# **DEDICATION**

**THIS WORK IS DEDICATED TO MY MOTHER**

# **ACKNOWLEDGMENTS**

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:

* The Resident Representative of IAI-Cameroon, **Mr. ABANDA ARMAND CLAUDE**, for his support, words of encouragement and the different advices on how to approach situations.
* The chief executive officer of Welldone planet, **Mr. BOM GABRIEL** and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Our professional supervisor, **Mr. Tiomela Jou Daniel** for his moral support, advice and professional expertise he shared with us.
* Our academic supervisor **Mrs. ONGUENE EBENYE** for her moral support, effort and advices throughout the academic year.
* To our academic teachers **Mr. MESSIO, Mr. AGBOR AND Mr. NDENGE** 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.
* For my brothers and sister for their love and support.
* 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 des matières

[**DEDICATION** 1](#_Toc174022737)

[**ACKNOWLEDGMENTS** 2](#_Toc174022738)

[**GLOSSARY** 5](#_Toc174022739)

[**ABSTRACT** 6](#_Toc174022740)

[**RESUME** 7](#_Toc174022741)

[**GENERAL INTRODUCTION** 8](#_Toc174022742)

[**PART I: INSERTION PHASE** 8](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022743)

[**INTRODUCTION** 10](#_Toc174022744)

[I. WELCOME AND INTERGRATION 10](#_Toc174022745)

[1. Welcome 10](#_Toc174022746)

[2. Integration 10](#_Toc174022747)

[II. GENERAL PRESENTATION OF WELLDONE PLANET 10](#_Toc174022748)

[1. History and Missions 10](#_Toc174022749)

[2. Presentation of the structure 11](#_Toc174022750)

[b. Material Safety Data Sheet 11](#_Toc174022751)

[C. Resources 12](#_Toc174022752)

[3. Visions, Objectives of Welldone Planet 14](#_Toc174022753)

[b. Goals 15](#_Toc174022754)

[c. Attributions 17](#_Toc174022755)

[**CONCLUSION** 18](#_Toc174022756)

[**PART TWO: SPECIFICATION BOOK** 19](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022757)

[**INTRODUCTION** 21](#_Toc174022758)

[I. CONTEXT AND JUSTIFICATION 22](#_Toc174022759)

[II. OBJECTIVES OF THE PROJECT 22](#_Toc174022760)

[A. General Objective 22](#_Toc174022761)

[B. Specific Objectives 22](#_Toc174022762)

[III. EXPRESSION OF NEEDS 22](#_Toc174022763)

[A. Functional needs 22](#_Toc174022764)

[B. Non-functional Needs 22](#_Toc174022765)

[IV. ESTIMATED COST OF THE PROJECT 23](#_Toc174022766)

[A. Hardware Resources 23](#_Toc174022767)

[B. Software Resources 24](#_Toc174022768)

[C. Human Resources 25](#_Toc174022769)

[D. Total Project Estimated Cost 25](#_Toc174022770)

[V. PROJECT PLANING 26](#_Toc174022771)

[A. Chronogram of activities 26](#_Toc174022772)

[B. Gantt Chart 27](#_Toc174022773)

[VI. LIST OF PARTICIPANTS AND DELIVERABLES 27](#_Toc174022774)

[A. List of participants 27](#_Toc174022775)

[B. Deliverable 27](#_Toc174022776)

[**CONCLUSION** 28](#_Toc174022777)

[**PART THREE: ANALYSIS PHASE** 29](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022778)

[**INTRODUCTION** 32](#_Toc174022779)

[I. DESCRIPTION OF THE EXISTING SYSTEM 32](#_Toc174022780)

[II. PRESENTATION OF THE ANALYSIS 32](#_Toc174022781)

[A. MODELING WITH UML 2.5 33](#_Toc174022782)

[a) Structural Diagrams : 34](#_Toc174022783)

[b) Behavioral Diagrams : 34](#_Toc174022784)

[c) UML 2.5 diagrams overviews 35](#_Toc174022785)

[B. THE PROCESS OF DEVELOPMENT OF A SOFTWARE 35](#_Toc174022786)

[a) A Unified Process 36](#_Toc174022787)

[b) The Two Track Unified Process (2TUP) 36](#_Toc174022788)

[III. JUSTIFICATION OF THE ANALYSIS 36](#_Toc174022789)

[a) The left branch (functional branch) 37](#_Toc174022790)

[b) The right branch (Technical branch) 37](#_Toc174022791)

[c) The middle branch (Realization or implementation branch) 37](#_Toc174022792)

[IV. MODELING OF THE SYSTEM 38](#_Toc174022793)

[A. CAPTURE OF FUNCTIONAL NEEDS 38](#_Toc174022794)

[1. *Use Case Diagram* 38](#_Toc174022795)

[2. ACTIVITY DIAGRAM 42](#_Toc174022796)

[3. State Machine Diagram 45](#_Toc174022797)

[**CONCLUSION** 47](#_Toc174022798)

[**PART FOUR: CONCEPTION PHASE** 47](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022799)

[**INTRODUCTION** 49](#_Toc174022800)

[I. TECHNICAL BRANCH 49](#_Toc174022801)

[A. GENERIC DESIGN 49](#_Toc174022802)

[a) Hardware diagram of the system 49](#_Toc174022803)

[High Level Architectural Diagram of the Software 50](#_Toc174022804)

[B. CAPTURES TECHNICAL NEEDS 50](#_Toc174022805)

[a) Physical architecture 50](#_Toc174022806)

[Logical architecture 50](#_Toc174022807)

[II. IMPLEMENTATION BRANCH 51](#_Toc174022808)

[A. PRELIMINARY DESIGN 51](#_Toc174022809)

[a) Component diagram 51](#_Toc174022810)

[Package Diagram 53](#_Toc174022811)

[Deployment Diagram 55](#_Toc174022812)

[B. DETAILED DESIGN 57](#_Toc174022813)

[**CONCLUSION** 57](#_Toc174022814)

[**PART FIVE: REALISATION PHASE** 58](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022815)

[**INTRODUCTION** 60](#_Toc174022816)

[I. ENTITY RELATIONSHIP DIAGRAM 61](#_Toc174022817)

[II. PRESENTATION OF DEVELOPMENT TOOLS 61](#_Toc174022818)

[A. MATERIAL RESOURCE 61](#_Toc174022819)

[B. TECHNOLOGICAL STACK 61](#_Toc174022820)

[**CONCLUSION** 61](#_Toc174022821)

[**PART SIX: USER GUIDE** 62](file:///D:\E-books\Cours%20Ecole\BA2\Stage\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc174022822)

[**INTRODUCTION** 64](#_Toc174022823)

[I. INSTALLATION OF THE APPLICATION 64](#_Toc174022824)

[II. SHOWCASES 64](#_Toc174022825)

[**GENERAL CONCLUSION** 64](#_Toc174022826)

[**ANNEXE** 64](#_Toc174022827)

[**BIBLIOGRAHY** 65](#_Toc174022828)

[**WEBOGRAPHY** 65](#_Toc174022829)

[**TABLE OF CONTENT** 65](#_Toc174022830)

# **GLOSSARY**

➢ 2TUP: Two Track Unified Process.

➢ AICS: African Institute of Computer Sciences.

➢ APK: Android Package Kit.

➢ ERD: Entity Relational Diagram.

➢ IPA: iOS App Store Package.

➢ MFS: Mantra Fingerprint Sensor

➢ MVC: Model View Controller.

➢ UML: Unified Modelling Language.

# **ABSTRACT**

# **RESUME**

# **GENERAL INTRODUCTION**

# **PART I: INSERTION PHASE**

**SUMMARY**

The integration file is the very first document written during the academic internship period. This is the part of the internship which presents the reception structure, its operation, as well as the reception of the student in the company. This induction phase generally lasts two weeks and allows the trainee to familiarize themselves with their new environment.

Preview

**INTRODUCTION**

1. **WELCOME AND INTERGRATION**
2. **GENERAL PRESENTATION OF WELLDONE PLANET**
3. **ORGANISATION OF WELLDONE PLANET**
4. **HARDWARE AND SOFTWARE RESOURCES**

**CONCLUSION**

# 

# **INTRODUCTION**

Professional integration is a process which allows an individual or a group of individuals to enter the labor market under conditions favorable to obtaining a job. As part of the academic internship, we were received as an intern within the premises of Welldone Planet for a period of three (03) months and the integration phase lasted 02 weeks. This part of the internship report presents the structure or internship was carried out, its operation, the reception of the student and the research theme during this period.

# I. WELCOME AND INTERGRATION

## Welcome

Within the Welldone Planet structure, we were welcomed on Monday July 1, 2024 by its Technical Director, Mr. TIOMELA JOU Daniel

## Integration

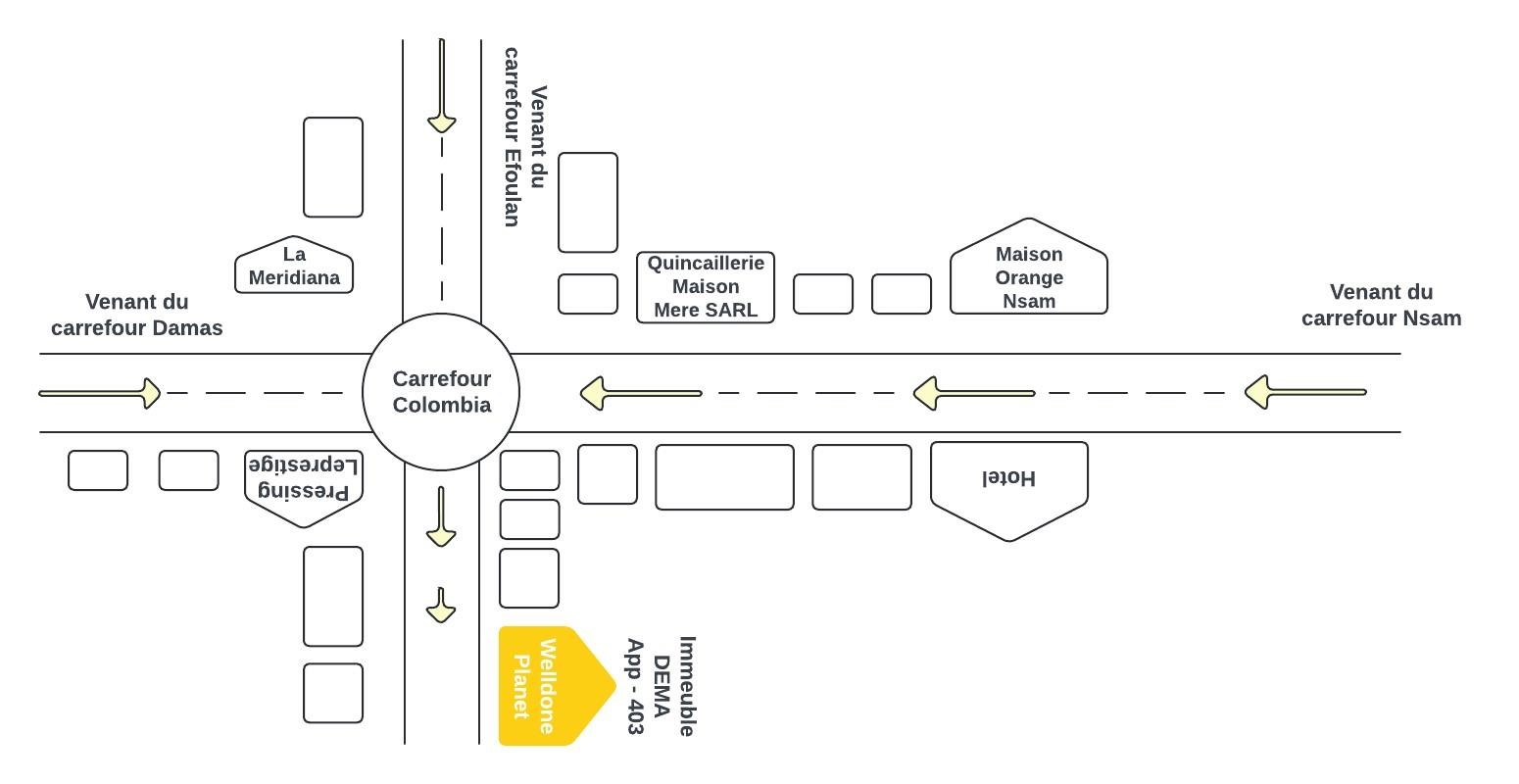
Mr. TIOMELA JOU Daniel organized a meeting during which he reminded us of the internal regulations of the establishment, its operation, its vision, its objectives, its services, its requirements and the development of the weekly schedule. Emphasis was placed on the conduct to be followed and compliance with the regulations in force. Several pieces of advice and recommendations were also given to us during various online sessions with the Technical Director. Then we argued about our theme.

# II. GENERAL PRESENTATION OF WELLDONE PLANET

## History and Missions

Welldone Planet was born from an ambitious project designed by a group of engineers graduated from the African Institute of Computer sciences (AIC) in Cameroon. Created in June 2023 and officially inaugurated on June 1, 2024 by the resident representative of AICS-CAMEROON, this company aims to become a leader in the provision of digital services. By advocating the emergence of the new information and communication technologies sector in Africa, and more particularly in Cameroon, Welldone Planet's mission is to inspire excellence and achieve innovation. Their slogan, "We believe that thanks to digital technology all dreams can come true. Let's Dream Big!" perfectly reflects their ambition to transform dreams into reality thanks to the power of digital technology. The company motto, “Inspire Excellence, Achieve Innovation, Dream Big,” embodies their commitment to creating a bright, technological future for the continent.

## Presentation of the structure



There

~~Efoulan~~

Coming from



DO

Welldone

Ann.

op

-

403

~~Building~~

a. Structure location plan

figur

e 1: Location plan

## b. Material Safety Data Sheet

table 1: Welldone Planet Material Safety Datan Sheet

|  |  |
| --- | --- |
| QUALITY | DESCRIPTION |
| The head office | Yaoundé (Nsam – Carrefour Colombia) |
| Legal form Type | SARL |
| of business | Private |
| Services | IOT/AI BIG DATA  Software Development  IT maintenance Web and Mobile development  Infographics  Cybersecurity  Digital Marketing IT training |
| Phone | (237) 677 223 241 |
| Website | www.welldoneplanet.com |
| E-mail | welldoneplanet@gmail.com |
| Languages used | French, English |

## C. Resources

i. Material resources

The company has numerous equipment listed in the following table.

table: Welldone Planet's material resources

|  |  |  |
| --- | --- | --- |
| DESIGNATION PICTURE | | |
| 3 HP desktops | |  |
| 3 Wifi Modem | |  |
| 3 plasma screens | |  |
| 1 fridge | |  |
| 1 canon printer | | Canoe |
| 1 HP laptop | |  |
| Ultiamte Kit for Rasberry | | OsYoo |
| Super starter kit | |  |
|  | Smart robots | SMART R |

. Software resources

table 2: Welldone Planet software resources

|  |  |
| --- | --- |
| DESIGNATION | PICTURE |
| Windows 10 operating system |  |
| Windows 11 operating system | ~~Home~~    ~~Windows11~~ |
| Adobe Creative Cloud Suite 2024 | ~~COLLECTION 0~~ |

# 3. Visions, Objectives of Welldone Planet

*a. Visions*

Welldone Planet's vision is to become a major catalyst in the field of information and communication technologies in Africa and Cameroon in particular, with a focus on innovation and excellence. Firmly believing that digital can turn dreams into reality, Welldone Planet is committed to creating forward-thinking technology solutions that meet the growing needs of society. By inspiring excellence and achieving innovation, the company aspires to propel Cameroon, and by extension Africa, towards a future where new technologies are at the heart of economic and social development.

## b. Goals

Welldone Planet sets ambitious goals to have a significant impact on the economic and social development in Africa. Based on a holistic approach and inclusive, the company aims to:

* Reduce poverty: By providing innovative technological solutions, Welldone Planet intends to create economic opportunities and improve living conditions of local populations, thus contributing to the reduction of poverty..
* Facilitate the development of human capacities: Through programs of quality training in computer science and information technology, the company aspires to strengthen the skills of individuals and professionals, allowing them to remain competitive in the job market.
* Holistic approach, including young people\*\*: Welldone Planet adopts a global which includes young people, whether they are educated or have dropped out of school, in order to fully integrate them into technological development and economically, providing them with opportunities for learning and growth.
* Support businesses in their growth\*: By using techniques based on optimization, bottleneck management and cost-effectiveness sustainable, Welldone Planet is committed to supporting businesses in their expansion, by improving their operational efficiency and economic performance

**4. Activities, Organization Chart and Responsibilities of**

**Welldone Planet**

a. Activities

Welldone Planet offers a full range of technology services to meet to the diverse needs of its customers. The company's main mission is to solve IT problems that businesses encounter on a daily basis in the areas following:

* IOT/IA BIG DATA
* Software Development
* IT maintenance
* Web and Mobile Development
* Infographics
* Cybersecurity
* Digital Marketing
* IT training

b. Organizational chart

The organization chart is above all a communication tool intended to

Facilitate understanding of existing relationships and links within society. To this end, it allows fora global view of the company in terms of services, divisions and more. **Welldone Planet** has six departments within it and we have been assigned to technical direction. This is the division colored in orange in the organization chart below.

***figure 2 : Welldone Planet organization chart***

## 

## c. Attributions

The Welldone Planet Company includes several executive-led divisions main competent persons:

* **The Human Resources Division (HRD)**  headed by a director of human resources which is responsible for personnel management, training and skill developpement
* **The Sales and Marketing Division (DCM**) headed by a director sales and marketing who is responsible for sales, marketing and development commercial.
* **The Communication Division (DCOM)** headed by a director of communication of the company which is responsible for sales, marketing and Business Development**.**
* **The Research and Development Division (DR&D)** headed by a director of research and development who is responsible for innovation and development of new products
* **The Financial Division (DRMF)** headed by a director of financialresources who is responsible for financial management, budgets and accounting.
*  **The Technical Division (DT)** headed by a technical director who is responsible for the technical management, production and maintenance**.**.
* **Secretariat** for invoicing, photocopying, customer registration

# **CONCLUSION**

The insertion phase was a very educative one, we got to know more about the history of the company, how it all started till the level it is right now, how it is 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 **“CONCEPTION AND REALISATION ”.** 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: SPECIFICATION BOOK**

SUMMARY

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.

Preview

**INTRODUCTION**

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

**CONCLUSION**

# **INTRODUCTION**

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

# CONTEXT AND JUSTIFICATION

# OBJECTIVES OF THE PROJECT

## General Objective

## Specific Objectives

# EXPRESSION OF NEEDS

## Functional needs

## Non-functional Needs

Non-functional requirements define how the system should perform, these include:

* **Usability:** The system should be easy to use and navigate for both customers and service providers.
* **Reliability:** The system should be available and reliable with minimal downtime.
* **Performance:** The system should respond quickly to user interactions and process orders efficiently.
* **Security:** User data and payment information should be protected.
* **Maintainability:** The system should be easy to update and maintain.
* **Scalability:** The system should be able to handle increasing numbers of users and orders.
* **Portability:** The system should be compatible with different mobile devices and platforms.
* **Accessibility:** The system should be accessible to users with disabilities.

# ESTIMATED COST OF THE PROJECT

## Hardware Resources

*Table 4:Hardware Resource (Source: https://www.scribd.com/document/561202966/mercuriale-2022)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RESOURCE**  **S** | **HARDWARE** **USAGE** **QUANTITY** | | | **UNIT**  **COST(FCFA)** |
| COMPUTER | DELL  Windows 10 pro 64 bits,  Latitude E5440, 8GB RAM, 500GB hard disk; intel core i5. | main resource for the  accomplishment of this project; report writing, designs, analysis, coding | 1 | 368000 |
| Mobile phone | SAMSUNG A02  4 GB RAM, 64 GB hard disk | Testing of functionalities. | 2 | 80000 |
| Removable disk | 8GB USB key | For file transfer from one computer to another. | 1 | 6325 |
| CD | CD-ROM |  | 3 | 3000 |
| MODEM | 4g LTE-advanced **mobile**  **Wi-**  **Fi** hotspot **mq531** 150mps | Used to share internet access amongst devices | 1 | 23000 |
| **Total 1** |  |  |  | **559862** |

## Software Resources

*Table 5:Software Resources(https://www.g2.com/categories/pricing)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | HARDWARE | USAGE | QUANTITY | UNIT COST(FCFA) |
| Development tool | Visual studio code | Code editing | 1 | Open source |
| Project planner | Gantt project | Project planning tool | 1 | Free software |
| Operating system | Microsoft windows 10 pro | Computer operating system | 1 | 103000 |
| Text Editor | Microsoft office word 2019 | For typing and formatting of  report | 1 | 161000 |
| Presentation | Microsoft office PowerPoint 2019 | For making  presentations | 1 | 84230 |
| Modelling tool | Visual-paradigm Modeler version | Modelling the system in uml | 1 | 3350 |
| Geo-location plan designing tool | Microsoft Visio 2007 | To draw  location plan | 1 | Free version |
| Web browser | Google chrome | For running and  testing application | 1 | Free version |
| Database management system | PostgreSQL | Communing  with the  database | 1 | Open source |
| **Total 2** |  |  |  | **351580** |

## Human Resources

*Table 6:Human Resource (Source: https://www.paylab.com/)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | NUMBER | COST PER DAY | NUMBER OF DAYS | COST (FCFA) |
| **Project manager** | 1 | **250000** | **60** | **15000000** |
| **Analyst** | 1 | **150000** | **30** | **4500000** |
| **Designer** | 1 | **130000** | **25** | **3250000** |
| **Programmer** | 1 | **100000** | **32** | **3200000** |
| **TOTAL 3** | **4** |  |  | **25950000** |

## Total Project Estimated Cost

*Table 7:Total Project Estimate Cost*

|  |  |  |  |
| --- | --- | --- | --- |
| Total 1 (FCFA) Total 2 (FCFA) | | Total 3 (FCFA) | Overall total (FCFA) |
| **559862** | **351580** | **25950000** | **26861442** |
| **Margin of error = 10% of Overall total**  **=2686144.2 FCFA**  **Final Total = Overall Total + margin of Error**  **=29547586.2 FCFA** | |  |  |

# PROJECT PLANING

## Chronogram of activities

*Table 8:Chronogram of activities*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE | OBJECTIVE | OUTPUT | DURATION | PERIOD |
| Insertion | Welcome and installation into the into the company, and the attribution of themes | Insertion report | 2weeks | 01st to 14th June 2022 |
| Specification | Assessment of  functional needs | Specification book | 1week 6days | 15th to 28th June 2022 |
| Analysis | Analysis of the system | Analysis book | 3weeks | 29th June to 19th July  2022 |
| Conception | Hardware and  software conception | Conception book | 2weeks | 20th July to 09th august  2022 |
| Realization | Realization of the system | Realization document | 4weeks 2days | 10th august to 09th  September |
| Deployment | Deployment of the system | Testing of the software and debugging | 1week 2days | 12th to 20th September |
| Writing user guide | Instructions on how to use the software and  hardware | User guide | 1week 3days | 21st to 30th September |

## Gantt Chart

# LIST OF PARTICIPANTS AND DELIVERABLES

## List of participants

|  |  |  |
| --- | --- | --- |
| NAME | TITLE ROLE | |
| Mrs. ONGUENE EBENYE | Lecturer at AICS-Cameroon | Academic supervisor |
| Mr. TIOMELA DANIEL | Computer science engineer at WELLDONE PLANET | Professional supervisor |
| TCHABAT TCHOUATE PETTANG VALIER | software engineering student at AICS-Cameroon | Analyst and developer |

## Deliverable

We are supposed to submit a report at the end of this project that includes;

* A user guides.
* We are also expected to submit CD ROM containing the mobile application IPA[[1]](#footnote-1) and APK[[2]](#footnote-2)

# **CONCLUSION**

The specification book helps us to site the different needs we need in order to implement our application for student follow-up, we saw list of participants and deliverables for our project. In time allocated for this section, we were able to accomplish the different task, we will move directly to the next phase which is the analysis phase. In the analysis phase we will study the existing system in detail and model our system with a modelling language and process.

# **PART THREE: ANALYSIS PHASE**

SUMMARY

After specification book, we have the Analysis phase which permits us to represent a detailed analysis of the limitations identified in our context, and our solution, through a software development process and modelling language.

Preview

**INTRODUCTION**

1. **DESCRIPTION OF THE EXISTING SYSTEM**
2. **PRESENTATION OF THE ANALYSIS METHOD**
3. **JUSTIFICATION OF THE ANALYSIS METHOD**
4. **MODELING OF THE SYSTEM**

**CONCLUSION**

# **INTRODUCTION**

The analysis book permits us to examine in an explicit way the existing system, it’s limitations and how we can remedy them. We will also describe in details the modeling language known as UML (Unified Modeling Language) which is coupled with Two Tract Unified Process (2TUP) to form a method and its justification why we decided to use it in preference of another. then we will dive directly into the modeling of the proposed solution consisting of diagrams that meets the requirements of the functional needs.

# DESCRIPTION OF THE EXISTING SYSTEM

# PRESENTATION OF THE ANALYSIS

We have studied in details some analysis methods by looking at its objectives, its structuring, its pros and cons, so as to choose an analysis method which is reliable and adaptive to our project, below are some analysis methods we studied and a brief explanation in order for you to understand why we made our choice;

* 1. **MERISE:** it is an information system design and development widely used in France. It was first early introduced in 1980s.
  2. **Agile:** this methodology is growing in popularity, thanks to highly competitive business environment and increased innovation. In general, agile methodologies prioritized shorter, interactive cycle and flexibility.
  3. **Scrum:** this is the most popular agile development framework because it is relatively simple to implement. It also solves so many problems that software developers struggle with in the past, convoluted development cycles, project plan, and shifting production schedules. This methodology allows for rapid development and testing, especially with small teams.
  4. **APF:** which stands for ADAPTIVE PROJECT FRAMEWORK, it grows from the difficulty in managing most IT projects using traditional project management methods due to uncertain and changing requirement. APF begins with a requirement breakdown structure (RBS) to define strategic goals based on productive requirements, functions, sub-function and features. The project proceeds in iterative stages, and at the end of each step, teams evaluate previous results to improve performance and practices.
  5. **XP:** which stands for Extreme Programming is a software development methodology that advocate frequent releases in short development cycles, which is intended to include checkpoints for the adoption of new customer requirements and improve productivity. This methodology takes its name from the idea that the traditional software engineering practices are taken to extreme levels.

## MODELING WITH UML 2.5

The unified modeling language (UML) is a general purpose, developmental modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The Unified Modeling Language (UML) was standardized in January 1997 by the Object Management Group (OMG) which is an American association created in 1989 and aims to promote and standardize the object model in all it forms. In 2005, UML was also published by the international organization for standardization (ISO) as an approved ISO standard. Uml since 2015 is in its version 2.5. This version consists of fourteen diagrams classified into structural and behavioral diagrams

### Structural Diagrams :

Structural diagrams represent the static components of a system; they emphasize on what should be in the system we are modeling. They include:

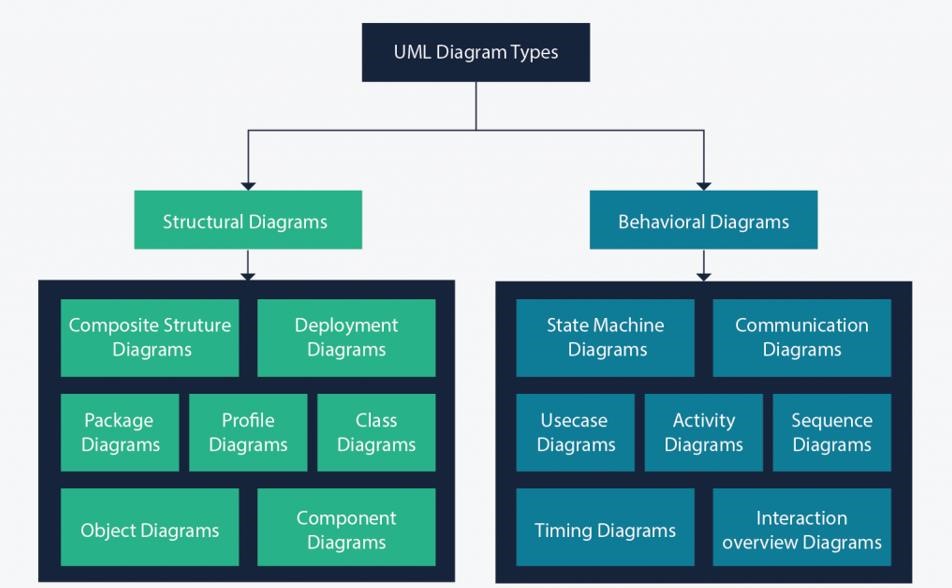
* Class diagram;
* Object diagram;
* Package diagram;
* Composite structural;
* Deployment diagram;
* Component diagram;
* Profile diagram.

### Behavioral Diagrams :

The behavioral diagrams capture the dynamic state of a system; they emphasize on what should happen in the system we are modeling. They are:

* + Use case diagram;
  + Activity diagram;
  + State machine diagram;
  + Sequence diagram;
  + Communication diagram;
  + Interaction overview diagram;
  + Timing diagram.

### UML 2.5 diagrams overviews



*Figure 4: UML 2.5 diagrams overview (source: https://creately.com/blog/diagrams/uml-diagram-types-examples/)*

It is important to note that uml is a modeling language and not a method or procedure. Hence, to give it an approach, we need to associate UML to a Unified process (UP) in order to give our conception a methodology to follow. A unified process is a generic method for developing software. This implies it is necessary to adapt the UP to the context of the project, team domain and or the organization. We will use the Two Tracks Unified Process (2TUP) throughout our project to implement our solution.

## THE PROCESS OF DEVELOPMENT OF A SOFTWARE

A process can be defined as a partially sequence of steps that permits us to obtain software systems or evolution of an existing one. The main objective of software development is the production of quality software that response to the needs of the users during a particular time and at a particular cost.

### 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 (2TUP)

2TUP is a unified process which is belt 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 bellow illustrates the branches of 2TUP.

# JUSTIFICATION OF THE ANALYSIS

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.

### 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 different diagrams are shown in the table below.

### 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.

**Detailed conception:** This is the detailed design of each feature of the system.

**Coding and testing:** This are 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.

# MODELING OF THE SYSTEM

## A. CAPTURE OF FUNCTIONAL NEEDS

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

## 1. Use Case Diagram

#### Definition

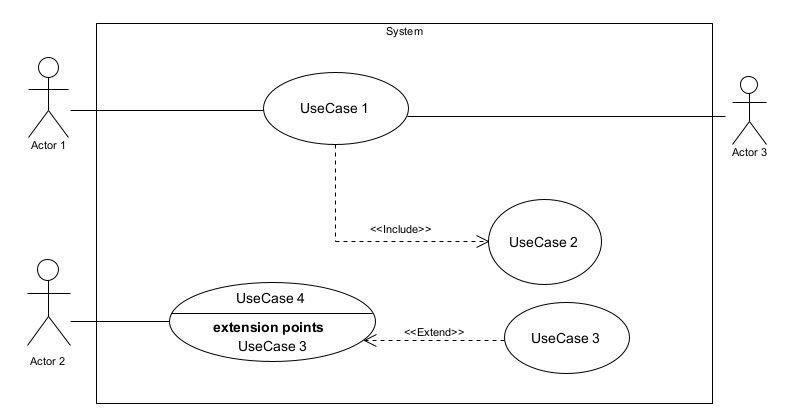
A use case diagram shows the functionalities of a system, their interdependencies and how they relate with actors of the system. A use case is a specification of behaviour.

The main objectives of the use case diagram are:

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

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

#### Formalism



*Figure 6: Use Case diagram formalism*

*Table 10:Use case diagram component*

|  |  |  |
| --- | --- | --- |
| 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 ase represents a |
|  | Use Case  1 | 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 Actor of our System

After the study of the current system in AICS-Cameroon, in other to implement our proposed solution, we identified the actors listed in the table below.

*Table 11:Actors of our system.*

|  |  |
| --- | --- |
| **Actor** | **Role** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **SMS API** |  |

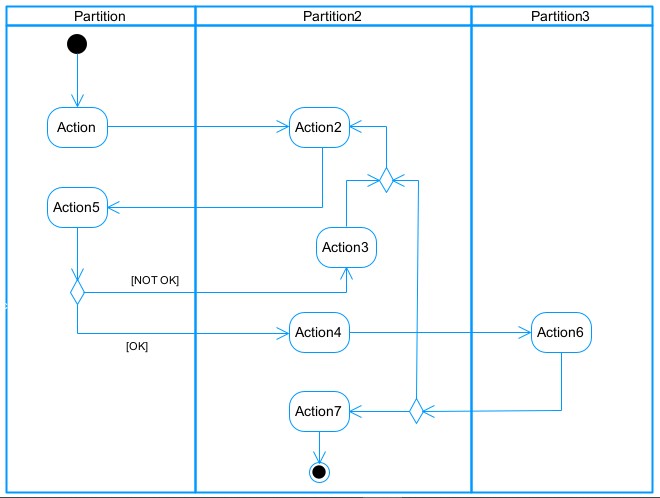
#### General Use Case Diagram

### 2. 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:



*Figure 10:Activity Diagram Formalism*

*Table 14:components of an Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Elements Diagrammatic Representation** | | **Description** |
| **Activity** |  | Use 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. |

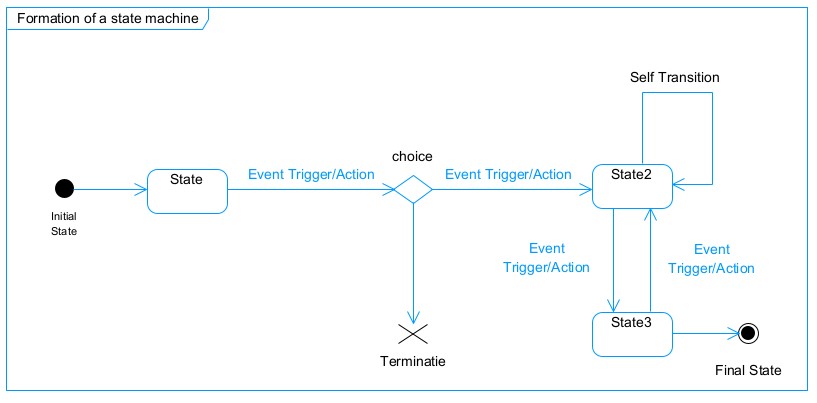
#### Activity Diagram

### 3. State Machine Diagram

#### Definition

A state machine diagram describes the behaviour 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 13:Formalism of a state machine diagram*

*Table 16 State Machine Diagram Components:*

|  |  |  |
| --- | --- | --- |
| **Element** | **Diagram Relationship** | **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. |

#### State Machine Diagram

# **CONCLUSION**

In the analysis phase, we chose a software development process and modelling language, after which we presented the existing system, its limitation and our proposed solutions. We went forth explaining 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 activity diagram which shows the workflow of our system, and lastly the State machine diagram which shows the behaviour of a single object in response to an event. We will now move to the conception phase in which we will present the other two branches of the 2TUP which are the Technical and the realization branch of our system.

# **PART FOUR: CONCEPTION PHASE**

SUMMARY

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.

Preview

**INTRODUCTION**

1. **TECHNICAL BRANCH**
2. **Generic Design**
3. **IMPLIMENTATION BRANCH**
4. **Preliminary Design**
5. **Detailed Design**

**CONCLUSION**

# **INTRODUCTION**

The conceptual phase will describe in details 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 component diagram, package diagram, communication diagram, class diagram and the object diagram.

# TECHNICAL BRANCH

## 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.

### High Level Architectural Diagram of the Software

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

## CAPTURES 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 multi-tier architecture This architecture separate it tiers from each other based upon the user and the manipulated data in the database. It is important to note that with the multi-tier architecture, only neighboring layers can communicate. Each layer has a well-defined communication interface and the evolution of the layer is independent of the other. The multi-tier of our system is made up of:

* The hardware tier, which represents our fingerprint sensor 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

### 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.

# IMPLEMENTATION BRANCH

Here we will see the preliminary conception, detailed conception and documentation of the system.

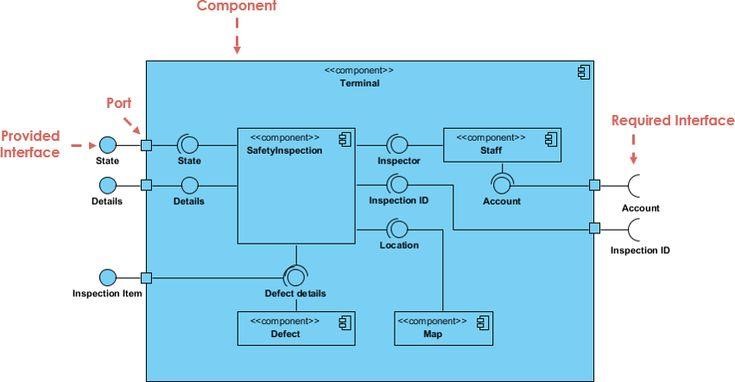
## PRELIMINARY DESIGN

### 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.

#### Formalism



*Figure 20:Formalism of a component diagram (Source: https://www.pinterest.com/pin/551128073157994549/)*

*Table 17: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. |
| **interface** |  | An interface is a circle or a semicircle attached to a stick which looks like a lollipop. It describes groups of operations provided or required by components. |
| **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. |

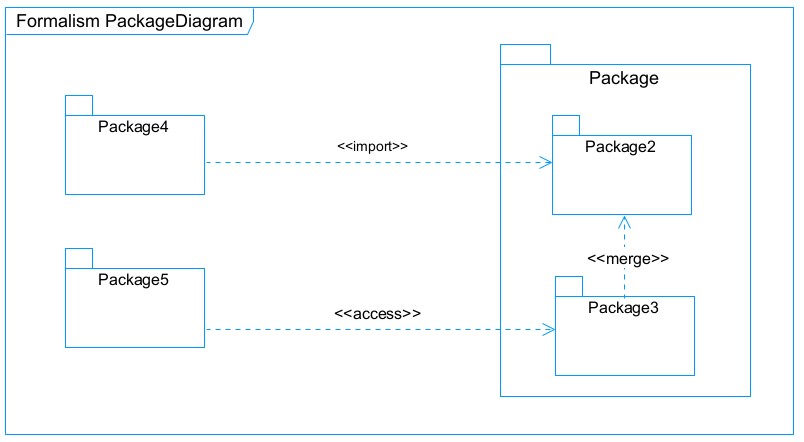
#### System component diagram

### 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 eve other packages.

#### Formalism



*Figure 21:Package Diagram Formalism*

*Figure 22:Package Diagram Components*

|  |  |  |
| --- | --- | --- |
| **NAME** | **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. |

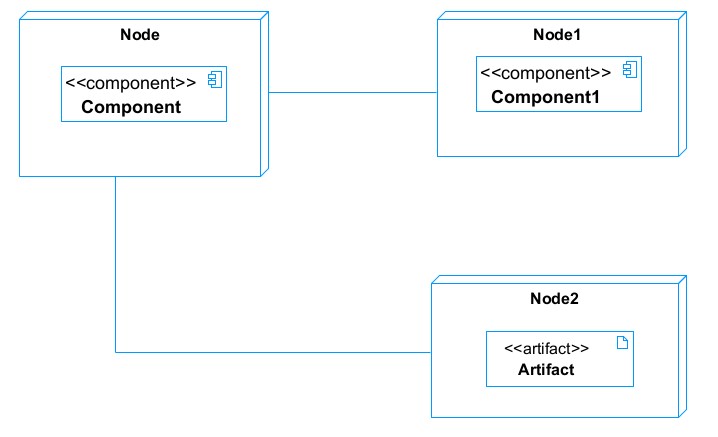
#### Mobile package Diagram

### 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.

#### Formalism



*Figure 25:Deployment Diagram Formalism*

*Figure 26:Deployment Diagram components*

|  |  |  |
| --- | --- | --- |
| **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** |  | An association helps to connect two nodes together which permits them to communicate  together |

#### System Deployment Diagram

## DETAILED DESIGN

# **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 taking into account 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 FIVE: REALISATION PHASE**

SUMMARY

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

Preview

**INTRODUCTION**

1. **THE ENTITY REALTIONAL DIAGRAM**
2. **PRESENTATION OF DEVELOPMENT TOOLS**

**CONCLUSION**

# **INTRODUCTION**

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

# ENTITY RELATIONSHIP DIAGRAM

# PRESENTATION OF DEVELOPMENT TOOLS

## MATERIAL RESOURCE

## TECHNOLOGICAL STACK

# **CONCLUSION**

In our realization phase, we implemented our application. In other to accomplish this phase, we made used of our analysis and conception phase. The Entity-Relational diagram. helped us to establish our database. We then presented the development tools we used for the development or realization of our system, we had material resource (visual studio code) and technology stack (React js, react-native js, Node js/express js). We will move to the user guide phase where we shall elaborate on how to install and use

# **PART SIX: USER GUIDE**

SUMMARY

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.

Preview

**INTRODUCTION**

1. **INSTALLATION OF THE APPLICATION**
2. **SHOW CASE**

**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 it futures, all this in a step by step manner to facilitate the setting up of the platform for the first time users. The steps of different processes will be accompanied by images. After we will showcase our application by viewing the different screens and some brief explanation.

# INSTALLATION OF THE APPLICATION

# SHOWCASES

# 

# **GENERAL CONCLUSION**

# 

# **ANNEXE**

# 

# **BIBLIOGRAHY**

# 

# **WEBOGRAPHY**

# 

# **TABLE OF CONTENT**

[**DEDICATION** 1](#_Toc173689683)

[**ACKNOWLEDGMENTS** 2](#_Toc173689684)

[**GLOSSARY** 5](#_Toc173689685)

[**ABSTRACT** 7](#_Toc173689686)

[**RESUME** 7](#_Toc173689687)

[**GENERAL INTRODUCTION** 7](#_Toc173689688)

[**PART I: INSERTION PHASE** 8](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689689)

[**INTRODUCTION** 10](#_Toc173689690)

[I. WELCOME AND INTERGRATION 10](#_Toc173689691)

[1. Welcome 10](#_Toc173689692)

[2. Integration 10](#_Toc173689693)

[II. GENERAL PRESENTATION OF WELLDONE PLANET 11](#_Toc173689694)

[1. History and Missions 11](#_Toc173689695)

[2. Presentation of the structure 12](#_Toc173689696)

[b. Material Safety Data Sheet 12](#_Toc173689697)

[C. Resources 13](#_Toc173689698)

[3. Visions, Objectives of Welldone Planet 16](#_Toc173689699)

[b. Goals 17](#_Toc173689700)

[c. Attributions 19](#_Toc173689701)

[**CONCLUSION** 20](#_Toc173689702)

[**PART TWO: SPECIFICATION BOOK** 21](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689703)

[**INTRODUCTION** 23](#_Toc173689704)

[I. CONTEXT AND JUSTIFICATION 24](#_Toc173689705)

[II. OBJECTIVES OF THE PROJECT 25](#_Toc173689706)

[A. General Objective 25](#_Toc173689707)

[B. Specific Objectives 25](#_Toc173689708)

[III. EXPRESSION OF NEEDS 25](#_Toc173689709)

[A. Functional needs 25](#_Toc173689710)

[B. Non-functional Needs 25](#_Toc173689711)

[IV. ESTIMATED COST OF THE PROJECT 26](#_Toc173689712)

[A. Hardware Resources 26](#_Toc173689713)

[B. Software Resources 27](#_Toc173689714)

[C. Human Resources 28](#_Toc173689715)

[D. Total Project Estimated Cost 29](#_Toc173689716)

[V. PROJECT PLANING 29](#_Toc173689717)

[A. Chronogram of activities 29](#_Toc173689718)

[B. Gantt Chart 30](#_Toc173689719)

[VI. LIST OF PARTICIPANTS AND DELIVERABLES 31](#_Toc173689720)

[A. List of participants 31](#_Toc173689721)

[B. Deliverable 31](#_Toc173689722)

[**CONCLUSION** 32](#_Toc173689723)

[**PART THREE: ANALYSIS PHASE** 33](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689724)

[**INTRODUCTION** 36](#_Toc173689725)

[I. DESCRIPTION OF THE EXISTING SYSTEM 36](#_Toc173689726)

[II. PRESENTATION OF THE ANALYSIS 36](#_Toc173689727)

[A. MODELING WITH UML 2.5 38](#_Toc173689728)

[a) Structural Diagrams : 38](#_Toc173689729)

[b) Behavioral Diagrams : 39](#_Toc173689730)

[c) UML 2.5 diagrams overviews 40](#_Toc173689731)

[B. THE PROCESS OF DEVELOPMENT OF A SOFTWARE 41](#_Toc173689732)

[a) A Unified Process 41](#_Toc173689733)

[b) The Two Track Unified Process (2TUP) 41](#_Toc173689734)

[III. JUSTIFICATION OF THE ANALYSIS 42](#_Toc173689735)

[a) The left branch (functional branch) 43](#_Toc173689736)

[b) The right branch (Technical branch) 43](#_Toc173689737)

[c) The middle branch (Realization or implementation branch) 43](#_Toc173689738)

[IV. MODELING OF THE SYSTEM 44](#_Toc173689739)

[A. CAPTURE OF FUNCTIONAL NEEDS 44](#_Toc173689740)

[1. *Use Case Diagram* 44](#_Toc173689741)

[2. ACTIVITY DIAGRAM 48](#_Toc173689742)

[3. State Machine Diagram 51](#_Toc173689743)

[**CONCLUSION** 53](#_Toc173689744)

[**PART FOUR: CONCEPTION PHASE** 54](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689745)

[**INTRODUCTION** 56](#_Toc173689746)

[I. TECHNICAL BRANCH 56](#_Toc173689747)

[A. GENERIC DESIGN 56](#_Toc173689748)

[a) Hardware diagram of the system 56](#_Toc173689749)

[High Level Architectural Diagram of the Software 56](#_Toc173689750)

[B. CAPTURES TECHNICAL NEEDS 57](#_Toc173689751)

[a) Physical architecture 57](#_Toc173689752)

[Logical architecture 57](#_Toc173689753)

[II. IMPLEMENTATION BRANCH 58](#_Toc173689754)

[A. PRELIMINARY DESIGN 58](#_Toc173689755)

[a) Component diagram 58](#_Toc173689756)

[Package Diagram 61](#_Toc173689757)

[Deployment Diagram 63](#_Toc173689758)

[B. DETAILED DESIGN 65](#_Toc173689759)

[**CONCLUSION** 66](#_Toc173689760)

[**PART FIVE: REALISATION PHASE** 67](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689761)

[**INTRODUCTION** 69](#_Toc173689762)

[I. ENTITY RELATIONSHIP DIAGRAM 70](#_Toc173689763)

[II. PRESENTATION OF DEVELOPMENT TOOLS 70](#_Toc173689764)

[A. MATERIAL RESOURCE 70](#_Toc173689765)

[B. TECHNOLOGICAL STACK 70](#_Toc173689766)

[**CONCLUSION** 70](#_Toc173689767)

[**PART SIX: USER GUIDE** 71](file:///I:\Sauvegarde%20Soutenance\ID%20app\Rapport%20de%20stage\Internship%20report.docx#_Toc173689768)

[**INTRODUCTION** 73](#_Toc173689769)

[I. INSTALLATION OF THE APPLICATION 74](#_Toc173689770)

[II. SHOWCASES 74](#_Toc173689771)

[**GENERAL CONCLUSION** 74](#_Toc173689772)

[**ANNEXE** 74](#_Toc173689773)

[**BIBLIOGRAHY** 74](#_Toc173689774)

[**WEBOGRAPHY** 74](#_Toc173689775)

[**TABLE OF CONTENT** 74](#_Toc173689776)

1. **IPA (iOS APPLICATION SOFTWARE PACKAGE)** An .ipa file is an iOS application archive file which stores an iOS app. [↑](#footnote-ref-1)
2. **APK (ANDRIOD APPLICATION PACKAGE)** It is a file format used by android operating system for

   the distribution and installation of mobile applications. [↑](#footnote-ref-2)