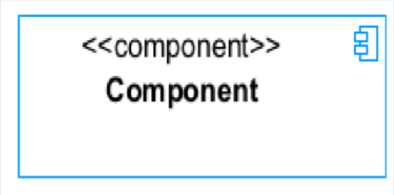
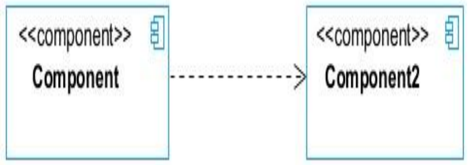
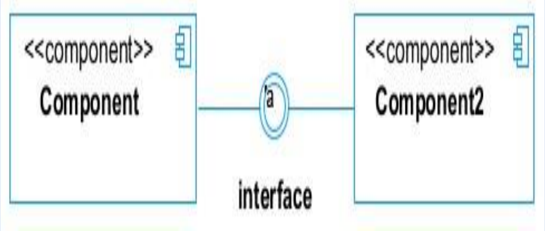
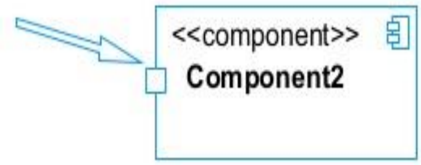


Figure 38 : Component diagram formalism

Table 22 : Component diagram components

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	 <<component>> Component a interface <<component>> Component2	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.

c) System component diagram

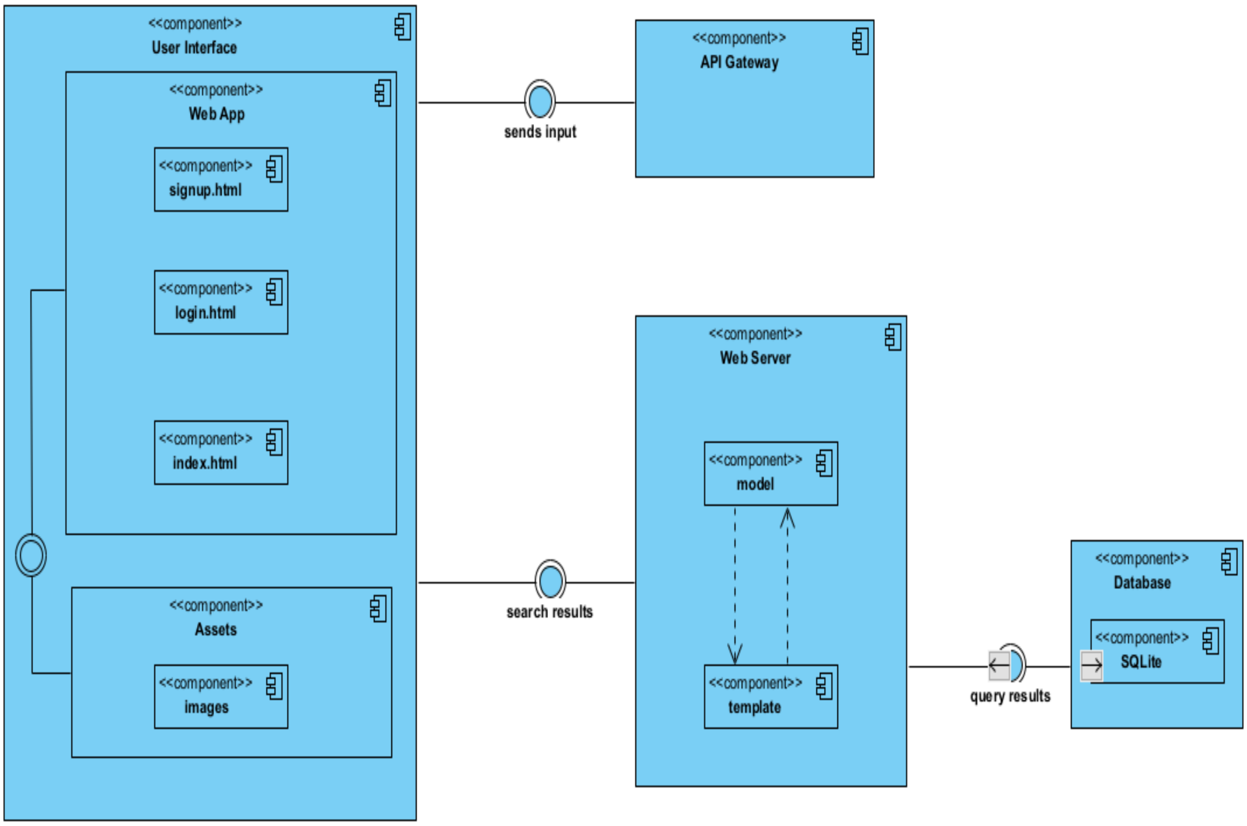


Figure 39 : System component diagram

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.

CHAPTER SIX : FUNCTIONAL TESTING

Preamble

The goal of this chapter is to ensure that each component and feature of the software application performs its intended function correctly.

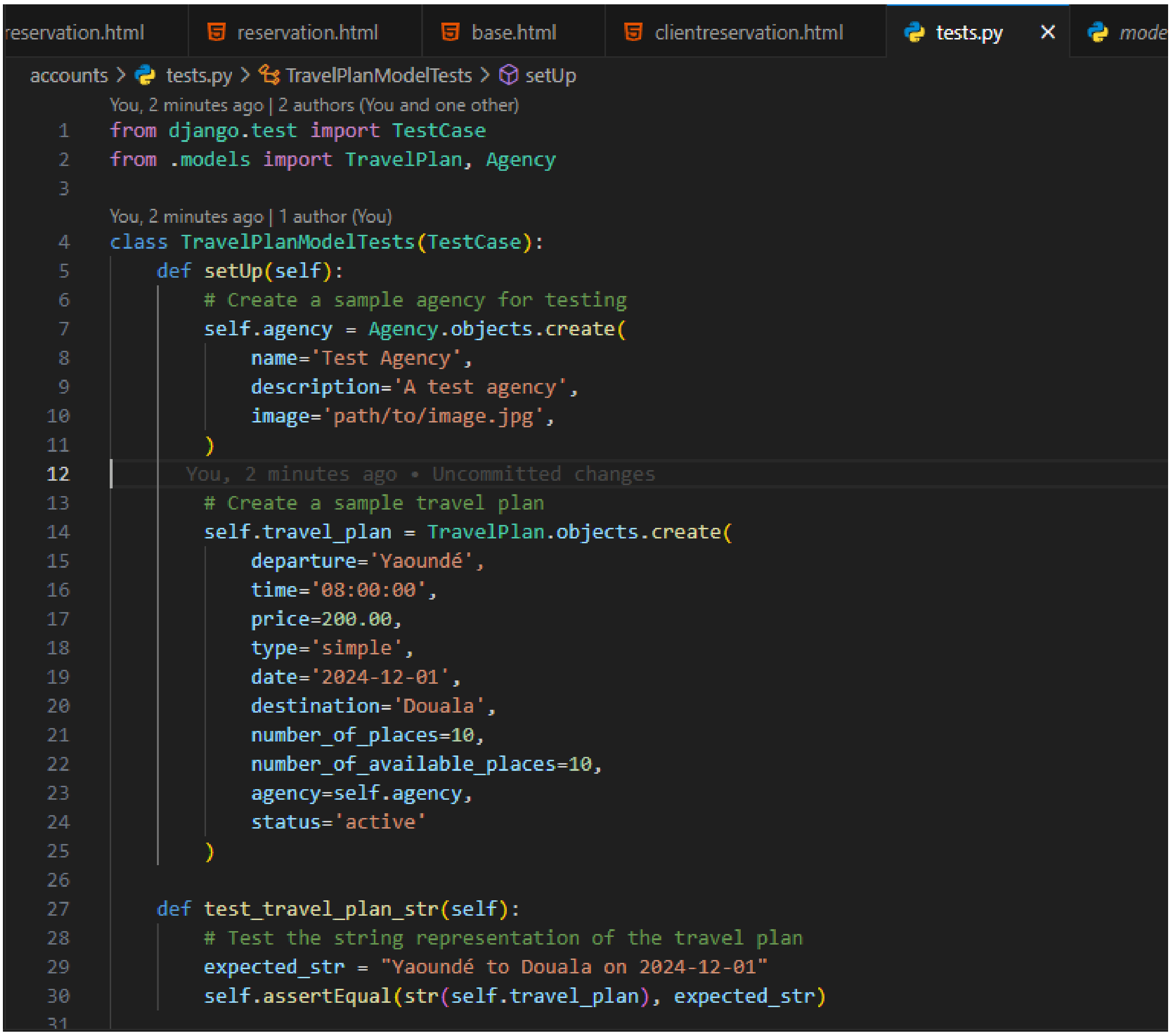
INTRODUCTION

Functional testing is a type of software testing that focuses on verifying that the system or application behaves as intended and meets the specified functional requirement. It is a crucial part of the software development life cycle and is typically performed by quality assurance teams. It involves testing various aspects, such as the user interface, input validation, data manipulation, and output generation, to validate that the software functions as expected.

I. Test

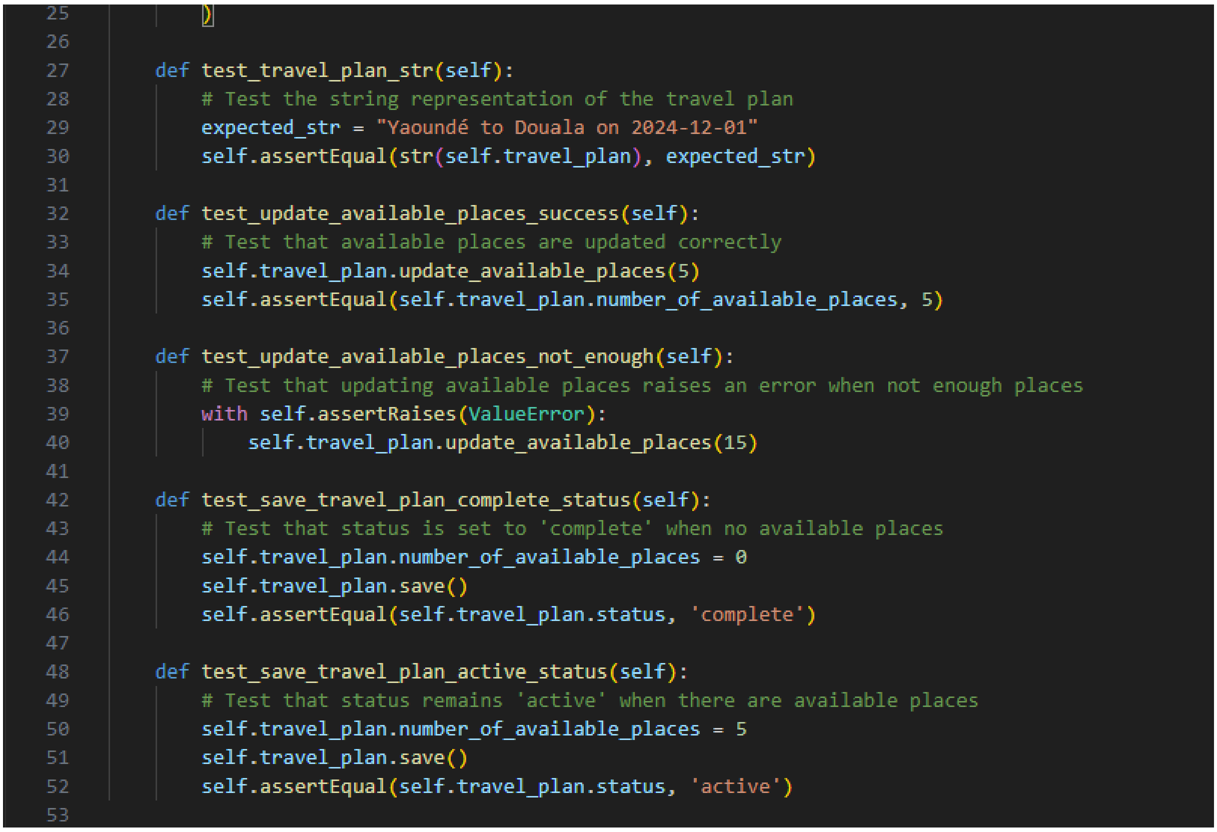
For the test of functionality of our web application, we’ve used the Django unit test. In Django, unit testing involves creating test cases to validate the behaviour of individual components like models, views, and forms. Django provides a built-in testing framework, which is based on Python's unittest module, and allows us to write tests to verify the functionality of our application.

TravelPlan model test



```
You, 2 minutes ago | 2 authors (You and one other)
from django.test import TestCase
from .models import TravelPlan, Agency

You, 2 minutes ago | 1 author (You)
class TravelPlanModelTests(TestCase):
    def setUp(self):
        # Create a sample agency for testing
        self.agency = Agency.objects.create(
            name='Test Agency',
            description='A test agency',
            image='path/to/image.jpg',
        )
          You, 2 minutes ago • Uncommitted changes
        # Create a sample travel plan
        self.travel_plan = TravelPlan.objects.create(
            departure='Yaoundé',
            time='08:00:00',
            price=200.00,
            type='simple',
            date='2024-12-01',
            destination='Douala',
            number_of_places=10,
            number_of_available_places=10,
            agency=self.agency,
            status='active'
        )

    def test_travel_plan_str(self):
        # Test the string representation of the travel plan
        expected_str = "Yaoundé to Douala on 2024-12-01"
        self.assertEqual(str(self.travel_plan), expected_str)
```

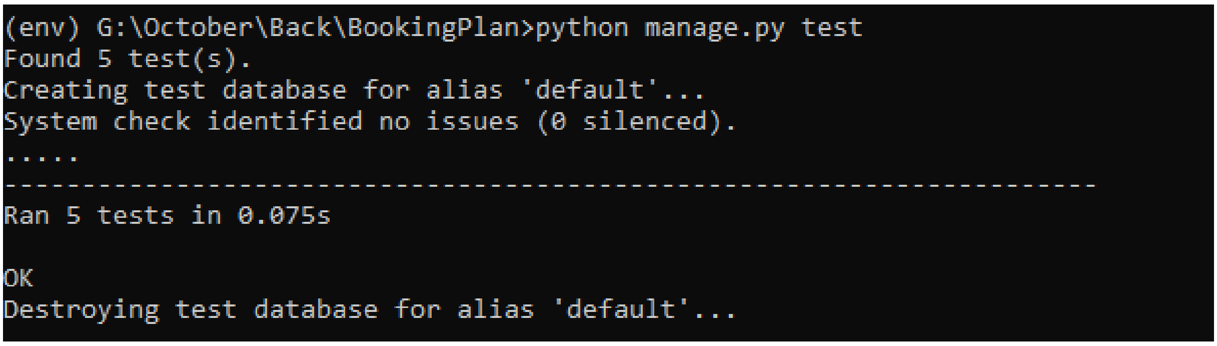
Figure 40 :TravelPlan model test (1)



```
        )

    def test_travel_plan_str(self):
        # Test the string representation of the travel plan
        expected_str = "Yaoundé to Douala on 2024-12-01"
        self.assertEqual(str(self.travel_plan), expected_str)

    def test_update_available_places_success(self):
        # Test that available places are updated correctly
        self.travel_plan.update_available_places(5)
        self.assertEqual(self.travel_plan.number_of_available_places, 5)

    def test_update_available_places_not_enough(self):
        # Test that updating available places raises an error when not enough places
        with self.assertRaises(ValueError):
            self.travel_plan.update_available_places(15)

    def test_save_travel_plan_complete_status(self):
        # Test that status is set to 'complete' when no available places
        self.travel_plan.number_of_available_places = 0
        self.travel_plan.save()
        self.assertEqual(self.travel_plan.status, 'complete')

    def test_save_travel_plan_active_status(self):
        # Test that status remains 'active' when there are available places
        self.travel_plan.number_of_available_places = 5
        self.travel_plan.save()
        self.assertEqual(self.travel_plan.status, 'active')

```

Figure 41 :TravelPlan model test (2)

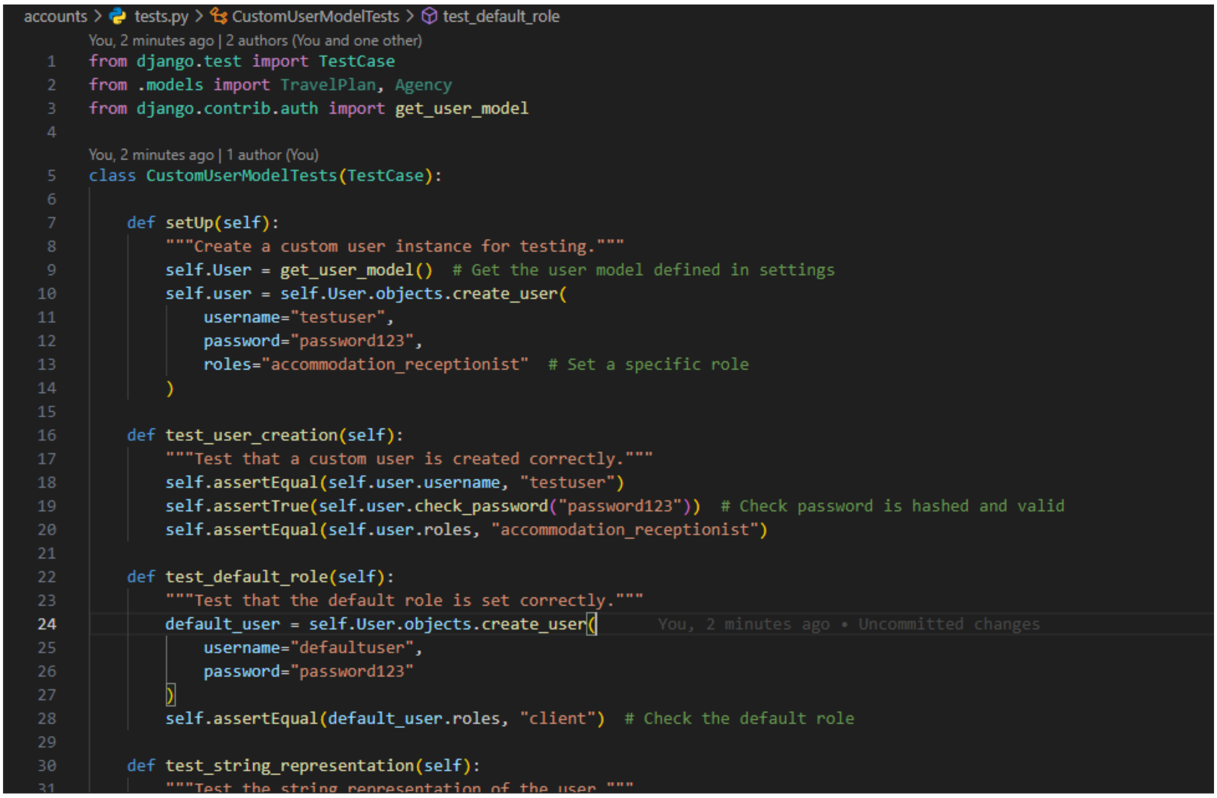
Result of the TravelPlan model test



```
(env) G:\October\Back\BookingPlan>python manage.py test
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
----------------------------------------------------------------------
Ran 5 tests in 0.075s

OK
Destroying test database for alias 'default'...
```

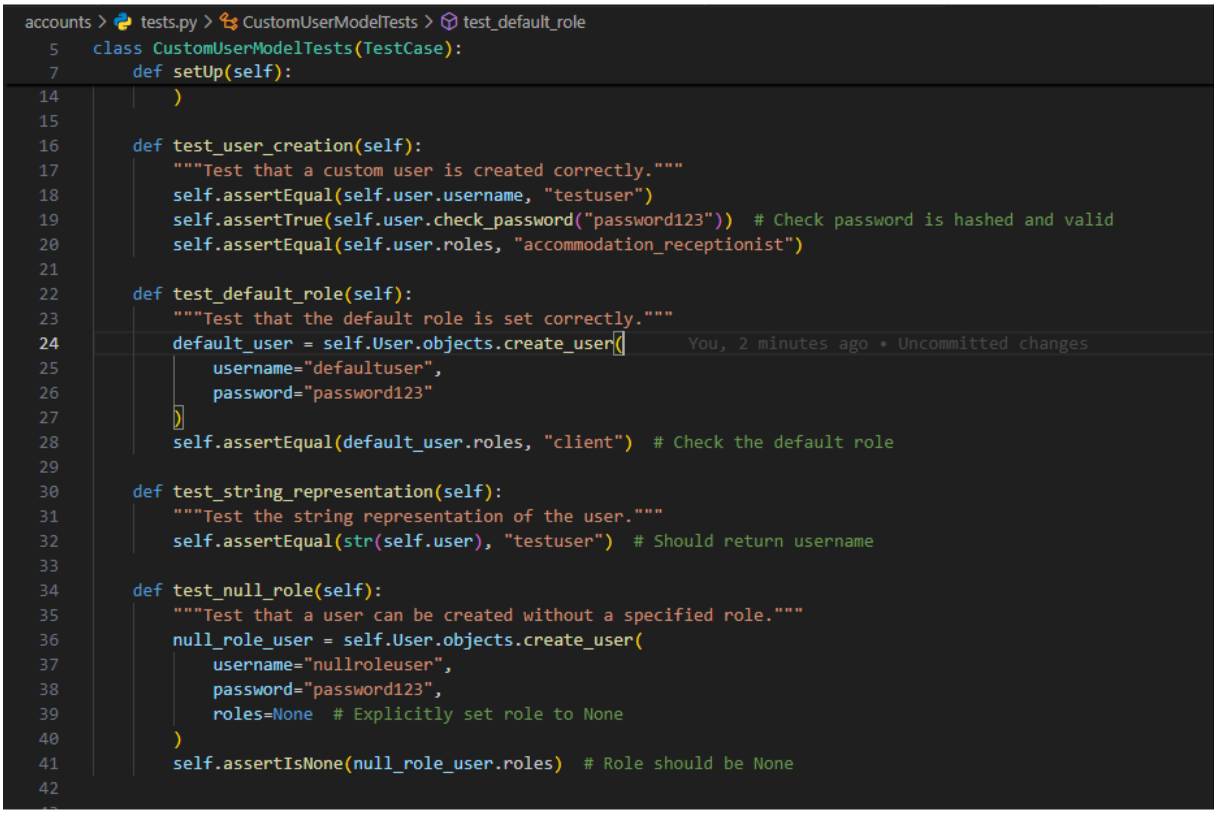
Figure 42 :Result of TravelPlan model test

<u>CustomUser model test</u>


accounts > tests.py > CustomUserModelTests > test_default_role
You, 2 minutes ago | 2 authors (You and one other)
You, 2 minutes ago | 1 author (You)
You, 2 minutes ago • Uncommitted changes

```python
from django.test import TestCase
from .models import TravelPlan, Agency
from django.contrib.auth import get_user_model

class CustomUserModelTests(TestCase):

    def setUp(self):
        """Create a custom user instance for testing."""
        self.User = get_user_model()  # Get the user model defined in settings
        self.user = self.User.objects.create_user(
            username="testuser",
            password="password123",
            roles="accommodation_receptionist"  # Set a specific role
        )

    def test_user_creation(self):
        """Test that a custom user is created correctly."""
        self.assertEqual(self.user.username, "testuser")
        self.assertTrue(self.user.check_password("password123"))  # Check password is hashed and valid
        self.assertEqual(self.user.roles, "accommodation_receptionist")

    def test_default_role(self):
        """Test that the default role is set correctly."""
        default_user = self.User.objects.create_user(
            username="defaultuser",
            password="password123"
        )
        self.assertEqual(default_user.roles, "client")  # Check the default role

    def test_string_representation(self):
        """Test the string representation of the user."""
```

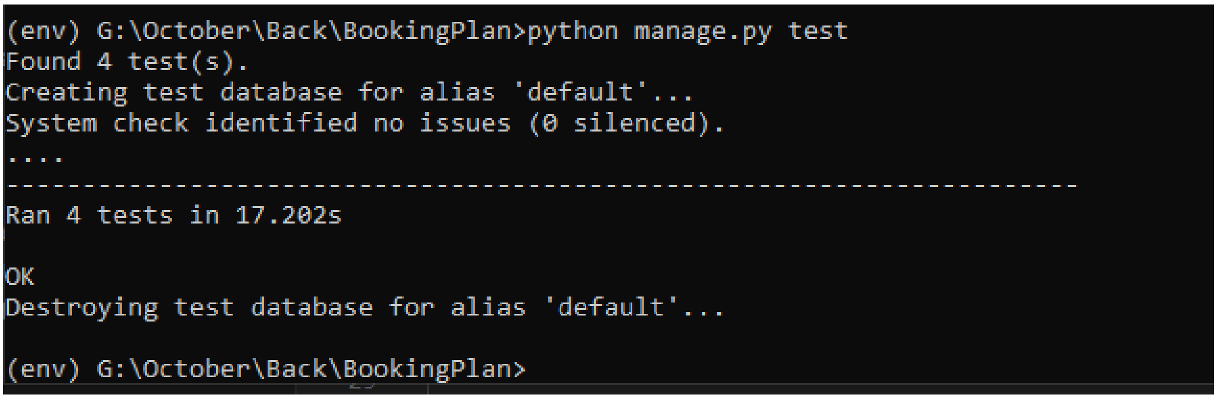
Figure 43 : CustomUser model test (1)


accounts > tests.py > CustomUserModelTests > test_default_role
You, 2 minutes ago • Uncommitted changes

```python
class CustomUserModelTests(TestCase):
    def setUp(self):
        )

    def test_user_creation(self):
        """Test that a custom user is created correctly."""
        self.assertEqual(self.user.username, "testuser")
        self.assertTrue(self.user.check_password("password123"))  # Check password is hashed and valid
        self.assertEqual(self.user.roles, "accommodation_receptionist")

    def test_default_role(self):
        """Test that the default role is set correctly."""
        default_user = self.User.objects.create_user(
            username="defaultuser",
            password="password123"
        )
        self.assertEqual(default_user.roles, "client")  # Check the default role

    def test_string_representation(self):
        """Test the string representation of the user."""
        self.assertEqual(str(self.user), "testuser")  # Should return username

    def test_null_role(self):
        """Test that a user can be created without a specified role."""
        null_role_user = self.User.objects.create_user(
            username="nullroleuser",
            password="password123",
            roles=None  # Explicitly set role to None
        )
        self.assertIsNone(null_role_user.roles)  # Role should be None
```

Figure 44 : CustomUser model test (2)

Result of the CustomUser model test



```
(env) G:\October\Back\BookingPlan>python manage.py test
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
....
----------------------------------------------------------------------
Ran 4 tests in 17.202s

OK
Destroying test database for alias 'default'...

(env) G:\October\Back\BookingPlan>
```

Figure 45 : Result of the CustomUser model test

CHAPTER SEVEN: USER GUIDE

Preamble

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

CONTENT OVERVIEW

INTRODUCTION

The user guide is a document for the various users of the application. Its purpose is to provide the user with the help he or she needs to use the application. In this section, we'll be presenting how to deploy the software and giving a detailed description of some of the pages developed in our application.

I. Installation Guide

Follow these steps to set up and run our system on your local machine.

Prerequisites

- Python 3.8
- Django 4.0
- Virtual Environment

1- Make sure you have the project in your local machine.

2- Create and activate a virtual environment.

“python -m venv venv”

“venv\Scripts\activate”

3- Install the required dependencies from the `requirements.txt` file.

“pip install -r requirements.txt”

4- Configure the database. By default, the system uses SQLite. If using a different database, update the `DATABASES` setting in `settings.py` accordingly.

5- Apply the migrations to set up the database schema.

“python manage.py migrate”

6- Create an admin account to access the Django admin interface.

“python manage.py createsuperuser”

7- Collect static files to serve them properly.

“python manage.py collectstatic”

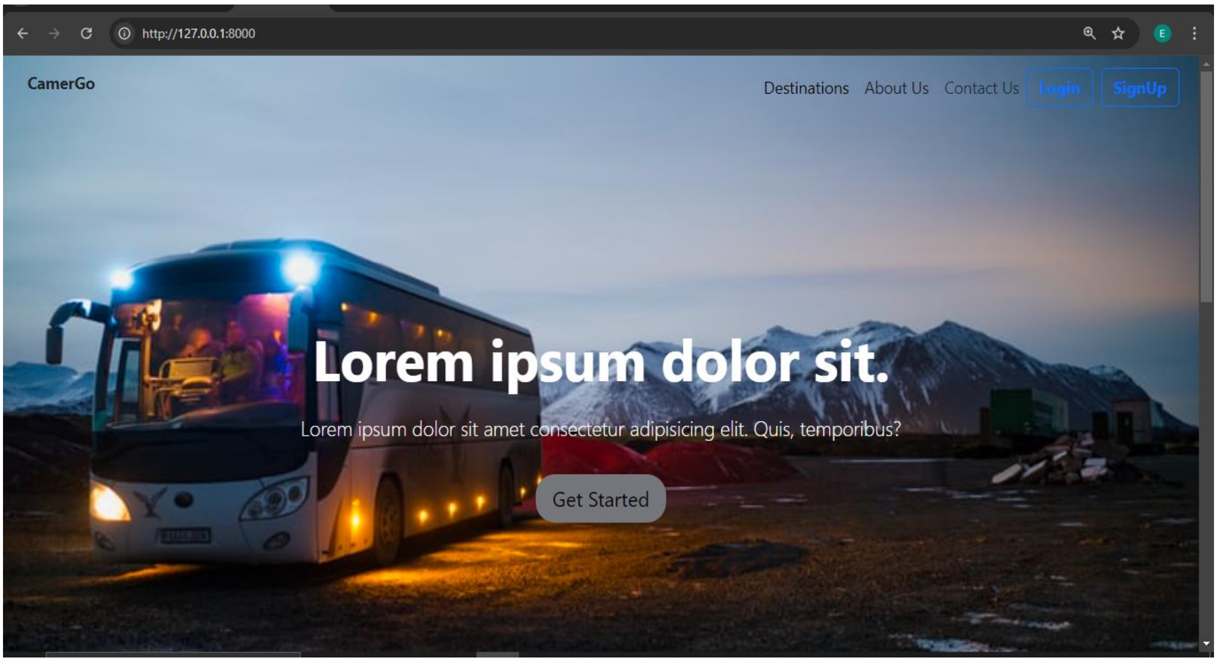
8- Start the Django Development Server

“python manage.py runserver”

9- Visit `**http://127.0.0.1:8000/**` in your browser to see the application.

II. User Guide

url: ‘127.7.0.0.1:8000/

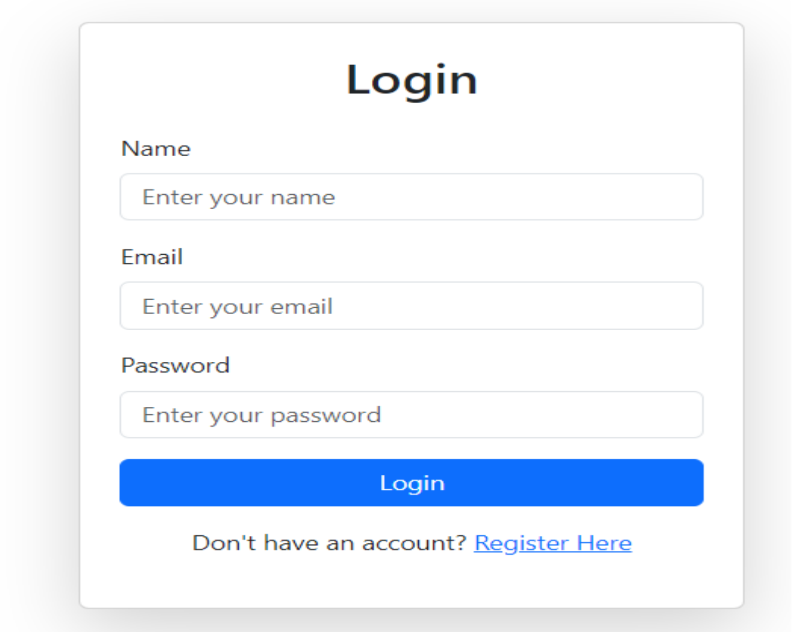


Lorem ipsum dolor sit.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis, temporibus?

Figure 46 : Home page

url: ‘127.7.0.0.1:8000/login/’


Login

Name

Enter your name

Email

Enter your email

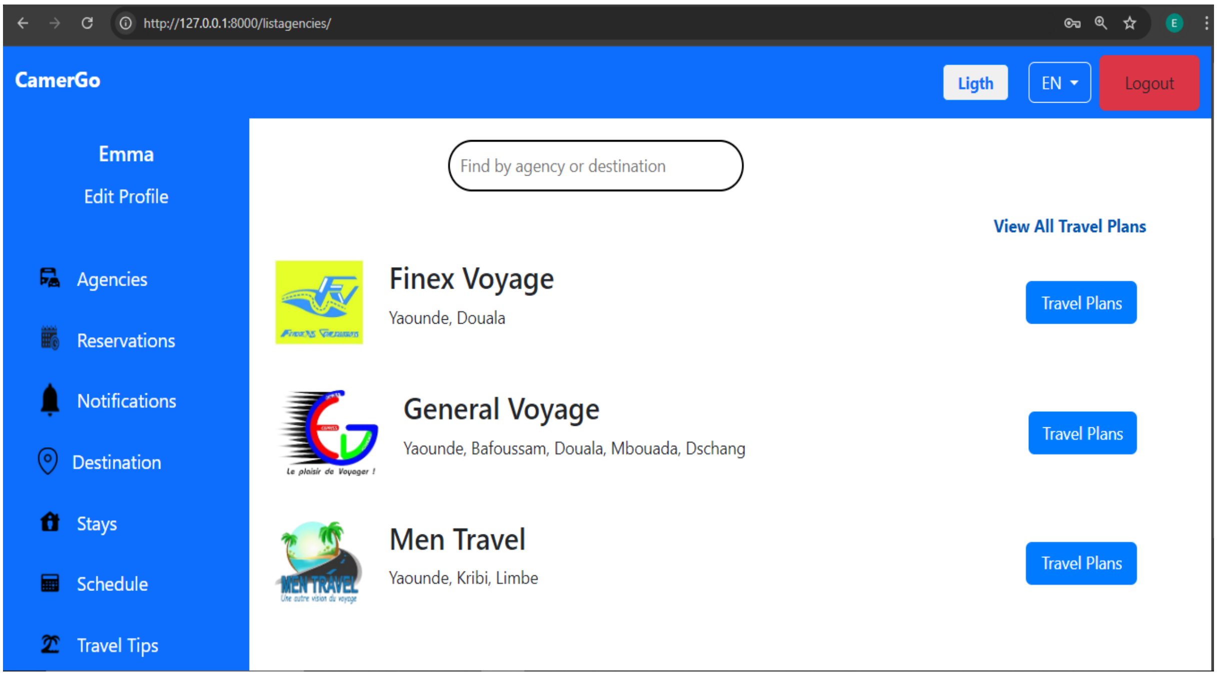
Password

Enter your password

Don't have an account? Register Here

Figure 47:login

url: ‘127.7.0.0.1:8000/listagencies/’



http://127.0.0.1:8000/listagencies/

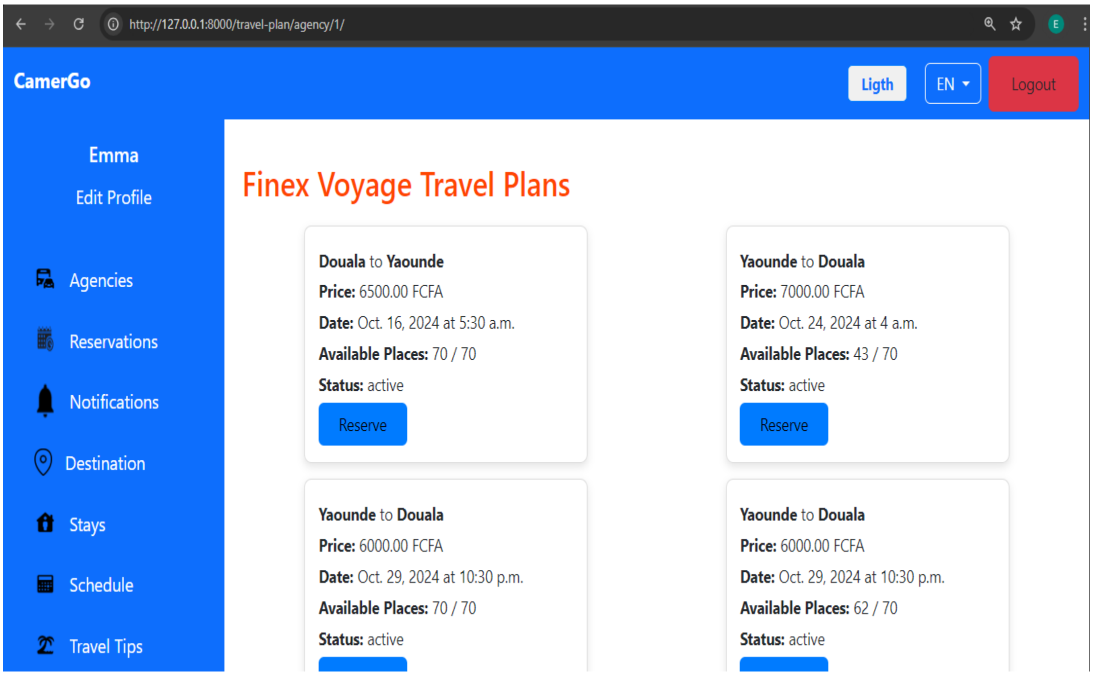
Finex Voyage
Yaounde, Douala

General Voyage
Yaounde, Bafoussam, Douala, Mbouada, Dschang

Men Travel
Yaounde, Kribi, Limbe

Figure 48 : Client Dashboard

url: ‘127.7.0.0.1:8000/travel-plan/agency/1’



http://127.0.0.1:8000/travel-plan/agency/1/

Finex Voyage Travel Plans

Douala to **Yaounde**
Price: 6500.00 FCFA
Date: Oct. 16, 2024 at 5:30 a.m.
Available Places: 70 / 70
Status: active

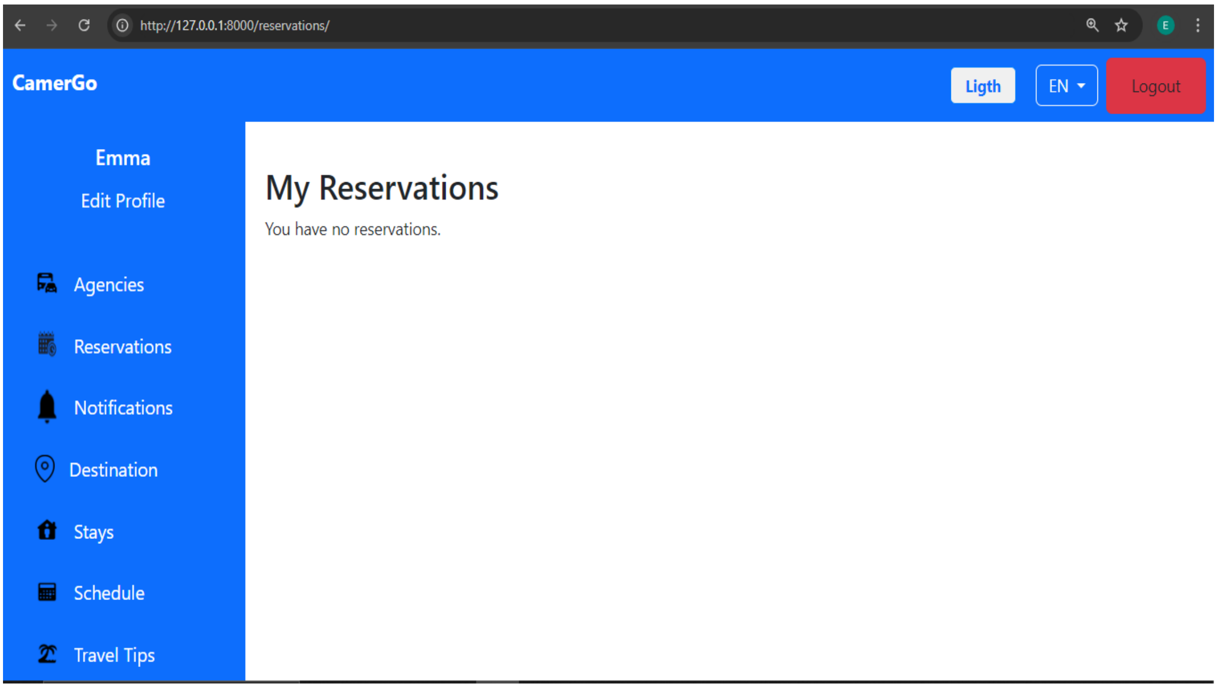
Yaounde to **Douala**
Price: 7000.00 FCFA
Date: Oct. 24, 2024 at 4 a.m.
Available Places: 43 / 70
Status: active

Yaounde to **Douala**
Price: 6000.00 FCFA
Date: Oct. 29, 2024 at 10:30 p.m.
Available Places: 70 / 70
Status: active

Yaounde to **Douala**
Price: 6000.00 FCFA
Date: Oct. 29, 2024 at 10:30 p.m.
Available Places: 62 / 70
Status: active

Figure 49:list of travel plans

url: ‘127.7.0.0.1:8000/reservations/’



My Reservations

You have no reservations.

Figure 50 : client reservations

CONCLUSION

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

GENERAL CONCLUSION

In conclusion, our travel management system represents a major advancement towards a more streamlined and user-friendly approach to travel planning and booking. Throughout this project, we have dedicated ourselves to creating a comprehensive platform that simplifies the travel experience for both travelers and agencies. The system offers a robust set of features, including travel plan creation, activity scheduling, and secure payment processing, all integrated within a single interface. It promotes efficiency, enhances communication, and ensures real-time updates, making travel arrangements more accessible and organized. We believe this system is a significant step towards modernizing the travel industry and providing a seamless, enjoyable experience for users. By continuing to expand and refine the platform, we aim to contribute to a more efficient and customer-focused travel ecosystem. We extend our gratitude to everyone who has supported this project, and hope that the system will be a valuable resource for all its users.

ANNEXE

BIBLIOGRAPHY

- **Mme. TCHINGA ALICE** UML course AICS-Cameroon 2022-2023
- OMG, OMG Unified Modelling Language 2.5, Object Management Group, March 2015, 794 pages.
- **Mr. NGUH PRINCE** python course (Django) AICS Cameroon 2022-2023.
- Mercurial 2024.
- Starting help on Visual Paradigm.
- (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 4, 2018

WEBOGRAPHY

- https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
- what is UML
- https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing) Here we got the 2TUP diagram
- https://thesai.org/Downloads/Volume9No4/Paper_2Unifying_Modeling_Language.pdf we got the differences between UML and MERISE.
- https://docs.djangoproject.com/en/5.1/topics/testing/#:~:text=The%20preferred%20way%20to%20write,for%20that%20kind%20of%20integration.
- https://stackoverflow.com/questions/58974983/django-query-object-values-list-as-parameter
- https://github.com/jameskomo/bus-reservation-system/blob/master/myapp/models.py
- https://forum.djangoproject.com/t/reservation-system/23684/3
- https://www.youtube.com/results?search_query=django+login+rendering+different+pages+base+on+the+user+roles

TABLE OF CONTENT