**REPUBLIC OF CAMEROON**Peace - Work – Fatherland



**INFINITY TECHNOLOGY – CAMEROON**

[**Tel:(237)**](Tel:(237)) **620 25 60 69 /654-486-985**

**E-mail :** [**infinitytechnoloqy.contactus@gmail.com**](mailto:infinitytechnoloqy.contactus@gmail.com)

**Website: www.infinity.com**

**REPUBLIC OF CAMEROON**Peace - Work - Fatherland



AFRICAN INSTITUTE OF COMPUTER  
SCIENCES  
CAMEROON OFFICE  
P.O. Box: 13719 Yaoundé  
Tel: +237.242.729.957 Fax: +237242729958  
E-mail: contact@iaicameroun.com  
Website: www.iaicameroun.com

**CONCEPTION AND REALIZATION OF AN INTELLIGENT JOB PORTAL AND CAREER MANAGEMENT PLATFORM: JOBBY**

Internship carried out from 1rd July to 15th August 2025  
In view of obtaining a **Bachelor degree** in Computer Sciences  
Option: **Software Engineering**

Written by:

**NKAYOU FRANCK LANDRY**

Level III Software Engineering student

**Supervision:**

Academic Supervisor:  
**Mr. NDIFOR**Software Engineer,  
Lecturer at AICS-Cameroon

Professional Supervisor:  
**Mr OSTIE JOSPIN**

Director,

Of Infinity Technology

**Academic Year 2024 - 2025**

# DEDICATION

**We dedicate this Work to the NSANGOU FAMILY, for their love, support and their unconditional  
sacrifices for my academic success.**

# ACKNOWLEDGEMENTS

As a prelude to this internship report we wish to express our sincere thanks to those who have assisted us and have contributed to the accomplishment of this report as well as the success of this great academic year. We would like specially to thank:

* The Resident Representative of the African Institute of Computer Sciences Paul Biya Technological Centre of Excellence Cameroon, **Mr. ABANDA Armand Claude.**
* Mr **Mr. OSTIE Jospin Mbita** the director of Infinity Technology and at the same time my professional supervisor for his kindness and youthfulness for permitting us to carry out our internship in their institution.
* We express our gratitude to our academic supervisor **Mr. NDIFOR** for his constant availability and support for the realization of this project.
* To my family and parents for their guidance and support during this internship period
* **Mr. AGBOR ANDERSON, MR NGUH PRINCE** and **Mr. MESSIO** teachers who always believed in the skills and potentials of each student, and gave us pieces of advises concerning our training.
* Our thanks also go to all the staff of AICS Cameroon who relentlessly have impacted knowledge for over three years to prepare us for this challenge.
* To all the **staff of INFINITY TECHNOLOGY** for their support throughout this internship period.
* To my classmates and school mates that contributed in one way or the other in the realization of this project.

.

# ABSTRACT

In an era marked by rapid technological advancement, the job market is evolving to meet the demands of a digital age. The emergence of intelligent job portals has transformed how candidates and employers interact, making the hiring process more efficient and user-friendly. This report outlines the conception and realization of JOBBY, an intelligent job portal designed to facilitate job searches, applications, and employer-candidate communication in Cameroon. Through a combination of rigorous research, user feedback, and modern technology stacks, we aim to enhance the employment landscape in the region. **“CONCEPTION AND REALIZATION OF AN INTELLIGENT JOB PORTAL”.** Following our theme, we did the feasibility studies, followed by the analysis of the system using the UML methodology associated with 2TUP process, from the analysis to the conception and realization phase. We successfully came out with an application called **JOBBY.**

**Keywords:**

* Job seeker
* Employer
* Project
* AI
* Technology
* Job recommendation

# RESUME

Dans une époque marquée par des avancées technologiques rapides, le marché du travail évolue pour répondre aux exigences d'un âge numérique. L'émergence de portails d'emploi intelligents a transformé la manière dont les candidats et les employeurs interagissent, rendant le processus de recrutement plus efficace et convivial. Ce rapport présente la conception et la réalisation de JOBBY, un portail d'emploi intelligent conçu pour faciliter les recherches d'emploi, les candidatures et la communication entre employeurs et candidats au Cameroun. Grâce à une combinaison de recherches rigoureuses, de retours d'utilisateurs et de technologies modernes, nous visons à améliorer le paysage de l'emploi dans la région. “CONCEPTION ET RÉALISATION D'UN PORTAIL D'EMPLOI INTELLIGENT”. Suivant notre thème, nous avons effectué des études de faisabilité, suivies de l'analyse du système en utilisant la méthodologie UML associée au processus 2TUP, allant de l'analyse à la phase de conception et de réalisation. Nous avons abouti avec succès a une application appelée **JOBBY**.

**Mot clés :**

* Candidat
* Employeur
* Projet
* IA
* Tecnologie
* Recommandation d'emploi

# SUMMARY

[DEDICATION i](#_Toc206462374)

[ACKNOWLEDGEMENTS ii](#_Toc206462375)

[ABSTRACT iii](#_Toc206462376)

[RESUME iv](#_Toc206462377)

[SUMMARY v](#_Toc206462378)

[LIST OF FIGURES vi](#_Toc206462379)

[LIST OF TABLES vii](#_Toc206462380)

[GENERAL INTRODUCTION 1](#_Toc206462381)

[PART ONE: INSERTION PHASE 3](#_Toc206462382)

[PART TWO: TECHNICAL PHASE 13](#_Toc206462387)

[BOOK ONE: EXISTING 14](#_Toc206462388)

[BOOK TWO: SPECIFICATION PHASE 25](#_Toc206462395)

[BOOK THREE: 43](#_Toc206462402)

[ANALYSIS PHASE 43](#_Toc206462403)

[BOOK FOUR: CONCEPTION PHASE 81](#_Toc206462415)

[BOOK FIVE: REALIZATION PHASE 102](#_Toc206462425)

[BOOK SIX: TEST OF FUNTIONALITIES 113](#_Toc206462431)

[BOOK SEVEN: USER GUIDE 117](#_Toc206462434)

[GENERAL CONCLUSION 127](#_Toc206462439)

[BIBLIOGRAPHY ix](#_Toc206462440)

[WEBOGRAPHY xi](#_Toc206462441)

[ANNEX xii](#_Toc206462442)

[CONTENT xiii](#_Toc206462443)

# LIST OF FIGURES

[Figure 1 : GEOGRAPHICAL LOCATION OF INFINITY TECHNOLOGY (source: INFINITY TECHNOLOGY) 7](#_Toc206462526)

[Figure 2: 2TUP Diagram (Source: www.memoireonline.com) 48](#_Toc206462527)

[Figure 3: UML 2.5 Diagrams Overview (source: http://www.uml-diagrams.) 52](#_Toc206462528)

[Figure 4: formalism of a Use Case diagram 53](#_Toc206462529)

[Figure 5:Use case Manage application 57](#_Toc206462530)

[Figure 6: Use case post job opening 58](#_Toc206462531)

[Figure 7: Use case Apply for job 59](#_Toc206462532)

[Figure 8: Formalism of a communication diagram 65](#_Toc206462533)

[Figure 9: Authentication Communication Diagram 67](#_Toc206462534)

[Figure 10: Create account communication diagram 68](#_Toc206462535)

[Figure 11: Post Job communication diagram 69](#_Toc206462536)

[Figure 12: Sequence Diagram Formalism 70](#_Toc206462537)

[Figure 13: Create account Sequence diagram 72](#_Toc206462538)

[Figure 14: Authentication Sequence diagram 73](#_Toc206462539)

[Figure 15: Post Job sequence diagram 74](#_Toc206462540)

[Figure 17 Activity diagram for create account 77](#_Toc206462542)

[Figure 18 Activity diagram for Login 78](#_Toc206462543)

[Figure 19: Activity diagram Post Job 79](#_Toc206462544)

[Figure 20:Hardware Diagram of the system 84](#_Toc206462545)

[Figure 21: MVC pattern diagram 86](#_Toc206462546)

[Figure 22: Class Diagram Formalism 87](#_Toc206462547)

[Figure 23: System class Diagram 90](#_Toc206462548)

[Figure 24:Formalism of a state machine diagram 91](#_Toc206462549)

[Figure 25: Post Job State machine diagram 93](#_Toc206462550)

[Figure 26: Formalism Package Diagram 94](#_Toc206462551)

[Figure 27: Package Diagram 97](#_Toc206462552)

[Figure 28: Object diagram formalism 98](#_Toc206462553)

[Figure 29:Component diagram formalism (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-iscomponent-diagram) 104](#_Toc206462554)

[Figure 30:Component Diagram 106](#_Toc206462555)

[Figure 31: Formalism Deployment diagram 107](#_Toc206462556)

[*Figure 32:* System deployment diagram 109](#_Toc206462557)

[Figure 33: JavaScript logo 111](#_Toc206462558)

[Figure 34 :React logo 111](#_Toc206462559)

[Figure 35 :CSS logo 111](#_Toc206462560)

[Figure 36 Nodejs logo 112](#_Toc206462561)

[Figure 37 MongoDB logo 112](#_Toc206462562)

[Figure 38 JOBBY WELCOME SCREEN 120](#_Toc206462563)

[Figure 39: Login Page 121](#_Toc206462564)

[Figure 40 registration form screen 122](#_Toc206462565)

[Figure 41: JOB POSTING 123](#_Toc206462566)

[Figure 42:Submit Application 124](#_Toc206462567)

# LIST OF TABLES

[Table 3:Criticism of the existing system 20](#_Toc201261507)

[Table 4 Hardware Resources of the project (Source: Mercurial 2023-2024) 35](#_Toc201261508)

[Table 5:Software ressources (source: Mercurial 2023/2024) 36](#_Toc201261509)

[Table 6:Software resources (source: Mercurial 2023/2024) 37](#_Toc201261510)

[Table 7: Total project Estimated cost 37](#_Toc201261511)

[Table 8: Estimation of required time for the project 38](#_Toc201261512)

[Table 9:Gantt chart 39](#_Toc201261513)

[Table 10:Project participants 40](#_Toc201261514)

[Table 11:Use case diagram components 54](#_Toc201261515)

[Table 12: Formalism of Textual Description 60](#_Toc201261516)

[Table 13: Textual description for register/create account 61](#_Toc201261517)

[Table 14: Textual description of authentication 62](#_Toc201261518)

[Table 15: Textual description of Review Application 63](#_Toc201261519)

[Table 16: Textual description Post Job Openings 64](#_Toc201261520)

[Table 17:Component of the Communication Diagram 66](#_Toc201261521)

[Table 18:Component of a Sequence Diagram 71](#_Toc201261522)

[*Tabl*e 19: Component of an Activity Diagram 76](#_Toc201261523)

[Table 20:Component of a Class diagram 88](#_Toc201261524)

[Table 21:Components of a State Machine diagram 92](#_Toc201261525)

[Table 22:Package Diagram Formalism 95](#_Toc201261526)

[Table 23:Components of an Object Diagram 99](#_Toc201261527)

[Table 24: Formalism of a component Diagram 104](#_Toc201261528)

[Table 25: Component of a Deployment Diagram 108](#_Toc201261529)

[Table 26: Material resources 110](#_Toc201261530)

[Table 27: Software resources 110](#_Toc201261531)

# GENERAL INTRODUCTION

In today’s rapidly evolving digital landscape, the integration of intelligent technology into job search processes has become not just an innovation, but a necessity. With the rising demand for efficiency, personalization, and accessibility in career development, mobile applications have emerged as powerful tools in reshaping how individuals find and apply for jobs.

One such advancement is the **conception and realization of "Jobby" an intelligent job search and career management platform**. Jobby is designed to bridge the gap between traditional job-hunting methods and the dynamic capabilities of modern technology. By harnessing the power of smart algorithms and real-time communication tools, the platform enables job seekers to discover tailored opportunities, submit applications seamlessly, and connect with employers—regardless of geographical limitations.

At the same time, Jobby offers recruiters a streamlined way to filter candidates based on skills, experience, and match scores, making the hiring process more efficient and informed. The platform's intelligent matching system, coupled with its intuitive user interface, offers a personalized and effective experience for all users.

This report explores the core features and advantages of this cutting-edge application and its potential to revolutionize the job search experience. Our approach is rooted in thorough research and user-centric design principles to deliver a practical and scalable solution.

For clarity and structure, this report is divided into **eight (8) parts**, each focusing on a specific aspect of the project:

1. **Insertion Document**:Here, we will present the company in which we spent our  
internship period and the way we were welcomed in the company. At the end of this  
part, we will present the theme given to us.

**2.** **Existing System**: Here, we shall present the already present system in place, that is the

one used for consultation and follow-up purposes  
**3. Specification Book:** In this book, we specify the needs of the user taking in to  
considerations the time and cost of the project.  
**4. Analysis Document:** Here, we will present the analysis method chosen together with  
the presentation of all the diagrams used for the analysis of the project.

**5. Conception phase:** This presents the generic and detailed conception of the project to bring out real world constituents  
**6. The Realisation phase:** This phase will permit us to visualise the implementation  
process of the solution.

**7. Test of functionalities:** In this phase, we shall present to you the different functionalities or modules of our application and how they work  
**8. The user Manual:** Which will present the conditions necessary to use the application  
and how to use it

# PART ONE: INSERTION PHASE

Preamble

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

Content Overview

INTRODUCTION

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

CONCLUSION

## INTRODUCTION

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

## WELCOME AND INTEGRATRION

On Tuesday, July 1, 2025, we went to the premises of INFINITY TECHNOLOGY located in Mvog-ada. We were welcomed with a conference led by the CEO of the company, Mr. OSTIE Jospin Mbita Messina, along with various members, including representatives from partner companies. These partner companies gave us a brief presentation of the services they offer while inviting us to visit their premises to discover any needs related to the services they provide. During this conference, emphasis was placed on respecting the company's internal regulations and on the proper attitude to adopt when writing our internship reports. The following Monday, we started the internship, which lasted one and half months, during which terms were assigned to us.

## GENERAL PRESENTATION OF THE COMPANY

The company INFINITY TECHNOLOGY is located in the city of Yaoundé, specifically at MVOG ADA HYSACAM CROSSROADS. It can be found using the location map below.

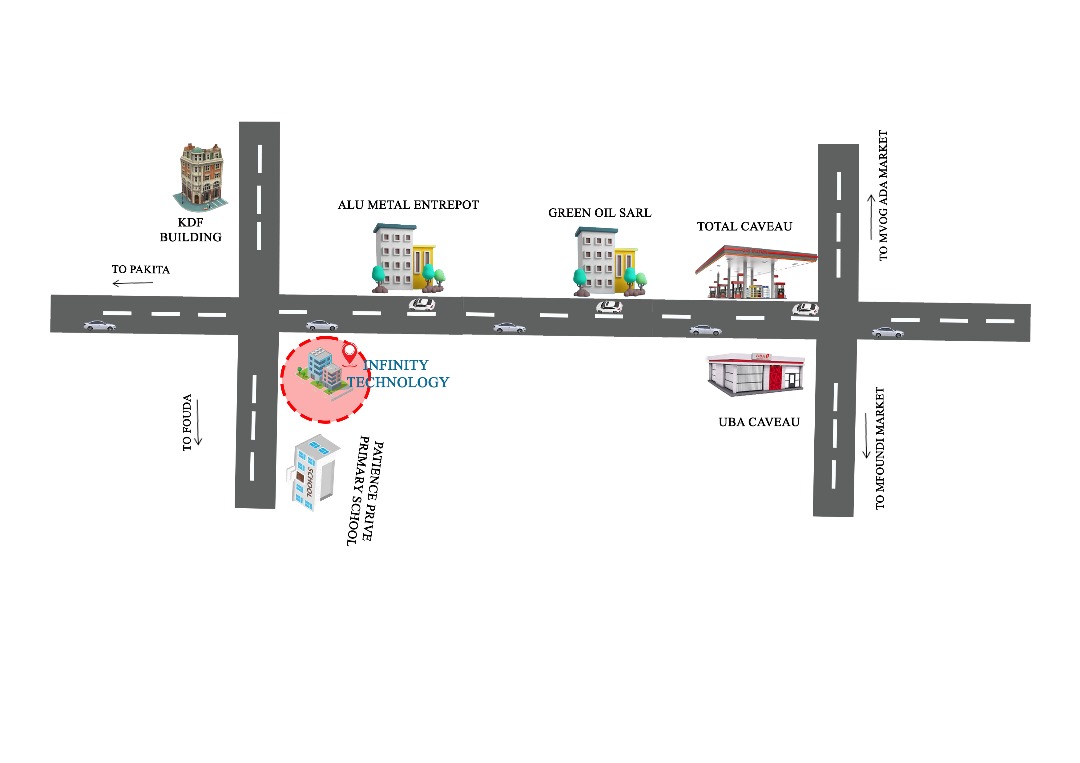


Figure 1 : GEOGRAPHICAL LOCATION OF INFINITY TECHNOLOGY (source: INFINITY TECHNOLOGY)

History

INFINITY TECHNOLOGY SARL is a project that stems from a series of reflections by CEO Mr. OSTIE JOSPIN Mbita, a software engineering student, digital marketing and SEO specialist, certified IT project manager from the United Kingdom Professional Development Academy, and a young IT entrepreneur. The company was founded in 2023.

Mission

Infinity Technology is a full-service technology solutions provider dedicated to offering businesses and individuals cutting-edge IT services and products. Their comprehensive offerings include IT project management, logo design, video editing, web development, graphic design, certified training, AI development, IT maintenance, and cybersecurity. Guided by a mission of excellence, Infinity Technology strives to deliver innovative and tailored solutions that open up infinite possibilities for their clients. With a commitment to quality, creativity, and customer service, the company aims to be the trusted partner in harnessing the power of technology to achieve growth, productivity, and success.

Functional organization of INFINITY TECHNOLOGY

The functional branch of infinity technology is organised as follows;

Technical Department

Human Resource Department

Financial

Department

Software Engineering department

Communication

Departure

General management

* + - 1. HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY

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

1. Hardware Resources

Table 1: Hardware resources of INFINITY TECHNOLOGY (source: MERCURIAL 2024)

* + - 1. Software Resources

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

As an enterprise specialised in IT, INFINITY TECHNOLOGY also comprises of software resources such as;

Table 2: Software resources of INFINITY TECHNOLOGY (source: MERCURIAL 2022)

## CONCLUSION

**In conclusion**, the theme *"***CONCEPTION AND REALIZATION OF AN INTELLIGENT JOB PORTAL AND CAREER MANAGEMENT PLATFORM***"* presents a unique opportunity to revolutionize the job market by digitizing and modernizing traditional recruitment methods. By integrating features such as intelligent job matching, personalized career guidance, skill assessment tools, and remote interview scheduling, the platform aims to streamline the hiring process for employers while offering job seekers tailored opportunities for career growth. Ultimately, this innovative solution addresses the growing need for efficiency and personalization in the employment sector, bridging the gap between outdated recruitment practices and advanced digital capabilities.

# PART TWO: TECHNICAL PHASE

# BOOK ONE: EXISTING

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
3. CRITISISM OF THE EXISTING
4. PROBLEMATICS
5. PROPOSED SOLUTION

CONCLUSION

## **INTRODUCTION**

The existing system refers to the system put in place to carry out the work done in the field on which our theme is based. Understanding this system is a great step in solving the problems that we might identify. It's of great importance that we take into consideration this old system before proposing a suitable solution

BRIEF PRESENTATION OF THE PROJECT THEME

Our chosen theme is**: “CONCEPTION AND REALIZATION OF AN INTELLIGENT JOB PORTAL AND CAREER MANAGEMENT PLATFORM**. In today’s competitive and fast-evolving job market, many job seekers face challenges in accessing the right opportunities due to lack of visibility, inefficient application processes, and limited access to relevant information. The traditional methods of job hunting relying on word-of-mouth, printed CVs, and in-person visits are increasingly outdated and often ineffective in a digitally connected world. To address this issue, we have developed Jobby, an intelligent job search and career management platform designed specifically to streamline and modernize the job-seeking experience. The platform leverages smart matching algorithms to connect job seekers with opportunities that align with their skills, qualifications, and interests. It also provides features such as remote applications, appointment booking for interviews, progress tracking, and employer communication making the entire job application journey more efficient and personalized. Jobby also benefits recruiters and employers by helping them identify ideal candidates quickly through intelligent filtering and data-driven recommendations. The platform ultimately aims to close the gap between job seekers and employment opportunities by making the hiring process more transparent, accessible, and technology-driven, particularly for the Cameroonian context and beyond.

### DESCRIPTION OF THE EXISTING SYSTEM

The current job search and recruitment process largely relies on traditional, manual methods that are often inefficient and fragmented. Job seekers and employers face several challenges due to the lack of integrated digital tools tailored to streamline these processes. Below is a detailed overview of how the existing system operates:

1. **Job Seeker Profile Management:**  
   Job seekers typically manage their resumes, qualifications, and application history manually or across multiple disconnected platforms. This often leads to inconsistent records, lost data, and difficulty in updating information efficiently.
2. **Job Application Process:**  
   Applications are usually submitted via email, physical documents, or through multiple job boards without synchronization. This results in a scattered process where tracking the status of multiple applications becomes challenging and prone to errors.
3. **Job Tracking and Follow-up:**  
   Most job seekers rely on memory or personal notes to track job openings they have applied for, upcoming interviews, or follow-up deadlines. This manual system increases the risk of missing important dates or neglecting promising opportunities.
4. **Communication with Employers:**  
   Communication often occurs through emails, phone calls, or various messaging platforms, leading to inconsistent and sometimes delayed exchanges. This fragmented communication can cause misunderstandings and lost opportunities.
5. **Interview Scheduling:**  
   Scheduling interviews is frequently done informally by phone or email, lacking a centralized system to efficiently coordinate availability. This can result in scheduling conflicts, missed interviews, or last-minute cancellations.
6. **Job Listing and Filtering:**  
   Job seekers often face difficulties in filtering relevant job offers from numerous listings scattered across multiple sites. Employers may struggle to reach the right candidates without efficient filtering or matching tools.
7. **Candidate Experience:**  
   The traditional recruitment process can be frustrating for job seekers due to lack of transparency, delayed feedback, and inefficient application management, leading to dissatisfaction and decreased engagement.
8. **Scalability and Efficiency Challenges:**  
   Employers and recruiters managing large volumes of applicants face challenges in organizing and filtering candidate data without automation. Similarly, job seekers juggling multiple applications lack tools to optimize their job search efficiently, hindering scalability and effectiveness.

### LIMITS OF THE EXISTING SYSTEM

Table 3:Criticism of the existing system

|  |  |  |
| --- | --- | --- |
| CRITICISMS | LIMITS | PROPOSE SOLUTION |
| Manual management of job seeker profiles and resumes across multiple platforms leads to inconsistencies and lost information. | - No centralized database for job seeker profiles. - Difficulties maintaining up-to-date application records. - Risk of losing important candidate data over time. | Implement a centralized digital profile management system where users can create, update, and store resumes, preferences, and application history in one secure platform. |
| Job applications submitted through email or multiple job boards result in scattered and untracked processes. | - Lack of unified application tracking. - Difficulty following up on application status. - High chance of missing deadlines or interview invites. | Develop an integrated application tracking system that allows users to monitor all application statuses in real-time and receive timely notifications. |
| Scheduling interviews manually via phone or email causes conflicts and missed opportunities. | - No centralized interview scheduling tool. - Overlapping or missed interview appointments. - Poor coordination between job seekers and employers. | Build an online interview scheduling system that synchronizes availability with automated reminders to reduce no-shows. |
| Communication between job seekers and employers is fragmented across different channels, causing delays and miscommunication. | - Informal and scattered communication methods. - No centralized messaging history. - Risk of misinterpretation and missed updates. | Introduce an in-app messaging platform to streamline communication, ensuring all exchanges are tracked and accessible. |
| Job seekers have difficulty filtering and finding relevant job listings amidst overwhelming and unorganized postings. | - Lack of effective job filtering and recommendation tools. - Users overwhelmed by irrelevant listings. - Reduced chances of matching suitable jobs. | Implement smart filtering and AI-powered job recommendations based on user profiles, skills, and preferences to present relevant opportunities. |
| Lack of transparency and feedback in the recruitment process causes frustration and disengagement. | - Delayed or absent updates on application progress. - Uncertainty about interview outcomes or next steps. | Provide a transparent status tracking dashboard with regular updates and feedback options, enhancing user trust and engagement. |

### PROBLEMATIC

With the description of the current job search process and its limitations clearly established, we can now formulate the central problem that our project seeks to address: **“HOW CAN WE INTEGRATE INTELLIGENT DIGITAL TOOLS INTO THE JOB SEARCH PROCESS TO ADDRESS INEFFICIENCIES IN APPLICATION TRACKING, ENHANCE CANDIDATE EXPERIENCE, AND IMPROVE OVERALL EMPLOYMENT ACCESSIBILITY FOR JOB SEEKERS?”** This key question guided our development process and was answered through a set of reliable and efficient digital solutions proposed in the later parts of this work. These solutions aim to eliminate the disorganization, communication gaps, and time-consuming steps found in traditional job-seeking methods—making job hunting smarter, faster, and more user-friendly through the Jobby platform.

### PROPOSED SOLUTION

1. **Centralized Job Seeker Profile Management**  
   Develop a digital system where users can create and manage their professional profiles, upload resumes, specify job preferences, and store application history. This reduces the inefficiencies of managing information across multiple platforms and ensures easy access to updated records.
2. **Smart Job Application Tracker**  
    Implement a smart dashboard that allows users to track the status of each job application in real-time—from submission to interview and offer stages. This promotes transparency and helps job seekers stay organized throughout the job search process.
3. **Online Appointment Scheduling with Recruiters**  
    Build an integrated scheduling tool that enables job seekers to book consultation or interview appointments with recruiters. The system synchronizes with calendars and sends automated reminders, improving time management and reducing missed opportunities.
4. **In-App Communication System**  
    Provide a messaging platform within the application that facilitates direct communication between job seekers and employers or career advisors. This feature ensures clear, centralized conversations, reducing the risk of miscommunication or lost messages.
5. **Career Profile History and Analytics**  
    Introduce a system that maintains each user's career journey—tracking job types applied to, match scores, response rates, and interview feedback. This allows for more personalized job recommendations and empowers users to make data-driven career decisions.

### DELIMITATIONS OF THE FIELD OF STUDY

**• Account Management:**

* User Registration: Job seekers and employers can register on the Jobby platform to create personalized accounts.
* Account Deletion: Users have the flexibility to delete their accounts at any time if they no longer wish to use the platform.
* Secure Logout: A secure logout function ensures user data privacy and account protection.

**• Job Application Management:**

* Job seekers can submit applications to various job postings and monitor the progress of each one through their dashboard.
* All submitted applications are stored for future reference, allowing users to build a comprehensive application history.

**➢ ADMIN**

* **Employer Account Management:**

Admins oversee employer accounts, including verification, activation/deactivation, and monitoring for platform compliance.

**• Application and Interview Oversight:**

Admins monitor the overall flow of job applications and scheduled interviews to ensure smooth platform operations**.**

**• Notifications and Communication:**

Admins can send push notifications to job seekers and employers regarding new job listings, interview reminders, system updates, or special announcements**.**

Conclusion

To conclude this section, we conducted a study of the traditional job search system by evaluating common practices and their limitations. Based on this analysis, we identified the challenges users face such as disorganized application tracking, inconsistent communication, and inefficient scheduling. We then proposed a modern digital solution in the form of a mobile application, **Jobby**, to resolve these issues. This step was crucial for understanding the current system's weaknesses so we could design a solution that meaningfully improves the job search experience for both candidates and employers.

# BOOK TWO: SPECIFICATION PHASE

Content

INTRODUCTION

1. CONTEXT AND JUSTIFICATION
2. OBJECTIVES OF THE PROJECT
3. EXPRESSIONS OF NEEDS
4. TARGET POPULATION AND BENEFICIARIES
5. ESTIMATED COST OF THE PROJECT
6. ESTIMATED OF TIME REQUIRED
7. CONSTRAINT
8. LIST OF PARTICIPANTS AND DELIVERABLES

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

#### Context

Job searching is a critical phase in every individual’s professional journey. Traditionally, the process has relied heavily on physical resumes, word-of-mouth recommendations, notice boards, and scattered job listings across various platforms. This outdated approach presents numerous challenges for both job seekers and employers. Job seekers often face difficulties tracking their applications, finding relevant opportunities that match their skills, or securing timely interview appointments. Employers, on the other hand, encounter inefficiencies in filtering through applicants, scheduling interviews, and managing candidate data effectively.

In today’s fast-paced, digital-first world, the expectations for recruitment have drastically changed. Job seekers demand platforms that are not only intuitive but also intelligent capable of recommending job opportunities tailored to their qualifications and preferences. Employers need tools that streamline the hiring process, reduce manual effort, and allow for data-driven decision-making. Unfortunately, existing systems often lack integration, automation, and user-friendliness, resulting in missed opportunities and inefficient workflows.

To overcome these challenges, the development of a modern, AI-powered platform **Jobby** has been proposed. Jobby redefines the job search and hiring experience by introducing a centralized system for managing job applications, candidate-employer communication, and appointment scheduling. With smart job matching, real-time tracking, and a user-friendly interface, Jobby bridges the gap between job seekers and recruiters, ensuring a faster, smarter, and more efficient employment process.

.

#### Justification

The development of the **Jobby** platform brings significant value to both job seekers and employers in the modern employment landscape:

1. **Efficiency:** Job seekers can easily create profiles, upload resumes, and apply to multiple job listings without the hassle of repeated form-filling. Employers, in turn, benefit from a streamlined candidate management system that allows them to post jobs, filter applicants based on match scores, and automate parts of the recruitment process. This reduces administrative workload and accelerates hiring.
2. **Accessibility:** Jobby provides users the flexibility to search for and apply to jobs from anywhere, at any time. This remote accessibility eliminates the need for physical job-hunting efforts and ensures that opportunities are within reach regardless of geographic location. It also allows employers to reach a broader talent pool.
3. **Enhanced Interaction:** With features like real-time application tracking, messaging, and appointment scheduling for interviews, Jobby strengthens communication between candidates and employers. This leads to quicker feedback, fewer missed opportunities, and a more transparent recruitment process.
4. **Data Organization and Personalization:** Jobby securely stores users’ application histories, preferences, and match scores, which helps tailor future recommendations. For employers, structured access to applicant data makes follow-up and decision-making easier and more informed.

Ultimately, **Jobby bridges the gap** between traditional job search methods and modern expectations by using intelligent algorithms, real-time data, and intuitive design to make the job search and hiring experience more efficient, fair, and user-centric.

#### OBJECTIVES OF THE PROJECT

#### General Objective

To design and implement **Jobby**, an intelligent and user-friendly job portal and career management system that simplifies recruitment processes, optimizes job searches, and supports continuous professional development.

#### Specific Objectives

* 1. Develop a smart job matching engine using AI algorithms.
  2. Enable recruiters to post and manage job offers seamlessly.
  3. Offer applicants tools like CV/resume builders, career analytics, and progress tracking.
  4. Include a calendar-based interview scheduler and appointment manager.
  5. Create messaging/chat systems for candidate-recruiter communication.
  6. Implement push/email notifications for job alerts, application updates, and interview reminders.
  7. Provide an admin interface to oversee platform activity, monitor user behavior, and ensure compliance.

#### EXPRESSION OF NEEDS

#### Functional Needs

Functional needs refer to the specific capabilities and features that the **Jobby** platform must provide to meet the requirements of the primary users — **job seekers**, **recruiters**, and **administrators**. These functionalities are essential to ensure a smooth and effective user experience across the job application and recruitment lifecycle.

1. **User Registration and Authentication**: The platform must support secure account creation and login for different user roles:

* **Job Seekers**: Register, log in, and manage personal profiles, resumes, and applications.
* **Recruiters**: Create and manage employer profiles, post jobs, and review applicants.
* **Admins**: Oversee user activities, verify accounts, and maintain platform security and integrity.

1. **Profile Management**:

* **Job Seekers**: Should be able to update and manage personal details, educational background, work experience, skills, certifications, and upload CVs/resumes or portfolios.
* **Recruiters**: Should manage company profiles, job listings, and receive statistics on candidate engagement.
* **Admins**: Can monitor and modify user profiles when necessary for compliance or moderation.

1. **Job Posting and Discovery**:

* **Recruiters**: Can post detailed job listings with descriptions, qualifications, deadlines, and locations.
* **Job Seekers**: Can search, filter, and browse job listings by categories such as industry, location, salary, experience, etc.
* **AI Suggestions**: The system provides intelligent job suggestions based on a user’s profile, past searches, and applications.

1. **Interview Scheduling and Calendar Management**:

* Integrated **calendar module** for recruiters to schedule interviews and sync with applicant availability.
* Notifications/reminders for upcoming interviews.
* Support for **in-app interview links or virtual meeting tools** (e.g., Zoom, Google Meet).

1. **Application Management and Tracking**:

* **Job Seekers**: Should be able to apply for jobs, track application statuses (e.g., pending, shortlisted, rejected), and receive interview updates.
* **Recruiters**: Can view applicant profiles, filter them by criteria, and update their application status.

1. **Notification System**:

* Real-time and scheduled **notifications** for job postings, application updates, messages, interviews, and system alerts.
* Support for **email, push notifications**, and in-app alerts.

1. **Reporting and Analytics**: The platform should generate reports and analytics for tailors, including order history, revenue tracking, and client statistics to help them make informed business decisions.

#### Non-Functional Needs

Non-functional needs refer to the system's overall performance, usability, and operational requirements. These address how the system should perform rather than what it should do.

1. **Scalability**: The platform must be designed to scale smoothly as the number of users grows including thousands of job seekers, recruiters, and job postings without negatively impacting system performance or user experience.
2. **Security**: Jobby must ensure robust security mechanisms to protect sensitive data such as personal details, resumes, employer credentials, and payment information. All data should be encrypted in transit and at rest. Role-based access control and multi-factor authentication should be implemented to prevent unauthorized access.
3. **Reliability**: The system must be reliable and available at all times, with minimal downtime. A reliable backup system should be in place to prevent data loss.
4. **Performance**: The platform should have fast loading times, quick response to user inputs, and efficient data processing to ensure a smooth user experience.
5. **Cross-Platform Compatibility**: The platform should be compatible with various devices and operating systems, including web browsers, Android, and iOS devices.
6. **User-Friendly Interface**: The UI/UX should be intuitive and clean, catering to users with different technical proficiencies. Clear navigation, consistent styling, and meaningful feedback (e.g., success/failure messages) should help users easily complete tasks like applying for jobs, posting listings, or updating profiles.
7. **Accessibility**: The platform should be accessible to all users, including those with disabilities, by following web accessibility standards (e.g., WCAG 2.1).
8. **Maintainability**: The codebase should be well-documented and modular, allowing for easy updates, bug fixes, and feature additions.

#### TARGET POPULATION AND BENEFICIARIES

**Job Seekers**:

* **Primary Users**: Individuals actively seeking employment opportunities across various sectors and experience levels, including students, recent graduates, professionals in transition, and freelancers.
* **Key Benefits**:

1. Access to a wide range of verified job postings.
2. Personalized job recommendations based on skills, experience, and career goals.
3. Tools to create and manage digital resumes/CVs and professional profiles.
4. Ability to apply directly to job offers and track application status in real time.
5. Career development features such as skill assessments, certifications, and interview preparation.

**Recruiters and Employers:**

* **Primary Users**: HR managers, recruiters, and organizations looking to post job offers, scout qualified candidates, and manage hiring pipelines.
* **Key Benefits**:

1. Ability to post job openings with filtering options to reach the most suitable candidates.
2. Access to a rich database of candidate profiles with advanced search and filtering tools.
3. Streamlined communication with applicants through the platform’s messaging system.
4. Tools to track the status of each hiring process and manage interviews efficiently.
5. Analytics dashboards to monitor recruitment metrics and optimize hiring strategies.

### ESTIMATED COST OF PROJECT

#### Hardware Resources

Table 4 Hardware Resources of the project (Source: Mercurial 2023-2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES HARDWARE USAGE QUANTITY | | | | UNIT COST (FCFA) |
| Computer | Hp, 20GB RAM, 1TB Hard Drive;  intel core i3 | Report writing and editing,  analysis, coding | 1 | 368000 |
| Removable  Drive | 8GB USB key | For file  transferring  from one computer to  another | 1 | 8338 |
| Mobile phone | Iphone XR | Used for emulating the  application | 1 | 100,000 |
| TOTAL 1 |  |  |  | **476, 338** |

#### Software Resources

Table 5:Software ressources (source: Mercurial 2023/2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES SOFTWARE | | USAGE | QUANTITY | UNIT COST (FCFA) |
| Development  tool | Visual Studio  Code | Code Editing | 1 | Open Source |
| Database  Management  System | SQL | Communication  with the  database | 1 | Open Source |
| Operating  System | Microsoft  Window 10 pro | Computer  Operating  System | 1 | 126 000 |
| Project Planner | Gantt Project | Project Planning  Tool | 1 | Free Software |
| Text Editor | Microsoft  Office Pro 2016 | For writing and structuring the internship report | 1 | 161 000 |
| Modelling Tool | Visual Paradigm | Modelling the system in UML | 1 | 439 450 |
| TOTAL 2 |  |  |  | **726 450** |

#### Human resources

Table 6:Software resources (source: Mercurial 2023/2024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RESOURCES | NUMBER | COST PER DAY | NUMBER OF DAY | COST (FCFA) |
| Project  Manager | 1 | 70000 | 70 | 4 900 000 |
| Analyst | 1 | 30000 | 31 | 930 000 |
| Designer | 1 | 40000 | 30 | 1 200 000 |
| Programmer | 1 | 30000 | 30 | 900 000 |
| TOTAL 3 |  |  |  | **7 930 000** |

#### Total Project Estimation

Table 7: Total project Estimated cost

|  |  |
| --- | --- |
| Designation | Cost (FCFA) |
| Human Resources | **7 930 000** |
| Software Resources | **726 450** |
| Hardware Resources | **476,338** |
| Unforseen | 900,000 |
| TOTAL | **10,132 788** |

### ESTIMATION OF TIME REQUIRED

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHASE OBJECTIVE | | OUTPUT | DURATION | PERIOD |
| Insertion | Welcome and integration in the company and attribution of internship themes. | Insertion  Report | 6 days | 01st to 08th July  2025 |
| Specification | Description of project functionalities | Specification book | 4 days | 08th to 11th July 2025 |
| Analysis | Analysis of the system | Analysis  Book | 5 days | 14st July to 18th  July 2025 |
| Conception | Hardware and software conception, | Conception  Book | 4 days | 18th July to 22th July  2025 |
| Realization | Realization of the system | Realization document | 6 days | 22th July to 29th July  2025 |
| Deployment | Deployment of the  system | Running system | 6 days | 1st August to 8th  August  2025 |
| Writing user guide | Instruction and indication on how to use the software | User guide | 4 days | 9st August to 12th  August  2025 |

This activity consists of determining tasks and putting them in order by presenting them in interval of time. We will present it on a table and a Gant diagram.

Table 8: Estimation of required time for the project

We have a period of three month to carry out our project. Project planning helps us schedule the different task of the project to permit the successful and on-schedule realization of the project. We made use of a timing activity and Gantt chart to illustrate the various tasks for our project. We will execute the tasks successively, with each task commenced upon the completion of its predecessor*.*

GANTT CHART

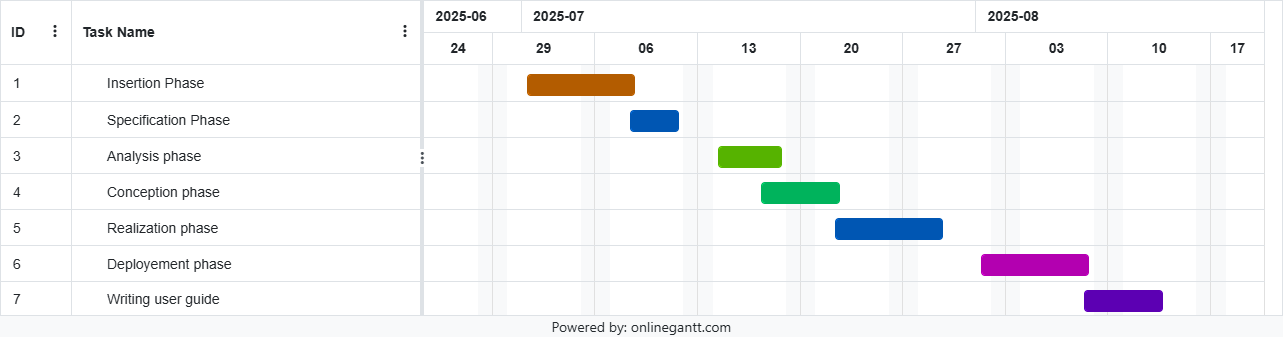
**

Table 9:Gantt chart

### CONSTRAINTS

#### Technical constraint

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

#### 2. Time Constraint

The project will be realized in 14 weeks starting from the beginning date coupled with many other school projects.

#### 3. Cost constraint

The realization of our project will require expenditures in human resources, material and software for a total of **10,132 788 FCFA**.

### LIST OF PARTICIPANTS AND DELIVERABLES

#### List of participants

|  |  |  |
| --- | --- | --- |
| Name Title Role | | |
| Mr OSTIE JOSPIN MBITA | Software Engineer | Professional Supervisor |
| Mr NDIFOR | Lecture at AICS-Cameroon | Academic Supervisor |
| Mr NKAYOU FRANCK LANDRY | Software Engineering  Student at AICS-Cameroon | Analyst and Developer |

Table 10:Project participants

#### Deliverables

In project management, any component materializing the result of a realization.

At the end of this project, we are expected to submit the following;

A complete report containing

* Insertion Book
* Specification Book
* Analysis Book
* Conception Book
* Realization Book
* Software Setup
* User’s guide

## CONCLUSION

Having come to the end of our specification book, we recall that for this part of our report, we had set as objectives: to clarify the scope of our project by defining its context and demonstrate its relevance through the justification. We were also to set objectives for our project, express the needs it will respond to, estimate the cost of the project and define a detailed plan we will follow to realize our project. In addition to the objectives set for this section of the report, we saw the list of participant and deliverable for our project. In the time allocated for this section, we were able to meet all the set goals which means we can advance to the next section of our report, 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.

# BOOK THREE:

# 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

1. PRESENTATION OF THE ANALYSIS APPROACH
2. MODELLING OF THE PROPOSEDSOLUTION

CONCLUSION

## INTRODUCTION

As Engineers our role in the society is problem-solving which is targeted as making life easier for everyone and in order to bring forth a solution, we ought to ask ourselves some questions to know why the problem exist in the first place and how it is being dealt with at the time. Most often we realize that there is an existing system put in place to solve the problem irrespective of its nature, we either need to bring forth a new system or bring improvement to the existing system. No matter which routes we decide to take we need to study and understand the functioning if the system our propose solution to really make a difference. System development can be thought of as having two major components; ***system* *analysis*** and ***system* *design*.** In system analysis, the emphasis is placed on the understanding the details of the existing system, propose a new one and then deciding whether the proposed system is desirable or not, or whether the existing system needs improvements

### PRESENTATION OF THE ANALYSIS METHOD

We studied multiple analysis methods in order to get a better orientation on which one to choose for our system. Below are some analysis methods we studied;

#### A. SOME ANALYSIS METHODS/APPROACH

1. ***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.
2. ***SCRUM:*** In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. This is why in Scrum development, for example, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, exit criteria (ETVX) and so on, as would be provided in most methodologies. Agile scrum methodology is a project management system that relies on incremental development. Each iteration consists of two- to four- week sprints, where each sprint's goal is to build the most important features first and come out with a potentially deliverable product. More features are built into the product in subsequent sprints and are adjusted based on stakeholder and customer feedback between sprints.

1. ***UP:*** The UP is an abbreviation of Unified Process. It is an iterative and incremental software development methodology. The Unified Process is an iterative, architecture centric software development process driven by use cases and geared towards reducing risk. It is a process pattern that can be adapted to a wide class of software systems, to different areas of application, to different types of businesses, to different skill levels and to different sizes of the business. Different data. It qualifies a process or a procedure that performs a group of operations repeatedly until a well-defined condition is met.

##### *4.* **DYNAMIC SYSTEM DEVELOPMENT METHOD(DSDM):** It is an

organized, common-sense process focused on delivering business solutions quickly and efficiently rather than just team creativity. It is similar in ways to SCRUM and XP, but it has its best uses where the time requirement is fixed.

### CHOICE OF THE ANALYSIS APPROACH

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

2TUP is a unified process, that is, it is a software development process built on UML. The 2TUP brings an answer to the constraint of continual changes imposed by the information system of an enterprise. In this sense, it enforces the control on the capacity of evolution and correction of the system. The 2TUP offers a Y-shaped process that divides the technical aspects giving two branches, namely the functional branch and the technical branch, which justifies why it is a two-track process. These two branches finally merge into the realization branch for the implementation of the system. The figure below illustrates the 2TUP:

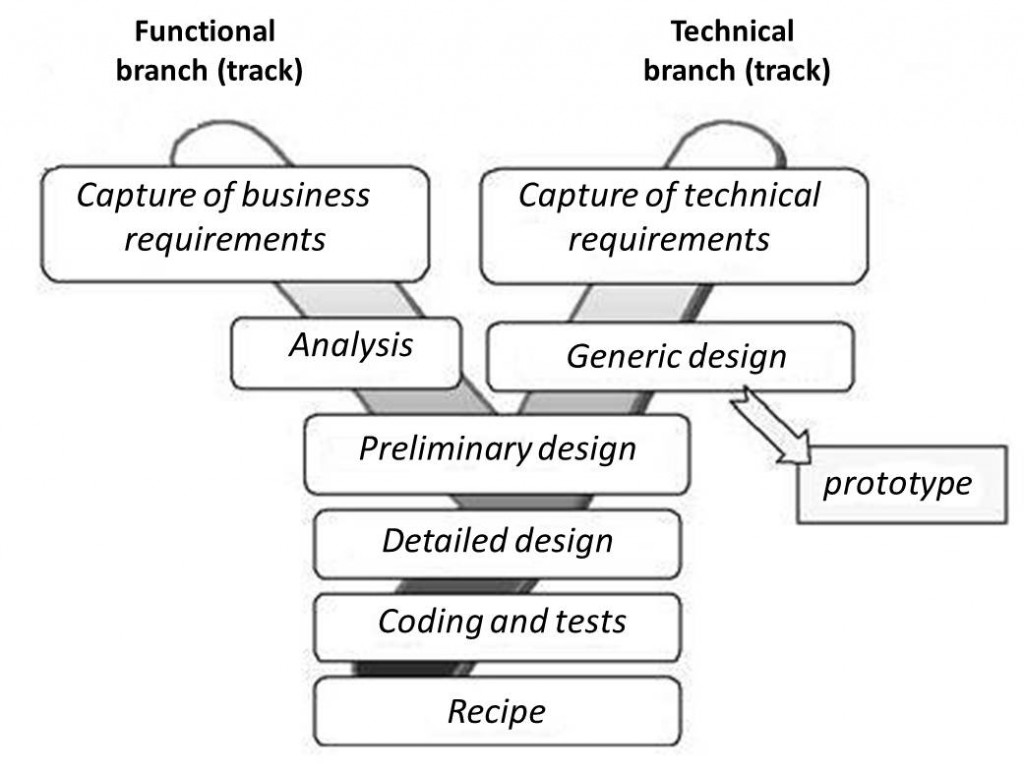


Figure 2: 2TUP Diagram (Source: www.memoireonline.com)

#### a) Functional Branch

The functional branch captures the functional needs of the system and analyse them. This phase specifies the elements of the preliminary study and does not depend on technology used to build the system.

#### b) The Technical Branch

The technical branch enumerates the technical needs and proposes a generic design validated by a prototype. The technical needs here include the tools, materials and technologies that will be used along with the different constraints such as worst-case scenario and integration with the existing controls.

#### c) Realization Branch

The realization track supports the following:

* The preliminary conception: This is the most sensitive step of the 2TUP. This is the meeting point between the functional and technical branches. It ends when the deployment model, the operating model, the logical model, interfaces and software configuration model are defined;
* The detail conception: This is the detailed design of each feature of the system;
* Coding and testing: This are the programming phase of the designed features, alongside testing of the coded features;
* The Recipe: This is the validation phase of the functions of the system developed

#### Modelling with UML 2.5

The 2TUP and UML work in close collaboration. Having already discuss briefly on 2TUP, UML being its foundation is not to be taken lightly. **UML** stands for **Unified Modelling Language**. It can be used to model a system independent of a platform language. UML is a graphical language for visualizing, specifying, constructing, and documenting information about software-intensive systems, some of which are object oriented. UML gives a standard way to write a system model, covering conceptual ideas, hence it is a privilege vector of communication between members of a team. Before proceeding, here are some key terms to understand:

**Model:** A model is an idealized, abstract and simplified representation of a real-world object or a simplified simulation of an entity.

**Why is UML Unified:** UML is said to be unified because it is a combination of three modelling approaches namely:

1. Object Modelling Technique (OMT) which provides a graphical representation of the static, dynamic and functional aspects of a system;
2. Booch approach which was excellent for design and implementation. It introduces the notion of packages;
3. OOSE (Object-Oriented Software Engineering) approach which focuses on design base on the user’s needs.

How is UML a language: The UML notations are a standard are widely in the professional milieu. The notations are a must, however the usage these notations in a software development approach are not, hence it is just a guide line.

UML 2.5 defines 14 diagrams that are classified into two main categories: Structural and Behavioural diagrams

#### a) Structural diagrams:

Structural diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts. In UML 2.5 there are seven types of structural diagrams as follows:

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

#### b) Behavioural diagrams

Behavioural diagrams show the dynamic behaviour of the objects in a system from the beginning of a task to its completion. Below are the behavioural diagrams in UML 2.5:

* Use case diagram;
* Activity diagram;
* State Machine diagram;
* Sequence diagram;
* Communication diagram;
* Interaction Overview diagram;
* Timing diagram



Figure 3: UML 2.5 Diagrams Overview (source: http://www.uml-diagrams.)

### VIII. MODELING OF THE PROPOSED SOLUTION

###### **Definition**

Use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

###### **Formalism**

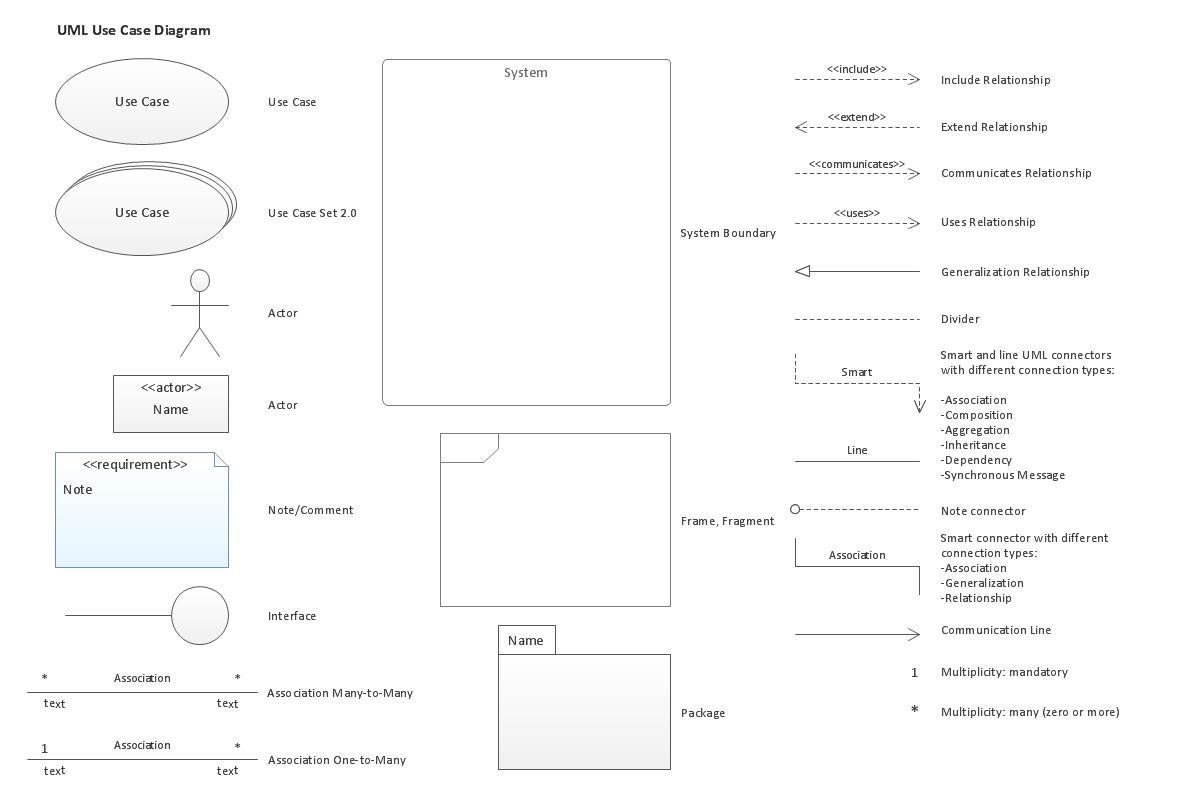
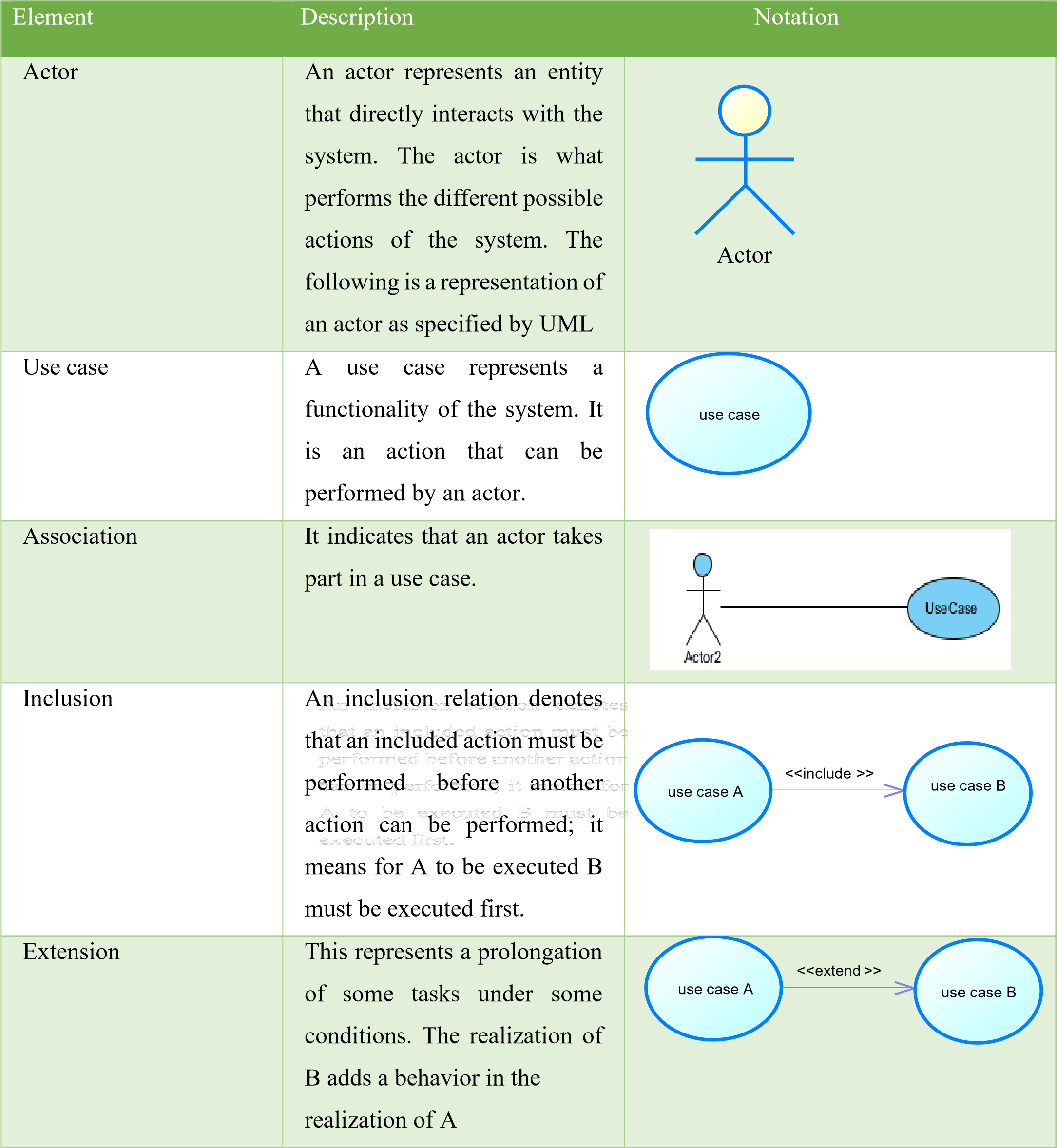


Figure 4: formalism of a Use Case diagram

The components of a use case diagram are illustrated below:

Table 11:Use case diagram components



|  |  |  |
| --- | --- | --- |
| Generalization | This show that an actor or 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 |  |

**A. LIST OF ACTORS:**

* **Administrator or Super user:** The administrator is responsible for system administration of users and transaction.
* **Job seeker:** Thes are parties interacting with the application in order to benefit from the services.
* **Employer:** This is verified personnel who post job openings, view job applications

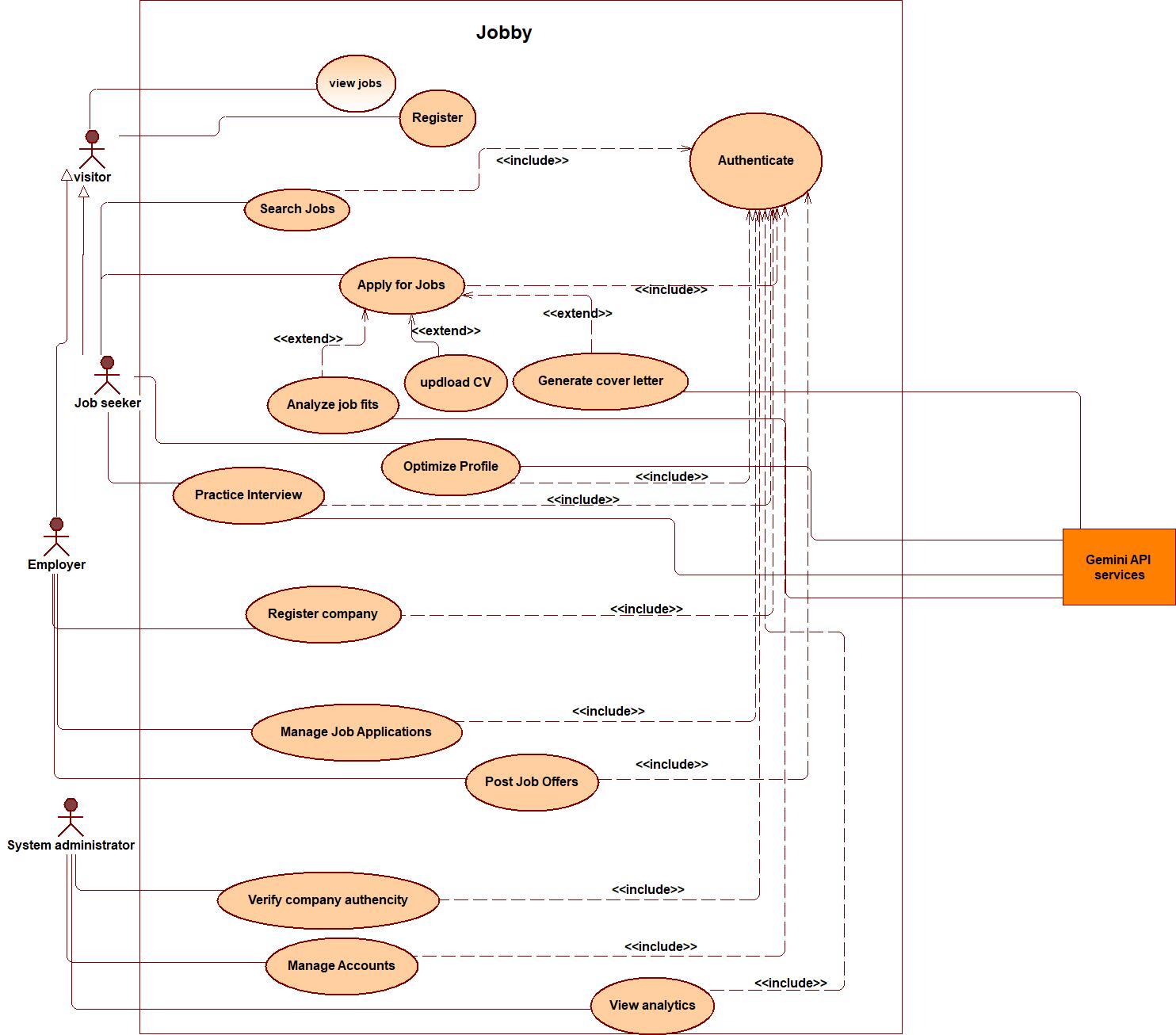
USE CASE DIAGRAM: GENERAL USE CASE

Figure 3:General Use case

## Use case: Manage application



Figure 5:Use case Manage application

## Use case: Post Job Openings

Figure 6: Use case post job opening

## Use case: Apply for Jobs

Figure 7: Use case Apply for job

b) Textual description of use cases

UML allows the execution of a use case to be described in a textual way, in a form called nominal scenario. A nominal scenario describes in more detail the execution of a use case by an actor until it is successfully completed.

***Formalism***

A textual description of a use case is represented in the following form

Table 12: Formalism of Textual Description

|  |  |
| --- | --- |
| NAME OF THE USE CASE | |
| Actors | Users |
| Objective | Aim of the use case |
| Presuppose | Set of actions that must be completed before the launching of the use case |
| Post condition | Set of actions that must be completed before the launching of the use case |
| Triggers | Element that triggers the use case |
| Principal Scenario | Description of the principal scenario |
| Alternative Scenario | Descriptions of alternative scenarios (where the nominal scenario is a failure) |
| Post condition | Set of mechanism that can lead to the end of the use case |

TEXTUAL DESCRIPTION OF REGISTER/CREATE ACCOUNT

Table 13: Textual description for register/create account

|  |  |  |
| --- | --- | --- |
|  | USE CASE REGISTER | |
| Actors |  | Employer, Job Seeker |
| Objective |  | Grant the user’s access to the application |
| Precondition |  | The system is working properly |
| Triggers |  | The user launches the application |
| Principal Scenario |  | 1. Users click on the application Icon 2. The system displays an onboarding screen 3. After the onboarding screen the users click on get started. 4. The Application display the login page 5. The user clicks on register 6. The Application displays the Registration page 7. The user fills the empty field with his credential 8. The user clicks on register button 9. The application check whether the data format entries are correct 10. The application sends a request to the database server which execute it and return a response. 11. The application analysis the respond return by the database server 12. The user is directed to his page |
| Alternative Scenario |  | 9-a) If at step 7 of the principal scenario the data format is not correct, an error message is return to the user and he return to step 4 of the principal scenario  11-a) If at step 10 of the principal scenario a user was found during the verification, the system returns an error message and we return to step 4 of the principal scenario |
| Post condition of success |  | The user’s account is created and he access his page |
| Post condition of failure |  | An error message is display and user does not have access his space. |

##### TEXTUAL DESCRIPTION OF AUTHENTICATION

Table 14: Textual description of authentication

|  |  |
| --- | --- |
| USE CASE AUTHENTICATE | |
| Actors | Job seeker, Employer, Admin |
| Objective | Grant the user’s access to the application |
| Precondition | The user should have an active account |
| Triggers | The user launches the application |
| Principal Scenario | 1. Users click on the application Icon 2. The system displays an onboarding screen 3. After the onboarding screen the users click on get started. 4. The Application server display the login page 5. The user fills the empty field with his credential 6. The user clicks on login button 7. The application server check whether the data format entries is correct 8. The application server sends a request authentication to the database server which execute it and return a response. 9. The application server analysis the respond return by the database server 10. The users are directed to his page |
| Alternative Scenario | 7-a) If at step 7 of the principal scenario the data format is not correct, an error message is return to the user and he return to step 4 of the principal scenario  9-a) If at step 9 of the principal scenario no users were found during the verification, the system returns an error message and we return to step 4 of the principal scenario |
| Post condition of success | The users have access to his page |
| Post condition of failure | An error message is display and user does not have access his space. |

##### TEXTUAL DESCRIPTION OF MANAGE APPLICATION

Table 15: Textual description of Manage Application

|  |  |
| --- | --- |
| USE CASE MANAGE APPLICATION | |
| Actors | Employers |
| Objective | Review applications submitted by job seekers |
| Precondition | The user must be authenticated |
| Triggers | The recruiter clicks on the "View Applications" button |
| Principal Scenario | 1. The recruiter clicks on a job post from their dashboard. 2. The system displays a list of submitted applications for the selected job post. 3. The recruiter selects one application to review. 4. The system displays the detailed profile of the job seeker, including resume, cover letter, and any additional documents. 5. The recruiter selects an action and confirms. 6. The system processes the server response and updates the UI accordingly. |
| Alternative Scenario | 4-a) If no applications are found for the selected job post, the system displays a message: *"No applications submitted yet."* and returns to step 1.  6-a) If at step 6 of the principal scenario no user is found during the verification, the system returns an error message and we return to step 5 of the principal scenario |
| Post condition of success | The recruiter has successfully reviewed and taken action on the selected application, and the application status is updated accordingly. |
| Post condition of failure | An error message is display and user does not have access his space. |

## TEXTUAL DESCRIPTION OF Post Job Openings

Table 16: Textual description Post Job Openings

|  |  |
| --- | --- |
| USE CASE POST JOB OPENINGS | |
| Actors | Employer |
| Objective | Post a new job offer on the platform |
| Precondition | The recruiter must be authenticated (logged into their account). |
| Triggers | The recruiter clicks on the **“Post a Job”** button. |
| Principal Scenario | 1. The recruiter clicks on the **“Post a Job”** button from their dashboard. 2. The system displays the job posting form. 3. The recruiter fills in the job details (title, description, category, location, salary, requirements, etc.). 4. The system checks whether the data entries are correct 5. The recruiter submits the form 6. The system sends a request to the server, which executes and returns a response 7. The system processes the server response and updates the UI accordingly |
| Alternative Scenario | 4-a) If at step 4 of the principal scenario, the data format is not correct, an error message is returned to the user, and they return to step 3 of the principal scenario 6-a) If at step 6 of the principal scenario, no valid user is found during the verification, the system returns an error message, and the process returns to step 4 of the principal scenario |
| Post condition of success | The job posting is successfully published on the platform and is visible to job seekers. |
| Post condition of failure | An error message is displayed, and the job post is not published until corrected and resubmitted. |

#### b) Communication Diagram

##### Definition

Communication diagram model the interaction between objects in a sequence. A communication diagram is more focused on showing the collaboration of objects rather than the time sequence. Communication diagrams are especially good at showing which links are needed between participants to pass an interaction’s message.

##### Formalism

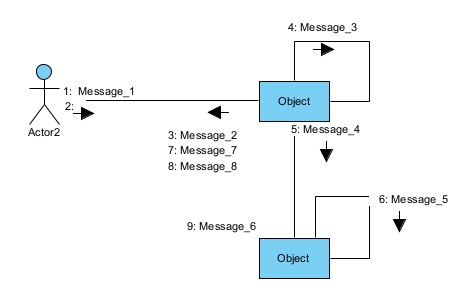


Figure 8: Formalism of a communication diagram

Table 17:Component of the Communication Diagram

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Call Message |  | A call message defines a particular communication between lifeline of an interaction |
| Dependency |  | A dependency is a relationship that signifies that a single or set of model elements require other model elements for their specification |
| Object |  | An object represents an individual participant in the interaction conversion |
| Generalization |  | A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. |

##### Authentication Communication Diagram

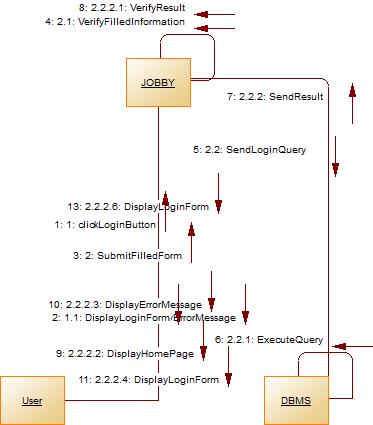
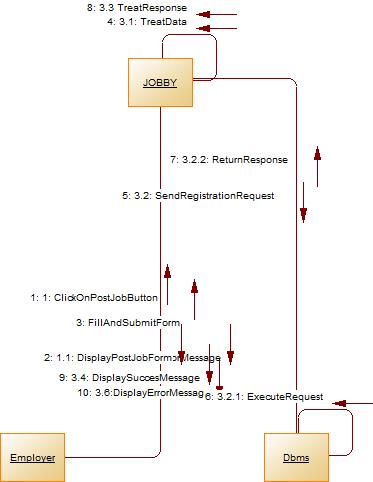


Figure 9: Authentication Communication Diagram

##### Create Account Communication Diagram

Figure 10: Create account communication diagram

Post Job communication diagram

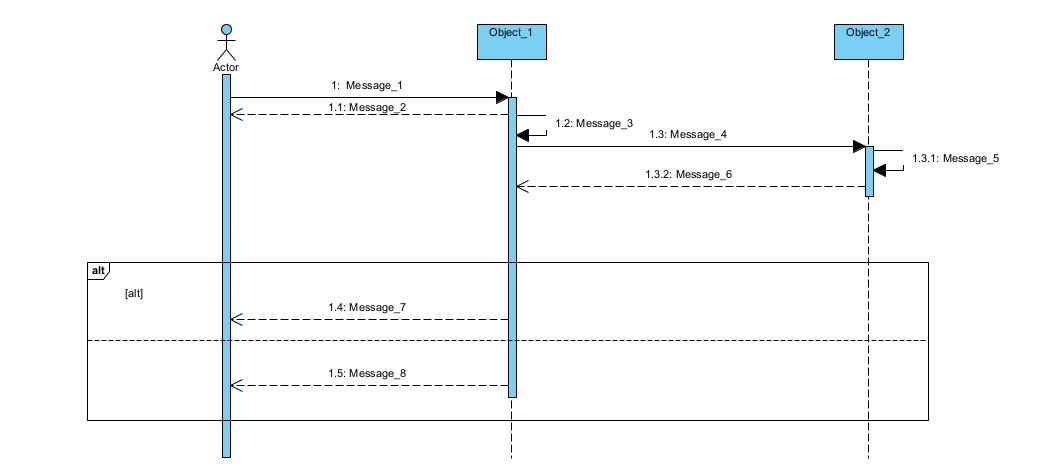
  
Figure 11: Post Job communication diagram

#### c) Sequence Diagram

##### Definition

A sequence diagram is a diagram that shows the interaction details, and how operations are carried out. Sequence diagrams focus on the message interchange between a number of lifelines.

##### Formalism



[

OK

]

[

NOT

Figure 12: Sequence Diagram Formalism

Table 18:Component of a Sequence Diagram

|  |  |  |
| --- | --- | --- |
| Element Representation Description | | |
| Lifeline |  | It represents the presence of an object over time in the execution of a modelled function. |
| Activation |  | It is used to represent a period during which an objective is active when a modelled use case is running |
| Message |  | Message are arrows that represent  communication between objects: Synchronous message: Message completed with feedback. Asynchronous message: Message complete without waiting for feedback. |
| Combined fragment |  | Used to group messages together to show conditional flow in a sequence diagram |

##### SEQUENCE DIAGRAM: Create Account/ Register

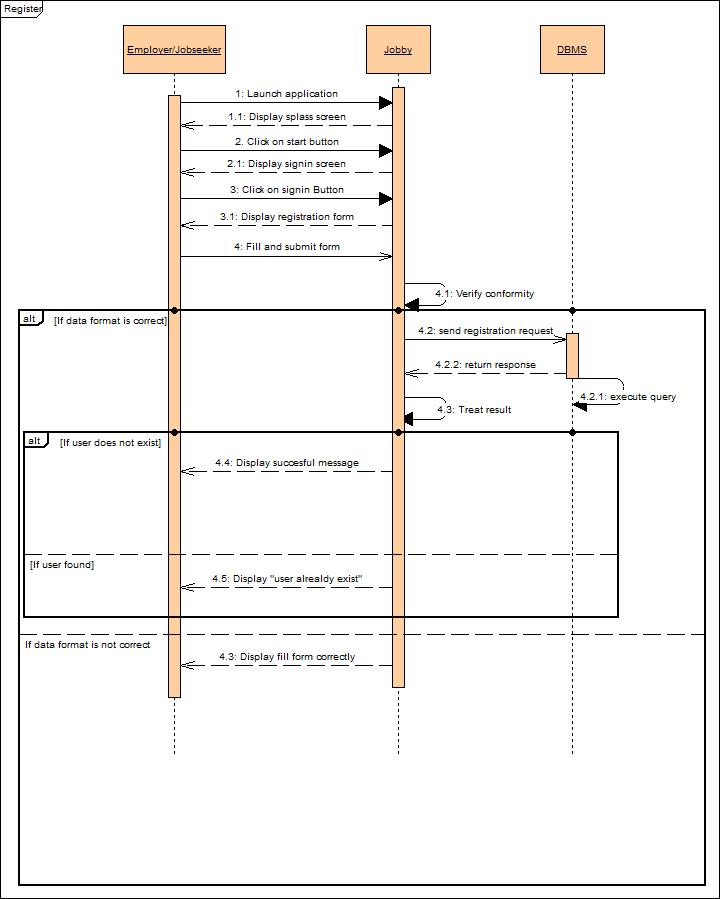


Figure 13: Create account Sequence diagram

##### SEQUENCE DIAGRAM: AUTHENTICATE

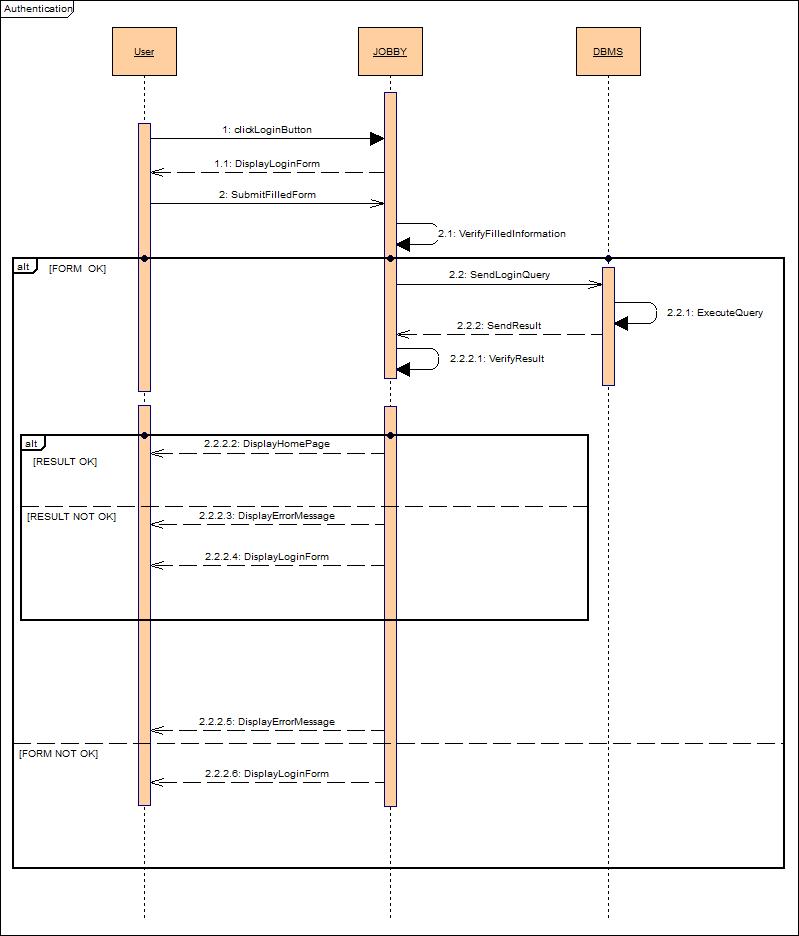


Figure 14: Authentication Sequence diagram

##### SEQUENCE DIAGRAM: Post Job

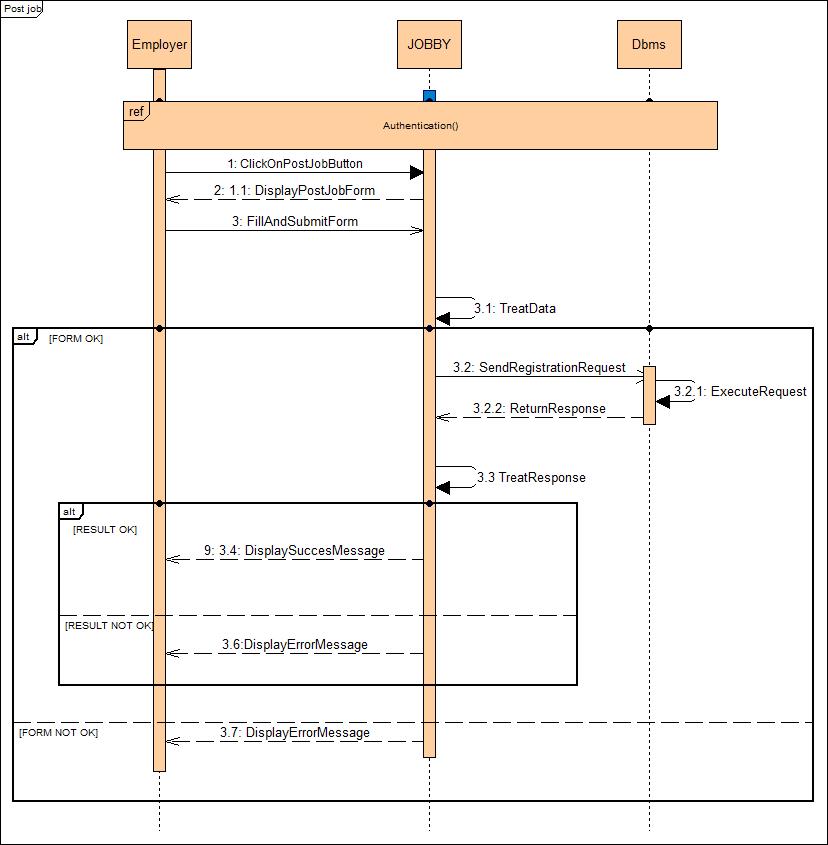


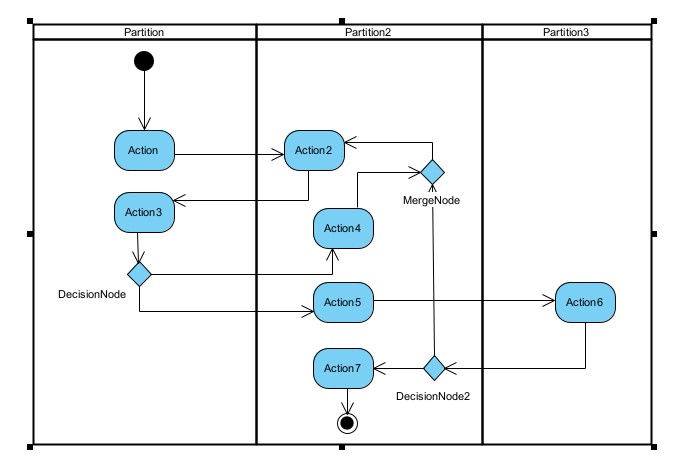
Figure 15: Post Job sequence diagram

##### d) ACTIVITY DIAGRAM

###### Definition

An activity diagram is a graphical representation of workflows that show the step needed in the realization of a process showing the details from start point to an end point through all the decision and actions that can possibly be performed. Activity diagrams are intended to model both the computational and organizational process. They show the overall flow, which is drawn from one operation to another. This flow can be sequential, branched or concurrent. Below is the activity diagram formalism.

Formalism*:*

 Figure 16: Activity Diagram Formalism

|  |  |  |
| --- | --- | --- |
| ELEMENT  DESCRIPTION | | NOTATION |
| Activity | An activity is shown as a rounded-concerned rectangle enclosing all the actions, control flow and all the element that make up the activity. |  |
| Action | It represents a single step within an activity. Actions are generally denoted by rounded cornered rectangles |  |
| Transition (Control Flow) | Control flows show the flow of control from one action to the next. It’s represented by a line with an arrow head |  |
| An Initial or Start node | It is depicted by a large black spot |  |
| Final Nodes | They are diamond shaped having control flows with guard conditions |  |
| Object Node | An object node is an activity node that indicate an instance of a particular classifier or a particular point in the activity |  |
| Swim Lanes | This is where we place activities. Items are listed inside it. |  |
| Flow Final node | This indicates the end of a flow |  |

*Table 19: Component of an Activity Diagram*

###### *ACTIVITY DIAGRAM: Create Account*

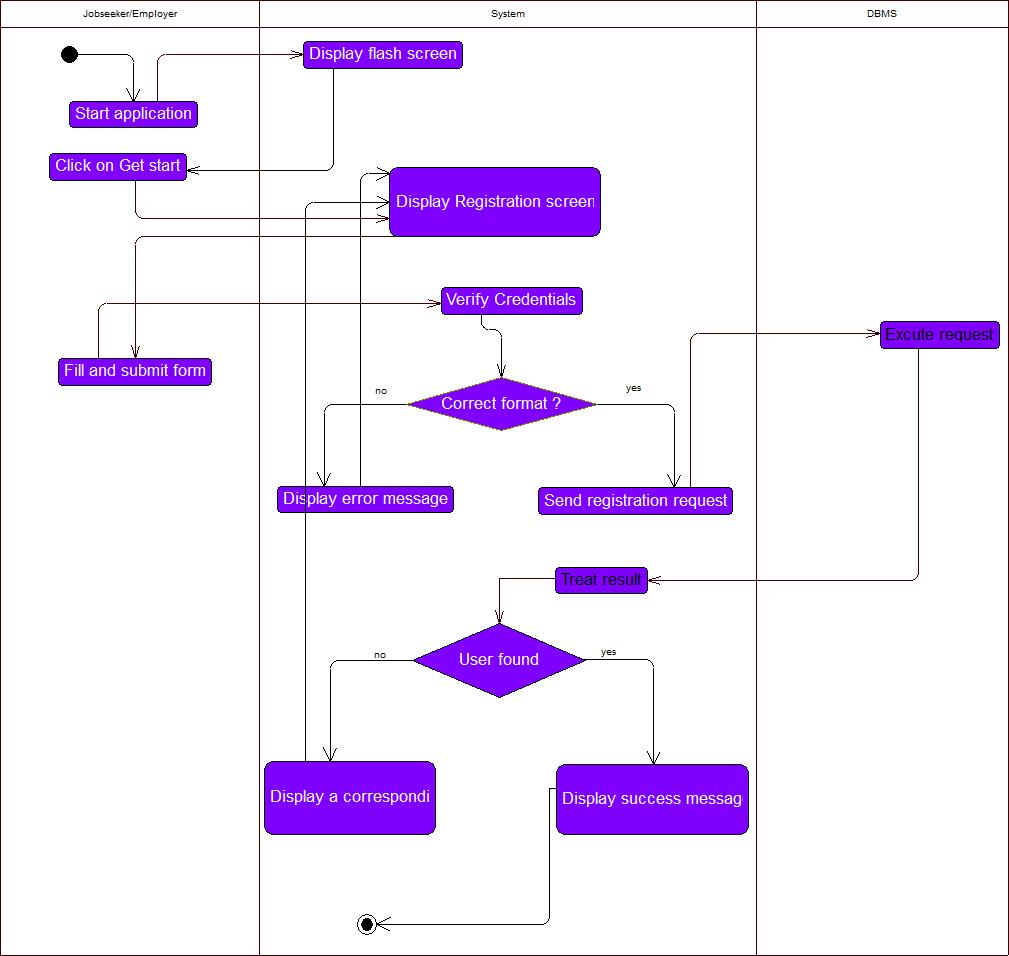


Figure 17 Activity diagram for create account

###### *ACTIVITY DIAGRAM: Login*

Figure 18 Activity diagram for Login

###### *ACTIVITY DIAGRAM: Post a Job*

Figure 19: Activity diagram Post Job

### CONCLUSION

The Analysis document has enabled us to sort out the problem at hand, and then proposed a solution that will lead to the realization of this Application. From here we will proceed to the conception phase where we shall see a detailed conception of the system.

# BOOK FOUR: CONCEPTION PHASE

#### Preamble

The conception phase is part of the document that shows the link between the analysis and the realization phase. It is a continuation of the analysis phase which represents the technical aspects used in modelling our system.

#### Content

Introduction

1. TECHNICAL BRANCH
2. CAPTURE TECHNICAL NEEDS
3. RELATED UML DIAGRAMS

Conclusion,,,,jjj

## INTRODUCTION

The conception phase consists of defining the necessary components to the construction for the technical architecture. This conception is completely independent of the functional aspects. In this phase, we are going to see the technical branch of 2TUP which consist of technical needs and capture of the various UML diagrams that are going to permit us to model our system.

## I. TECHNICAL BRANCH

Generic Design.

### ***Hardware Diagram of the System***

The hardware diagram shows in details how the application components of our system are deployed through the adequate computer network. Below is the hardware diagram of our system



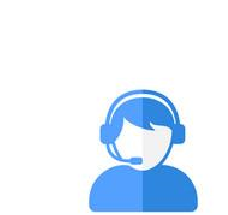
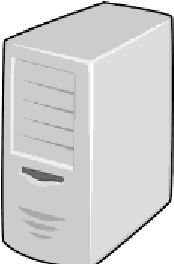
User 1

User

2

User

3



Main server

Database

server

Admin portal

Administrator

Internal domain

Firewall

External domain

*:*

Figure 20:Hardware Diagram of the system

### ***Physical Architecture of the System***

The physical architecture diagram shows the machines in which different application components of our system will be installed. We will use a n-tier achitecture

The tiers of our system include:

* The client tier, that runs on a mobile phone,
* The application tier, which runs on a Java web server;
* The data tier, that runs on a database server.
* The Gemini API

### ***Logical Architecture of the System***

To avoid a lack of maintainability, testability and scalability, which are drawbacks of the traditional approach of programming which works on Input-Processes-Output, we opted for the MVC architecture of our application. MVC is an Acronym for **Model View** and **Controller**. The MVC architecture is a design pattern that is used in software engineering to separate the application’s logic from the user interface. As the name implies, the MVC is described in three layers, namely:

* The Model (business logic and access to data);
* The View (user interface);
* The Controller (request handler) performs the action of invoking the Model and sending data to the View.

The MVC layers are independent from one another. A change in the content of one will have no effect on the others. This is a great advantage in that it facilitates maintenance and follow up of the software. Below are derails concerning the layers:

* **Model**: Model objects are the parts of the application that implement the logic for the application’s data land registry. Model objects retrieves and store model state in a database. The model itself can be sub divided into many levels but this decomposition is not shown at the MVC level
* **View:** The view has components that display the application’s user interface (UI).

It is the presentation layer used to display the Model data fetched by the controller.

* **Controller:** This layer acts as an interface between View and Model. It receives requests from the View layer and processes them, including the necessary validations. The request is further sent back to the Controller and then displayed on the View.

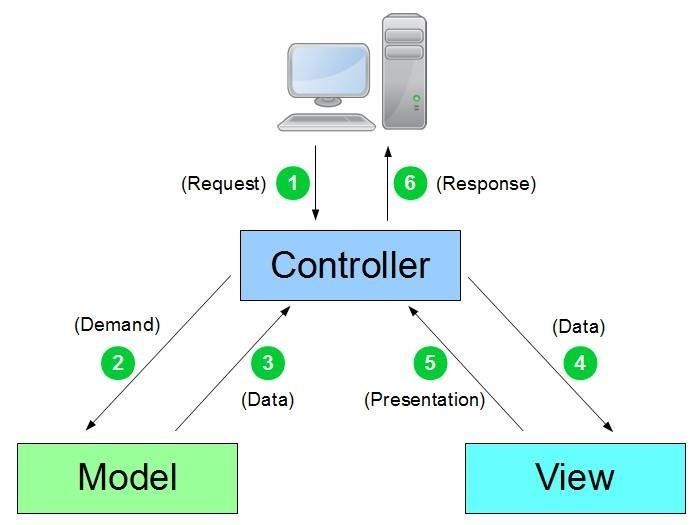


Figure 21: MVC pattern diagram

#### Advantages of MVC Design Pattern

* **Simultaneous development:** MVC architecture makes it possible for multiple developers to work simultaneously on the model, controller and views.
* **High cohesion:** MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
* **Low coupling:** The very nature of the MVC frame work is such that there is low coupling among models, views and controllers.
* **Ease of modification:** Because of the separation of responsibilities, future development or modification is easier, that is scalability of the product is increased.
* **Multiple views for a model**: Models can have multiple views.

## RELATED UML DIAGRAMS

#### A) Class Diagram

##### Definition

In UML, a class diagram is a type of static structure diagram that describe the structure of a system by showing the system’s classes, their attributes, their methods or operations and their relationship among objects. Classes are represented with boxes that contain three compartments:

* The top compartment containing the class name;
* The middle compartment containing the attributes of the class;
* The button compartment containing the operations the class can execute.

##### Formalism

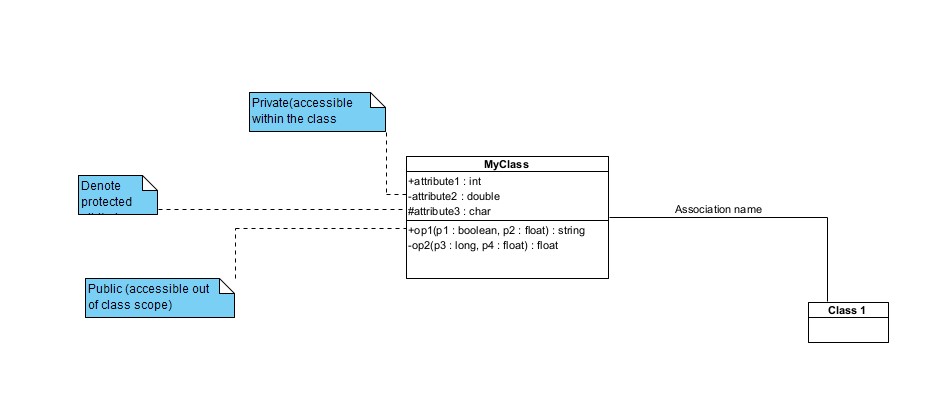


Figure 22: Class Diagram Formalism

Table 20:Component of a Class diagram

|  |  |  |
| --- | --- | --- |
| Element Representation Description | | |
| Inheritance or  Generalization |  | A  generalization is used to  indicate inheritance. It shows a parent class generalizing a child class |
| Association |  | It is the  general relationship type between elements. This connector may include named roles at each end,  cardinality, direction and attributes. |
| Aggregation |  | If the parent of aggregate is deleted, Usually the children are not deleted. |
| Composition |  | If the Parent of a composite is deleted, usually, all of its parts are deleted within it. | |
| Class |  | A class is an element that defines the attributes and behaviour that an object is able to generate. | |
| Dependency |  | Exists  between two classes if changes in the definition of one may cause changes to the other, but not the other way around. | |

##### Class Diagram

Figure 23: System class Diagram

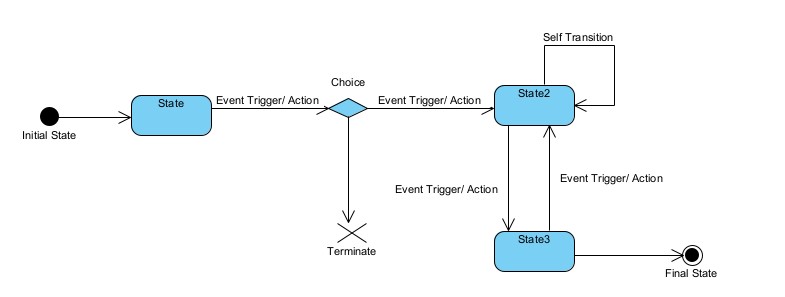
###### B) State Machine Diagram

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 chart diagram, it models the dynamic flow of control from the state of a particular object within a system.

Formalism

*:*



State Machine Diagram

Figure 24:Formalism of a state machine diagram

Components of State Machine Diagram

Table 21:Components of a State Machine diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | DESCRIPTION | NOTATION |
|  |
| States | A state is denoted by a rounded cornered rectangle with the name of the state written inside it. |  |
| Initiates states | The initial states is denoted by a field circle and may be labelled with name. |  |
| Transitions | Transitions from one state to the next are denoted by lines with arrow heads. A transition may have a trigger, a guard and an effect. |  |
|  |  |  |
| Junction | Junction vertices are semantic free vertices that are used to chain together multiple transitions. They are used to construct compound transition paths between states |  |
| Final states | The final state is denoted by a circle with a dot inside |  |
|  |  |  |

STATE MACHINE DIAGRAM: Application

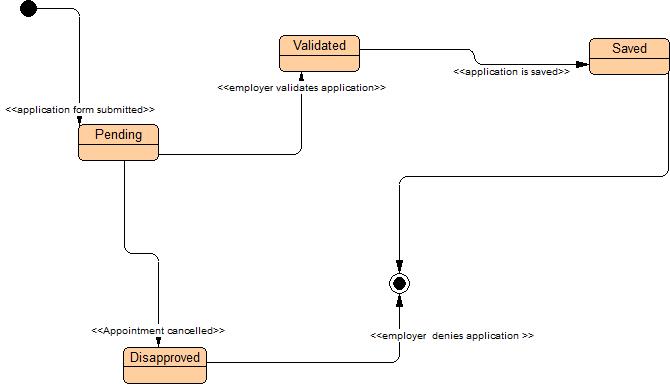


Figure 25: Post Job State machine diagram

#### C) Package Diagram

##### Definition

Package diagrams serve the purpose of reflecting the organization of package and their elements. When used to represent class element, package diagram provides a visualization of the namespaces. The most common use for package diagram is to organize use case diagram and class diagram although the use of package diagram is not limited to these elements.

##### Formalism

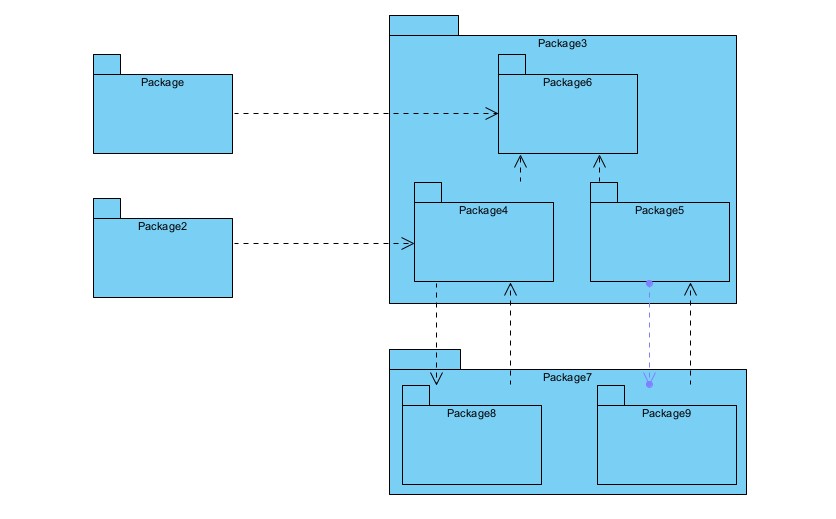


Figure 26: Formalism Package Diagram

Table 22:Package Diagram Formalism

|  |  |  |
| --- | --- | --- |
| Name Representation Description | | |
| Package |  | A package is a namespace used to group together elements that are semantically related and might change  together. It is a general-  purpose mechanism to organize elements into groups to provide a better structure for system model |
| Package  Merge |  | It is a direct relationship between two packages that indicates that the content of the target (merged package) is combined into the source (receiving/merging package). |
| Package  Import |  | This is a direct relationship between an importing namespace and imported package, that allows the use of unqualified names to refer to the package members from other namespaces. |
| Package access |  | It is a direct relationship used to show that a UML element or a set of elements require(s) or depend(s) on another model element for  implementation |

Package Diagram

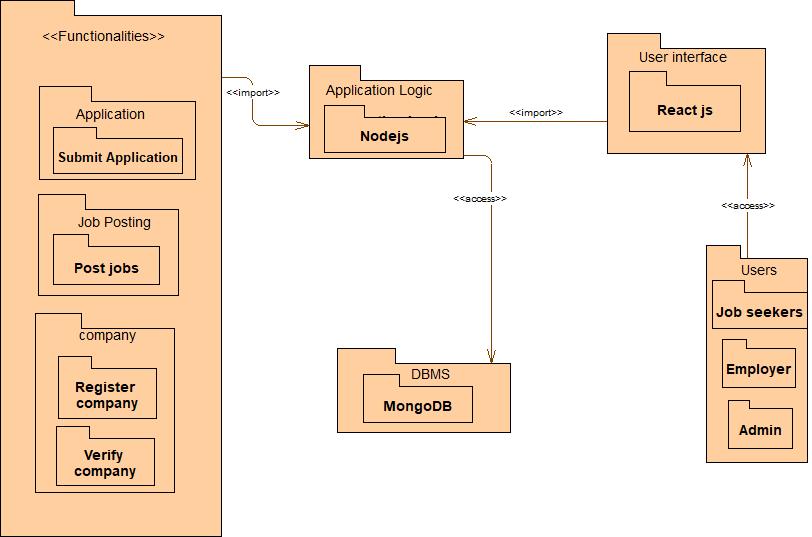


Figure 27: Package Diagram

#### D) OBJECT DIAGRAM

##### Definition

An object diagram is an instance of a class in a particular moment in runtime that can have its own state and data values. It shows a snapshot of the detailed state of the system at a point in time, thus an object diagram encompasses objects and their relationships whish may be considered a special case of a class diagram.

##### Formalism

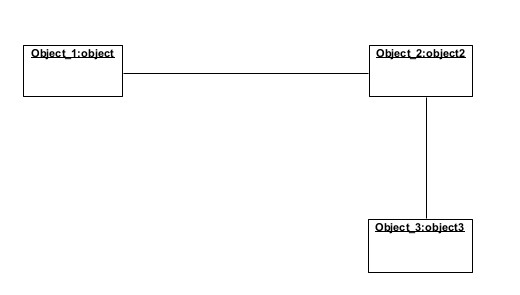


Figure 28: Object diagram formalism

Table 23:Components of an Object Diagram

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| Object |  | An object is an instance of a  class |
| Link |  | Links between  objects  correspond to  associations  between the  object’s classes Thus, a link is an instance of an association. |

### System object diagram

Figure 28:system Object diagram

## CONCLUSION

From the conception document, we can now conceive in detail the proposed solution on the structural plan, organizational, materialization of various models which are the use case to have a view of how the realization of our solution to the problem posed will be implemented. Hence the next step will be essentially be consecrated to the realization of the solution.

# BOOK FIVE: REALIZATION PHASE

### Preamble

The realization phase contains the fact that are strongly related to the analysis phase and aim at the physical implementation of the software

### Content

INTRODUCTION

1. COMPONENT DIAGRAM
2. DEPLOYEMENT DIAGRAM
3. CHOICES OF TECHNOLOGIES

CONCLUSION

### a) Component Diagram

##### Definition

A component diagram represents modular aspect of an object-orientated system that encapsulate their content and whose manifestation is replaceable within their environment. The modular aspect includes: run-time, executable and source code components.

##### Formalism

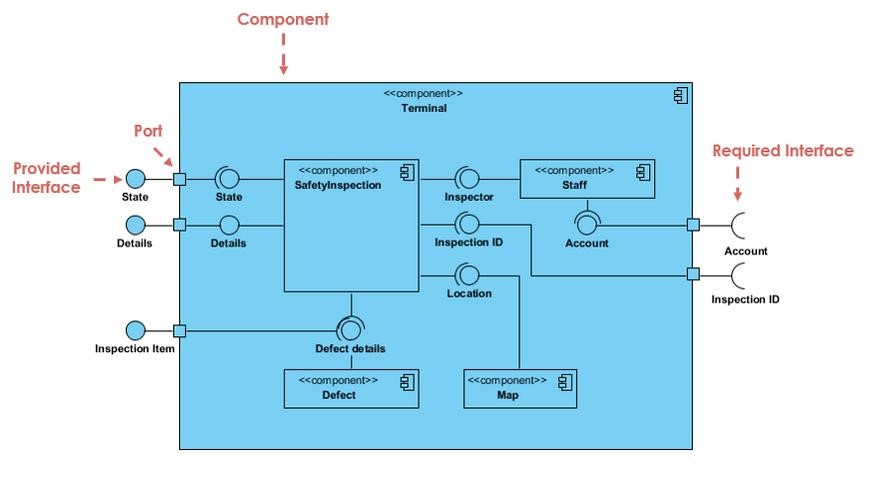


Figure 29:Component diagram formalism (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-iscomponent-diagram)

Table 24: Formalism 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 a smaller rectangle in the upper right corner with tabs, or the word written above the name of the  component |
| Dependency |  | Dependency is a directed relationship which is used to show that some component or set of components depend on other component elements for specification and implementation. It is represented with dashed arrows |
| Interface |  | An interface (small circle or semi-circle on a sick) describes a group of operation required or provided by components |

##### *Component Diagram*

Figure 30:Component Diagram

#### b) Deployment Diagram

##### Definition

The deployment diagram is a structural diagram that shows the architecture of a system as distribution of software artefacts to deployment targets. It involves modelling the hardware configuration together with the software component that live on them.

##### Formalism

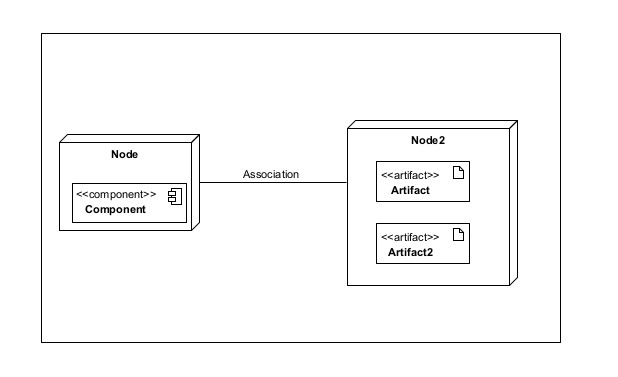


Figure 31: Formalism Deployment diagram

Table 25: Component of a Deployment Diagram

|  |  |  |
| --- | --- | --- |
| Elements | Representation | Description |
| Node |  | A node is either a hardware or a software |
| Artifact |  | An artifact is a product of a software development process or the operation of a system. |
| Component |  | It represents a modular part of a system that encapsulates its content and whose manifestation is replaceable within its environment |
| Association |  | An association represents a communication path between nodes. |

##### B) Deployment Diagram

*Figure 32: System deployment diagram*

# CHOICES OF TECHNOLOGIES

#### MATERIAL RESOURCES

Table 26: Material resources

|  |
| --- |
| HARDWARE RESOURCES |
| * HP Computer, Intel core i3 1TB , RAM 20GO * WIFI HOME BOX |

#### SOFTWARE RESOURCES

Table 27: Software resources

|  |  |  |  |
| --- | --- | --- | --- |
| SOFTWARE NAME | VERSION | USAGE | LOGO |
| OS window 10 | 18.32 | The operating system we worked on is Windows 10 |  |
| |  |  |  | | --- | --- | --- | | SybasePowerAMC |  |  | | 16,5 | The software engineering workshop used for solution modeling is called "Enterprise Architect." We utilized this tool to create various diagrams for our system modeling. |  |
| Visuel studio code | 1.93.1. | The text editor used to enter the lines of code that will be interpreted by the browser is called a "code editor." |  |
| Postman | 10.24 | It is a platform that aims to simplify every step of the API lifecycle and streamline collaboration, in order to create better APIs more easily and quickly |  |

### 3.LANGUAGE USED

* **JavaScript** (JS) was indeed developed by a team at Netscape Navigator, led by Brendan Eich. Created in 1995, JavaScript is a scripting language that is embedded within an HTML document. Historically, it is the first scripting language for the web. Numerous frameworks are based on JavaScript, both for web and mobile development.



Figure 33: JavaScript logo

* **Reactjs**: A JavaScript framework for building the web app,

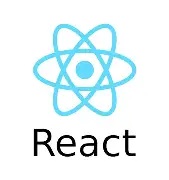


Figure 34 :React logo

* **CSS (Tailwind CSS)**: Used for styling the app's components via the twrnc (Tailwind React Native Class names) library.



Figure 35 :CSS logo

* **Node.js**: JavaScript runtime used on the backend for handling server-side operations.

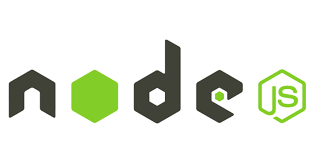


Figure 36 Nodejs logo

* **MongoDB Query Language (MQL)**: For managing and querying data in the MongoDB database, connected to the backend.



Figure 37 MongoDB logo

# BOOK SIX: TEST OF FUNTIONALITIES

**Preamble**

In this phase, we will present the various functionalities of **JOBBY**, our job and career management application. This chapter focuses on demonstrating the core features of the app, explaining how they benefit the user, and how they support the app's intended use.

**INTRODUCTION**

The **test of functionalities** phase allows us to evaluate the performance and usability of our solution, whether web-based or mobile. It highlights the various functionalities or modules present in the application, detailing how each contributes to the user experience. In this chapter, we will explore the essential features of **JOBBY** emphasizing their advantages for the users.

**APPLICATION FUNCTIONALITIES**

1. **Authentication**

This functionality allows job seekers and employers to securely access their dashboards. Users can sign up for an account or log in with existing credentials.

1. **Receive Alerts**

Users receive important notifications about new job postings, application status updates, and interview schedules, ensuring job seekers stay informed about their applications and opportunities.

1. **User Management**

Employers can efficiently manage their job postings and applicants by adding, editing, and storing job details, applicant information, and interview schedules in the database. This enhances organization and improves hiring efficiency.

1. **Job Application Submission**

Job seekers can easily apply for positions directly through the platform. Employers can view, validate, and track applications, streamlining the hiring process**.**

1. **Job Posting Management**

Employers can create, update, and manage job postings, including details such as job title, description, requirements, and application deadlines. This allows for effective job visibility and management.

1. **Profile Management**

Users (job seekers and employers) can edit their profile details, such as name, email, phone number, and LinkedIn profile. This feature enables users to keep their information current and improve networking opportunities.

1. **Notifications**

Job seekers receive notifications regarding their application status, new job matches based on their profile, and reminders for upcoming interviews. This keeps users engaged and informed.

1. **Application Insights**

Employers can analyze applicant data and monitor overall application statuses. This functionality provides insights into hiring patterns, candidate demographics, and application trends to enhance recruitment strategies.

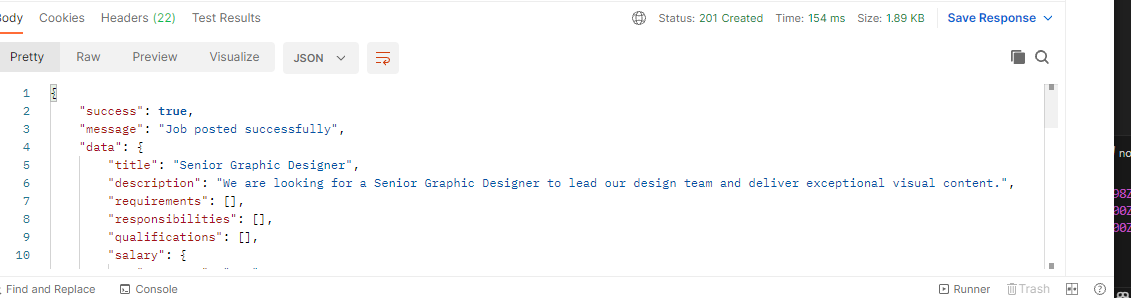
* + - 1. API test for Post Jobs

Endpoint: post / api/employers/jobs

Method: Post

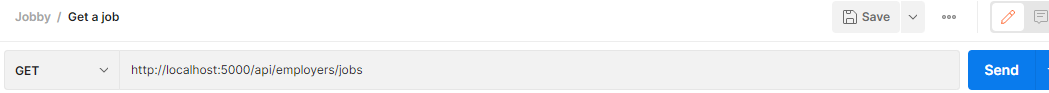


Purpose: This endpoint is used by employer to post job offer



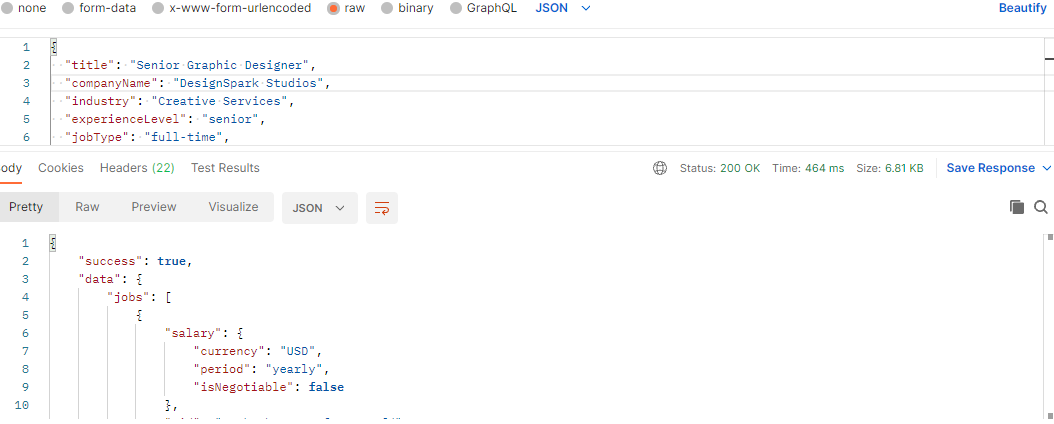
* + - 1. Get Clients API Test

Method: GET



Endpoint: GET /api/jobs

Purpose: Fetch a list of jobs



### CONCLUSION

Having put in place the platform, it was not sufficient for we had to produce a manual that will help its various users. That is why we presented the different tools to be installed and how they are to be installed in order to run this application without any problem and how the users will use this platform once the environment is set up.

### 

# BOOK SEVEN: USER GUIDE

### Preamble

This phase actually aims at show the installation process of the application and how it functions.

### Content

INTRODUCTION

1. INSTALLATION GUIDE
2. USER INTERFACE GUIDE

CONCLUSION

## INTRODUCTION

The user guide is the final phase of our report. In this phase, we will walk through the requirements for using our system, the necessary installation processes and accessing our system and its features, all this in a step-by-step manner to facilitate the setting up of the system for first-time users. The steps of the different processes will be accompanied by visuals. After the processes involving setting up our system, we will go through a showcase of the key functionalities our system offers.

WEB Application

To be able to use JOBBY web application, just connect to the internet and navigate to the website to either create an account or login.

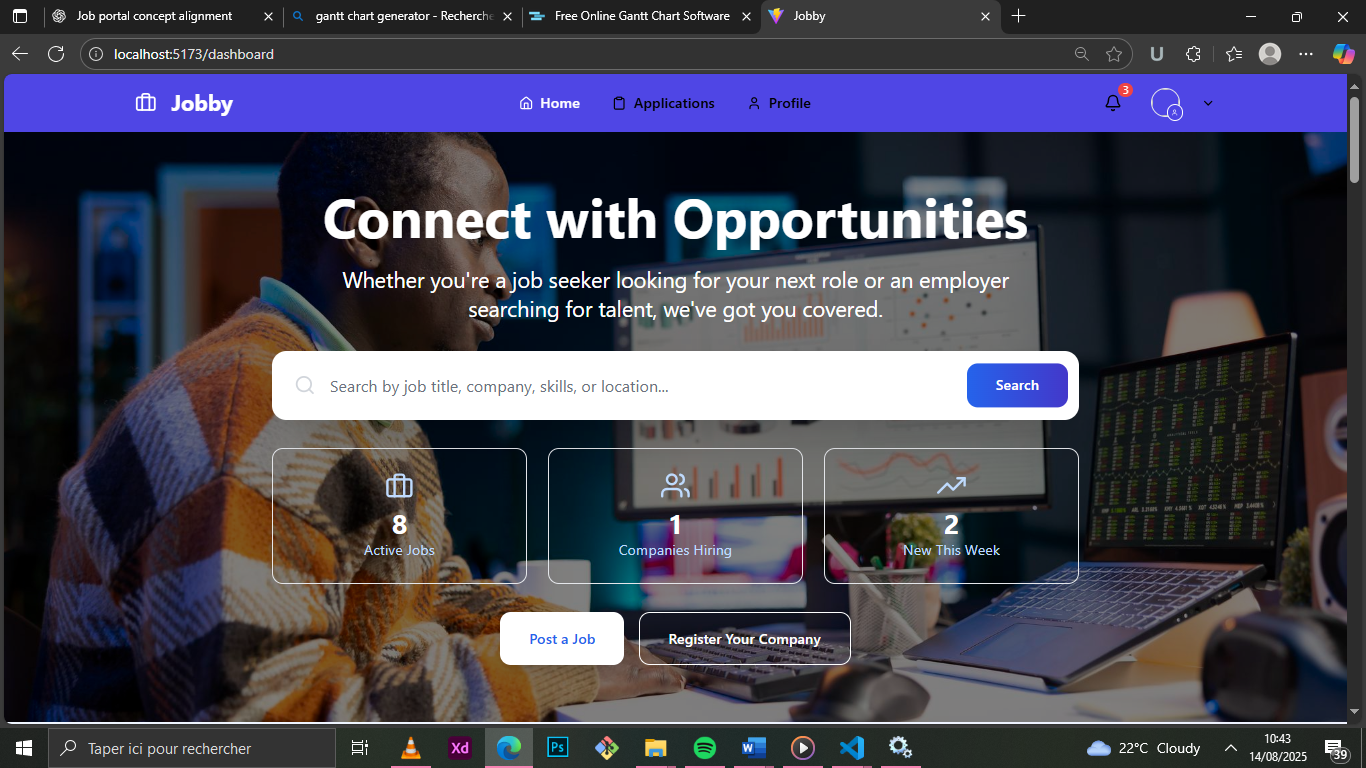
* + - 1. SHOW CASES

Figure 38 JOBBY WELCOME SCREEN

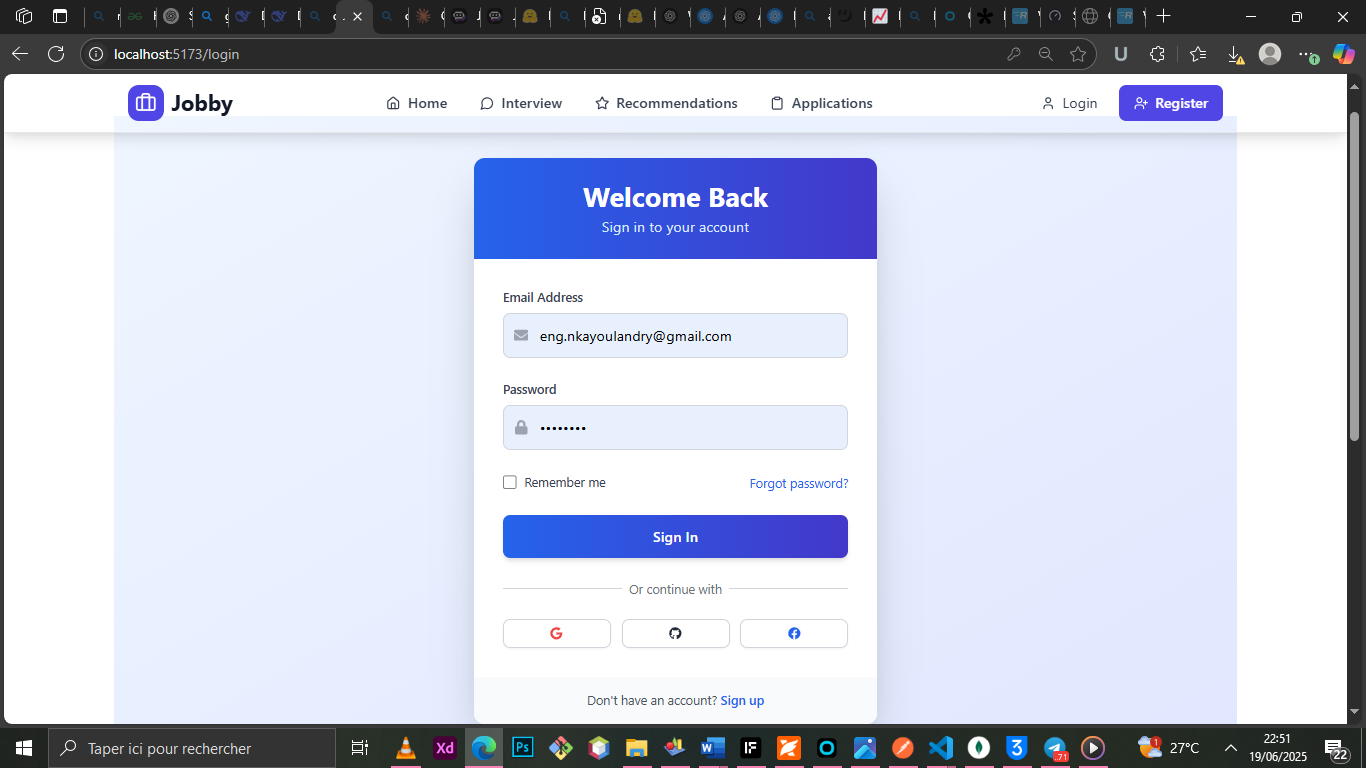
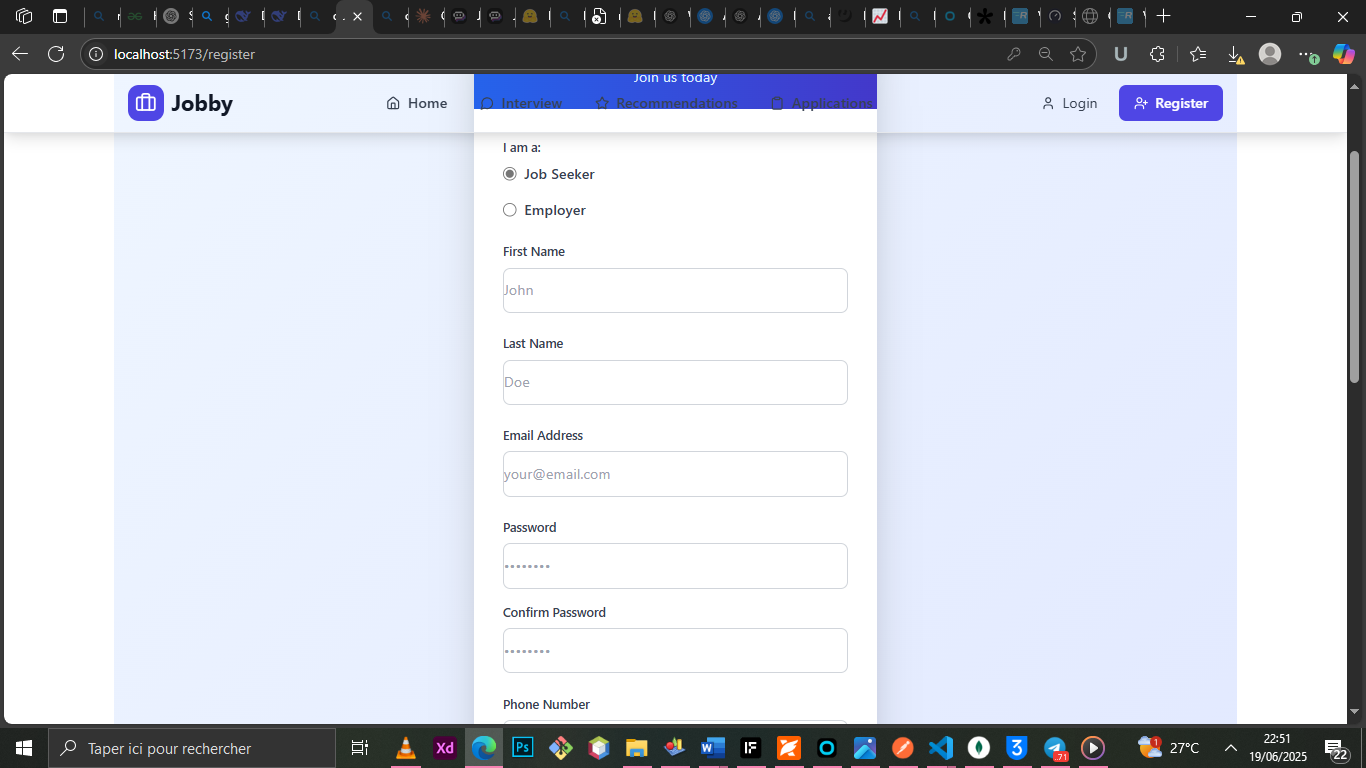


Figure 39: Login Page

Figure 40 registration form screen



**4. Key Functionalities**

**1. JOB POSTING**

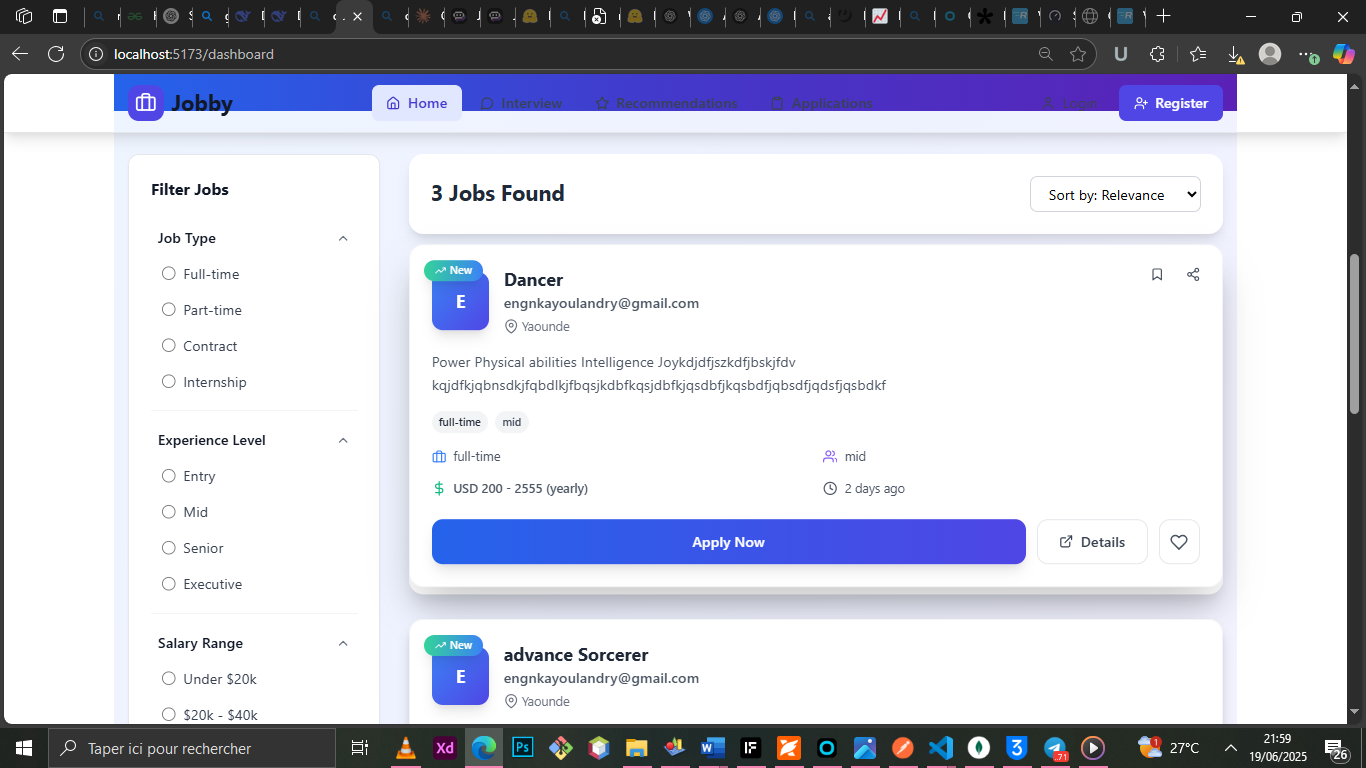
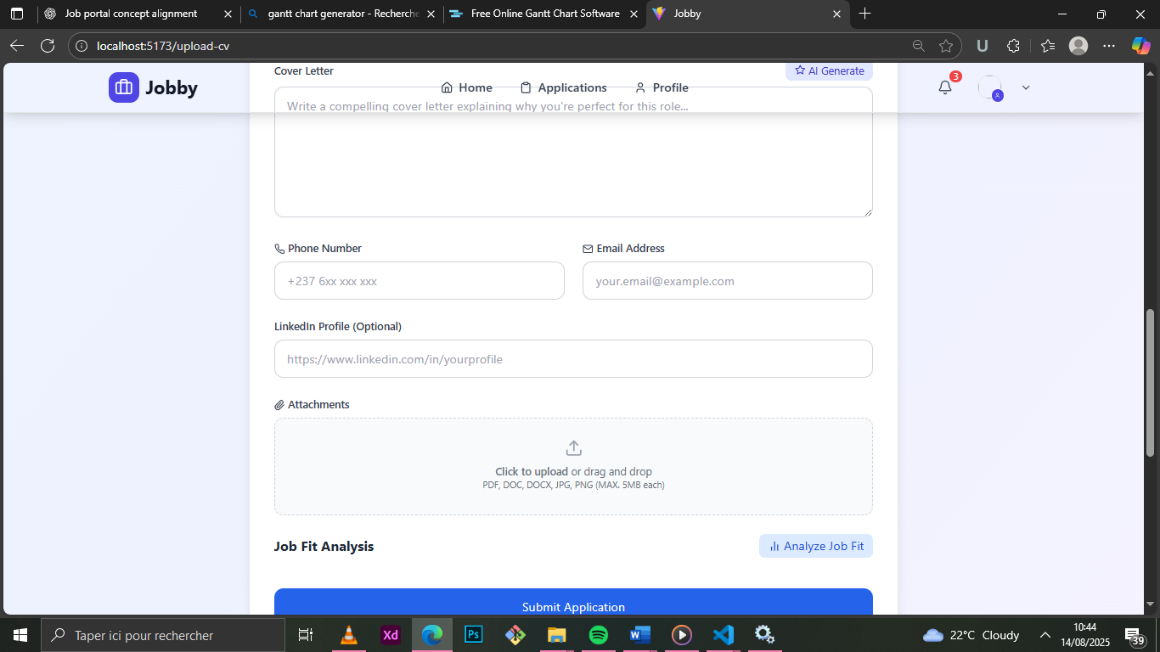


Figure 41: JOB POSTING

Submit application



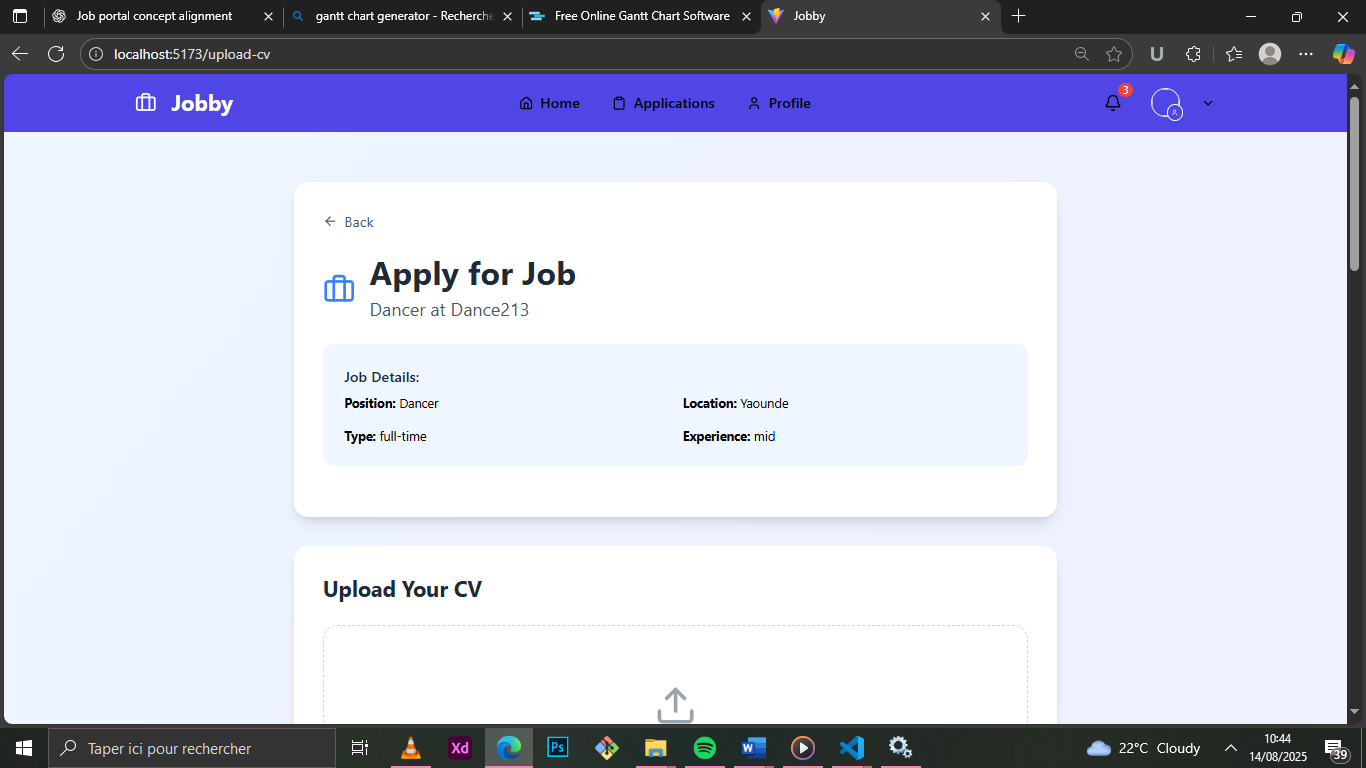


Figure 42:Submit Application

##### CONCLUSION

Having put in place the platform, it was not sufficient for we had to produce a manual that will help its various users. That is why we presented the different tools to be installed and how they are to be installed in order to run this application without any problem and how the users will use this platform once the environment is set up.

## PERSPECTIVES

To enhance your application, focus on improving user experience through better navigation, customizable dashboards, and real-time updates. Expand functionality by integrating advanced filtering, AI-driven insights, and a profile completion meter. Optimize the back-end with API enhancements, database optimization, and robust security measures, while implementing monitoring and logging for performance tracking. Leverage the existing admin section to streamline enterprise verification and application tracking processes. Ensure code quality through refactoring, comprehensive testing, and thorough documentation to maintain scalability and readability. These measures will significantly improve usability, functionality, and maintainability of the application. enhance user experience and improve operational efficiency. One key feature is **AI-driven job matching**, which leverages machine learning algorithms to analyze user profiles and job postings, suggesting the best job matches for candidates based on their skills, experiences, and preferences. This personalization can significantly increase the chances of successful placements, benefiting both job seekers and employers.

# GENERAL CONCLUSION

The job application management system developed in this project aims to streamline the hiring process for employers while enhancing the job-seeking experience for candidates. By integrating functionalities such as user authentication, job posting management, and application tracking, the platform creates a user-friendly environment that supports efficient interactions between job seekers and employers. The inclusion of real-time notifications and application insights ensures that both parties remain informed and engaged throughout the hiring journey, promoting a more effective recruitment process. Moreover, this project emphasizes the importance of data management and user experience, leveraging modern technologies like Node.js and MongoDB to create a robust backend, complemented by a responsive frontend built with React. By focusing on key functionalities such as client management, appointment scheduling, and order management, the system not only addresses the immediate needs of job seekers and employers but also provides valuable analytics for continuous improvement. Ultimately, this project positions itself as a comprehensive solution in the competitive landscape of job recruitment platforms, fostering connections that lead to successful employment outcomes.

# BIBLIOGRAPHY

Author not listed. *Méthode d’Etude et de Réalisation Informatique pour des Systèmes d’Entreprise (MERISE)*.

Alistair Cockburn. *Agile Software Development: The Cooperative Game*. Addison-Wesley, 2nd Edition, 2006.

Schwaber, Ken. *Agile Project Management with Scrum*. Microsoft Press, 2004.

Ivar Jacobson, Grady Booch, James Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, 1999.

Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3rd Edition, 2003.

GLOSSARY

* + 1. **API (Application Programming Interface):** A set of rules and tools for building software and allowing different applications to communicate with each other.
    2. **Backend:** The server-side of an application, where data is stored, processed, and managed. It includes the server, database, and application logic that support the app's functionality.
    3. **CV (Curriculum Vitae)**: A document that summarizes a person's educational background, work experience, skills, and accomplishments, used for job applications.
    4. **Database Management System (DBMS)**: Software that interacts with the database to manage data storage, retrieval, and manipulation.
    5. **Job Application**: The process where candidates submit their CVs and other required information to be considered for a job position.
    6. **Job Posting**: An advertisement created by employers that describes a job opportunity, including its responsibilities and qualifications.
    7. **3-tier Architecture:** A software architecture model where the application is separated into 3 logical layers, usually including the presentation (frontend), application logic (backend), and data storage (database) tiers.
    8. **UI (User Interface):** The visual elements that users interact with on an application, such as buttons, forms, and menus.
    9. **UX (User Experience):** The overall experience and satisfaction a user has when interacting with an application, including how easy and intuitive it is to use.

# WEBOGRAPHY

* "Visual Paradigm UML Guide". Accessed from <https://www.visual-paradigm.com/guide>. **Accessed on:** 12/07/2024 at 10:45am
* "UML 2.5 Diagrams Overview". UML Diagrams Website. Accessed from <http://www.uml-diagrams.org>. **Accessed on:** 28/07/2024 at 9h30
* Realize Cameroon. Company website and resources. Accessed from [www.realize-cm.com](https://www.realize-cm.com). **Accessed on:** 17/08/2024 at 10h00
* "Expo Documentation for React Native". Expo Official Site. Accessed from https://docs.expo.dev. **Accessed on:**18/08/2024 at 9h0
* "Scrum Framework Overview". Scrum.org. Accessed from https://www.scrum.org/resources/what-is-scrum. **Accessed on:** 06/09/2024 at 10h45am
* "MongoDB Documentation". MongoDB. Accessed from <https://docs.mongodb.com>. **Accessed on:** 06/09/2024 at 8h45am
* "Express.js Web Framework". Express.js Official Site. Accessed from <https://expressjs.com>. **Accessed on:** 18/09/2024 at 9h:30
* "Tailwind CSS Documentation". Tailwind CSS Official Site. Accessed from <https://tailwindcss.com/docs>. **Accessed on:** 20/09/2024 at 10h:00

.

.

# ANNEX

# CONTENT

[DEDICATION i](#_Toc206462303)

[ACKNOWLEDGEMENTS ii](#_Toc206462304)

[ABSTRACT iii](#_Toc206462305)

[RESUME iv](#_Toc206462306)

[SUMMARY v](#_Toc206462307)

[LIST OF FIGURES vi](#_Toc206462308)

[LIST OF TABLES vii](#_Toc206462309)

[GENERAL INTRODUCTION 1](#_Toc206462310)

[PART ONE: INSERTION PHASE 3](#_Toc206462311)

[INTRODUCTION 5](#_Toc206462312)

[WELCOME AND INTEGRATRION 6](#_Toc206462313)

[GENERAL PRESENTATION OF THE COMPANY 7](#_Toc206462314)

[CONCLUSION 12](#_Toc206462315)

[PART TWO: TECHNICAL PHASE 13](#_Toc206462316)

[BOOK ONE: EXISTING 14](#_Toc206462317)

[**INTRODUCTION** 16](#_Toc206462318)

[I. DESCRIPTION OF THE EXISTING SYSTEM 18](#_Toc206462319)

[II. LIMITS OF THE EXISTING SYSTEM 20](#_Toc206462320)

[III. PROBLEMATIC 21](#_Toc206462321)

[IV. PROPOSED SOLUTION 22](#_Toc206462322)

[V. DELIMITATIONS OF THE FIELD OF STUDY 23](#_Toc206462323)

[BOOK TWO: SPECIFICATION PHASE 25](#_Toc206462324)

[INTRODUCTION 27](#_Toc206462325)

[V. ESTIMATED COST OF PROJECT 35](#_Toc206462326)

[VI. ESTIMATION OF TIME REQUIRED 38](#_Toc206462327)

[VII. CONSTRAINTS 40](#_Toc206462328)

[VIII. LIST OF PARTICIPANTS AND DELIVERABLES 40](#_Toc206462329)

[CONCLUSION 42](#_Toc206462330)

[BOOK THREE: 43](#_Toc206462331)

[ANALYSIS PHASE 43](#_Toc206462332)

[Preamble 44](#_Toc206462333)

[INTRODUCTION 45](#_Toc206462334)

[VI. PRESENTATION OF THE ANALYSIS METHOD 46](#_Toc206462335)

[VII. CHOICE OF THE ANALYSIS APPROACH 48](#_Toc206462336)

[VIII. MODELING OF THE PROPOSED SOLUTION 53](#_Toc206462337)

[Use case: Manage application 57](#_Toc206462338)

[Use case: Post Job Openings 58](#_Toc206462339)

[Use case: Apply for Jobs 59](#_Toc206462340)

[TEXTUAL DESCRIPTION OF Post Job Openings 64](#_Toc206462341)

[Post Job communication diagram 69](#_Toc206462342)

[CONCLUSION 80](#_Toc206462343)

[BOOK FOUR: CONCEPTION PHASE 81](#_Toc206462344)

[INTRODUCTION 83](#_Toc206462345)

[I. TECHNICAL BRANCH 84](#_Toc206462346)

[***Hardware Diagram of the System*** 84](#_Toc206462347)

[***Physical Architecture of the System*** 84](#_Toc206462348)

[***Logical Architecture of the System*** 85](#_Toc206462349)

[RELATED UML DIAGRAMS 87](#_Toc206462350)

[Package Diagram 97](#_Toc206462351)

[System object diagram 100](#_Toc206462352)

[CONCLUSION 101](#_Toc206462353)

[BOOK FIVE: REALIZATION PHASE 102](#_Toc206462354)

[Preamble 103](#_Toc206462355)

[Content 103](#_Toc206462356)

[a) Component Diagram 104](#_Toc206462357)

[c) CHOICES OF TECHNOLOGIES 110](#_Toc206462358)

[3.LANGUAGE USED 111](#_Toc206462359)

[BOOK SIX: TEST OF FUNTIONALITIES 113](#_Toc206462360)

[CONCLUSION 116](#_Toc206462361) [117](#_Toc206462362)

[BOOK SEVEN: USER GUIDE 117](#_Toc206462363)

[Preamble 118](#_Toc206462364)

[Content 118](#_Toc206462365)

[INTRODUCTION 119](#_Toc206462366)

[PERSPECTIVES 126](#_Toc206462367)

[GENERAL CONCLUSION 127](#_Toc206462368)

[BIBLIOGRAPHY ix](#_Toc206462369)

[WEBOGRAPHY xi](#_Toc206462370)

[ANNEX xii](#_Toc206462371)

[CONTENT xiii](#_Toc206462372)