REPULIQUE DU CAMEROUN Paix-Travail-Patrie



**REALIZE**

YAOUNDE – VALLEE AMADOU

Tel: +237 654486985/695250691 E-mail: realize.contactus@gmail.com

REPUBLIC OF CAMEROON Peace-Work-Fatherland



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

P.O. Box: 13719 Yaoundé Tel: +237.242.729.957 Fax: 22729958 E-mail: contact@iaicameroun.com web site: contact@iaicameroun.com

INTERNSHIP REPORT

DESIGN AND IMPLEMENTATION OF A FOOD COMMAND AND DELIVERY PLATFORM

Internship carried out from 1st July to 30th September 2024 at Realize as part of my pursuit to obtain a

**Higher technician diploma in computer science**

Option**: software engineering**

**Written by:**

**BONG SHALOM NGWI**

Level II Software Engineering students at AICS Cameroon

ACADEMIC YEAR 2023-2024

Professional Supervisor:

**Mr. NGUH prince**

IT engineer

Academic Supervisor:

**Mr. NGUH Prince** Lecturer at AICS/IAI-CAMEROUN

# DEDICATIONS

This Work Is Dedicated To My Family For Their Support And Encouragement Towards My Academic Success.

# ACKNOWLEDGEMENT

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

* God almighty for the gift of life and good health during the intership period.
* The Resident Representative of IAI-Cameroon, Mr. Armand Claude ABANDA, for his support, words of encouragement and the different advices on how to approach situations.
* The chief executive officer of Realize, Mr. NDELOGAKEH Daniel and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Our professional supervisor, Mr. NGUH Prince for his moral support, advice and professional expertise he shared with us.
* Our academic supervisor Mr. AGBOR Donald Anderson for his moral support, effort and advices throughout the academic year.
* To our academic teachers Mrs. TCHOUTCHUO Issabelle, Mr. AGBOR Donald Anderson AND Mr. MESSIO for their advices and assistance in realizing this document.
* To all my teachers of AICS CAMEROON for the knowledge they impacted me with for me to be where I am today.
* To my mother TAMBA Claudia Mbuh for her financial and moral support towards my academic success.
* To
* The countless contributors of open-source programming community, for their great help in learning basic skills and detecting and solving bugs.
* To all my classmates for their collaborative work throughout the academic year.

# TABLE OF CONTENT

[DEDICATIONS i](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800713)

[ACKNOWLEDGEMENT ii](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800714)

[TABLE OF CONTENT iii](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800715)

[LIST OF TABLES ix](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800716)

[LIST OF FIGURES x](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800717)

[GLOSSARY xii](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800718)

[ABSTRACT xiii](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800719)

[RESUME xiv](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800720)

[GENERAL INTRODUCTION 1](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800721)

[PART ONE: 2](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800722)

[INSERTION PHASE 2](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800723)

[INTRODUCTION 4](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177800724)

[I. WELCOME AND INTEGRATION 4](#_Toc177800725)

[1. Welcome 4](#_Toc177800726)

[2. Integration 4](#_Toc177800727)

[II. GENERAL REPRESENTATION OF THE COMPANY 5](#_Toc177800728)

[I. Geographical location 5](#_Toc177800729)

[II. History 6](#_Toc177800730)

[III. Mission 6](#_Toc177800731)

[IV. Vision 6](#_Toc177800732)

[V. Activities 6](#_Toc177800733)

[III. ORGANISATION OF THE ENTERPRISE 7](#_Toc177800734)

[1. Administrative Organization of REALIZE 7](#_Toc177800735)

[a. The general management 7](#_Toc177800736)

[b. Human resource department 7](#_Toc177800737)

[c. Communication Department 7](#_Toc177800738)

[d. Department of Financial Affairs 8](#_Toc177800739)

[e. Technical department 8](#_Toc177800740)

[f. Software Engineering Department 8](#_Toc177800741)

[2. Functional organization of REALIZE 9](#_Toc177800742)

[IV. HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY 10](#_Toc177800743)

[1 Hardware Resources 10](#_Toc177800744)

[2 Software Resources 11](#_Toc177800745)

[V. BRIEF PRESENTATION OF THE PROJECT THEME 12](#_Toc177800746)

[CONCLUSION 13](#_Toc177800747)

[INTRODUCTION 16](#_Toc177800748)

[I. PRESENTATION OF THE PROJECT THEME 17](#_Toc177800749)

[II. DESCRIPTION OF THE EXISTING SYSYTEM 18](#_Toc177800750)

[III. LIMITATION OF THE EXISTING SYSTEM 20](#_Toc177800751)

[IV. PROBLEMATIC 21](#_Toc177800752)

[V. PROPOSED SOLUTION 21](#_Toc177800753)

[CONCLUTION 22](#_Toc177800754)

[INTRODUCTION 25](#_Toc177800755)

[I. CONTEXT AND JUSTIFICATION 26](#_Toc177800756)

[A. CONTEXT 26](#_Toc177800757)

[B. JUSTIFICATION 27](#_Toc177800758)

[II. OBJECTIVES OF THE PROJECT 28](#_Toc177800759)

[A. GENERAL OBJECTIVE 28](#_Toc177800760)

[B. SPECIFIC OBJECTIVES 28](#_Toc177800761)

[III. EXPRESSION Of NEEDS 29](#_Toc177800762)

[1. Functional Needs 29](#_Toc177800763)

[2. Non-functional Needs 30](#_Toc177800764)

[IV. PROJECT PLANNING 31](#_Toc177800765)

[a. Chronogram of activities 31](#_Toc177800766)

[b. Gantt Project planning 33](#_Toc177800767)

[IV. ESTIMATED COST OF THE PROJECT 34](#_Toc177800768)

[c. Software Resources 34](#_Toc177800769)

[e. Hardware Resources 35](#_Toc177800770)

[f. Human resources 36](#_Toc177800771)

[g. Global Estimation 36](#_Toc177800772)

[V. CONSTRAINTS 37](#_Toc177800773)

[VI. LIST OF PARTICIPANTS AND DELIVERABLES 37](#_Toc177800774)

[a. List of participants 37](#_Toc177800775)

[b. Deliverables 38](#_Toc177800776)

[CONCLUSION 38](#_Toc177800777)

[INTRODUCTION 41](#_Toc177800778)

[I. METHODOLOGY 42](#_Toc177800779)

[1. COMPARATIVE STUDY OF UML AND MERISE 42](#_Toc177800780)

[a. MERISE 42](#_Toc177800781)

[b. UML 42](#_Toc177800782)

[I. STATIC OR STRUCTURAL DIAGRAMS 42](#_Toc177800783)

[ii. BEHAVIOURAL OR DYNAMIC DIAGRAMS 43](#_Toc177800784)

[2. COMPARATIVE STUDY OF UNIFIED PROCESS 45](#_Toc177800785)

[a. A Unified Process 45](#_Toc177800786)

[b. The Two Track Unified Process (2 TUP) 45](#_Toc177800787)

[II. CHOICE OF THE ANALYSIS METHOD 48](#_Toc177800788)

[III. MODELLING OF THE PROPOSED SOLUTION 48](#_Toc177800789)

[A. Capture Of The Functional Needs 48](#_Toc177800790)

[1. USE CASE DIAGRAM 49](#_Toc177800791)

[a. Definition 49](#_Toc177800792)

[b. Formalism 49](#_Toc177800793)

[c. The Actors of our System 52](#_Toc177800794)

[d. General Use Case Diagram 53](#_Toc177800795)

[e. Place order specific use case diagram 54](#_Toc177800796)

[f. Textual description 54](#_Toc177800797)

[2. COMMUNICATION DIAGRAM 57](#_Toc177800798)

[a. Definition 57](#_Toc177800799)

[b. Formation 57](#_Toc177800800)

[c. Components of a communication Diagram 58](#_Toc177800801)

[d. Authentication communication diagrams 59](#_Toc177800802)

[e. Place command communication diagram 59](#_Toc177800803)

[3. SEQUENCE DIAGRAM 60](#_Toc177800804)

[a. Definition 60](#_Toc177800805)

[b. Formalism 60](#_Toc177800806)

[c. Components of a sequence diagram 61](#_Toc177800807)

[d. Authentiication sequence diagram 63](#_Toc177800808)

[e. Place ordeer sequence diagram 64](#_Toc177800809)

[f. Upload sequence diagram 65](#_Toc177800810)

[g. Rate sequence diagram 66](#_Toc177800811)

[h. Track delivery sequence diagram 67](#_Toc177800812)

[i. ACTIVITY DIAGRAM 68](#_Toc177800813)

[a. Definition 68](#_Toc177800814)

[b. Formalism 68](#_Toc177800815)

[c. Elements of an activity diagram 69](#_Toc177800816)

[d. Get menu activity diagram 71](#_Toc177800817)

[CONCLUTION 72](#_Toc177800818)

[INTRODUCTION 75](#_Toc177800819)

[A. GENERIC DESIGN 76](#_Toc177800820)

[a. Hardware diagram of the system 76](#_Toc177800821)

[b. High Level Architecture of the System 76](#_Toc177800822)

[B. CAPTURE OF TECHNICAL NEEDS 77](#_Toc177800823)

[a. Physical Architecture 77](#_Toc177800824)

[b. Logical Architecture 78](#_Toc177800825)

[C. RELATED UML DIAGRAMS 79](#_Toc177800826)

[1. CLASS DIAGRAM 79](#_Toc177800827)

[a. Definition 79](#_Toc177800828)

[b. Formalism 79](#_Toc177800829)

[c. Components of a class diagram 80](#_Toc177800830)

[d. Easy-eats class diagram 82](#_Toc177800831)

[e. Business Rules 83](#_Toc177800832)

[2. STATE MACHINE DIAGRAM 84](#_Toc177800833)

[a. Definition 84](#_Toc177800834)

[b. Formalism 84](#_Toc177800835)

[c. Components of a state machine diagram 85](#_Toc177800836)

[d. Place command state machine 87](#_Toc177800837)

[3. PACKAGE DIAGRAM 88](#_Toc177800838)

[a. Definition 88](#_Toc177800839)

[b. Formalism 88](#_Toc177800840)

[c. Elements of a package diagram 89](#_Toc177800841)

[d. Easy-eats package diagram 90](#_Toc177800842)

[CONCLUSION 91](#_Toc177800843)

[INTRODUCTION 94](#_Toc177800844)

[1. DEPLOYMENT DIAGRAM 94](#_Toc177800845)

[a. Definition 94](#_Toc177800846)

[b. Formalism 94](#_Toc177800847)

[c. Elements of a deployment diagram 95](#_Toc177800848)

[d. System deployment diagram 96](#_Toc177800849)

[2. COMPONENT DIAGRAM 97](#_Toc177800850)

[a. Definition 97](#_Toc177800851)

[b. Formalism 97](#_Toc177800852)

[c. Elements of a component diagram 98](#_Toc177800853)

[d. System component diagram 100](#_Toc177800854)

[CONCLUSION 101](#_Toc177800855)

[INTRODUCTION 104](#_Toc177800856)

# LIST OF TABLES

[Table 1: Hardware resources of REALIZE (source: MERCURIAL 2024) 10](#_Toc177734881)

[Table 2: Software resources of REALIZE (source : MERCURIAL 2024) 11](#_Toc177734882)

[Table 3: limitations of the existing sysytem 20](#_Toc177734883)

[Table 4: chronogram of activities 31](#_Toc177734884)

[Table 5: software resource(source: matricule 2024) 34](#_Toc177734885)

[Table 6: hardware resource(source: https://www.scribd.com/document/561202966/mercuriale-2024) 35](#_Toc177734886)

[Table 7: human resource (source : Mercurial 2024) 36](#_Toc177734887)

[Table 8: Global Estimation 36](#_Toc177734888)

[Table 9:List of participants 37](#_Toc177734889)

[Table 10: difference between MERISE and UML 44](#_Toc177734890)

[Table 11: elements of a usecase diagram(source: matricule 2024) 50](#_Toc177734891)

[Table 12: actors of the system(source: Matricule 2024) 52](#_Toc177734892)

[Table 13: Textual description of authentication 55](#_Toc177734893)

[Table 14: textual description of use case 56](#_Toc177734894)

[Table 15: components of the communication diagram 58](#_Toc177734895)

[Table 16: components of a sequence diagram 61](#_Toc177734896)

[Table 17: Elements of an activity diagram 69](#_Toc177734897)

[Table 18: components of a class diagram 80](#_Toc177734898)

[Table 19: Class diagram of the system 82](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734899)

[Table 20: components of a state machine diagram 85](#_Toc177734900)

[Table 21: Elements of a package diagram 89](#_Toc177734901)

[Table 22: Elements of the deployment diagram 95](#_Toc177734902)

[Table 23: Elements of a component diagram 98](#_Toc177734903)

# LIST OF FIGURES

[Figure 1: realize geolocalisation 5](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734839)

[Figure 2:organisation of realize 9](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734840)

[Figure 3: time spent getting food 18](#_Toc177734841)

[Figure 4: time spent cooking food 19](#_Toc177734842)

[Figure 5: gantt planning 33](#_Toc177734843)

[Figure 6: Y-shaped 2TUP 46](#_Toc177734844)

[Figure 7: formalism of a usecase diagram 49](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734845)

[Figure 8: the general usecase diagraam 53](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734846)

[Figure 9: place order specific use case diagram 54](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734847)

[Figure 10: formalism of the communication diagram 57](#_Toc177734848)

[Figure 11: place command communication diagram 59](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734849)

[Figure 12: formalism of the sequence diagram 60](#_Toc177734850)

[Figure 13: Authentication sequence diagram 63](#_Toc177734851)

[Figure 14: place command sequence diagram 64](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734852)

[Figure 15: Upload sequence diagram 65](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734853)

[Figure 16: rate sequence diagram 66](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734854)

[Figure 17: track delivery sequence diagram 67](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734855)

[Figure 18: formalism of the activity diagram 68](#_Toc177734856)

[Figure 19: Get menu activity diagram 71](#_Toc177734857)

[Figure 20: Get menu activity diagrm 71](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734858)

[Figure 21: hardware design of the system 76](#_Toc177734859)

[Figure 22: Physical architecture 77](#_Toc177734860)

[Figure 23: Logical architecture 78](#_Toc177734861)

[Figure 24: formalism of the class diagram 79](#_Toc177734862)

[Figure 25: formalism of the state machine diagram 84](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734863)

[Figure 26: place command state machine diagram 87](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734864)

[Figure 27: formalism of the package diagram 88](#_Toc177734865)

[Figure 28: system package diagram 90](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734866)

[Figure 29: formalism of the deployment diagram 94](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734867)

[Figure 30: System deployment diagram 96](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734868)

[Figure 31: Formalism of the component diagram 97](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734869)

[Figure 32: system component diagram 100](file:///C:\Users\USER\Desktop\REPORT\report%20project.docx#_Toc177734870)

# GLOSSARY

* 2TUP: Two Track Unified Process.
* AICS: African Institute of Computer Sciences.
* APK: Android Package Kit.
* DBMS : Database Management System
* GUI: Graphical User Interface
* IDE: Integrated Development environment
* IPA: iOS App Store Package.
* JSON: JavaScript Object Notation
* MVC: Model View Controller.
* UML: Unified Modelling Language

# ABSTRACT

In today's fast-paced world, convenience is key, and access to quality food without the need for physical travel has become increasingly important. The global shift towards on-demand services has sparked a revolution in the food industry. With busy work schedules, traffic congestion, and lifestyle changes, people increasingly rely on online platforms for fast, reliable, and efficient access to meals. Whether it's ordering a quick lunch during a hectic workday, hosting a spontaneous dinner party, or trying out new dishes from distant restaurants, the convenience of food delivery has become an indispensable part of modern life. **EASY-EATS** is a mobile application developed to bridge the gap between businesses, such as restaurants and food vendors, and their customers. The motivation for developing EASY-EATS stems from the challenges faced by both customers and businesses. Customers often have limited time to visit restaurants, and some may live in areas with limited access to diverse food options. On the business side, restaurants often face difficulties reaching a wider customer base, particularly those living farther away. These challenges inspired us to create an easy-to-use platform that connects both parties seamlessly. This application allows users to conveniently browse menus, place food orders, and have their meals delivered directly to their doorsteps. In order to accomplish this project, we used UML (Unified Modelling Language) as our modelling language coupled with 2TUP (2 Track Unified Process) in order to form a method.

**Keywords:**

* Food industry
* Easy-eats
* Food order
* Delivery
* Businesses
* Customers

# RESUME

Dans le monde d'aujourd'hui, où tout va à toute vitesse, la commodité est essentielle, et l'accès à une alimentation de qualité sans avoir à se déplacer est devenu de plus en plus important. Le virage mondial vers les services à la demande a déclenché une révolution dans l'industrie alimentaire. Avec des horaires de travail chargés, des embouteillages et des changements de mode de vie, les gens se tournent de plus en plus vers les plateformes en ligne pour un accès rapide, fiable et efficace aux repas. Qu'il s'agisse de commander un déjeuner rapide pendant une journée de travail bien remplie, d'organiser un dîner improvisé ou de découvrir de nouveaux plats provenant de restaurants éloignés, la commodité de la livraison de repas est devenue une partie indispensable de la vie moderne. **EASY-EATS** est une application mobile développée pour combler l'écart entre les entreprises, telles que les restaurants et les vendeurs de nourriture, et leurs clients. La motivation pour développer EASY-EATS découle des défis auxquels sont confrontés à la fois les clients et les entreprises. Les clients ont souvent peu de temps pour se rendre dans les restaurants, et certains peuvent vivre dans des zones où l'accès à des options alimentaires variées est limité. Du côté des entreprises, les restaurants rencontrent souvent des difficultés à atteindre une clientèle plus large, notamment ceux vivant à distance. Ces défis nous ont inspirés à créer une plateforme facile à utiliser qui connecte les deux parties de manière fluide. Cette application permet aux utilisateurs de parcourir les menus en toute simplicité, de passer des commandes de repas et de se faire livrer directement à leur domicile. Pour mener à bien ce projet, nous avons utilisé UML (Unified Modelling Language) comme langage de modélisation, associé à 2TUP (2 Track Unified Process) pour former une méthode.

**Mots-clés :**

* Industrie alimentaire
* Easy-eats
* Commande de repas
* Livraison
* Entreprises
* Clients

# GENERAL INTRODUCTION

Technology is advancing at an incredible rate, spanning almost every sector from business to agriculture and other sectors. Due to the huge technological advancement, we now live in a fast-paced society where everything is gradually being digitalized making life easier. As a developing country, most businesses in Cameroon strive to make advantage of the digital economy to grow their businesses thereby boosting the economy as well. To achieve such a goal, enterprises need qualified personnel having skills in computer sciences and related fields. Second year students in AICS Cameroon, are required to carry out a three (3) months’ academic internship at an enterprise to put into practice the knowledge gotten from school in the professional milieu. For this reason, we applied for an internship at Realize, which specializes in providing IT solutions. While there, we were attributed the theme “ONLINE FOOD COMMAND DELIVERY PLATFORM”. We divided this report into eight (8) parts which are as follows:

1. **Insertion Document:** In this book, we shall present the company in which we spent our internship period and the integration of interns.
2. **Existing System:** Here, we shall present the already present system in, that is the one used for consultation and follow-up purposes.
3. **Place Specification Book:** In this book, we specify the needs of the user taking into considerations the time and cost of the project.
4. **Analysis Document:** Here, we shall present the analysis method chosen together with the presentation of all the diagrams used for the analysis of this project.
5. **Conception phase:** This presents the generic and detailed conception of the project to bring out real world constituents.
6. **The Realization phase:** This phase will permit us to visualize the implementation process of the solution.
7. **Test of functionalities:** In this phase, we shall present to you the different functionalities or modules of our application and how they work.
8. **The User Guide:** This elaborates on all conditions necessary to use the application and how to use it.

# PART ONE:

# INSERTION PHASE

Preamble

This phase presents the details of how we were integrated in the host company, the company presentation and organization.

Content overview

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF THE COMPANY
3. ORGANISATION OF THE XOMPANY
4. HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY
5. BRIEF PRESENTATION OF THE PROJECT THEME

CONCLUSION

# INTRODUCTION

The insertion phase is a period (generally of 02 weeks) reserve for the different interns to discover and to familiarize with the working environment. Here, we got to know about the staffs, the different hardware and software resources used, the different departments which constitute the enterprise, how the company function both internally and externally and we were introduced to our work space. During this period, we were also attributed an internship master often called professional supervisor and a theme. We also had a time to discuss amongst us interns on topics like what we love doing most, what we dislike, our believes and experiences. We shared about different realizations and failures in life.

## WELCOME AND INTEGRATION

### Welcome

We arrived at REALIZE on Monday July 2024 at 8:00am. We were welcomed, by **Mr TANUE Monette BUKWATI** the enterprise frontend developer, who introduced us to our work space and his collaborators. Later on, he gave us the task to do some findings on the possible internship themes that we can work on.

### Integration

A working day at REALIZE start from 07:30am to 04:30pm. Our tasks in the company generally follow a predefine routine. Every day we were to write five pages of our internship report, to complete the assigned task for the day. It was also an opportunity to present our difficulties and challenges to everyone and receive directives.

## II. GENERAL REPRESENTATION OF THE COMPANY

### Geographical location

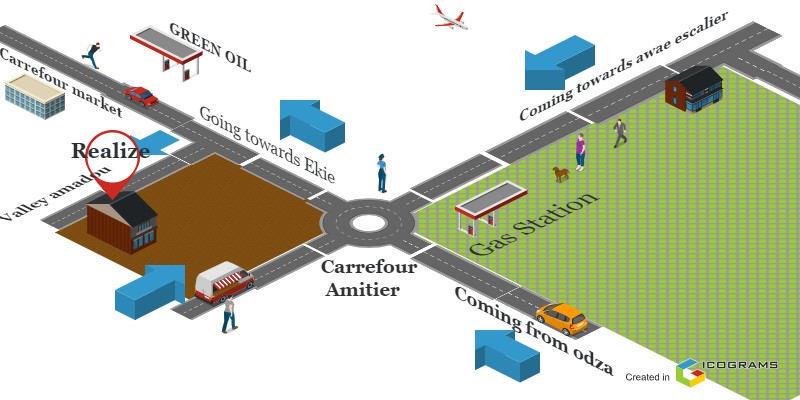


Figure 1: realize geolocalisation

### History

Realize is a Cameroon based non-governmental tech start-up institute founded by **NDELOGAKEH DANIEL** in 2022 which proposes IT solutions and empowers the use on new technologies in Cameroon.

### Mission

The mission of Realize mainly relates to the empowerment and the perpetuation of Technology use. These missions include:

* Design and realize mobile-oriented software for companies and individuals.
* Assist new born companies for quick growth using computer sciences.
* 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.

### Vision

At Realize, we believe that the true potential of every organization and individual can be unlocked through the harmonious fusion of analysis, development, realization, and data analytics. Our vision propels us forward, guiding us to make a lasting impact on the digital landscape and shape a future where innovation knows no bounds.

### Activities

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

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

## ORGANISATION OF THE ENTERPRISE

## Administrative Organization of REALIZE

REALIZE is administratively organized as follows;

### **The general management**

This is the highest level of the company, which takes care of the following:

* Proper functioning of each department.
* Define project strategies.
* Provide leadership and guidance to the company’s employees.
* Makes critical decision that could affect the company’s operations or reputation ➢ Severe as a point of contact for key stakeholders such as investors etc.

### **Human resource department**

This department is in charge of the following:

* Recruitment and Hiring.
* Acts as a liaison between employees and management;
* Manages employee compensation and benefits programs;
* Develops and implements company policies and procedures, ensuring compliance with employment laws and regulations.
* Manages administrative tasks related to employee records, data management, and HR systems.

### c. **Communication Department**

This department is in charge of the following:

* Handles public relations activities, which involve managing the company’s reputation and image.
* Responsible for crafting and delivering messages to external stakeholders such as customers etc.
* It fosters effective communication within the company.
* They create and manage content across different platforms and channels.

### d. **Department of Financial Affairs**

This department is in charge of the following:

* Responsible for developing and managing the company’s financial plans and budgets.
* Prepare and present accurate and timely financial reports to management, stakeholders, and regulatory authorities.
* Responsible for managing costs and expenses within the organization.
* Ensures compliance with tax laws and regulations. Etc.

### **Technical department**

This department is in charge of the following:

* Managing the company’s technological infrastructure, including networks and hardware.
* Play a role in managing and maintaining the company’s data.
* Implementing and maintaining cybersecurity measures to protect the company’s digital assets from potential threats.
* Interacts with technology vendors and manages relationships with external service providers. Etc.

### **Software Engineering Department**

This department is in charge of the following:

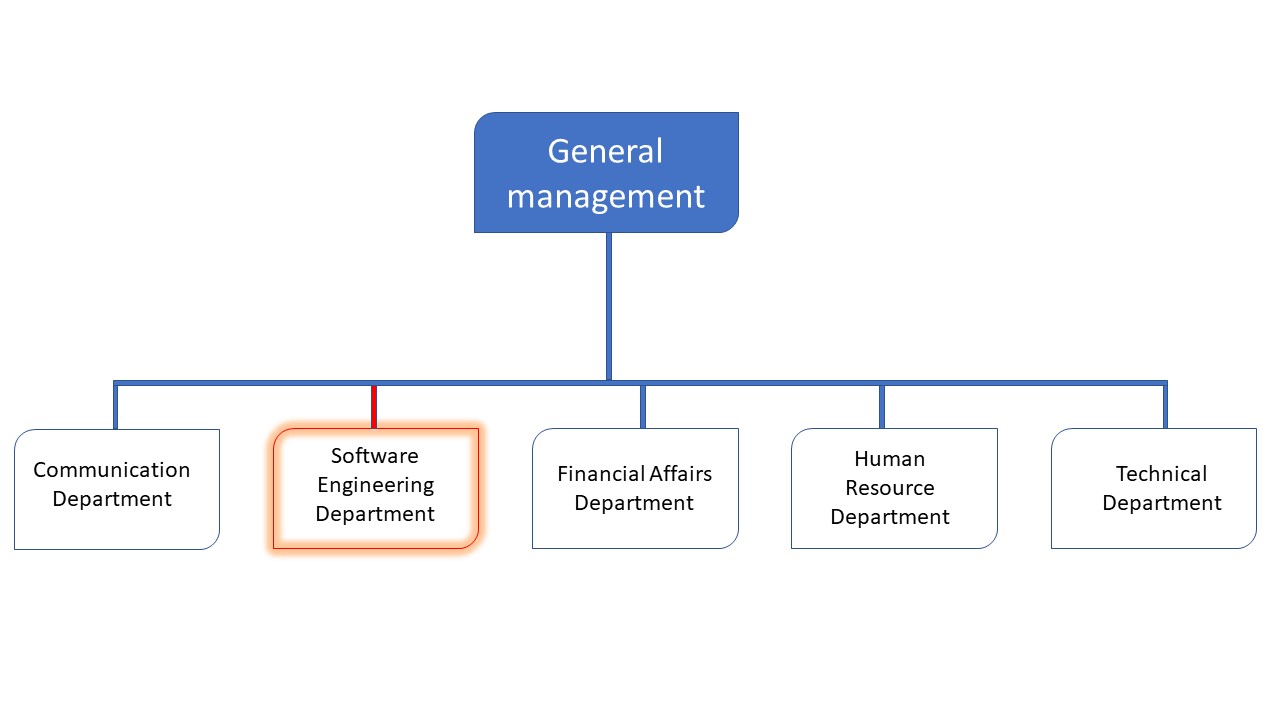
* Primarily responsible for developing software applications and systems.
* Responsible for maintaining and supporting software applications throughout their lifecycle.

Evaluation and realization of projects etc.

## Functional organization of REALIZE

The functional branch of REALIZE is organized as follows;

Figure 2:organisation of realize



## HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY

### Hardware Resources

Table 1: Hardware resources of REALIZE (source: MERCURIAL 2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Designation | Quantity | Characteristics | Observation |
| 1 | MacBook Pro | 1 | APPLE | Good |
| 2 | Modem | 1 | CAMTEL | Good |
| 3 | LCD Screen | 1 | HP | Excellent |
| 4 | Training equipment and office furniture | / | / | / |

### Software Resources

As an enterprise specialized in IT, REALIZE also comprises of software resources such as;

Table 2: Software resources of REALIZE (source : MERCURIAL 2024)

|  |  |
| --- | --- |
| Designation | Software |
| Operation system | Windows 10, Mac OS, parrot Linux |
| Design Tools | Photoshop |
| Integrated development environment (IDE) | Visual studio code, |
| Text editor | Sublime text, notepad++ |
| Database management system (DBMS) | MongoDB, PostgreSQL, MySQL |
| Web browser | Google chrome, Microsoft edge |
| Document editor | Microsoft office word |
| Presentation | Microsoft office PowerPoint |

## BRIEF PRESENTATION OF THE PROJECT THEME

During our insertion phase in REALIZE, we were asked to look for themes which solves a specific problem and something innovative. The professional supervisor took some time to look into what we proposed as themes and he also proposed some themes ideas. Finally, we had to go for **“REAL-TIME FOOD COMMAD, DELIVERY PLATFORM”**. This platform bridges the gap between businesses, such as restaurants and food vendors, and their customers willing to buy but don’t know where and or do not want to displace themselves.

## CONCLUSION

The insertion phase was a very educative one, we got to know more about the history of the company, how it all started, how it is structured and one of its main realizations as a young company is **oneLink**: which is an application which helps individuals from various field to exploit readymade template for their presentations, then we discovered that it’s a very serious and goal-oriented company. We got to Know the staff and other interns, in fact we felt at home. Finally, we decided to work on the theme **“REAL-TIME FOOD COMMAND, DELIVERY PLATFORM”.** Now that we have chosen the theme we will work on; we can move on to the specification book of our report where we will identify the objectives and requirements for our project.

PART TWO: EXISTING SYSTEM

Preamble

The existing system is a document that provides a view of the system currently in place, that is how it carries out its different activities, also it provides a deep understanding of this system associated to the various limitations, the problems that result from these and the solution we propose.

Content overview

INTRODUCTION

1. PRESENTATION OF THE THEME
2. STUDY OF THE EXISTING SYSTEM
3. CRITICISMS OF THE EXISTING SYSTEM
4. PROBLEMATICS
5. PROPOSED SOLUTION

CONCLUSION

## INTRODUCTION

As Engineers our role in the society is problem-solving which is targeted at making life easier for everyone and in order to bring forth a solution, we ought to ask ourselves some questions to know why the problem exist in the first place and how it is being dealt with at the time. The existing system refers to the system put in place to carry out the work done in the field on which our theme is based. Understanding this system is a great step in solving the problems that we might identify. It’s of great importance that we take into consideration this system before proposing a suitable solution that will ameliorate or replace the current one.

## PRESENTATION OF THE PROJECT THEME

Our theme is “ONLINE FOOD, COMMAND DELIVERY PLATFORM”. This application (Easy-Eats) bridges the gap between business personnel or restaurants and consumers, particularly in scenarios where the business is willing to deliver food to the consumer in a specific location range. This platform does not only connect consumers with businesses but also allows customers to rate businesses, order from a variety of options, and enables businesses to track customer locations in real-time until delivery is completed. This is advantageous to both parties. The popular slogan, “the reason every human works is because of food”, is more of a fact as days goes by. Considering this, getting food should be done in comfortable conditions.

## DESCRIPTION OF THE EXISTING SYSYTEM

In our world today, a 24hour day is congested with so many activities such that we have less time to do the neglectable yet most relevant. Good Health is very cumbersome to maintain since it deals with activity, mentality and most importantly food. Consider a software engineer who works in a software development company in Amadou. He leaves home at 7:00am, often without having breakfast. At work, he gets a 30-minute break at 12:30., but there are no nearby restaurants to provide a substantial meal. He only manages to get a complete meal after getting home at 5:30 pm. When he decides to get food, he takes approximately 30 mins a day to think of what to eat and walk to where to get it.

Base on this statistics the following time is spent in 2years

Figure 3: time spent getting food

From the data above, we noticed that although this time spent seems significantly minor, when accumulated for a year or two, 10,080 minutes a year or 20,160 in 2 years is wasted. In some scenarios, he tries to verify his WhatsApp contact status for any avalable seller.

What about individuals who have to cook on daily bases, they need to go to the market or groocery store before cooking. The approximate time per day is 1 hour, considering two years,

Figure 4: time spent cooking food

Based on the figure above, we notice that approximately 720 hours is spent on cooking in 2 years, this is challenging exspecially for males. Considering both figures, the time spent is very significant.

On the other hand, there are cooks and small businesses or restaurant owners in our community who are eager to sell their food. However, they often struggle to reach customers effectively and suffer losses from unsold leftovers. They do this at random and mostly do not make maximum sales. These situations are far from ideal, which led us to develop Easy-Eats.

## LIMITATION OF THE EXISTING SYSTEM

At the end of our observation and collection of information concerning individuals in terms of getting food, we noticed that the system has downsides that should be considered as minor as though they seems. However, we thought it wise to suggest some enumerations to this system.

|  |  |  |
| --- | --- | --- |
| LIMITATIONS CONSEQUENCE | | PROPOSE SOLUTION |
| Individuals who need food but do not want to displace themselves are limited the option of getting food in location | -Starvation or eating unhealthy and getting health issues. | A system that connects the individuals to personnels willing to deliver |
| Business personnel or restaurants are limited to customers, most exspecially personnel willing to displace themselves | -Food spoilage and decay  -Loss of profit | An online system which increases their visibility to customers |
| Individuals spend A lot of time is spent on deciding what to get. | -less time to complete daily tasks | A system in which user can choose from a vareity, and get it at location. |
| In areas with near by restaurants, individuals eat unhealthy fast food to quench hunger. | -Unhealthy food reduces work performance  -Affects daily mood. | A system that connects individuals to personnels close areas who can bring the food at their disposal. |

Table 3: limitations of the existing sysytem

## PROBLEMATIC

Since food is a main neccesity for survival, it is very important that it is gotten at favorable conditions and in comfort. Like the old saying ‘’health is wealth‘’ maintaining health should be a priority. To see this through, we ask ourselves the question **HOW CAN WE CONNECT BUSINESSES WITH LIMITED CUSTOMER VISIBILITY TO CUSTOMERS WILLING TO BUY ?** We will try to offer a solution through our application.

## PROPOSED SOLUTION

After our study and criticism of the existing situation, we propose to design a mobile application that will allow the individuals (business personnels and customers) of Cameroon to:

**Business personnels**

* Have a vareity of customers.
* Be aware on the real time location for delivery.
* See ratings on their meal.

**Customer**

* Order from a vareity of meals.
* Get food at the comfort of their homes.
* Minimize time spent to purchase or cook food.
* See other customer review and rates.

## CONCLUTION

Having reached the end of this part, we studied the existing system by questioning different individuals in different scenarios which enabled us to come out with the limitations of the system that leads to the consequences and we proposed solutions to the various limitations. Finally, we proposed our solution which is a mobile application. This step is necessary because we need to understand how the system put in place currently works before we can facilitate or ameliorate the processes been carried out in this system.

PART THREE: THE SPECIFICATION BOOK

Preamble

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

Content overview

INTRODUCTION

1. CONTEXT AND JUSTIFICATION
2. OBJECTIVES OF THE PROJECT
3. EXPRESSION OF NEEDS
4. EXTIMATED COST OF THE PROJECT
5. PROJECT PLANNING
6. LIST OF PARTICIPANTS AND DELIVERABLES

CONCLUSION

## INTRODUCTION

The specification book of our report helps us provide details about our theme to improve our understanding of it and increase the likelihood of it succeeding. To delimitate the scope of our project, we will specify the context of our theme. From the context, we will list the problems we have decided to address throughout the project. After presenting our solution, we will talk about the objective we have set 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

As the years progress, the desire for doorstep delivery services has significantly increased, reflecting a broader shift in consumer behavior. People increasingly prefer the convenience and comfort of receiving goods and services directly at their current location, often their home. This trend, which began to gain momentum over the past decade, has been propelled by the rise of online shopping and the rapid evolution of e-commerce.

A study conducted online, based on website data, reveals the following trend in the percentage of people demanding doorstep delivery from 2014 to 2024:

* **2014:** 20% - Online shopping was growing, but only a small percentage of the global population regularly demanded doorstep delivery.
* **2015:** 25% - The popularity of e-commerce giants like Amazon led more people to expect home delivery.
* **2016:** 30% - Mobile shopping started contributing significantly to e-commerce, increasing demand for doorstep delivery.
* **2017:** 35% - Same-day delivery options became more common, boosting demand further.
* **2018:** 40% - Continued growth as more markets and demographics gained access to online shopping and delivery services.
* **2019:** 45% - Expansion of logistics and delivery infrastructure globally.
* **2020:** 60% - The COVID-19 pandemic significantly accelerated the demand for doorstep delivery.
* **2021:** 70% - The trend persisted post-pandemic as consumers became accustomed to home delivery.
* **2022:** 75% - Further innovations in delivery services and faster shipping times fueled demand.
* **2023:** 80% - Doorstep delivery became normalized in everyday life.
* **2024:** 85% - By 2024, doorstep delivery is a standard expectation for the vast majority of online shoppers.

This research clearly demonstrates a substantial increase in the demand for doorstep delivery over the past years.

### JUSTIFICATION

The food service industry is experiencing a significant transformation driven by the rising demand for convenience and efficiency in food delivery. With the growth of e-commerce and online shopping, consumers have come to expect the same level of convenience when ordering food. This shift in consumer behavior has created a pressing need for businesses to adapt and offer delivery services that meet these expectations. The traditional methods of ordering food, such as dining in or picking up orders in person, are becoming less appealing to consumers who prioritize convenience. The COVID-19 pandemic further accelerated this trend, making home delivery a preferred option for many. However, the food industry faces challenges in meeting this demand effectively, including managing orders, ensuring timely deliveries, and maintaining customer satisfaction.

* **For Customers**: Easy-Eats offers a seamless and user-friendly experience, allowing customers to order food from their favorite restaurants without leaving the comfort of their home. The app’s rating system helps customers find the best menus, ensuring a satisfying dining experience.
* **For Businesses**: The app supports businesses by expanding their reach, enabling them to cater to a wider audience. Easy-Eats also streamlines the order management process, reducing errors and improving efficiency. By offering delivery services, businesses can increase their revenue and stay competitive in the market.

## OBJECTIVES OF THE PROJECT

### GENERAL OBJECTIVE

The main objective of this project is to connect food businesses and consumers, enabling the business to have a larger customer range and the customers to get their meals at any location.

### SPECIFIC OBJECTIVES

* The customer will be able to place a command from a vareity of meals avalable from different businesses.
* The customer will be able to rate and give review on a meal and view other reviews by different customers.
* The business will be able to upload more than one meal(image) to their menu and have visibility.
* The business will be able to recieve notifications on a command placed on their meal.
* The business will be able to see customer location in real time until delivery is made.

## EXPRESSION Of NEEDS

## Functional Needs

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

❖ **The administrator (admin) should be able to:**

* **Account Creation**: The restaurant or business and customers or consumers is responsible for creating accounts. This allows for efficient management of orders and deliveries.
* **Account Management**: Users can manage their accounts by updating their information, such as contact details, delivery preferences, and payment methods.
* **User Authentication**: The application should allow users (both customers and businesses) to register or log in securely. It also provides a mechanism for password recovery and requires account verification to ensure the security and identity of the user.
* **Menu Management**: Businesses can upload, update, and delete their menus, including adding new items, descriptions, prices, and images.
* **Order Management**: The application facilitates the management of incoming orders, enabling businesses to process, and track orders. Customers can view their order status, make modifications if needed, and receive notifications about their order’s progress.
* **Customer Feedback and Ratings**: Customers can rate and provide feedback on their orders and overall experience. This feedback is visible to businesses, helping them improve their services and build credibility.

## Non-functional Needs

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

Performance, Scalability and Security

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

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

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

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

* The application should provide a strong security mechanism to reassure user’s that they can trust it for their sensitive information.
* Some of the security principles include reassuring confidentiality, accountability, Integrity, Authentication.

## PROJECT PLANNING

## Chronogram of activities

Table 4: chronogram of activities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE OBJECTIVE | | OUTPUT | DURATION | PERIOD |
| INSERTION | Collection of information on the enterprise | Insertion book | 2 weeks | 03rd July to  14th July |
| EXISTING SYSTEM | Study of the existing system | Existing System | 5 days | 17th July to  21st July |
| SPECIFICATION BOOK | Specification of the user needs | Specification Book | 5 days | 24th July to  28th July |
| ANALYSIS | Capture of needs Use case and textual description  Modelling | Analysis Book | 2 weeks | 31st July to 11th August |
| CONCEPTION | Preliminary conception and  Detailed conception | Conception book | 2 weeks | 14th August to 25th  August |
| REALIZATION | Implementation Unitary test  Integration Test  Development,  Deployment,  Component diagrams | Realization book | 3 weeks | 28th August to 15th  September |
| TEST OF FUNCTIONALITIES | Testing of the software and debugging | Test of functionalities | 5 days | 18th  September to 22nd  September |
| INSTALLATION AND USER GUIDE | Documenting software | User Guide | 5 days | 25th  September to 29th  September |

## Gantt Project planning

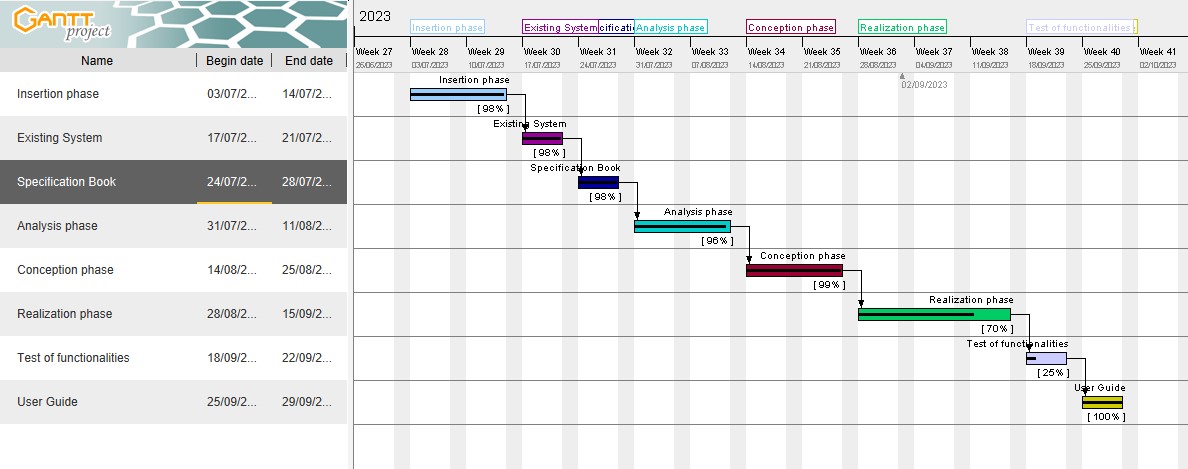


Figure 5: gantt planning

## ESTIMATED COST OF THE PROJECT

## Software Resources

Table 5: software resource(source: matricule 2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES DESIGNATION | | USAGE | QUANTITY | UNIT COST  (FCFA) |
| Formatting | Microsoft Office  365 | Writing of the report | 1 | **47 998** |
| Cloud Storage | Google Cloud | Saving our report | 1 | **Freemium** |
| Web browser | Google Chrome | View web pages | 1 | **Freeware** |
| Code Editor | Visual Studio  Code | For writing the code of the application | 1 | **sFreeware** |
| Project planning | Gantt Project | For building a  Gantt chart | 1 | **Freemium** |
| Illustrating tool | Ichogram | For building a geographical  location of the enterprise | 1 | **Freemium** |
| UML  Analysis | Visual paradigm | For drawing UML diagrams | 1 | **Freemium** |
| Testing Mobile App | Expo Go | For testing the mobile application | 1 | **Freemium** |
| TOTAL 1 |  |  | **8** | * + 1. **98** |

## Hardware Resources

Table 6: hardware resource(source: https://www.scribd.com/document/561202966/mercuriale-2024)

|  |  |  |  |
| --- | --- | --- | --- |
| RESOURCES | HARDWARE | Quantity | UNIT COST  (FCFA) |
| Computer | LAPTOP HP ELITEBOOK  8540P CORE I5, DISQUE DUR 500 GO, RAM 4 GO, ECRAN LARGE 15" HD | **1** | **402 500** |
| Printer | Printers | **1** | **546 250** |
| Network | Local network installation | **1** | **300 000** |
| Removable Disk | Removable Disk | **1** | **11 555** |
| Smart phone | Smart phone | **2** | **150 000** |
| TOTAL 2 |  | **6** | **1 410 305** |

## Human resources

Table 7: human resource (source : Mercurial 2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ROLE | NUMBER OF DAYS | QUANTITY | COST PER DAY | TOTAL PRICE (FCFA) |
| Project Manager | **90** | **01** | **30 000** | **2 700 000** |
| Analyst | **21** | **01** | **25 000** | **525 000** |
| UI/UX Designer | **07** | **01** | **20 000** | **140 000** |
| Programmer | **30** | **01** | **15 000** | **450 000** |
| Tester | **14** | **02** | **10 000** | **140 000** |
| Margin error | **/** | **/** | **/** | **1,475,000** |
| TOTAL 3 |  |  |  | 1. **840 000** |

## Global Estimation

Table 8: Global Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| OVERALL TOTAL TOTAL 1(FCFA) TOTAL 2(FCFA) TOTAL 3(FCFA) (FCFA) | | | |
| 47 998 | **1 410 305** | **4 840 000** | **6 298 303** |
| SIX MILLION TWO HUNDRED AND NINETY-EIGHT THOUSAND THREE  HUNDRED AND THREE | | | |

## CONSTRAINTS

1. **Technical constraint**

For the development of our system, we have sufficiently robust tools to guarantee a minimum of security, extensibility and excellent scalability. Moreover, the programming phase will have to follow all the technical standards for a better performance in a reduced execution time, this is why the choice of the development technologies is crucial.

1. **Time Constraint**

The project will be realized in 13 weeks (3 months) starting from the beginning date, July 3rd, 2023.

1. **Cost constraint**

The realization of our project will require expenditures in human resources, material and software a total cost of 6 298 303 FCFA.

## LIST OF PARTICIPANTS AND DELIVERABLES

## List of participants

Table 9:List of participants

|  |  |  |
| --- | --- | --- |
| NAME | FUNCTION | ROLE |
| Mr. TANUE Monette | Follows up interns at the company level | Professional supervisor |
| Mr. AGBOR Anderson | Follows up student at the academic level | Academic Supervisor |
| KAMENI SEPDEU Ange  Chris | AICS Student Intern | Student at AICS |

## Deliverables

* + 1. In project management, any component materializing the result of a realization service is called a deliverable. In the case of our project, the deliverables are: A report composed of the following document
    - The application.
    - The user guide.
    - The powerpoint.

## CONCLUSION

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

PART FOUR: THE ANALYSIS BOOK

Preamble

The main objective of the analysis phase is to capture the user’s need, the delimitation of the field of study and to have a clear understanding of the system in study. To achieve this, we will use UML (Unified Modelling Language) with the 2TUP (2 Track Unified Process) as method applied to UML to analyze the system. We will start by doing a comparison between UML and MERISE, of various unified processes and lastly, we will present the modelling of the solution we propose (Our software).

Content overview

INTRODUCTION

1. METHODOLOGY
2. COMPARATIVE STUDY OF UML
3. COMPARATIVE STUDY OF UNIFIED PROCESS
4. CHOICE OF THE ANALYSIS METHOD
5. MODELLING OF THE PROPOSED SOLUTION

CONCLUSION

## INTRODUCTION

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

## METHODOLOGY

## COMPARATIVE STUDY OF UML AND MERISE

### MERISE

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

### UML

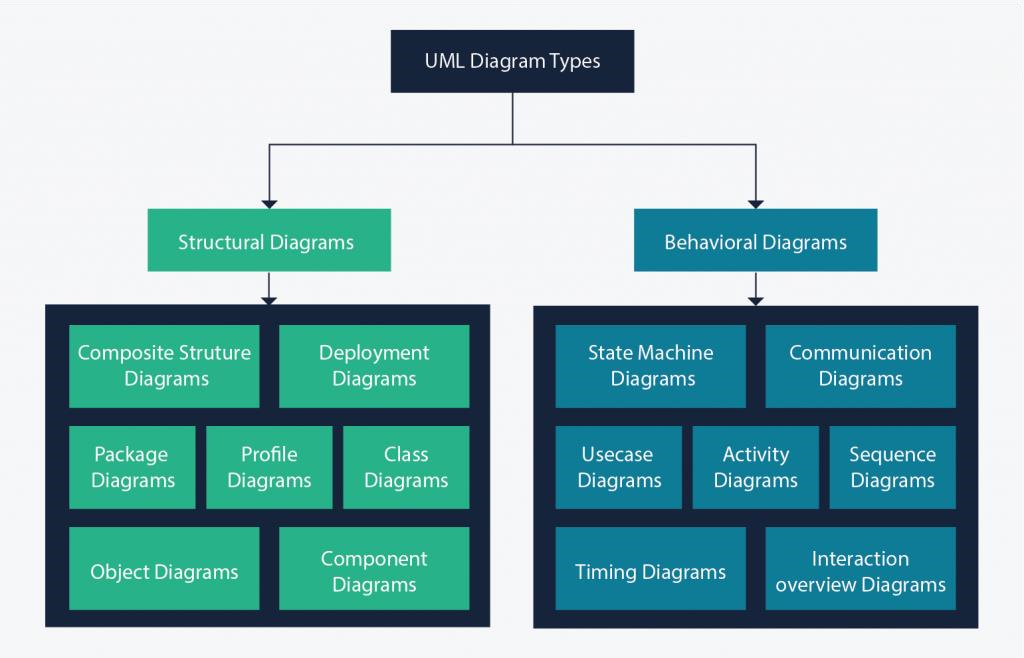
UML (Unified Modelling Language) is a standard notation for the modelling of real world objects as a first step in developing an object-oriented design methodology. Its notation is derived from and unifies the notations of three object-oriented design and analysis methodologies: Grady Booch's methodology for describing a set of objects and their relationships, James Rumbaugh's Object-Modelling Technique (OMT), Ivar Jacobson's approach which includes a use case methodology. Other ideas also contributed to UML, which was the result of a work effort by Booch, Rumbaugh, Jacobson, and others to combine their ideas, working under the sponsorship of Rational Software. UML captures information about the static and dynamic view of a system. UML 2.5 comprises of 14 diagrams which represent the different views of a system. The 14 diagrams can be subdivided into two, Static or structural and Dynamic diagrams. These diagrams include;

#### STATIC OR STRUCTURAL DIAGRAMS

* Class diagram;
* Object diagram;
* Component diagram;
* Deployment diagram;
* Composite Structure diagram;
* Package diagram;
* Profile diagram;

#### BEHAVIOURAL OR DYNAMIC DIAGRAMS

* Use case diagrams;
* Activity diagram;
* State machine diagram;
* Sequence diagram;
* Communication diagram;
* Global Interaction diagram;
* Timing diagram ;



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

Table 10: difference between MERISE and UML

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

## COMPARATIVE STUDY OF UNIFIED PROCESS

### A Unified Process

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

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

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

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

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

### The Two Track Unified Process (2 TUP)

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. The diagram below illustrates the branches of 2TUP.

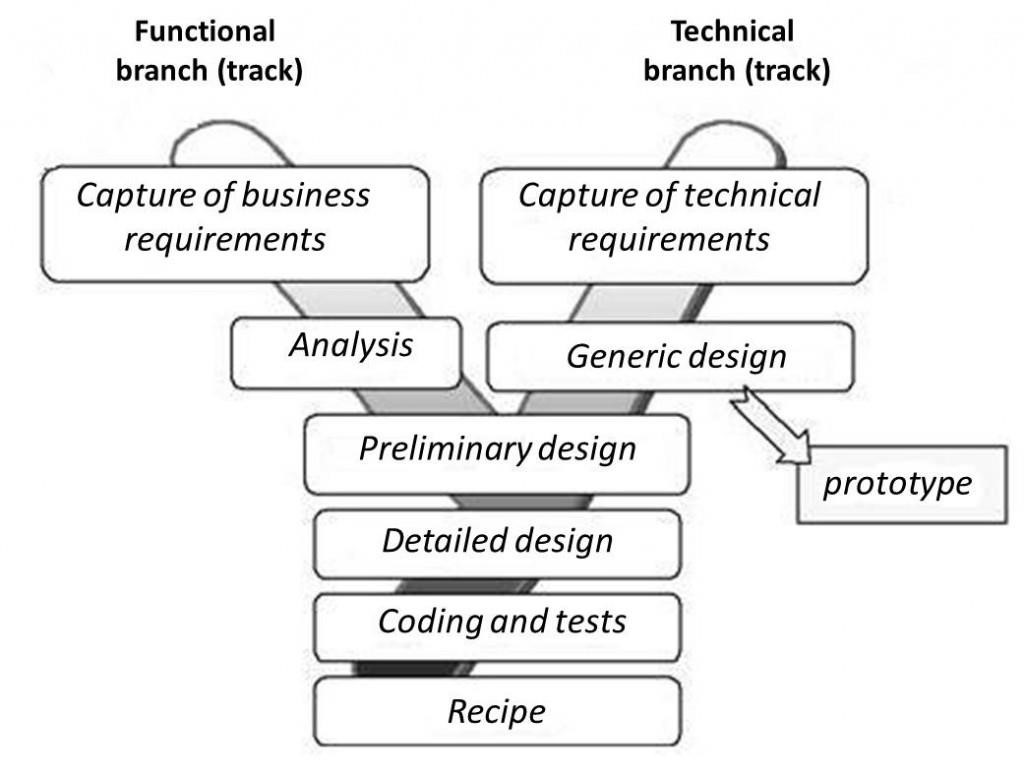


Figure 6: Y-shaped 2TUP

* The left branch (functional branch)

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

* The right branch (Technical branch)

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

* + The middle branch (Realization or Implementation branch)

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

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

* Component Diagram;
* Deployment Diagram;
* Package Diagram;
* Composite Structure Diagram;

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

* Class;
* Object;
* Sequence;
* Timing Diagram;

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

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

## CHOICE OF THE ANALYSIS METHOD

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

UML is the current standard for programming in an object-oriented language. For this reason, it is widely understood and well known making it easy for a new programmer to join the project and be productive from the very first day.

UML diagrams allow teams to virtualize how a project is or will be working, and they can be used in any field, not just software engineering. The diagrams will allow teams to virtualize together how a system, or a process will work or did work. It can provide new ideas for how teams have to collaborate to achieve the goal of the workflow process.

2TUP is centered around the creation and maintenance of a model, rather than the production of mountain documents.

2TUP is user oriented as it permits the development of software that responds to the needs of the users through the study of the user needs.

2TUP is iterative and incremental, hence it enables the project team to produce refined amelioration if necessary and easily integrate it in the already existing system.

2TUP by permitting the project team identify and test the key functionalities of the system limits the risk related to building the system.

## MODELLING OF THE PROPOSED SOLUTION

## Capture Of The Functional Needs

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

### USE CASE DIAGRAM

### Definition

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

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

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

### Formalism

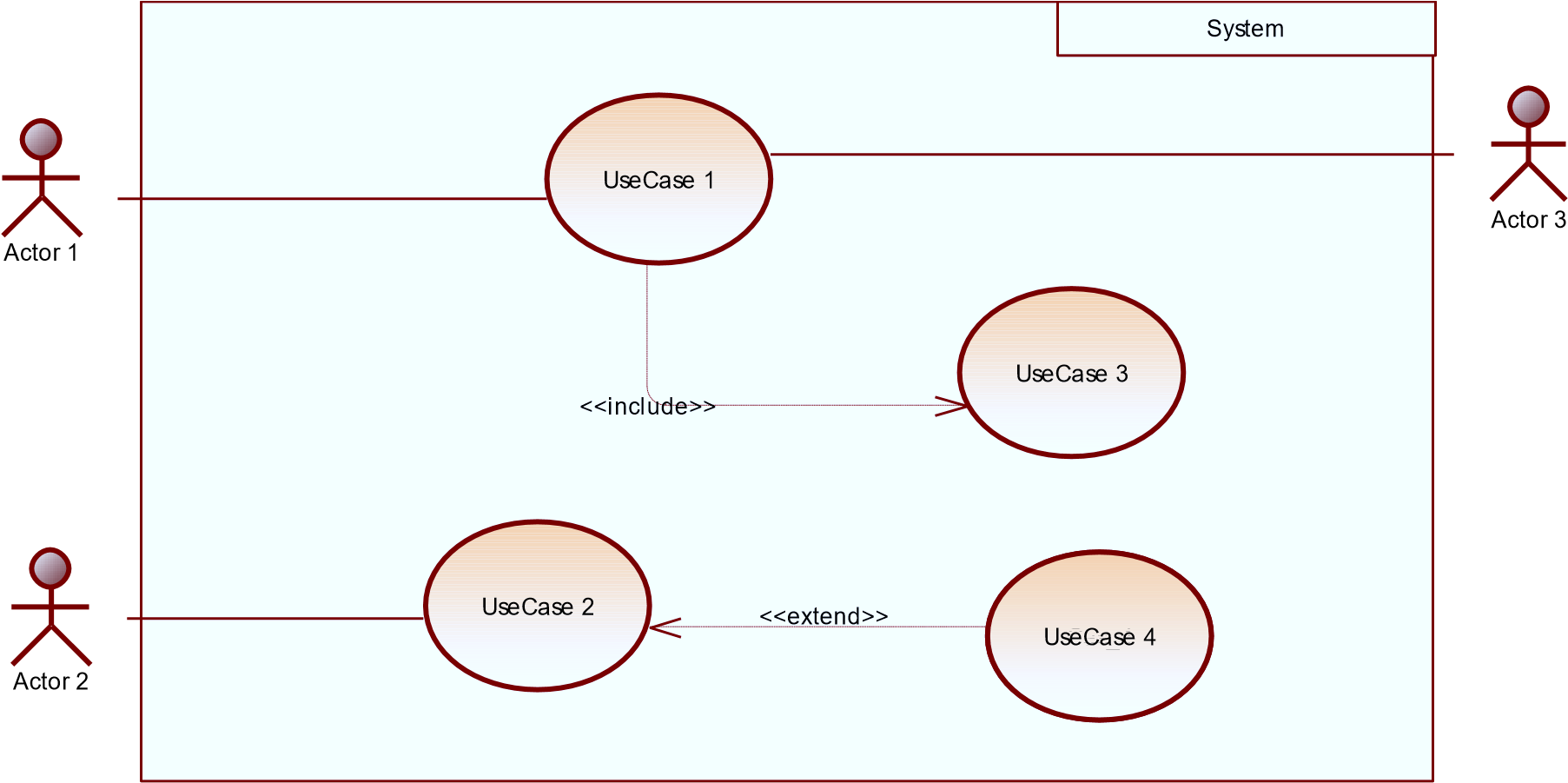


Figure 7: formalism of a usecase diagram

Table 11: elements of a usecase diagram(source: matricule 2024)

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

### The Actors of our System

Table 12: actors of the system(source: Matricule 2024)

|  |  |
| --- | --- |
| Actor | Role |
| Administrator | The administrator is responsible for managing the accounts of businesses/restaurants (validating, deleting accounts) and overseeing the general functioning of the system. |
| Customer | The user who browses menus, places food orders, and receives delivery services from restaurants or businesses through the app. |
| Business/Restaurant Admin | Verified personnel in charge of managing their business’s profile, menus, orders, and deliveries. |
| Geolocation API | Assists in the geographical location of customers and delivery personnel, providing real-time tracking for food deliveries and allowing businesses to optimize routes. |
| SMS API | Provides a service to send SMS to users, notifying them of updates on your account. |

### General Use Case Diagram

Figure 8: the general usecase diagraam

### Place order specific use case diagram

### f. Textual description

Figure 9: place order specific use case diagram

Table 13: Textual description of authentication

|  |  |
| --- | --- |
| Title | Authenticate |
| Summary | The user needs to authenticate |
| Actors | Customer, business, admin, delivery guy |
| Date | September 15, 2024 |
| Stakeholder | Easy-eats manager |
| Version | 1.0 |
| Precondition (s) | 1. The app is launched. 2. The actor has an account on the platform. |
| Triggers | The user clicks on login button |
| Nominal Scenario | 1. The system displays the login form. 2. The actor fills and submits the form. 3. The system verifies conformity of the form fields 4. The system sends the data to the dbms. 5. The dbms returns result of the query. 6. The system displays a success message to the actor. |
| Alternative Scenarios | 1. At step 4 of the nominal scenario, the user enters mismatched or missing information. 2. The system displays an error message then returns to step 2 of the nominal scenario. |
| Postcondition of success | The user has access to his/her dashboard |
| Postcondition of failure | The user does not have access to the platform |
| Non-functional requirement | Entering the password must not be visible on the screen |

Table 14: textual description of use case

|  |  |
| --- | --- |
| Title | Place order |
| Summary | The customer places an order for items |
| Actors | Customer |
| Date | September 15, 2024 |
| Stakeholder | Easy-eats manager |
| Version | 1.0 |
| Precondition (s) | The customer is authenticated |
| Triggers | The customer clicks the order button |
| Nominal Scenario | 1. The customer logs in to the system. 2. The system displays the product catalog. 3. The customer searches for items. 4. The system displays search results. 5. The customer selects an item for details. 6. The system shows item details. 7. The customer adds items to the cart. 8. The system updates the cart. 9. The customer reviews the cart and proceeds to checkout. 10. The system provides payment options. 11. The customer selects a payment method. 12. The system processes the payment. 13. The system confirms the order and displays a summary. |
| Alternative Scenarios | * At step 3, if no items match the search, the system displays a "no results found" message. * At step 11, if payment fails, the system prompts the customer to retry or choose another method. |
| Postcondition of success | The order is successfully placed and recorded |
| Postcondition of failure | The order is not completed, the cart remains unchanged |
| Non-functional requirement | * There must be a stable internet connection. * The payment API is operational. |

### COMMUNICATION DIAGRAM

### Definition

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

### Formation

11

Message\_

8.3:

12

Message\_

8.4:

13

8.5:

Message\_



Objet\_1



Obje

t\_2



Objet\_3



Objet\_4

Figure 10: formalism of the communication diagram

### Components of a communication Diagram

Table 15: components of the communication diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Message |  | Designs a particular communication between lifelines. |
| Connectors |  | It represents the relationships that exist between lifelines |
| Dependency |  | A dependency is a relationship that signifies a single or a set of model elements for their specification |
| Lifeline |  | An object represents an individual participant in the interaction conversation. |

### **Authentication communication diagrams**

7.2:

displays user does not exist error message

7.1:

displays error message

fills and submit form

3:

2:

displays login form

send authentication request

1:

6:

treats query result

process and sends query results

5:

4:

sends form data



user



system



DBMS

### Place command communication diagram

Figure 11: place command communication diagram

### SEQUENCE DIAGRAM

### Definition

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

### Formalism

sequence diagram formalism



Objet\_1



Objet\_2



Objet\_3



ok

not ok

alt

Message\_6

Message\_5

Message\_4

Message\_3

Message\_2

Message\_1

Figure 12: formalism of the sequence diagram

### Components of a sequence diagram

Table 16: components of a sequence diagram

|  |  |  |
| --- | --- | --- |
| ELEMENTS | NOTATION | DESCRIPTIONS |
| Actor |  | Represents an external entity, often a user, that interacts with the system. Typically illustrated as a stick figure. |
| Lifeline |  | The vertical dashed line extending from an object or actor, representing the existence of that entity over time |
| Message |  | Depicts the communication or interaction between objects or actors. It’s shown as horizontal arrow with the arrow’s direction indicating the message path. |
| Return Message |  | An arrow depicting the response sent back to the originating object or actor after a message. Usually represented as a dashed arrow |
| Activation Bar |  | A thin rectangle drawn on the lifeline, representing the period an object is active or executing something in response to a message. |
| Destroy |  | A large X on a lifeline, indicating the point at which an object is destroyed or ceases to exist. |
| Combined Fragment |  | Used to represent conditions, loops or other control structures. Often a rectangle that encompasses a portion of the lifelines and contains labels like ‘alt’ for alternatives or ‘loop’ for repetitive sequences. |
| Note |  | A comment or annotation that can be attached to any part of the diagram to provide additional information or clarification. |

### Authentiication sequence diagram

Authenticate



user



system



DBMS



correct format

Incorrect format

alt

if user credentials are valid

If user credentials are not valid

alt

displays invalid format error message

2.3:

2.2.2.3:

display invalid credentials error message

2.2.2.2:

displays user dashboard

2.2.2.1:

treate validation result

2.2.2:

return validation result

sends user verification query

2.2:

2.2.1:

executes user verification query

verify form conformity

2.1:

2:

fill login details and submit

1.1:

display login form

1:

click on login button

Figure 13: Authentication sequence diagram

### Place ordeer sequence diagram

Figure 14: place command sequence diagram

### Upload sequence diagram

Figure 15: Upload sequence diagram

### Rate sequence diagram

Figure 16: rate sequence diagram

### Track delivery sequence diagram

Figure 17: track delivery sequence diagram

### ACTIVITY DIAGRAM

### Definition

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

### Formalism



Partition 1



Partition 2



Partition 3



action 1



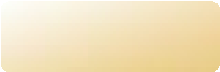
action 3



action 2



action 4



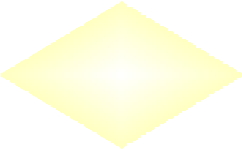
action 5



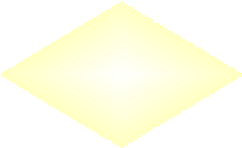
action 7



action 8



Decision\_1



Decision\_2



[

ok

]

[

]

not ok



action 6

[

ok

]

[

not ok

]

Figure 18: formalism of the activity diagram

### Elements of an activity diagram

Table 17: Elements of an activity diagram

|  |  |  |
| --- | --- | --- |
| Element Diagrammatic Representation | | Description |
| Activity |  | Used to represent a set of actions. |
| Action |  | Represent a task to be performed. |
| Activity edge |  | A directed connection between two activity nodes through which tokens may flow |
| Initial node |  | Shows the beginning of an activity or set of actions. |
| Final node |  | Stops all controls and object flows in an activity. |
| Object node |  | Represents an object connected to a series of object flows. |
| Decision node |  | Represents a test condition that slits an incoming activity edge into opposite outgoing activity edges. |
| Merge node |  | Reunite different decision  paths created using a decision node. |
| Fork node |  | Slits behaviour into parallel or concurrent flows of activities (or actions). |
| Join node |  | Unites a set of parallel or concurrent flows of activities or actions. |
| Swimlane and partition |  | A way of grouping activities performed by the same actor in an activity diagram or to group actions in the same thread. |

### Get menu activity diagram

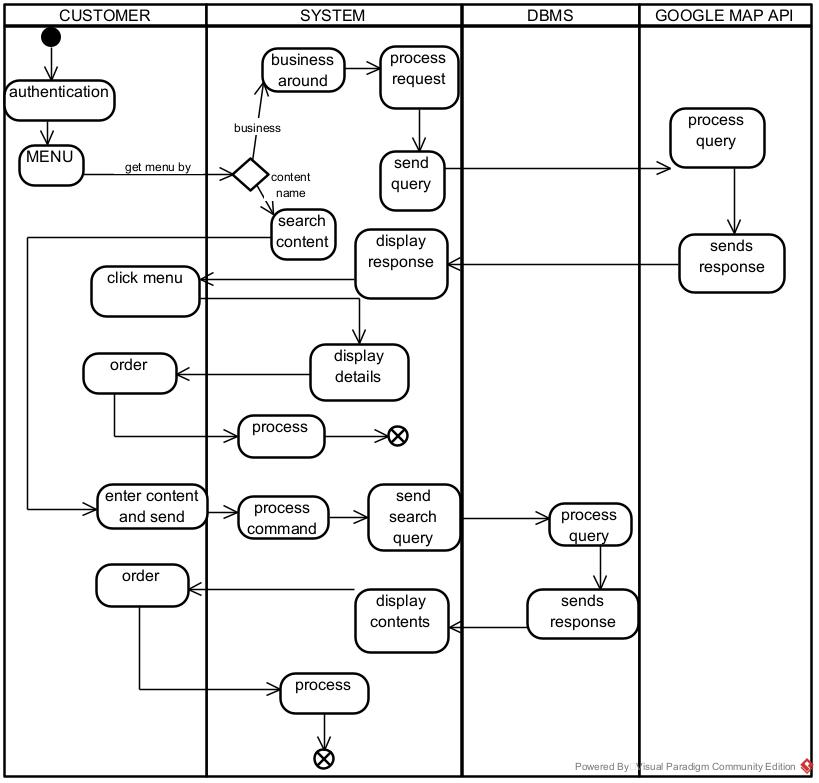


Figure 19: Get menu activity diagram

Figure 20: Get menu activity diagrm

### CONCLUTION

In the analysis phase, we chose a software development process and modelling language, after which we explained the functional need of our system, we saw the use case diagram which shows the relationship between the actors and use case (the action the actor can perform on the system),we saw the communication diagram which represents the architecture of the system based on object oriented programming, we saw the sequence diagram which represents the flow of messages between elements in the system, and lastly the activity diagram which shows the workflow of our system. We will now move to the conception phase in which we will present the Technical branch of our system together with related diagrams.

PART FIVE: THE CONCEPTION PHASE

Preamble

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

Content overview

INTRODUCTION

1.TECHNICAL BRANCH

1. GENERIC DESIGN
2. CAPTURE OF THE TECHNICAL NEEDS
3. RELATED UML DIAGRAMS
4. CLASS DIAGRAM
5. STATE MACHINE DIAGRAM
6. PACKAGE DIAGRAM

CONCLUSION

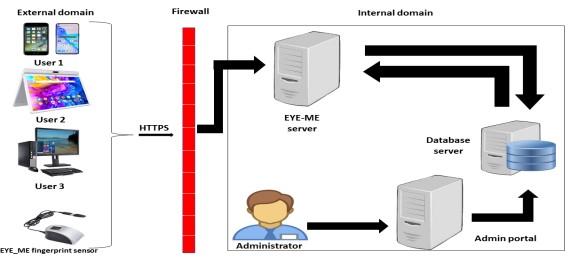
## INTRODUCTION

The conceptual phase will describe in detail the necessary specifications, features and operations that will satisfy the functioning requirements of the proposed system as modelled in the analysis phase. This phase is meant to identify and consider essential components (hardware /or software), structure (network capabilities), processes and procedures for the system to accomplish it objectives. We will look at some diagrams such as the class diagram, state machine diagram and package diagram.

## GENERIC DESIGN

### Hardware diagram of the system

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



iKare server

Figure 21: hardware design of the system

### High Level Architecture of the System

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

## CAPTURE OF TECHNICAL NEEDS

### Physical Architecture

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

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

The hardware tier, which represents our IOT device to collect vitals and it is connected directly to our mobile phone using a wired connection.

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

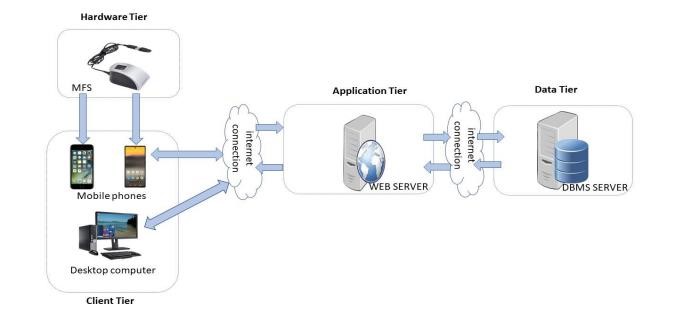


Figure 22: Physical architecture

### Logical Architecture

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

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

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

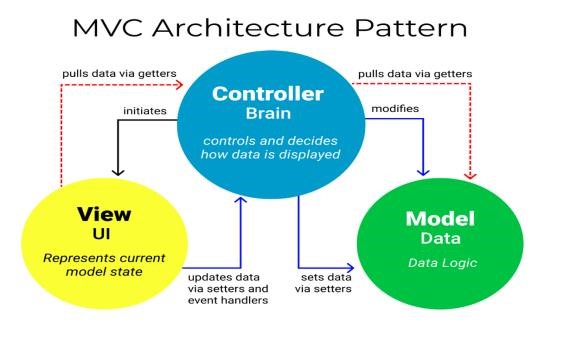


Figure 23: Logical architecture

## RELATED UML DIAGRAMS

### CLASS DIAGRAM

### Definition

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

### Formalism

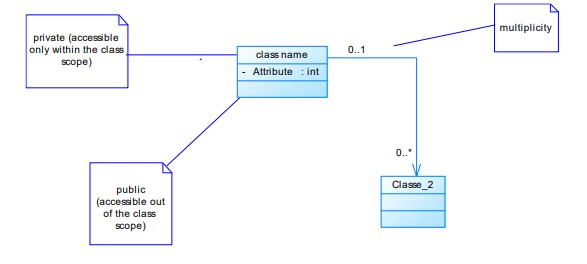


Figure 24: formalism of the class diagram

### Components of a class diagram

Table 18: components of a class diagram

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

### Easy-eats class diagram

Table 19: Class diagram of the system

### Business Rules

R1: A user must have a unique email and password for authentication.

R2: A business can upload one or more menus, and each menu is associated with only one business.

R3: A menu contains one or more items, and each item must have a unique item ID, name, and price.

R4: A customer can place one or more orders, and each order is linked to a specific customer.

R5: A customer can rate and review multiple businesses, and each rating is associated with one business and many customers.

R6: Both businesses and customers must have valid geographic coordinates (longitude and latitude) for location-based services.

R7: A cart holds one or more items, and each cart is associated with a single customer.

### STATE MACHINE DIAGRAM

### Definition

A state machine diagram describes the behaviors of a single object in response to a series of events in a system. Also known as the state machine diagram, it models the dynamic flow of control from the state of a particular object within a system.

### Formalism

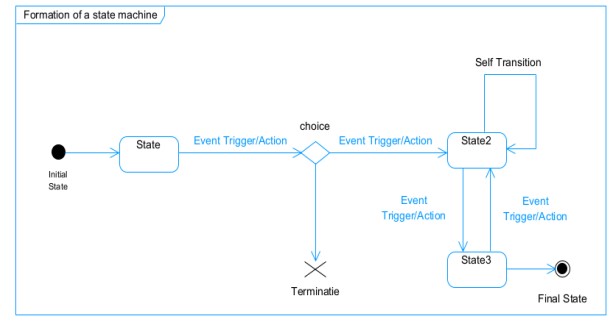


Figure 25: formalism of the state machine diagram

### Components of a state machine diagram

Table 20: components of a state machine diagram

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

### Place command state machine

Figure 26: place command state machine diagram

### PACKAGE DIAGRAM

### Definition

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

### Formalism

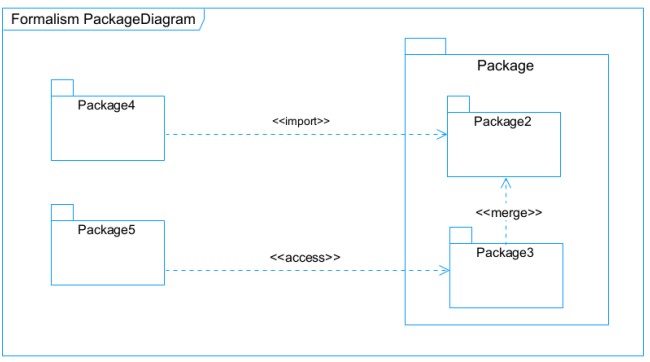


Figure 27: formalism of the package diagram

### Elements of a package diagram

Table 21: Elements of a package diagram

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

### d. Easy-eats package diagram

Figure 28: 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 on appropriate deployment targets. We finished this phase by looking at interactions between the various aspects and actors of our system. The next phase of our report is the realization phase where we will look at aspects concerning the implementation of our system.

PART SIX: THE REALIZATION PHASE

Preamble

In this phase we will to straight forward in the implementation of our solution, we will base ourselves on the analysis and conception phases and also present the component and deployment diagrams.

Content overview

INTRODUCTION

1. DEPLOYMENT DIAGRAM
2. COMPONENT DIAGRAM

CONCLUSION

## INTRODUCTION

Here in the realization phase, we will see some diagrams related to the physical aspect pf the system like libraries, documents, as well as the physical topology of the components of the system when the software is been deployed.

### DEPLOYMENT DIAGRAM

### Definition

Deployment diagram is a structural diagram used to visualize the topology of the physical components of a system, where the software is deployed. They consist of nodes and their relationship. It is related to the component diagram because the components are deployed using the deployment diagram. A deployment diagram consists of nodes.

Nodes are nothing but physical hardware used to deploy the application.

### b. Formalism

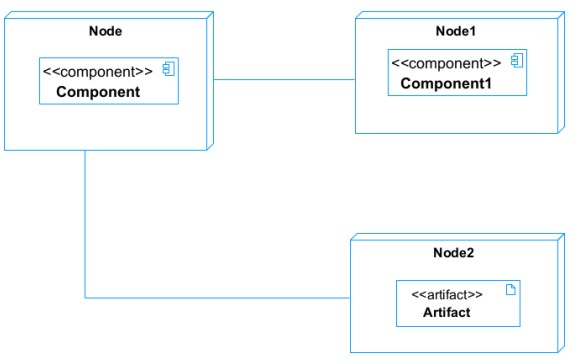


Figure 29: formalism of the deployment diagram

### c. Elements of a deployment diagram

Table 22: Elements of the deployment diagram

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

### System deployment diagram

Figure 30: System deployment diagram

### COMPONENT DIAGRAM

### Definition

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

### b. Formalism

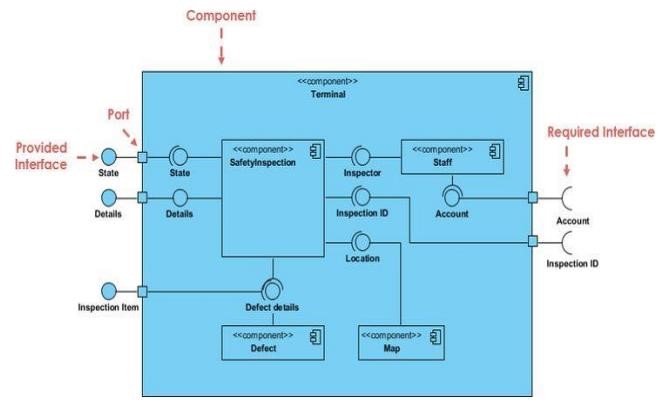


Figure 31: Formalism of the component diagram

### Elements of a component diagram

Table 23: Elements of a component diagram

|  |  |  |
| --- | --- | --- |
| Name Representation Description | | |
| A Component |  | A component is an abstract logical unit block of a system.it is represented as a rectangle with smaller rectangle in the upper right corner which saves as it icon for recognition. |
| Dependency |  | Dependency is a directed relationship which is used to show that some components are dependent on others for their correct functioning. |
| Required Interface |  | It is a straight line from the component box with an attached half circle representing interfaces where a component requires information in order to perform its own functions. |
| Provided Interface |  | It is a straight line from the component box with an attached circle representing interfaces where a component produces information used by required interfaces |
| Port |  | A port (represented by a small square at the end of a required or provided interface) is used when the components delegate the interfaces to an internal class. |

### System component diagram

Figure 32: system component diagram

## CONCLUSION

In our realization phase, we implemented our application. In other to accomplish this phase, we made used of our analysis and conception phase. We also drew the deployment and component diagrams which depict the structure of our system in terms of modules, files, assets, how the different elements interact with each other. We will move to the test of functionalities phase, where we will examine the different modules present in our app and how beneficial they are to its different users

PART SEVEN:

TEST OF FUNTIONALITIES

Preamble

In this phase, we will present the various functionalities of our application.

Content overview

INTRODUCTION

1. APPLICATION FUNCTIONALITIES
2. TESTS SHOWCASES

CONCLUSION

## INTRODUCTION

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