# DEDICATION

# DEDICATION

THIS WORK IS DEDICATED TO MY FAMILY. FOR THEIR LOVE, SUPPORT AND ENCOURAGEMENT TOWARDS MY ACADEMIC SUCCESS.

ACKNOWLEDGEMENTSTHIS WORK IS DEDICATED TO MY FAMILY. FOR THEIR LOVE, SUPPORT AND ENCOURAGEMENT TOWARDS MY ACADEMIC SUCCESS.

# ACKNOWLEDGEMENTS

# CONTENTACKNOWLEDGEMENTS

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.
* Our academic supervisor **Mr. MOHAMADOU Alhadji** for her normal support, effort and advices throughout through-out the academic year.
* To all my teachers of AICS CAMEROON for the knowledge they impacted me with for me to be where I am today.
* For my brother and sisters for their love and support (**ABENG Irene Vanessa**, **NDOUM Assembe Gaetan AND MBENGON Neily Linda**).
* The countless contributors of open-source programming community, for their great help in learning basic skills and detecting and solving bugs.
* To my mother and father (**MENGBWA Germaine and AKONO Jean-Marcel**) for their financial and moral support.
* To all my classmates for their collaborative work through-out the academic year.

# CONTENT

# LIST OF FIGURESCONTENT

[DEDICATION i](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005067)

[DEDICATION i](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005068)

[ACKNOWLEDGEMENTS ii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005069)

[CONTENTACKNOWLEDGEMENTS ii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005070)

[CONTENT iii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005071)

[LIST OF FIGURESCONTENT iii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005072)

[LIST OF FIGURES v](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005073)

[LIST OF TABLESLIST OF FIGURES v](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005074)

[LIST OF TABLES vi](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005075)

[ABBREVRATIONLIST OF TABLES vi](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005076)

[GLOSSARY vii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005077)

[ABSTRACTABBREVRATION vii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005078)

[ABSTRACT viii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005079)

[RESUMEABSTRACT viii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005080)

[RESUME ix](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005081)

[GENERAL INTRODUCTIONRESUME ix](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005082)

[GENERAL INTRODUCTION 1](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005083)

[CHAPTER 1: EXISTING SYSTEMGENERAL INTRODUCTION 1](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005084)

[PART 1: EXISTING SYSTEM 2](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005085)

[PART 2: SPECIFICATION BOOK 7](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005086)

[PART 3: ANALYSIS PHASE 19](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005087)

[B. CHOICE OF THE ANALYSIS METHOD 27](#_Toc171005088)

[C. MODELLING OF THE PROPOSED SOLUTION 28](#_Toc171005089)

[CHAPTER 4: CONCEPTION PHASE 36](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005090)

[PART 4: CONCEPTION PHASE 46](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005091)

[PART 5: REALISATION PHASE 61](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005092)

[CONCLUSION 68](#_Toc171005093)

[CHAPTER 6: TEST OF FUNCTIONALITIES 69](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005094)

[INTRODUCTION 71](#_Toc171005095)

[CONCLUSION 74](#_Toc171005096)

[PART 7: USER GUIDE 75](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005097)

[INTRODUCTION 77](#_Toc171005098)

[CONCLUSION 81](#_Toc171005099)

[PERSPECTIVES 82](#_Toc171005100)

[GENERAL CONCLUSION 83](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005101)

[ANNEXES 84](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005102)

[BIBLIOGRAPHY 85](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005103)

[WEBOGRAPHY 86](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005104)

[TABLE OF CONTENT 87](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005105)

# LIST OF FIGURES

# LIST OF TABLESLIST OF FIGURES

[Figure 1: Gantt Chart 14](#_Toc171005106)

[Figure 2; UML 2.5 diagrams overview(source: https://creately.com/blog/diagrams/uml-diagram-types-examples/) 23](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005107)

[*Figure 3: 2TUP diagram (source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based on-cloud-computing).* 26](#_Toc171005108)

[Figure 4:Use Case Diagram formalism 29](#_Toc171005109)

[Figure 5: System Use Case Diagram 32](#_Toc171005110)

[Figure 6: Communication Diagram Formalism 34](#_Toc171005111)

[Figure 7: Authentication Sequence Diagram 38](#_Toc171005112)

[Figure 8: Sequence Diagram for job application 39](#_Toc171005113)

[Figure 9: Formalism of Activity diagram 40](#_Toc171005114)

[Figure 10: Authentication Activity Diagram 43](#_Toc171005115)

[Figure 11: Job Application Diagram 44](#_Toc171005116)

[Figure 12: Structure of an application Source:WWW.APPVENTUREZ.COM 49](#_Toc171005117)

[Figure 13: MVT Architecture 51](#_Toc171005118)

[Figure 14: Formalism of Class Diagram 52](#_Toc171005119)

[Figure 15: Account state machine diagram 57](#_Toc171005120)

[Figure 16: Job application state machine diagram 58](#_Toc171005121)

[Figure 17: Formalism of package diagram 58](#_Toc171005122)

[Figure 18: System package Diagram 60](#_Toc171005123)

[Figure 19: System Deployment Diagram 65](#_Toc171005124)

[Figure 20: Formalism of component diagram (Source: https://www.pinterest.com/pin/551128073157994549/) 66](#_Toc171005125)

[Figure 21: System Component Diagram 67](#_Toc171005126)

[Figure 22: Unit Testing 73](#_Toc171005127)

[Figure 23: Django Administration login 77](#_Toc171005128)

[Figure 24: Job Listing page 78](#_Toc171005129)

[Figure 25: register page 78](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005130)

[Figure 26: Login page 79](#_Toc171005131)

[Figure 27: Job application page 79](#_Toc171005132)

[Figure 28: User Profile page 80](#_Toc171005133)

# LIST OF TABLES

# ABBREVRATIONLIST OF TABLES

[Tableau 1: Project Planning 13](#_Toc171005134)

[Tableau 2: Hardware Resources of the Estimated Cost of the Project(Source: https://www.scribd.com/document/561202966/mercuriale-2022) 14](#_Toc171005135)

[Tableau 3: Software Resources (https://www.g2.com/categories/pricing) 15](#_Toc171005136)

[Tableau 4: Human Resource (Source: https://www.paylab.com/) 16](#_Toc171005137)

[Tableau 5: Total Project Estimated Cost 16](#_Toc171005138)

[Tableau 6: List of participants 17](#_Toc171005139)

[Tableau 7: Differences between UML and Merise 24](#_Toc171005140)

[Tableau 8: Use Case Diagram Component 29](#_Toc171005141)

[Tableau 9: Use Case Diagram Component 31](#_Toc171005142)

[Tableau 10: Authenticate Textual Description 33](#_Toc171005143)

[Tableau 11: Elements of a communication diagram 34](#_Toc171005144)

[Tableau 12: Elements of a sequence Diagram 37](#_Toc171005145)

[Tableau 13: Elements of activity diagramdiagram 40](#_Toc171005146)

[Tableau 14: Elements of class diagram 52](#_Toc171005147)

[Tableau 15: Elements of state machine diagram 56](#_Toc171005148)

[Tableau 16: Elements of a Package Diagram 59](#_Toc171005149)

[Tableau 17: Elements of Deployment Diagram 64](#_Toc171005150)

[Tableau 18: Elements of a component diagram 66](#_Toc171005151)

# GLOSSARY

# ABSTRACTABBREVRATION

* 2TUP: Two Track Unified Process.
* AICS: African Institute of Computer Sciences.
* APK: Android Package Kit.
* IPA: iOS App Store Package.
* MVT: Model View Template.
* UML: Unified Modelling Language.

# ABSTRACT

# RESUMEABSTRACT

The "Job Board Platform" project aims to develop an efficient and user-friendly online platform to connect job seekers and employers. This platform addresses the challenges faced by both parties in the job search and hiring process, providing a seamless and integrated experience. Employers can post job vacancies, search for suitable candidates, and manage applications, while job seekers can create profiles, upload resumes, search for jobs, and apply to openings.

The platform includes features such as real-time job notifications, advanced search filters, and a robust recommendation system to match job seekers (which still need to be implemented) with relevant opportunities. Additionally, it offers tools for scheduling interviews, tracking application statuses, and providing feedback.

The development of this project utilizes the Django framework for the backend, PostgreSQL for the database, and a combination of HTML, CSS, and JavaScript for the frontend. The system ensures data security and privacy, providing a reliable and scalable solution for the job market.

This project aims to bridge the gap between employers and job seekers, streamlining the recruitment process, and enhancing the overall user experience. The platform is designed to be adaptable and scalable to accommodate the evolving needs of the job market.

# RESUME

# GENERAL INTRODUCTIONRESUME

Le projet "Plateforme de Recherche d'Emploi" vise à développer une plateforme en ligne efficace et conviviale pour connecter les chercheurs d'emploi et les employeurs. Cette plateforme répond aux défis rencontrés par les deux parties dans le processus de recherche d'emploi et de recrutement, offrant une expérience intégrée et fluide. Les employeurs peuvent publier des offres d'emploi, rechercher des candidats qualifiés et gérer les candidatures, tandis que les chercheurs d'emploi peuvent créer des profils, télécharger des CV, rechercher des emplois et postuler aux offres.

La plateforme inclut des fonctionnalités telles que des notifications d'emploi en temps réel, des filtres de recherche avancés et un système de recommandation robuste pour mettre en relation les chercheurs d'emploi avec des opportunités pertinentes. De plus, elle propose des outils pour programmer des entretiens, suivre le statut des candidatures et fournir des retours d'information.

Le développement de ce projet utilise le Framework Django pour le backend, PostgreSQL pour la base de données, et une combinaison de HTML, CSS et JavaScript pour le frontend. Le système assure la sécurité et la confidentialité des données, offrant une solution fiable et évolutive pour le marché de l'emploi.

Ce projet vise à combler le fossé entre les employeurs et les chercheurs d'emploi, en rationalisant le processus de recrutement et en améliorant l'expérience utilisateur globale. La plateforme est conçue pour être adaptable et évolutive afin de répondre aux besoins changeants du marché de l'emploi.

# GENERAL INTRODUCTION

# CHAPTER 1: EXISTING SYSTEMGENERAL INTRODUCTION

In today's world, mobile devices are integral to our lives, serving various purposes, including educational objectives. Our case study focuses on the development of a job board platform for third-year students at the AICS Cameroon. This personal project aims to create the "JOB BOARD PLATFORM, Case Study AICS Cameroon." This platform will enable students and graduates to search for job opportunities in real-time and help employers post and manage job listings more efficiently. To achieve these objectives, we followed six main sections or phases:

1. **Existing System:** Here, we shall present the already present system in place, that is the one used for consultation and follow-up purposes.
2. **Specification Book**: Identifying the needs of future system users and highlighting the different constraints of the project.
3. **Analysis Phase**: Choosing our analysis method and presenting all the diagrams used for the project analysis.
4. **Design Phase**: Presenting the generic and detailed design of the project to bring out real-world components.
5. **Implementation Phase**: Presenting the choice of technologies and techniques necessary for implementing our solution.
6. **User Guide**: Providing a user-friendly and graphical description of each functionality of the application.

# PART 1: 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. CRITICISM OF THE EXISTING SYSTEM
4. PROBLEMATICS
5. PROPOSED SOLUTION

CONCLUSION

### Introduction

In the realm of job search and recruitment, traditional and existing systems in Cameroon often struggle to meet the evolving needs of job seekers and employers. This phase of our report delves into the study of an existing job platform, analyzes its shortcomings, and proposes an enhanced solution through our project, the "REAL-TIME JOB BOARD PLATFORM" for AICS Cameroon.

### Presentation of the Theme

The theme of this study revolves around the development of an online job board platform that caters specifically to the students and graduates of AICS Cameroon. The goal is to provide real-time job opportunities and a streamlined process for employers to post and manage job listings, thereby bridging the gap between education and employment.

### Study of the Existing System

The existing system we studied is a simple job platform commonly used in Cameroon. This platform allows employers to post job vacancies and job seekers to browse and apply for these opportunities. While it provides basic functionality, it has several limitations that affect its efficiency and user satisfaction.

#### Functionality

1. **Job Posting**: Employers can create job listings with details about the position, requirements, and application process.
2. **Job Search**: Job seekers can search for job opportunities using various filters such as location, industry, and job type.
3. **Application Process**: Job seekers can apply for jobs directly through the platform by submitting their resumes and cover letters.

### Criticism of the Existing System

While the existing job platform serves its basic purpose, it has notable shortcomings:

1. **Lack of Real-Time Updates**: Job listings and application statuses are not updated in real-time, causing delays and misinformation.
2. **Limited User Interaction**: The platform lacks interactive features such as notifications, messaging, and feedback mechanisms between employers and job seekers.
3. **Inadequate Filtering Options**: The search and filter functionalities are basic, making it difficult for users to find relevant job opportunities.
4. **Manual Processes**: Many processes, such as verifying job postings and managing applications, are manual, leading to inefficiencies and errors.
5. **Poor User Interface**: The user interface is outdated and not user-friendly, affecting the overall user experience.

### Problematics

Given the criticisms of the existing system, several problems have been identified:

1. **Delayed Information**: The absence of real-time updates results in outdated job listings and application statuses, leading to frustration among users.
2. **Inefficient Communication**: The lack of interactive features hinders effective communication between employers and job seekers.
3. **Difficulty in Finding Relevant Jobs**: Limited filtering options make it challenging for job seekers to find suitable job opportunities quickly.
4. **Time-Consuming Manual Processes**: Manual handling of job postings and applications increases the workload for administrators and introduces the potential for errors.
5. **Unappealing User Experience**: An outdated and unfriendly user interface discourages continued use of the platform.

### Proposed Solution

To address the problems identified in the existing system, we propose the development of the "REAL-TIME JOB BOARD PLATFORM" for AICS Cameroon. Key features of the proposed solution include:

1. **Real-Time Updates**: Implement real-time updates for job listings and application statuses to ensure users receive the most current information.
2. **Enhanced User Interaction**: Introduce interactive features such as notifications, messaging, and feedback mechanisms to improve communication between employers and job seekers.
3. **Advanced Filtering Options**: Develop comprehensive search and filter functionalities to help job seekers find relevant opportunities efficiently.
4. **Automated Processes**: Automate processes such as job posting verification and application management to reduce administrative workload and minimize errors.
5. **Modern User Interface**: Design a user-friendly and visually appealing interface to enhance the overall user experience.

### Conclusion

By addressing the limitations of the existing system, our proposed platform aims to provide a more efficient, interactive, and user-friendly experience for both job seekers and employers. This will ultimately facilitate better job matching and improve the employment outcomes for students and graduates of AICS Cameroon.

# PART 2: 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 as the conditions necessary to realize the project.

Content overview

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.

## A. CONTEXT AND JUSTIFICATION

In the modern age, the efficiency and productivity of an organization are paramount. For the AICS Cameroon, connecting students and graduates with potential employers effectively is a crucial step towards ensuring successful career transitions and fostering economic growth. This project aims to address the challenges faced by the current job platforms in Cameroon, specifically tailored to the needs of AICS Cameroon.

The main issue identified is the lack of a centralized, real-time job board platform that provides accurate and up-to-date information about job opportunities and application statuses. Current systems fail to deliver real-time updates, interactive features, and efficient data management, leading to delays, misinformation, and a subpar user experience.

This problem is significant because timely and accurate information can greatly influence job seekers' chances of securing employment and employers' ability to find suitable candidates. Without a reliable system, students and graduates may miss out on opportunities, and employers may struggle to fill positions promptly, ultimately affecting the overall productivity and economic growth.

#### Key Questions Addressed:

* **How can we centralize all job-related data and activities for AICS Cameroon in real-time?**
* **How can we manage job postings and applications efficiently and accurately?**
* **How can we provide real-time updates on job statuses and application progress?**
* **How can we improve the communication between job seekers and employers?**
* **How can we ensure a user-friendly interface that encourages continuous use and satisfaction?**

## B. OBJECTIVES OF THE PROJECT

### General Objective

The primary objective of this project is to develop a job board platform that enables real-time access to job opportunities and application statuses for students and graduates of AICS Cameroon. This platform aims to streamline the job search process, enhance communication between job seekers and employers, and provide accurate, up-to-date information.

### Specification Objectives

 **Real-Time Job Posting and Updates:** The platform will allow employers to post job openings and provide real-time updates on application statuses, ensuring that job seekers have the most current information.

 **Enhanced Communication:** The system will facilitate direct communication between job seekers and employers, enabling prompt responses to inquiries and applications.

 **Comprehensive Job Search Tools:** Users will have access to advanced search and filter options, making it easier to find relevant job opportunities based on various criteria such as location, industry, and experience level.

 **Detailed Job Analytics:** The platform will provide analytics for employers to track the performance of job postings, including the number of views, applications received, and the progression of candidates through the hiring process.

 **Access Control and User Management:** The system will include features to limit access to sensitive information, ensuring that only authorized users (e.g., university administrators and employers) can view and manage certain data.

 **Profile Management:** Job seekers will be able to create and manage their profiles, upload resumes, and track their application history, making it easier to apply for multiple jobs and monitor their job search progress.

 **Integration with University Services:** The platform will integrate with existing university services, such as career counseling and alumni networks, to provide additional support and resources to job seekers.

 **Feedback and Rating System:** Employers will be able to provide feedback on candidates, and job seekers can rate their experience with employers, contributing to a transparent and accountable job market.

## C. EXPRESSION OF NEEDS

## Functional Needs

Functional needs describe the capabilities and actions that users can perform within the system. The following are identified functional needs for the job board platform:

* **Job Alert System:** Implement a notification system to alert job seekers about new job postings matching their preferences and qualifications.
* **User Account Management:** Provide administrative capabilities to manage user accounts, including adding, deleting, editing, and assigning privileges to users such as employers, recruiters, and job seekers.
* **Real-Time Job Posting and Updates:** Enable employers to post job vacancies in real-time and update job status as applications are received and processed.
* **Advanced Search and Filtering:** Offer job seekers advanced search and filtering options based on criteria such as job title, location, industry, and required skills.
* **Application Tracking:** Allow job seekers to track their applications, view status updates, and receive notifications about changes in application status.
* **Analytics and Reporting:** Provide employers with analytics and reporting tools to track job posting performance, including application metrics and candidate demographics.

## Non-Functional Needs

Non-functional needs define constraints and requirements that shape how the system performs and interacts with users:

### **Hardware Requirements:**

* + **Portability:** Ensure the platform is compatible and accessible across different devices and operating systems without compatibility issues.

### **Software Requirements:**

* + **Accessibility:** Design the platform to be user-friendly and accessible for all users, including individuals with disabilities.
  + **Documentation:** Provide comprehensive user manuals and documentation to guide users on how to navigate and utilize the platform effectively.
  + **Extensibility:** Design the system architecture to support integration with other systems and future enhancements without disrupting existing functionalities.
  + **Response Time:** Ensure that the system responds to user requests promptly, aiming for a minimal response time of under 1 second for accessing critical resources.
  + **Scalability:** Architect the system to accommodate growth and increased usage, allowing for the addition of new features and functionalities without compromising performance.
  + **Security:** Implement robust security measures to protect sensitive information, including encryption of data during transmission and setting access controls to restrict unauthorized access to system resources.

D. PROJECT PLANNING

Tableau 1: Project Planning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE OBJECTIVE | | OUTPUT | DURATION | PERIOD |
| EXISTING SYSTEM | Study of the existing system | Existing System | 5 days | 17th May to  21st May |
| SPECIFICATION BOOK | Specification of the user needs | Specification Book | 5 days | 24th May to  28th May |
| ANALYSIS | Capture of needs Use case and textual description  Modelling | Analysis Book | 2 weeks | 31st May to 11th June |
| CONCEPTION | Preliminary conception and  Detailed conception | Conception book | 2 weeks | 14th June to 25th  june |
| REALIZATION | Implementation Unitary test  Integration Test  Development,  Deployment,  Component diagrams | Realization book | 3 weeks | 28th June to o4th  July |
| 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 |

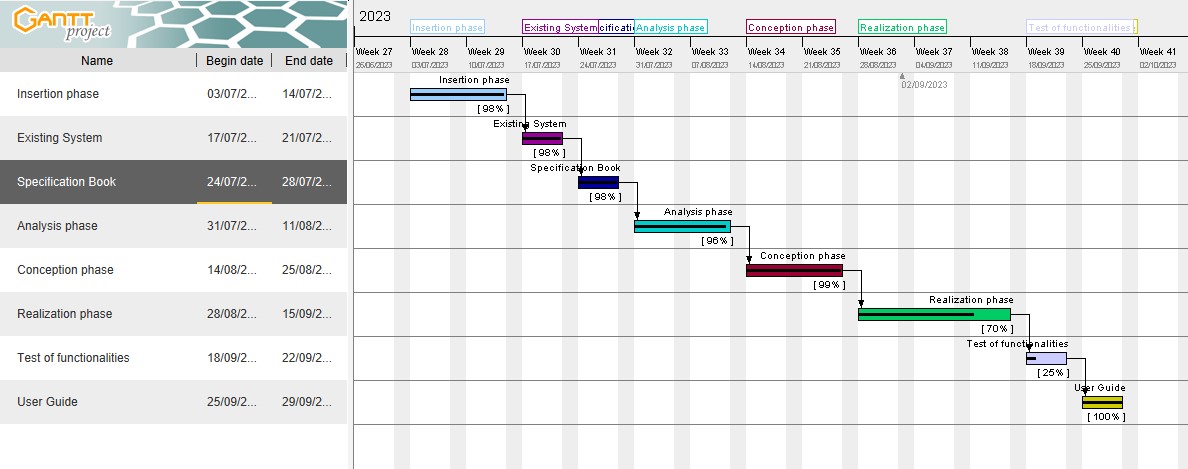


Figure 1: Gantt Chart

## E. ESTIMATED COST OF THE PROJECT

### Hardware Resources

Tableau 2: Hardware Resources of the Estimated Cost of the Project(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 E6520, 8GB RAM, 500GB hard disk; intel core i5. | main resource for the  accomplishment of this project; report writing, designs, analysis, coding | 1 | 368000 |
| 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 |  |  |  | **400,325** |

### Software Resources

Tableau 3: 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 |
| Operating system | Microsoft windows 10 pro | Computer operating system | 1 | 103000 |
| Text Editor | Microsoft office word 2016 | For typing and formatting of  report | 1 | 161000 |
| Presentation | Microsoft office PowerPoint 2016 | For making  presentations | 1 | 84230 |
| Modelling tool | Visual-paradigm Modeler version | Modelling the system in uml | 1 | 3350 |
| 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 |  |  |  | **352,580** |

### Human Resources

Tableau 4: 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

Tableau 5: Total Project Estimated Cost

|  |  |  |  |
| --- | --- | --- | --- |
| Total 1(FCFA) Total 2(FCFA) | | Total 3(FCFA) | Overall total (FCFA) |
| 400,325 | **352580** | **25950000** | **26701905** |
| Margin of error = 10% of Overall total  =2670190.5 FCFA  Final Total = Overall Total + margin of Error  =29547586.5 FCFA | |  |  |

## F. CONSTRAINTS OF THE PROJECT

### Time Constraint

The project will be realized in 12 weeks (3months) starting from the beginning date 7th January to the 29 May 2024.

### Cost Constraint

The realization of our project will require expenditure in human resources, materials and software a total cost **29547586.5 FCFA.**

## G. LIST OF THE PARTICIPANTS AND DELIVERABLES

### List of participants

Tableau 6: List of participants

|  |  |
| --- | --- |
| Participants | Functions |
| Mr. MOHAMADOU ALHADJI | Lecturer at AICS-Cameroon |
| MINSILI MINETTE DIANE | Project manager and developer |

### Deliverables

* A user guides.
* A power point of the application.

## CONCLUSION

The specification book helps us to site the different needs we need in order to implement our application for stock management 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 3: ANALYSIS PHASE

Preamble

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.

Content overview

INTRODUCTION

I.

METHODOLOGY

1.

COMPARATIVE STUDY OF UML

AND MERISE.

2.

COMPARATIVE

STUDY

OF

UNIFIED PROCESSES.

II.

CHOICE OF THE ANALYSIS METHOD

III.

MODELLING OF THE

PROPOSED

SOLUTION

CONCLUSION

## INRODUCTION

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

### What is a methodology?

It is the systematic and theoretical process so that all the methods and principles associated with the field of knowledge is explored well. This process collects information and data about the particular subject and ponders it well, analysis is made and methods to find the solution is found.

## Comparatives between 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,
* Composite Structure diagram,
* Package diagram,
* Profile Diagram,

## BEHAVIOURAL OR DYNAMIC DIAGRAMS

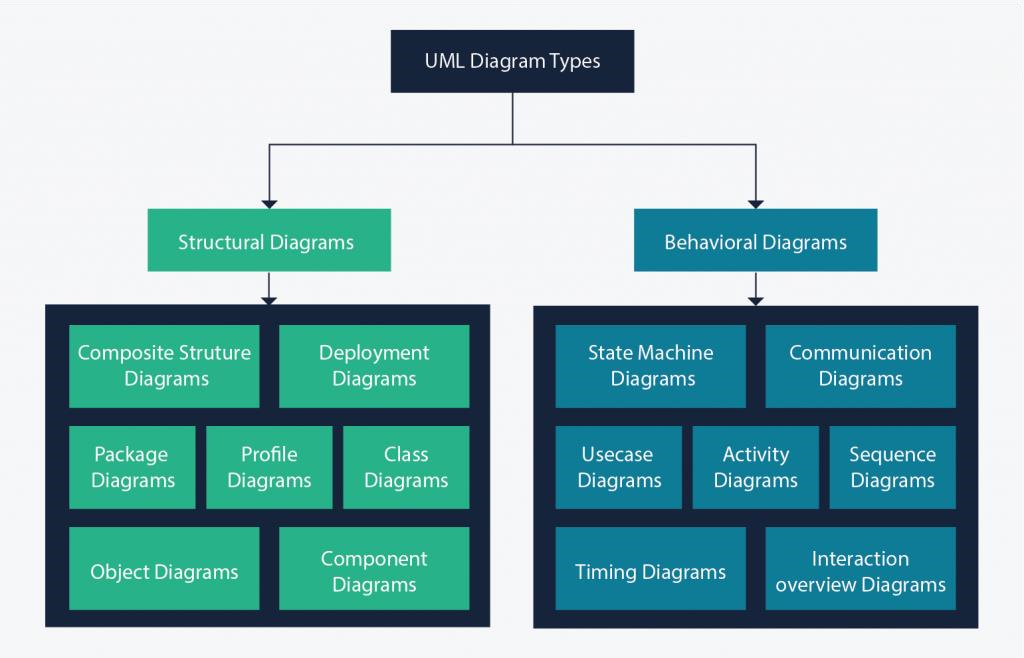
* Use case diagrams;
* Activity diagram;
* State machine diagram;
* Sequence diagram;
* Communication diagram;
* Global Interaction diagram;
* Timing Diagram.

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

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.

Tableau 7: Differences between UML and Merise

|  |  |
| --- | --- |
| 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 |

B. COMPARATIVE STUDY OF UNIFIED PROCESSES

### a. A Unified Process

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

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

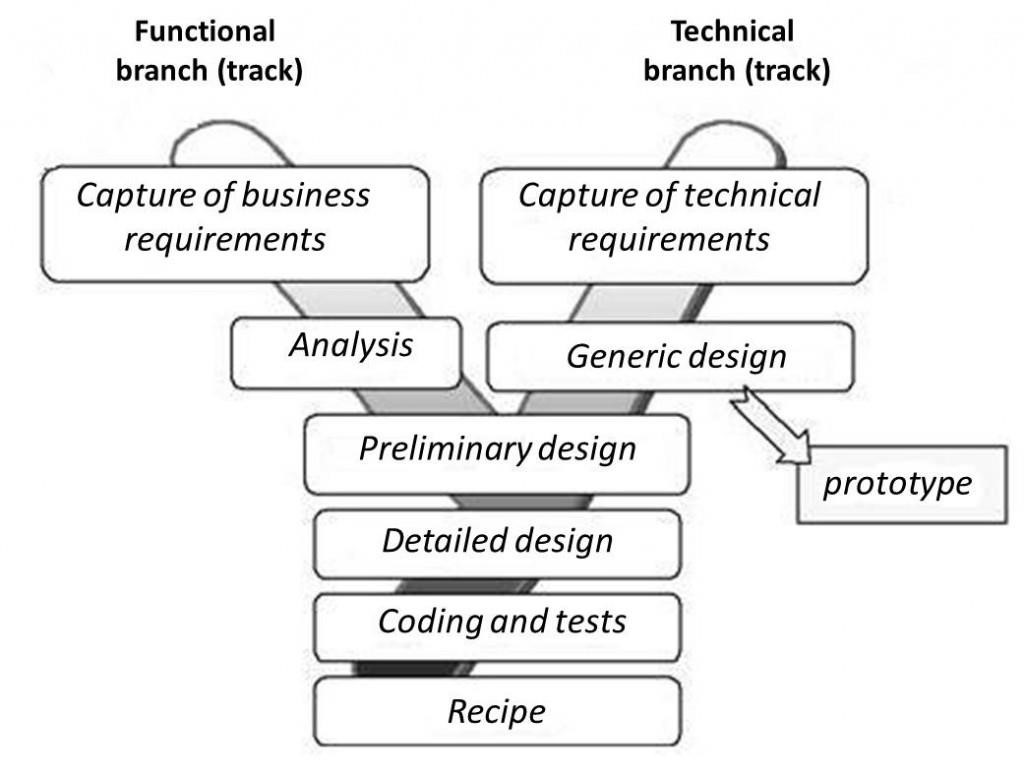
**An increment** is the difference between two released products at the end of two iterations. Each iteration that the group is capable of integrating the technical environment in order to develop a final product and give users the possibility of having tangible results.

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

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

### b. 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 3: 2TUP diagram (source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based on-cloud-computing).*

## **The left branch (functional branch)**

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

## **b. The right branch (Technical branch)**

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

### c. The middle branch (Realization or Implementation branch)

In this branch, we study the preliminary conception, detailed conception, and documentation of the system. The realization branch supports the following: **Preliminary conception:** This is the most sensitive step of 2TUP as it is the confluence of the functional and technical branch. It is completed when the deployment model, the operating model, the logical model, interphases and the software configuration model are defined. We have the following diagrams:

 Component Diagram ;

 Deployment Diagram ;

 Package Diagram ;

 Composite Structure Diagram ;

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

Class Diagram;



 Object Diagram;

 Sequence Diagram;

Timing Diagram;



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

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

# 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.

# C. MODELLING OF THE PROPOSED SOLUTION

A. Capture of Functional Needs

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

## 1. USE CASE DIAGRAM

## a) Definition

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

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

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

## b) Formalism

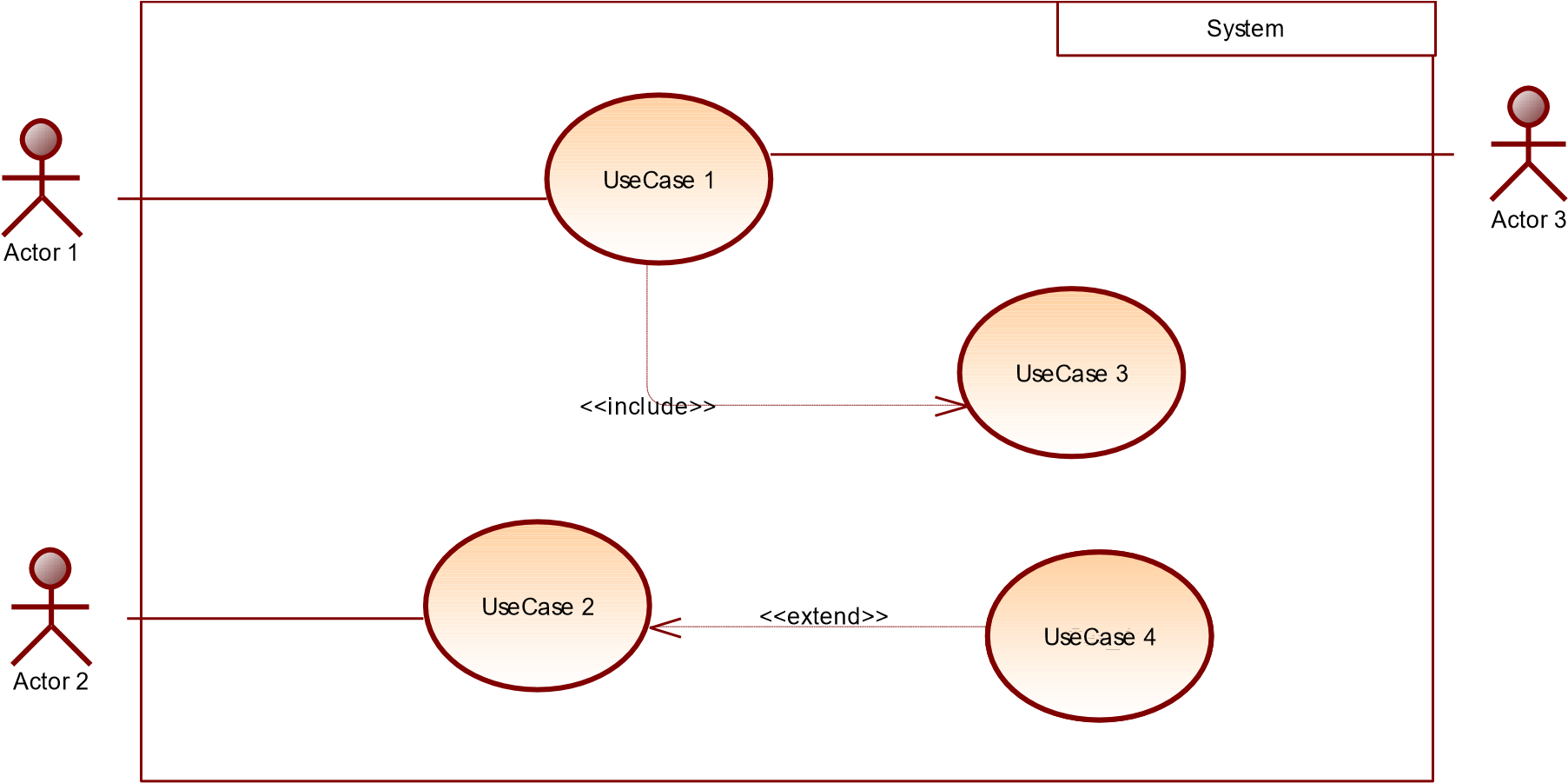


Figure 4:Use Case Diagram formalism

Tableau 8: 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 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 |

### Actors of our System

Tableau 9: Use Case Diagram Component

|  |  |
| --- | --- |
| Actor | Role |
| Administrator | The administrator is responsible of managing the accounts of jobseekers and health emploers (validating, deleting accounts) and general functioning of the system. |
| Super Administrator | He also manages the whole system and all the users (including the administrator) |
| Jobseeker | The user that views and applies for jobs. |
| Employer | The user that posts job opportunities, and views different applicants |

### General Use case Diagram



Figure 5: System Use Case Diagram

### Textual description of use case

Tableau 10: Authenticate Textual Description

|  |  |
| --- | --- |
| Title | Authenticate |
| Summary | The user needs to authenticate |
| Actors | Jobseeker, Employer, Admin |
| Date | May 11, 2024 |
| Stakeholder | MiBoard Directors |
| 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 |

## *2.* COMMUNICATION DIAGRAM

## **a. Definition**

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

* 1. **Formalism**

11

Message\_

8.3:

12

Message\_

8.4:

13

8.5:

Message\_



Objet\_1



Obje

t\_2



Objet\_3



Objet\_4

Figure 6: Communication Diagram Formalism

* 1. Components of a Communication Diagram

Tableau 11: Elements of a 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. |

* 1. Authenticate Communication Diagram

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

*Figure 5:Auhenticate Communication Diagram*

## *3.* SEQUENCE DIAGRAM

## a. Definition

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

## 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 6: Formalism of sequence diagram*

# CHAPTER 4: CONCEPTION PHASE

Tableau 12: Elements of a sequence Diagram

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Lifeline |  | An individual participant in a sequence diagram, it is positioned at the top of the diagram. |
| Combined fragment |  | It represents a choice of behavior in which at most one operand will be chosen. |
| Messages |  | These are arrows that shows the direction of message flow. We have the synchronous, the asynchronous and the self-messages. |
| Activation |  | It describes the time in which an operation is performed by an element. |

* 1. Authentication Sequence Diagram

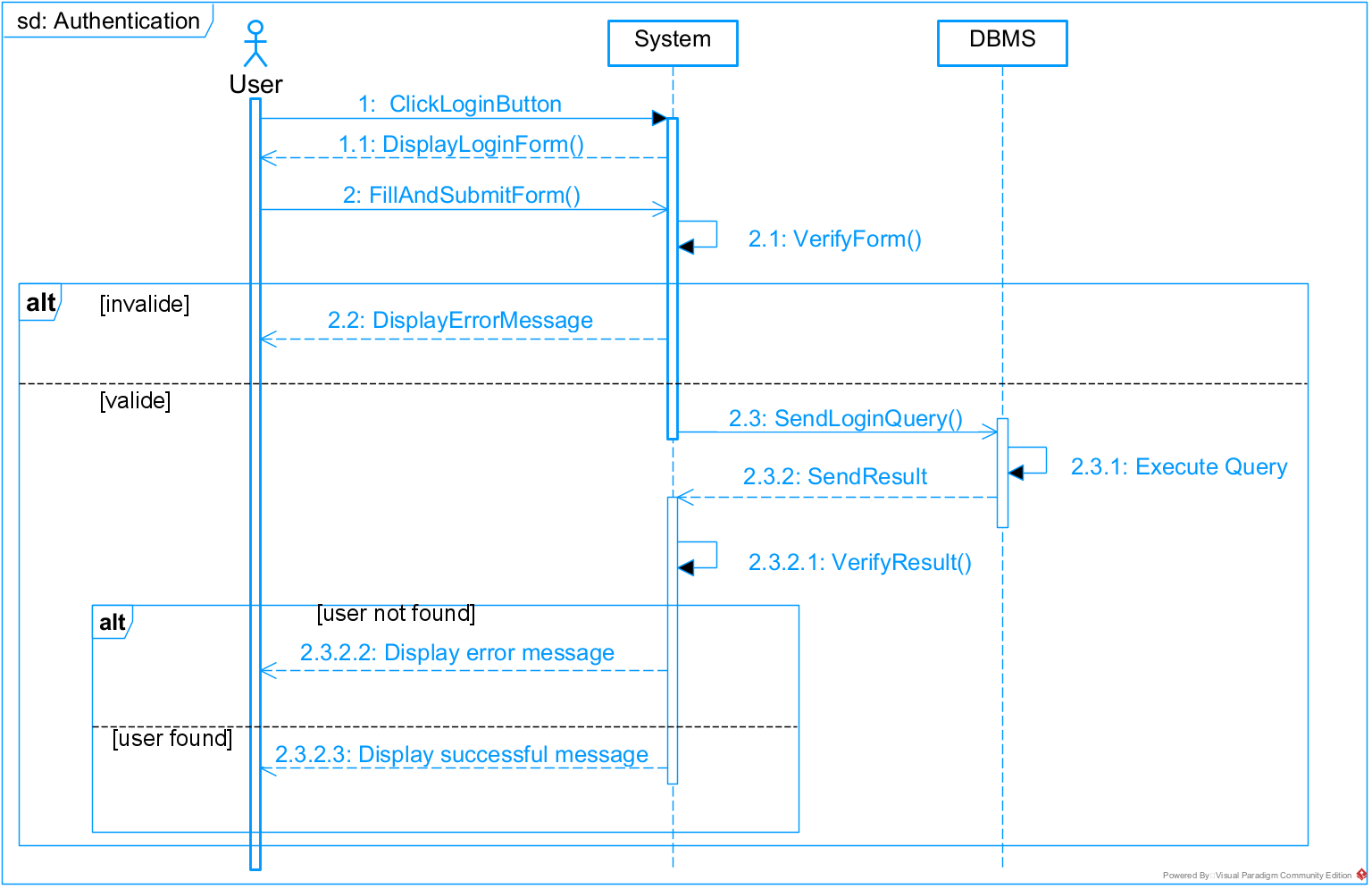


Figure 7: Authentication Sequence Diagram

### Sequence Diagram of the System



Figure 8: Sequence Diagram for job application

*4.* ACTIVITY DIAGRAM

## a. Definition

An activity diagram is a graphical representation of workflows that show the steps needed in the realization of a process; showing the details from a start point to an end point through all decisions and actions that can 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.

**b. Formalism**



Partition 1



Partition 2



Partition 3



action 1



action 3



action 2



action 4



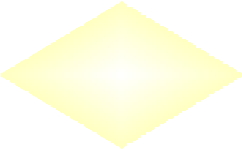
action 5



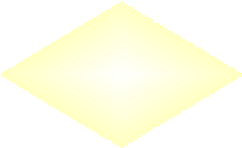
action 7



action 8



Decision\_1



Decision\_2



[

ok

]

[

]

not ok



action 6

[

ok

]

[

not ok

]

Figure 9: Formalism of Activity diagram

Tableau 13: Elements of activity diagramdiagram

|  |  |  |
| --- | --- | --- |
| 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. |

### Authentication Activity Diagram

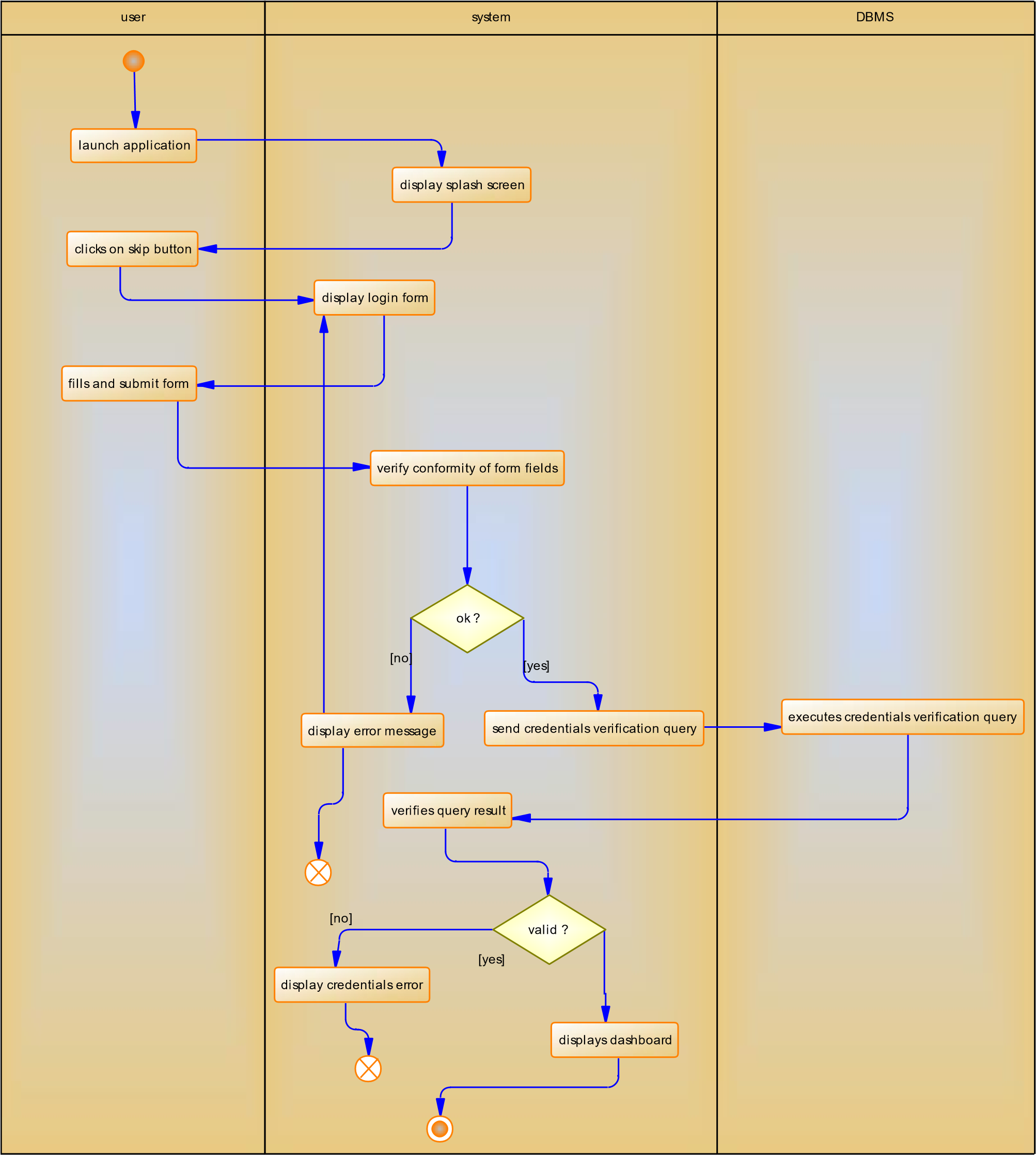


Figure 10: Authentication Activity Diagram

### Job Application Activity Diagram



Figure 11: Job Application Diagram

## CONCLUSION

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 4: CONCEPTION PHASE

Preamble

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

It bridges the gap between the analysis phase and the realization phase.

Content overview

**TEST OF**

**FUNCTIONALITIES**

INTRODUCTION

1.

TECHNICAL BRANCH

A.

GENERIC DESIGN

B.

CAPTURE

OF

TECHNICAL

NEEDS

C.

RELATED

UML

DIAGRAMS

i.

CLASS DIAGRAM

ii.

STATE

MACHINE

DIAGRAM

iii.

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.

## TECHNICAL BRANCH

### Generic Design

#### Structure of the application

Briefly, the **web application architecture** is a "skeleton" or layout that displays the interactions between application components, middleware systems, user interfaces, and databases. This kind of interaction allows a number of applications to work together simultaneously. Once a user opens a web app, the server sends specific data to the browser as a response to the user's request. To be precise, a web client (or user agent) may request information through a web server. Then, with these minimal manipulations, the requested information appears. After that, the interaction between the user and the web app starts.

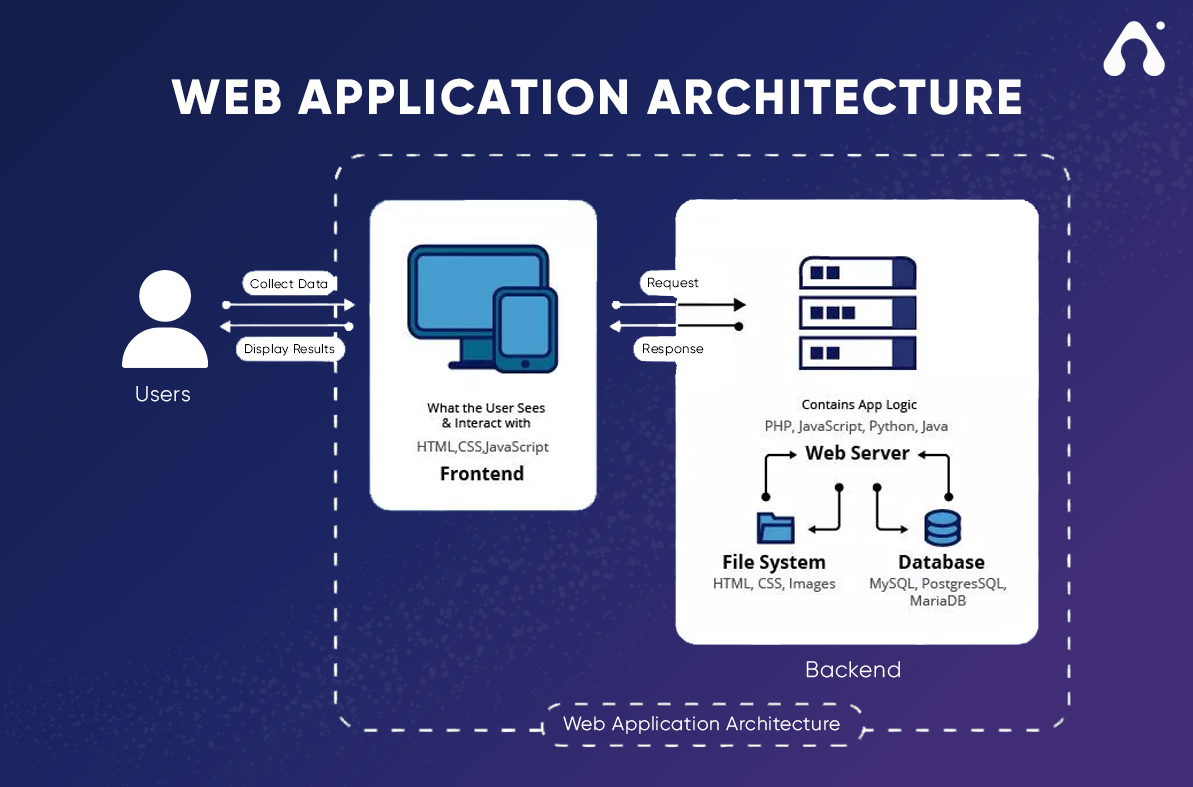


Figure 12: Structure of an application Source:WWW.APPVENTUREZ.COM

## B. CAPTURE OF TECHNICAL NEEDS

### a. Physical Architecture

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

Within the scope of our project, we made use of the 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.

### **Logical Architecture**

The job board platform will be developed using Django's Model-View-Template (MVT) architecture, which is a variation of the Model-View-Controller (MVC) design pattern. The MVT architecture separates the application into three interconnected components, promoting scalability, maintainability, and reusability. The components of the MVT pattern for this project are:

#### **Model:**

The Model represents the data structure and business logic of the application. It is responsible for maintaining data, managing the rules of data manipulation, and ensuring data consistency.

#### **View:**

The View is responsible for processing the data and executing business logic. It interacts with the Model to fetch data and send it to the Template for rendering.

#### **Template:**

The Template is responsible for rendering the data into a format that can be displayed to the user. It defines how the data is presented in the user interface.

The MVT pattern isolates the application logic from the user interface, allowing developers to work on the application logic (Model and View) independently from the design and user experience (Template). It also supports separation of concerns, making it easier to manage and scale the application.

In the job board platform, the View will receive all requests for the application, interact with the Model to fetch or update the necessary data, and then use the Template to render the final response to the user. This process ensures a clean and efficient flow of data and interactions within the application.

The MVT architecture can be visualized as follows:

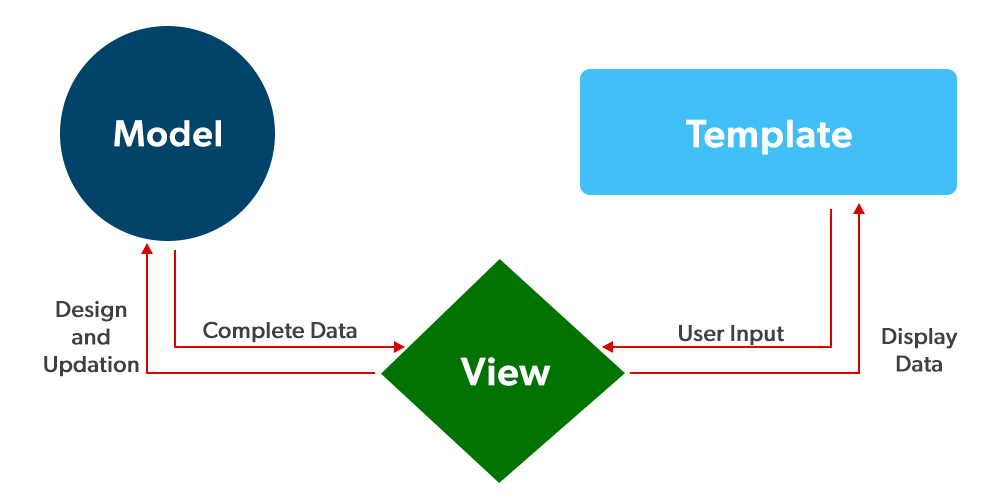


Figure 13: MVT Architecture

C. RELATED UML DIAGRAMS

## 1. CLASS DIAGRAM

### a. Definition

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

### b. Formalism

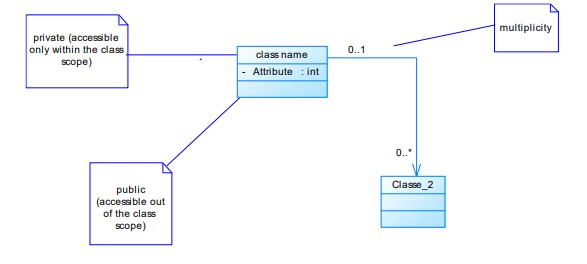


Figure 14: Formalism of Class Diagram

Tableau 14: Elements of 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. |

### Job Board Class Diagram



### Business Rules

**R1: A user creates and manages one** or more job postings, and a job posting is created and managed by only one user.

**R2:** A user holds one or more roles (e.g., job seeker, employer), and a role can be held by one or more users.

**R3:** An employer posts one or more job vacancies, and a job vacancy is posted by one employer.

**R4:** An admin oversees one or more job postings and ensures compliance, and each job posting is overseen by one admin.

**R5:** A job seeker applies to one or more job postings, and each job posting can receive applications from one or more job seekers.

**R6:** A job seeker has one or more saved job searches, and a saved job search belongs to one job seeker.

**R7:** An employer reviews one or more job applications, and a job application is reviewed by one or more employers.

**R8:** A job seeker receives notifications for one or more job postings, and a job posting sends notifications to one or more job seekers.

**R9:** An admin can manage one or more user accounts, and each user account is managed by one admin.

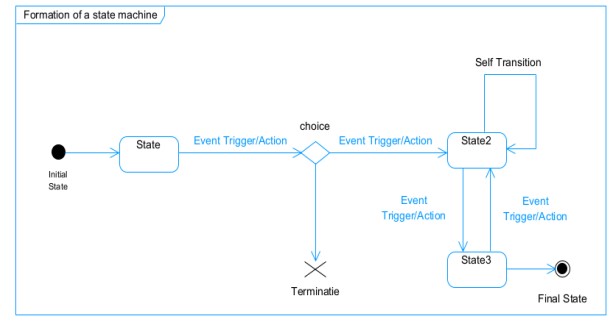
**R10:** An employer receives one or more job applications, and each job application is sent by one job seeker.

## *2.* STATE MACHINE DIAGRAM

### a. 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.

### b. Formalism



*Figure 32: Formalism of state machine diagram*

Tableau 15: Elements of 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. |

### c. Account state machine diagram

>>

user creates account

<<

<<

>>

admin validates account

>>

admin suspends account

<<

>>

admin reactivates account

<<

>>

admin deletes account

<<

<<

denies user

>>



pending



activate



Delete



suspend



Figure 15: Account state machine diagram

### Job Application State Machine



Figure 16: Job application state machine diagram

## *3.* PACKAGE DIAGRAM

### a. Definition

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

### b. Formalism

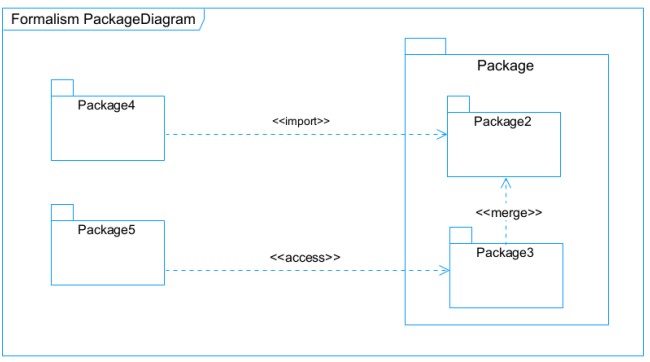


Figure 17: Formalism of package diagram

Tableau 16: 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. |



Figure 18: 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 5: REALISATION 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

The realization document contains the development of most important features of the application. It completes the preceding documents and any mistake made here will certainly alter the final application and might even underline work of the previous documents. So, it will thus be a question here of presenting data model of the application, the tools used for the realization of the application, the programming language chosen as well as the architecture of the application.

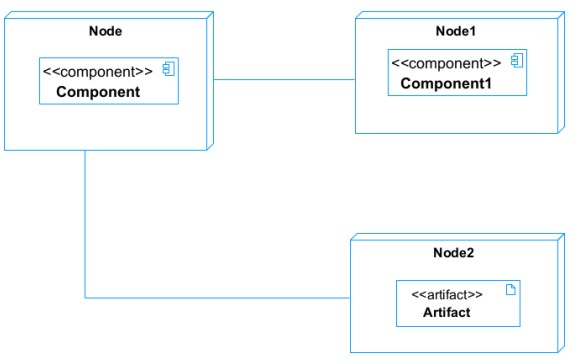
1. DEPLOYMENT DIAGRAM

## a. Definition

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

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

b. Formalism



*Figure 38: Formalism of 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 |

Tableau 17: Elements of Deployment Diagram

* 1. System Deployment Diagram



Figure 19: System Deployment Diagram

2. COMPONENT DIAGRAM

### a. Definition

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

### b. Formalism

c.

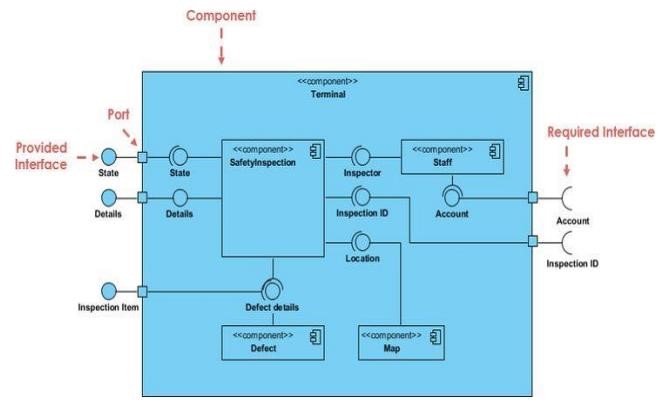


Figure 20: Formalism of component diagram (Source: https://www.pinterest.com/pin/551128073157994549/)

Tableau 18: 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. |

### c) System Web Component Diagram



Figure 21: 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.

# CHAPTER 6: TEST OF FUNCTIONALITIES

Preamble

Here, we will present the essential functionalities of the application.

Content overview

INTRODUCTION

1. APPLICATION MODULES
2. UNIT TEST
3. SHOW CASES

CONCLUSION

# INTRODUCTION

The test of functionalities phase helps us to know more about the solution we are building be it web or mobile. It provides the different functionalities or modules found in our application and how they are beneficial to the users. Hence, we are going to explore the different functionalities present in MiBoard.

## APPLICATION MODULES

 **User Registration and Authentication:**

* Module responsible for user registration, authentication, and account management. It includes features like registration form, login, logout, password reset, and profile management.

 **Job Listings:**

* Module for managing job listings. It allows employers to post job openings with details such as job title, description, location, requirements, and application deadline.

 **Job Applications:**

* Module handling job applications submitted by job seekers. Features typically include viewing job listings, applying for jobs, attaching resumes, adding cover letters, and tracking application status.

 **Employer Dashboard**:

* Module providing employers with tools to manage job listings, review applications, communicate with applicants, and update company profiles.

 Job Seeker Dashboard:

* Module for job seekers to manage their profiles, view applied jobs, track application statuses, and receive notifications about application updates.

** Notifications:**

* Module to notify users about important events such as application updates, new job listings matching their preferences, and system-related messages.

### ** Admin Dashboard:**

* Module accessible to super users (administrators) for managing the entire platform. It includes features like user management, content moderation, analytics, and system configuration.

 **Company Profiles:**

* Module to display information about companies posting jobs. It may include company descriptions, locations, industries, and links to related job listings.

## UNIT TEST

### What is unit test?

A unit test is a functional test of an application’s smallest possible source code unite. It aims to test the individual components of the software independent of the other parts of the code.

Unit testing are automated tests, In other words, unit testing is performed by software (such as unit testing framework or unit testing tool) and not manually by a developer. This means unit tests allow automated, repeatable continuous testing.

Settings Up and Running Tests

* Django provides a manage.py command to run tests located in your project. Command to run the tests: *python manage.py test.*

This command will automatically discover and execute all test cases defined in your project.

* Running Specific Tests: To run a specific test, we type the following command;

*python manage.py test your\_app\_name*

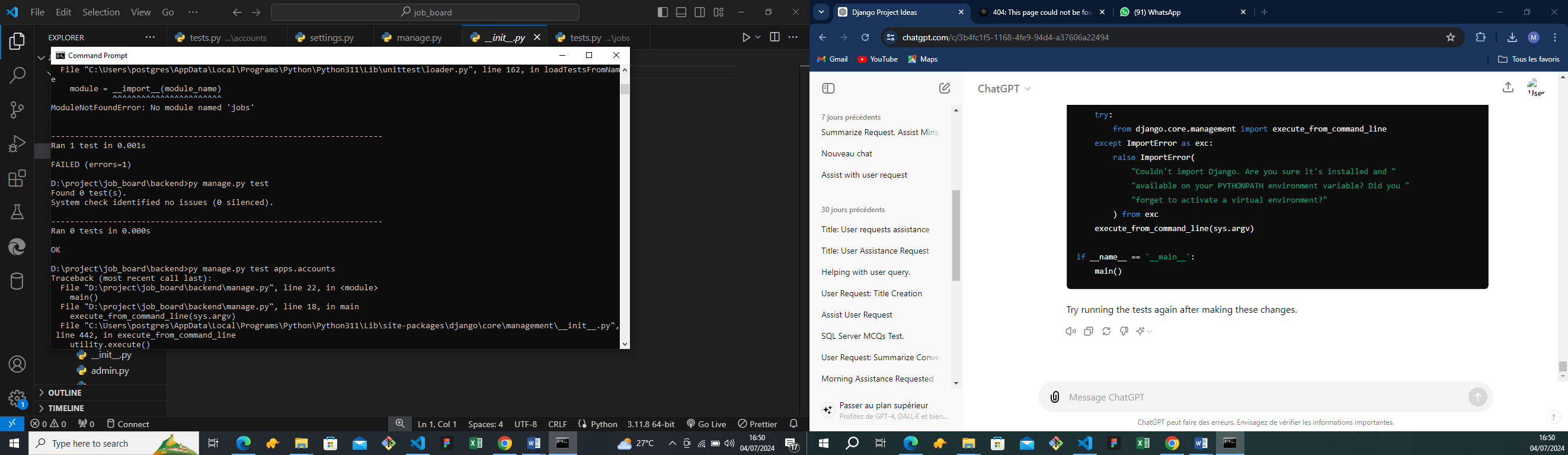


Figure : Unit Testing

# CONCLUSION

In the test of functionalities phase we explained the different functionalities of our application and the benefits to the various users (patient, health professional, admin, lab technician, super admin). This phase is essential in the understanding of the application. We will move to the last phase which is the installation and user guide where we will elaborate on how to install and use **MiBoard.**

## 

# PART 7: USER GUIDE

Preamble

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

Content overview

INTRODUCTION

* SHOW CASES

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.

## Usage Guide

### **Access the Admin Interface:**

* + Open a web browser and go to http://localhost:8000/admin/.
  + Log in using the superuser credentials created earlier.

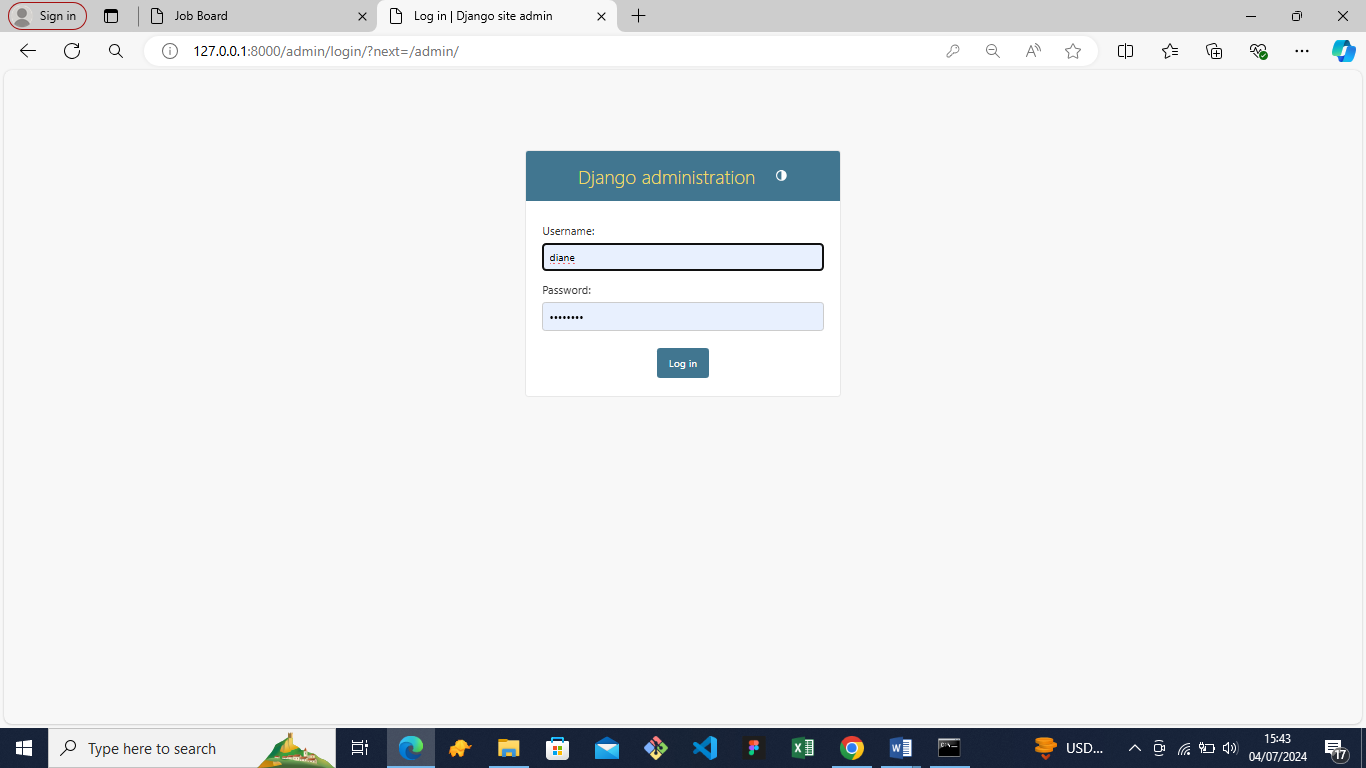


Figure 23: Django Administration login

### **Manage Job Listings:**

* + Navigate to "Jobs" to view, add, edit, or delete job listings.

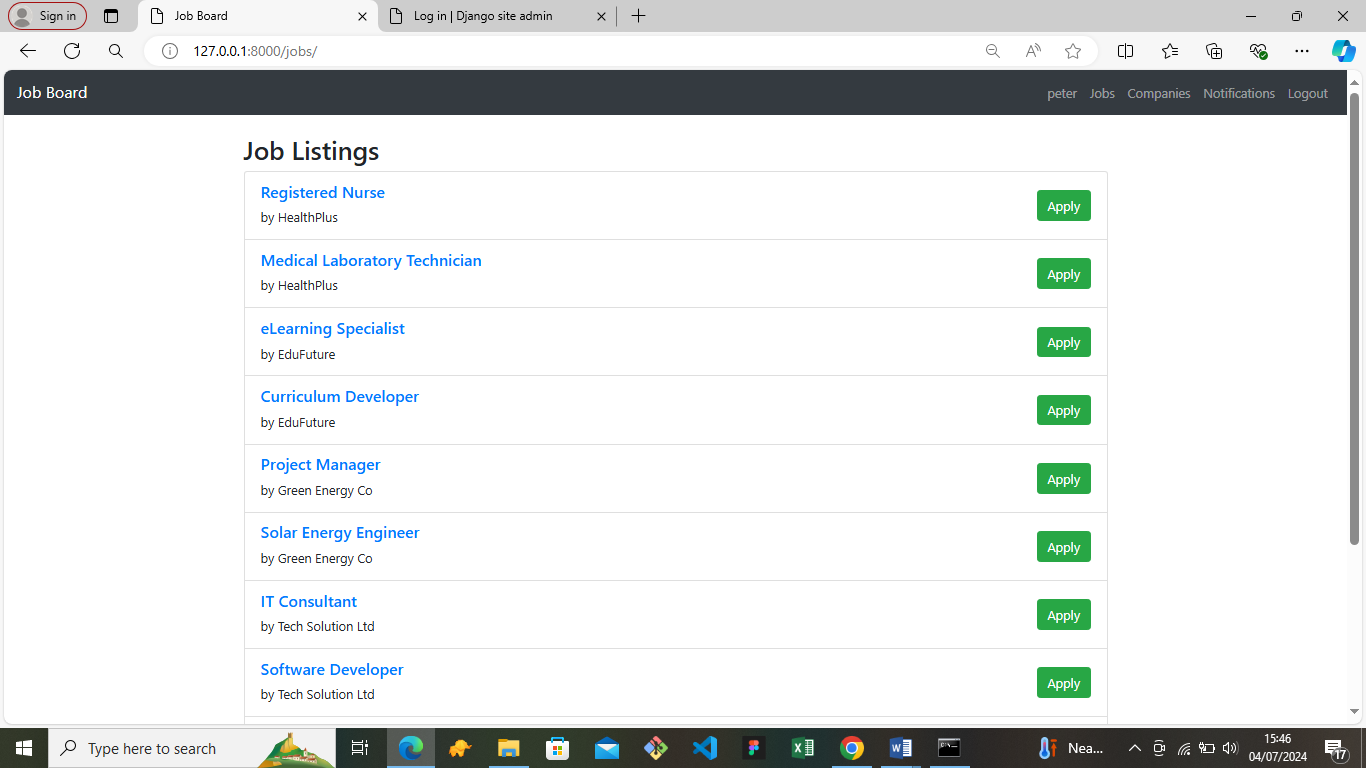


Figure 24: Job Listing page

### **User Registration and Login:**

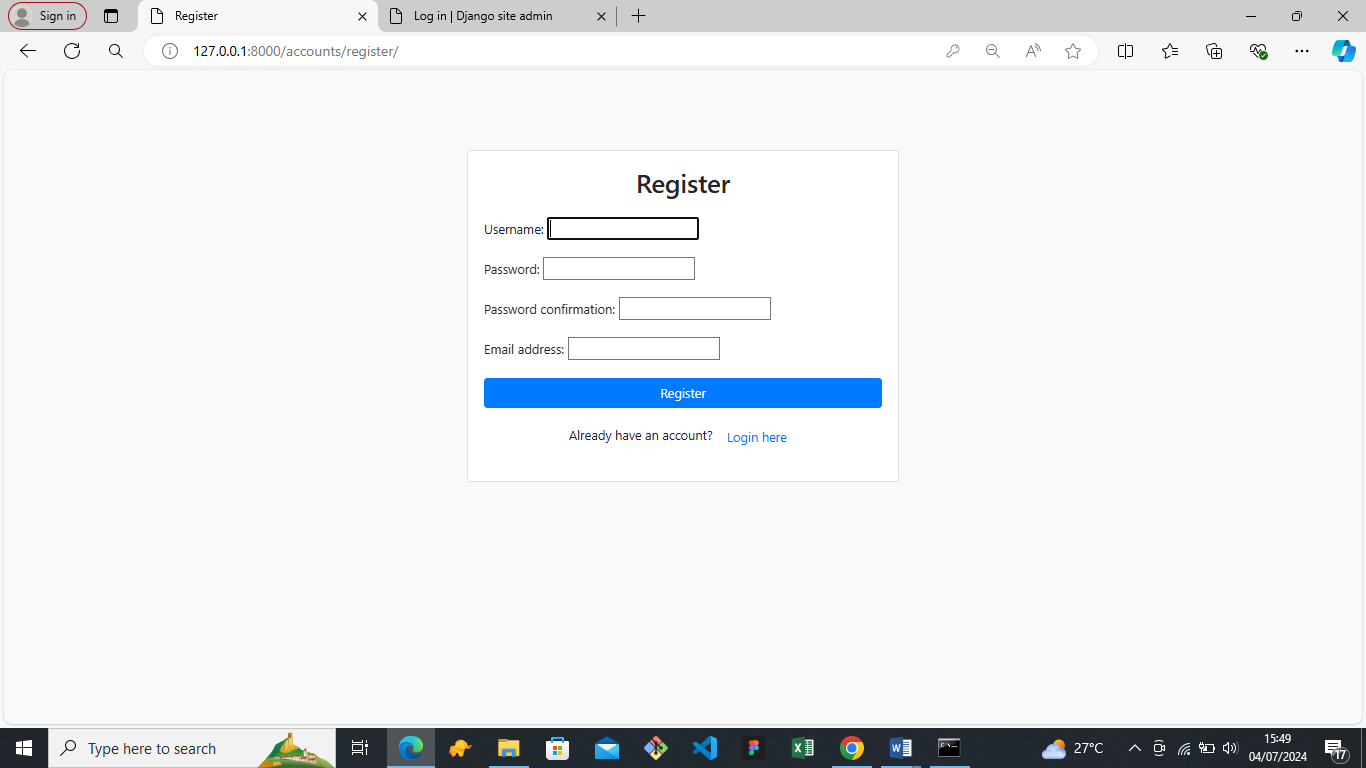
* + Users can register by clicking "Register" and filling out the registration form.
  + After registration, they can log in using their credentials via the "Login" link.

Figure : register page

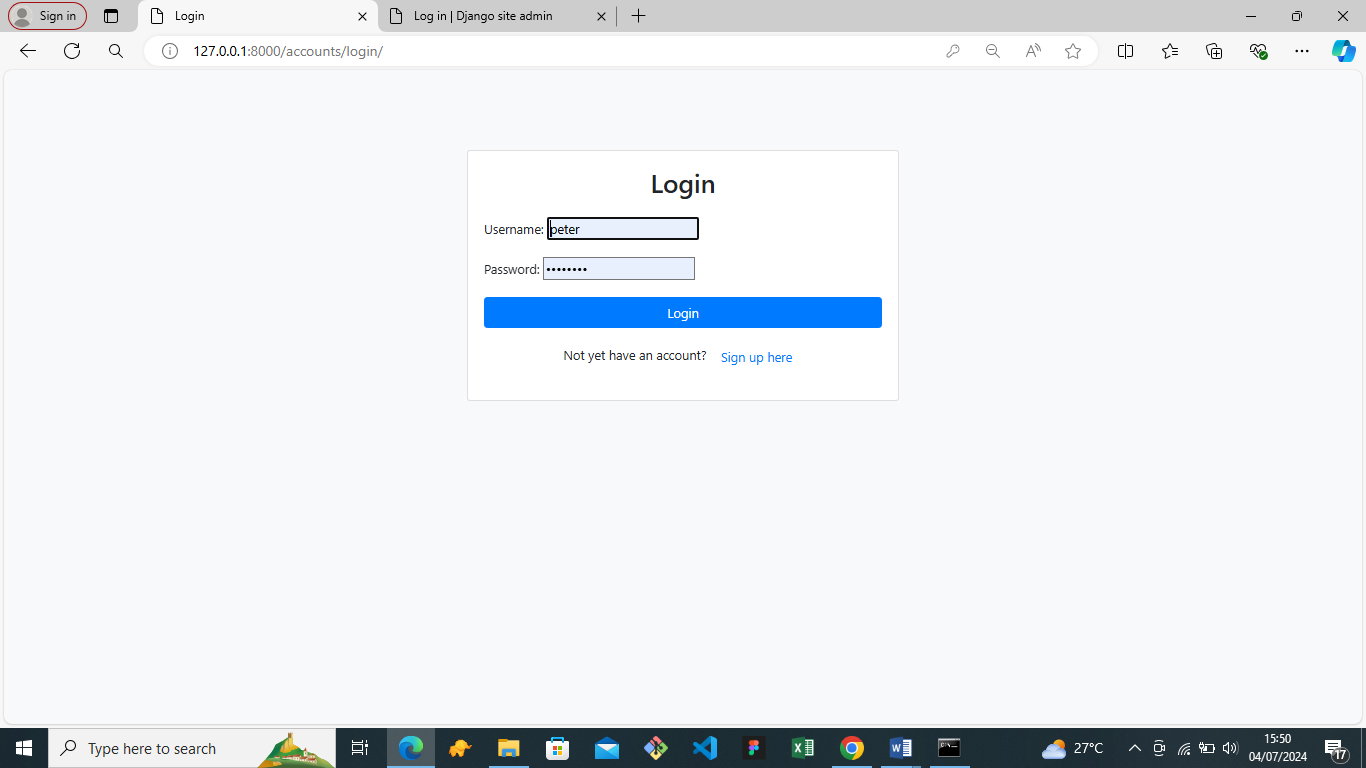


Figure 26: Login page

### **Job Application:**

* + Jobseekers can view job listings and apply for jobs by clicking "Apply".
  + Employers can manage job applications through the admin interface under "Applications".

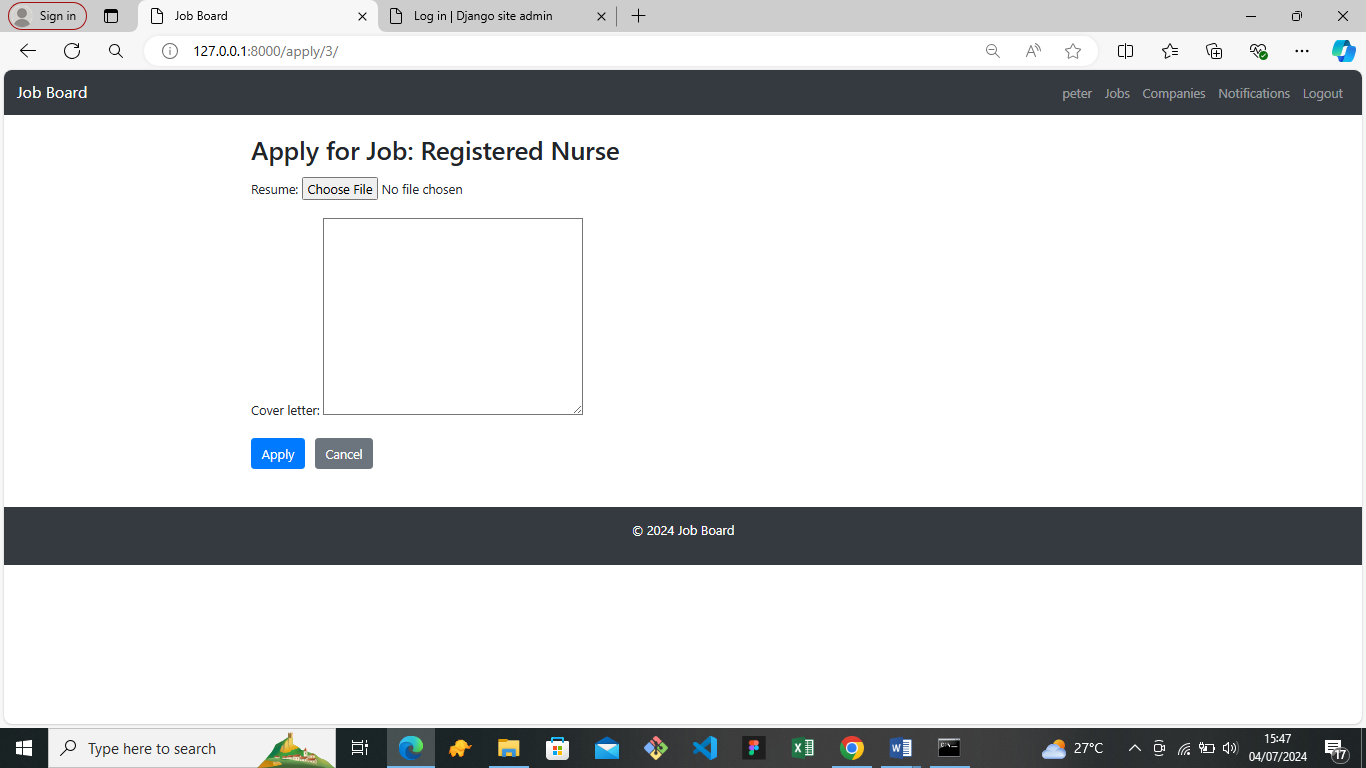


Figure 27: Job application page

### **Profile Management:**

* + Logged-in users can view and update their profiles by clicking on their username and selecting "Update Profile".

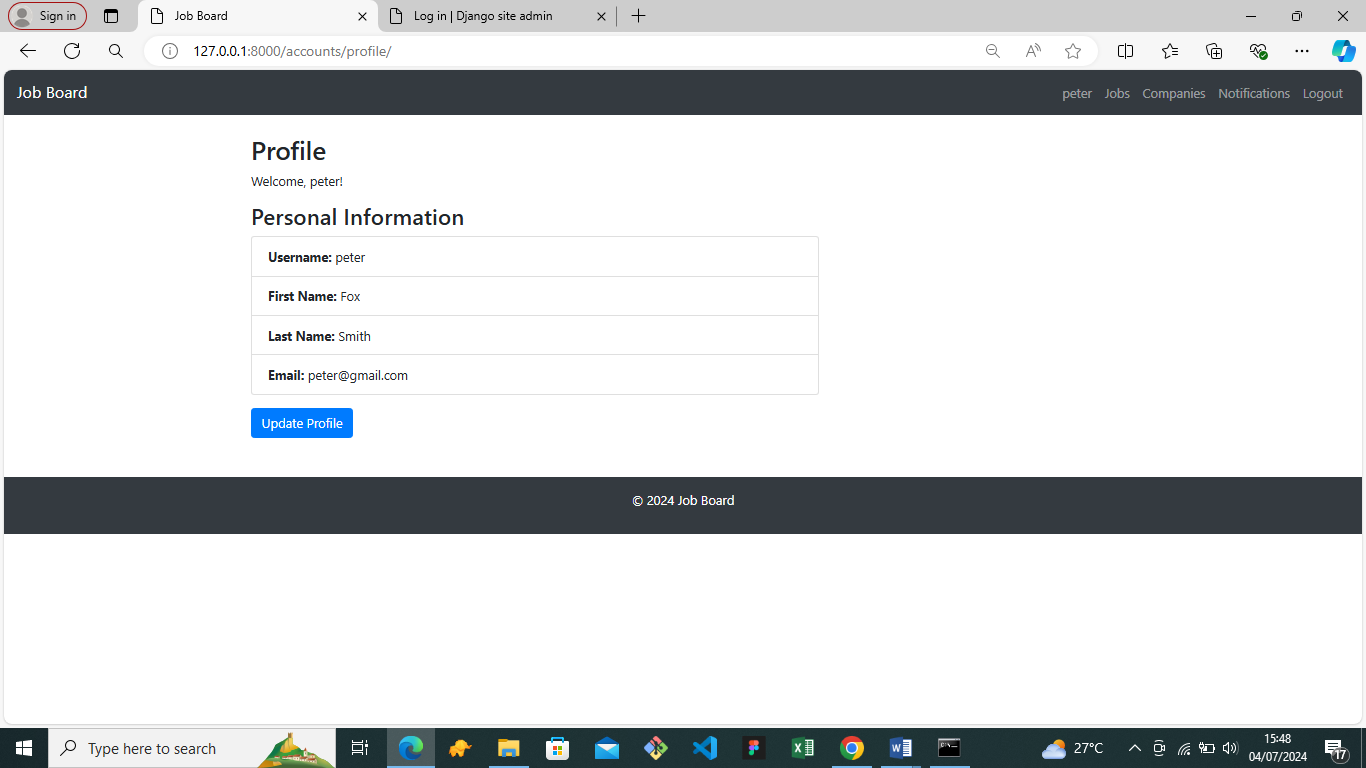


Figure 28: User Profile page

### **Logout:**

* + Users can log out from the platform using the "Logout" link.

# CONCLUSION

At this level, we can say further that the future users will not have a lot of difficulties understanding and using the application. It is at this point that we will move to the general conclusion of our work.

# PERSPECTIVES

Considering that this project can be further enhanced, the following perspectives can be taken into account for future improvements:

1. **Implementing AI-Driven Job Matching:**
   * Developing advanced AI algorithms to provide more accurate job recommendations for job seekers based on their skills, experience, and preferences. Employers can also receive more refined candidate suggestions, improving the hiring process.
2. **Introducing Real-Time Communication Tools:**
   * Adding chat and video interview functionalities to the platform to facilitate direct communication between job seekers and employers, making the recruitment process more efficient and interactive.
3. **Mobile Application Development:**
   * Creating a mobile application to increase accessibility and allow users to search for jobs, manage applications, and communicate with potential employers on the go.
4. **Integrating Skills Assessment Modules:**
   * Implementing online assessment tools to allow job seekers to demonstrate their skills through tests and practical tasks, providing employers with a clearer picture of candidates' capabilities.

# GENERAL CONCLUSION

The development of the Job Board Platform has been a significant step toward enhancing the employment landscape in Cameroon. This platform addresses critical inefficiencies in the job market by streamlining the connection between job seekers and employers. By providing a centralized and accessible online system, we aim to facilitate smoother interactions, quicker application processes, and better management of job postings and applications.

Through the implementation of this project, we have created a robust system that allows job seekers to easily find and apply for job opportunities, while employers can efficiently manage job postings and candidate applications. The platform's features, such as real-time notifications, role management, and comprehensive user account control, ensure that both job seekers and employers have a seamless experience.

Furthermore, the inclusion of advanced search capabilities and personalized job recommendations enhances the user experience, making it easier for job seekers to find relevant opportunities and for employers to find suitable candidates. The platform also provides valuable insights and reports, helping employers make informed hiring decisions and allowing administrators to monitor and improve the platform's performance.

In conclusion, the Job Board Platform not only bridges the gap between job seekers and employers but also contributes to the overall efficiency and transparency of the job market. By leveraging modern technology and innovative solutions, we have created a system that promises to bring lasting positive impact to the employment ecosystem in Cameroon. This project serves as a testament to the potential of digital platforms in transforming traditional processes and improving accessibility and effectiveness in various domains.

# ANNEXES

# BIBLIOGRAPHY

* Programming Microsoft Visual Basic.Net of Francesco Balema.
* Visual Studio Code Distilled of Alessandro Del Sole.
* Mercurial 2022.
* International Journal of Database Management Systems ( IJDMS ) Vol.10, No.2, April 208.
* Head First JavaScript Programming; A brain Friendly Guide of Elizabeth Robison.

# WEBOGRAPHY

* <https://www.makeuseof.com/tag/what-is-apk-file/>what is APK? August 28st 2023 10:13.
* <https://docs.expo.dev/get-started/installation/>Expo documentation February 07th 2024 8:35PM.
* <https://console.twilio.com/?frameUrl=/console>SMS API information. February 07th 2024 5:30PM.
* <https://formik.org/docs/guides/validation>FORMIK documentation. February 09th 2024 7:27PM.
* [https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programmingOOP#](https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP) Object Oriented programming May 15th 2024 8:35PM.
* <https://advice.writing.utoronto.ca/planning/intros-and-conclusions>How to write a good
* introduction and conclusion. May, 6th 2024. 07:20 AM.
* <https://builtin.com/software-engineering-perspectives/unit-testing> What is unit testing?
* https://www.guru99.com/javascript-books.html: (Tuesday 14th May 2024 at 9:48) to learn more about JavaScript.
* 7 Best New Visual Studio Books to Read in 2023(Friday 15th 2024 May at 10:30) to learn more on coding using visual studio.
* W3Schools Online Web Tutorials (Monday 18th June 2024 at 10:00) to get more knowledge of different programming languages.
* Tutorials List - Java point (Thursday 16th June 2024 at 14:30) to get more knowledge of different programming languages.
* https://www.bing.com/search?q=merise+definition&qs=n&form (Monday 20th June 2024 at 10:25) helped with some definitions.
* Share and Discover Knowledge on Slide Share (Tuesday 20th June 2024 at 09:00) taught us how to create the PowerPoint for our presentation.

# TABLE OF CONTENT

[DEDICATION i](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004927)

[DEDICATION i](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004928)

[ACKNOWLEDGEMENTS ii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004929)

[CONTENTACKNOWLEDGEMENTS ii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004930)

[CONTENT iii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004931)

[LIST OF FIGURESCONTENT iii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004932)

[LIST OF FIGURES v](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004933)

[LIST OF TABLESLIST OF FIGURES v](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004934)

[LIST OF TABLES vi](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004935)

[ABBREVRATIONLIST OF TABLES vi](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004936)

[GLOSSARY vii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004937)

[ABSTRACTABBREVRATION vii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004938)

[ABSTRACT viii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004939)

[RESUMEABSTRACT viii](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004940)

[RESUME ix](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004941)

[GENERAL INTRODUCTIONRESUME ix](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004942)

[GENERAL INTRODUCTION 1](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004943)

[CHAPTER 1: EXISTING SYSTEMGENERAL INTRODUCTION 1](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004944)

[PART 1: EXISTING SYSTEM 2](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004945)

[Introduction 4](#_Toc171004946)

[1. Presentation of the Theme 4](#_Toc171004947)

[2. Study of the Existing System 4](#_Toc171004948)

[Functionality 4](#_Toc171004949)

[3. Criticism of the Existing System 5](#_Toc171004950)

[4. Problematics 5](#_Toc171004951)

[5. Proposed Solution 6](#_Toc171004952)

[Conclusion 6](#_Toc171004953)

[PART 2: SPECIFICATION BOOK 7](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004954)

[INTRODUCTION 9](#_Toc171004955)

[A. CONTEXT AND JUSTIFICATION 10](#_Toc171004956)

[Key Questions Addressed: 10](#_Toc171004957)

[B. OBJECTIVES OF THE PROJECT 10](#_Toc171004958)

[1. General Objective 10](#_Toc171004959)

[2. Specification Objectives 11](#_Toc171004960)

[C. EXPRESSION OF NEEDS 11](#_Toc171004961)

[1) Functional Needs 11](#_Toc171004962)

[2) Non-Functional Needs 12](#_Toc171004963)

[ **Hardware Requirements:** 12](#_Toc171004964)

[ **Software Requirements:** 12](#_Toc171004965)

[E. ESTIMATED COST OF THE PROJECT 14](#_Toc171004966)

[a) Hardware Resources 14](#_Toc171004967)

[b) Software Resources 15](#_Toc171004968)

[c) Human Resources 16](#_Toc171004969)

[d) Total Project Estimated Cost 16](#_Toc171004970)

[F. CONSTRAINTS OF THE PROJECT 16](#_Toc171004971)

[i. Time Constraint 16](#_Toc171004972)

[ii. Cost Constraint 17](#_Toc171004973)

[G. LIST OF THE PARTICIPANTS AND DELIVERABLES 17](#_Toc171004974)

[a) List of participants 17](#_Toc171004975)

[b) Deliverables 17](#_Toc171004976)

[CONCLUSION 18](#_Toc171004977)

[PART 3: ANALYSIS PHASE 19](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171004978)

[INRODUCTION 21](#_Toc171004979)

[A. METHODOLOGY 22](#_Toc171004980)

[What is a methodology? 22](#_Toc171004981)

[i. Comparatives between UML and MERISE 22](#_Toc171004982)

[a. MERISE 22](#_Toc171004983)

[b. UML 22](#_Toc171004984)

[ii. STATIC OR STRUCTURAL DIAGRAMS 23](#_Toc171004985)

[i. BEHAVIOURAL OR DYNAMIC DIAGRAMS 23](#_Toc171004986)

[a. A Unified Process 25](#_Toc171004987)

[b. The Two Track Unified Process (2 TUP) 25](#_Toc171004988)

[**a.** **The left branch (functional branch)** 26](#_Toc171004989)

[**b. The right branch (Technical branch)** 26](#_Toc171004990)

[c. The middle branch (Realization or Implementation branch) 26](#_Toc171004991)

[B. CHOICE OF THE ANALYSIS METHOD 27](#_Toc171004992)

[C. MODELLING OF THE PROPOSED SOLUTION 28](#_Toc171004993)

[1. USE CASE DIAGRAM 28](#_Toc171004994)

[a) Definition 28](#_Toc171004995)

[b) Formalism 29](#_Toc171004996)

[c) Actors of our System 31](#_Toc171004997)

[d) General Use case Diagram 32](#_Toc171004998)

[e) Textual description of use case 33](#_Toc171004999)

[*2.* COMMUNICATION DIAGRAM 34](#_Toc171005000)

[**a. Definition** 34](#_Toc171005001)

[*3.* SEQUENCE DIAGRAM 36](#_Toc171005002)

[a. Definition 36](#_Toc171005003)

[b. Formalism 36](#_Toc171005004)

[CHAPTER 4: CONCEPTION PHASE 36](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005005)

[d. Sequence Diagram of the System 39](#_Toc171005006)

[a. Definition 39](#_Toc171005007)

[e. Authentication Activity Diagram 43](#_Toc171005008)

[f. Job Application Activity Diagram 44](#_Toc171005009)

[CONCLUSION 45](#_Toc171005010)

[PART 4: CONCEPTION PHASE 46](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005011)

[INTRODUCTION 48](#_Toc171005012)

[I. TECHNICAL BRANCH 49](#_Toc171005013)

[A. Generic Design 49](#_Toc171005014)

[ Structure of the application 49](#_Toc171005015)

[B. CAPTURE OF TECHNICAL NEEDS 50](#_Toc171005016)

[a. Physical Architecture 50](#_Toc171005017)

[**b.** **Logical Architecture** 50](#_Toc171005018)

[**1.** Model: 50](#_Toc171005019)

[**2.** **View:** 51](#_Toc171005020)

[**3.** **Template:** 51](#_Toc171005021)

[1. CLASS DIAGRAM 52](#_Toc171005022)

[a. Definition 52](#_Toc171005023)

[b. Formalism 52](#_Toc171005024)

[c. Job Board Class Diagram 54](#_Toc171005025)

[d. Business Rules 55](#_Toc171005026)

[*2.* STATE MACHINE DIAGRAM 55](#_Toc171005027)

[a. Definition 55](#_Toc171005028)

[b. Formalism 56](#_Toc171005029)

[c. Account state machine diagram 57](#_Toc171005030)

[e. Job Application State Machine 58](#_Toc171005031)

[*3.* PACKAGE DIAGRAM 58](#_Toc171005032)

[a. Definition 58](#_Toc171005033)

[b. Formalism 58](#_Toc171005034)

[PART 5: REALISATION PHASE 61](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005035)

[INTRODUCTION 63](#_Toc171005036)

[a. Definition 63](#_Toc171005037)

[a. Definition 65](#_Toc171005038)

[b. Formalism 66](#_Toc171005039)

[c) System Web Component Diagram 67](#_Toc171005040)

[CONCLUSION 68](#_Toc171005041)

[CHAPTER 6: TEST OF FUNCTIONALITIES 69](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005042)

[INTRODUCTION 71](#_Toc171005043)

[A. APPLICATION MODULES 72](#_Toc171005044)

[** Notifications** 72](#_Toc171005045)

[** Admin Dashboard:** 72](#_Toc171005046)

[B. UNIT TEST 73](#_Toc171005047)

[What is unit test? 73](#_Toc171005048)

[CONCLUSION 74](#_Toc171005049)

[75](#_Toc171005050)

[PART 7: USER GUIDE 75](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005051)

[INTRODUCTION 77](#_Toc171005052)

[Usage Guide 77](#_Toc171005053)

[1. **Access the Admin Interface:** 77](#_Toc171005054)

[2. **Manage Job Listings:** 78](#_Toc171005055)

[3. **User Registration and Login:** 78](#_Toc171005056)

[4. **Job Application:** 79](#_Toc171005057)

[5. **Profile Management:** 80](#_Toc171005058)

[6. **Logout:** 80](#_Toc171005059)

[CONCLUSION 81](#_Toc171005060)

[PERSPECTIVES 82](#_Toc171005061)

[GENERAL CONCLUSION 83](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005062)

[ANNEXES 84](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005063)

[BIBLIOGRAPHY 85](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005064)

[WEBOGRAPHY 86](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005065)

[TABLE OF CONTENT 87](file:///C:\Users\postgres\Desktop\School\report\rMimi\Academic_report.docx#_Toc171005066)