République Tunisienne Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Technologie

Université de Kairouan Institut Supérieur d'Informatique et de Gestion de Kairouan

Institut Supérieur d'Informatique et de Gestion de Kairouan
I.S.I.G.K

Mémoire de ********
Élaboré par: ********

En vue de l'obtention du diplôme National********

Spécialité : *********

****

Encadrante :********

Soutenue le /2021, devant les membres du jury

Président du jury :
Rapporteur :

Année universitaire 2023 / 2024

Contents

1 Preliminary study **2**

2 Sprint 0 Planning and Architecture **12**

3 Sprint 1 **28**

3.1 Introduction 29

3.2 Sprint Backlog 29

3.3 Analysis 30

3.3.1 General use case for Sprint 1 30

3.3.1.1 Refinement of general use case «register» 31

3.3.1.2 Refinement of use case «login» 32

3.3.1.3 Refinement of use case «Forget Password» 33

3.3.1.4 Refinement of use case «Consult Profile» 35

3.3.1.5 Refinement of use case «Search User» 36

3.3.1.6 Refinement of use case «Add Job Post» 38

3.3.1.7 Refinement of use case «Consult Job Posts» 39

3.3.1.8 Refinement of use case «Apply Job Post» 40

3.3.1.9 Refinement of use case «Consult Job Application» 41

3.4 Design 42
3.4.1 Sequence diagram 42
3.4.1.1 Sequence diagram «login» 42
3.4.1.2 Sequence diagram «delete user» 43
3.4.1.3 Sequence diagram «consult user» 43
3.4.1.4 Sequence diagram «search user» 44
3.4.1.5 Sequence diagram «Add job Post» 44
3.4.1.6 Sequence diagram «delete job Post» 45
3.4.1.7 Sequence diagram «edit job post» 45
3.4.2 Diagram du class 45
3.5 Implementation 46
3.5.1 Register Interface 46
3.5.2 Login Interface 47
3.5.3 Reset Password Interface 48
3.6 Testing 49

4 ******** 55**

Conclusion générale et perspectives 56

List of Figures

1.1 Waterfall Methodology 6
1.2 V-Shaped Methodology 7
1.3 Agile Methodology 9
1.4 Sprint cycle 10

2.1 Theoretical sprint estimation 20
2.2 MVC architecture 25
2.3 three-layer architecture 26

3.1 Caption 31
3.2 Caption 31
3.3 Caption 32
3.4 Caption 33
3.5 Caption 35
3.6 Caption 37
3.7 Caption 38
3.8 Caption 39
3.9 Caption 40
3.10 Caption 41

3.11 Caption 42
3.12 Caption 43
3.13 Caption 43
3.14 Caption 44
3.15 Caption 44
3.16 Caption 45
3.17 Caption 45
3.18 Caption 46
3.19 Interface «Register» 47
3.20 Wrong Input Interface 47
3.21 Interface «Login» 48
3.22 Wrong Input Interface 48
3.23 Inerface «Reset Password» 49
3.24 Testing the Get Profile Data API 49
3.25 Testing the Update Profile API 50
3.26 Testing the job Post API 50
3.27 Testing the Retrieve Job Post Information 51
3.28 Testing the delete of job Post API 51
3.29 Testing the update post information API 52
3.30 Testing the application on Job Post API 52
3.31 Testing the abandoned on job application 53
3.32 Testing the Retrieve of all job application History 53

List of Tables

2.1 Identification of actors 13

2.2 Project Backlog 20

3.1 Project Backlog 30

3.2 Sign up Use Case 32

3.3 Sign up Use Case 33

3.4 Sign up Use Case 34

3.5 Sign up Use Case 36

3.6 Sign up Use Case 37

3.7 Sign up Use Case 38

3.8 Sign up Use Case 40

3.9 Sign up Use Case 41

3.10 Sign up Use Case 42

Résumé

Abstract

Acronymes

Introduction générale

Chapter 1

Preliminary study

I. Introduction

A preliminary study is an important first step in the planning and implementation process, as it helps to identify potential risks and constraints, and provides valuable insights that can inform decision-making and the development of a more comprehensive plan.

This chapter aims to evaluate and propose a solution to a critical component of the hiring process based on resume analysis and test evaluation. The study will present the host organization and its relevant background information, critique the existing approach to the problem, present the proposed solution and discuss the methodology chosen for the study.

II. Presentation of the host organization (ETC)

ETC TUNISIE is a well-established custom web development agency with over 20 years of experience. Its team of skilled developers provides customized technology solutions that align with each client's objectives like a fully personalized approach, delivering efficient, cost-effective, and high-quality solutions that meet the specific requirements of each business. The agency has served companies across various industries such as industry, commerce, and consulting organizations, with a proven record of delivering high-quality solutions.

III. Analysis of the existing

1. Problem statement

Existing job search platforms such as Tanitjobs and Optioncarriere primarily focus on the needs of job seekers and provide them with a platform to search and apply for job opportunities. However, these platforms do not cater to the needs of companies looking to hire the right candidate for a specific job position.

Companies often struggle to find suitable candidates for their job openings and may end up with a large number of irrelevant applications. This not only wastes the company's time and resources but also results in a poor candidate experience, which can harm the company's reputation.

Moreover, companies may have specific requirements for their job openings and may want to reach out to potential candidates who meet those requirements directly, rather than wait for them to apply.

Therefore, there is a need for a platform that focuses on the needs of companies and allows them to search for and directly contact potential candidates who meet their requirements. This platform should provide companies with a targeted approach to hiring, and enable them to send job offers to specific candidates, rather than waiting for applications to come in.

2. Proposed Solution

To address the challenges faced by job seekers and employers in the job search process, we propose the development of an online job search platform that leverages AI technology to parse resumes, monitor who viewed or applied to job offers, search engine matching skills, and allow candidates to see who visited their profile. Additionally, the platform will enable job seekers to apply for job offers with quizzes or tests to showcase their skills and knowledge.

The platform will use natural language processing algorithms to parse resumes and extract relevant information, such as work experience, education, and skills. The system will also monitor who viewed or applied to job offers, enabling employers to track the effectiveness of their job postings and optimize their recruitment efforts accordingly.

The search engine matching skills will use machine learning algorithms to match job seekers with relevant job opportunities based on their skills, experience, and preferences. This will allow job seekers to receive personalized job recommendations that match their qualifications, increasing their chances of finding suitable job opportunities.

Additionally, the platform will enable job seekers to see who visited their profile, allowing them to gain insights into which employers are interested in their qualifications and skills. This will provide job seekers with valuable information that can help them tailor their job search efforts and improve their chances of success.

Moreover, the platform will allow job seekers to apply for job offers with quizzes or tests, enabling them to showcase their skills and knowledge related to the job position. The results of the quizzes or tests will be used to provide employers with a more comprehensive view of a candidate's qualifications and suitability for the role, improving the efficiency and effectiveness of the hiring process.

In summary, the proposed job search platform will leverage AI technology to provide a

comprehensive solution that parses resumes, monitors who viewed or applied to job offers, use a search engine to match skills, allows candidates to see who visited their profile, and enables job seekers to apply to job offers with quizzes or tests. This will enhance the efficiency and effectiveness of the job search process for both job seekers and employers, leading to better job matches and more successful hiring outcomes.

IV. Choice of methodology:

1. Comparison of the existing methods

Waterfall: The Waterfall Model is a sequential, linear approach to software development that follows a series of well-defined steps. Each step in the Waterfall Model is completed before moving on to the next. [1]

Advantages:

- Clear Process: The Waterfall Model provides a clear and well-defined process for software development, making it easy to understand and follow.
- Predictable Outcomes: The model provides a predictable and structured approach to software development, making it easy to plan and budget for projects.
- Emphasis on Planning: The Waterfall Model places a strong emphasis on planning and requirements gathering, helping to ensure that the final product meets the customer's needs.

Disadvantages:

- Inflexible: The Waterfall Model is a sequential and rigid approach to software development, which can make it difficult to adapt to changing requirements or technologies.
- Lack of Feedback: The model does not provide regular feedback or review from the customer, which can make it difficult to ensure that the final product meets their needs.
- Late Discovery of Defects: Testing and verification are typically done at the end of the development process after the implementation has been completed. This can lead to the discovery of defects and issues late in the project, which can be time-consuming and expensive to resolve.

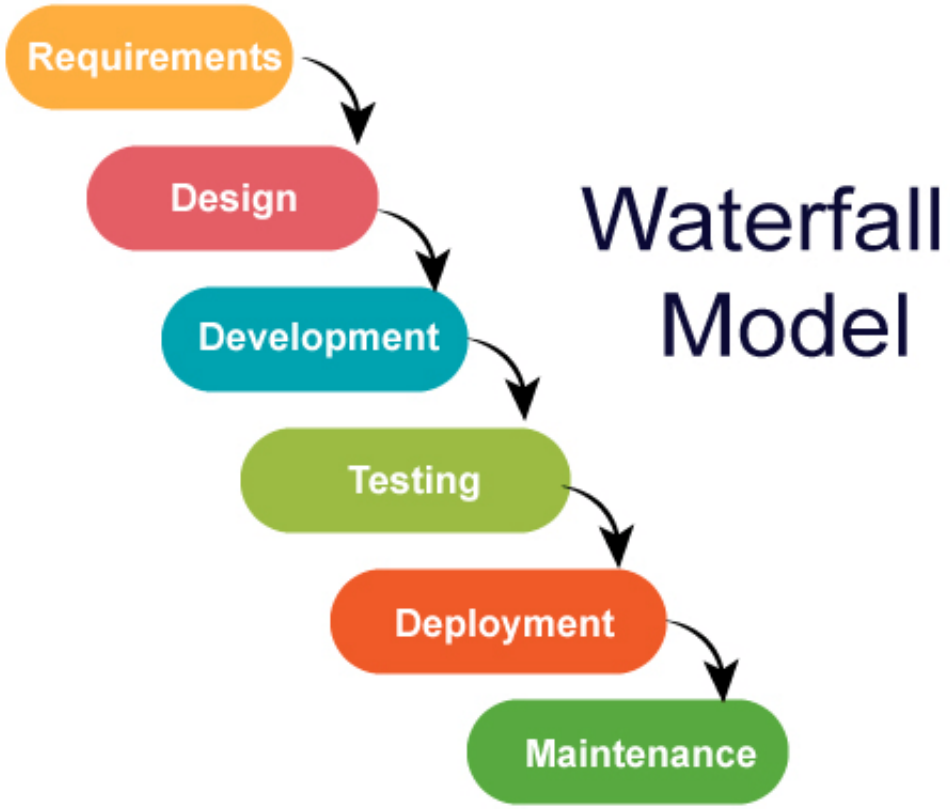


Figure 1.1: Waterfall Methodology
[2]

V-Shaped: The V-Shaped Model is a linear, sequential approach to software development that follows a step-by-step process with two main phases: development and testing. It emphasizes testing and verification at every stage to ensure a high-quality final product that meets customer requirements. [3]

Advantages:

- Emphasis on Testing: The model emphasizes the importance of testing and verification at every stage of the development process, helping to ensure that the final product meets the customer's requirements and is of high quality.
- Predictable Outcomes: The V-Shaped Model provides a predictable and structured approach to software development, making it easy to plan and budget for projects.

- Clear Process: The V-Shaped Model provides a clear and detailed process for software development, making it easy to understand and follow.

Disadvantages:

- Slow Process: The model's step-by-step approach can lead to a slow and time-consuming process, particularly for projects with complex requirements or large teams.
- Limited Creativity: The V-Shaped Model's focus on a structured and predictable process can limit creativity and innovation in the development process.
- Slow Process: The model's step-by-step approach can lead to a slow and time-consuming process, particularly for projects with complex requirements or large teams.

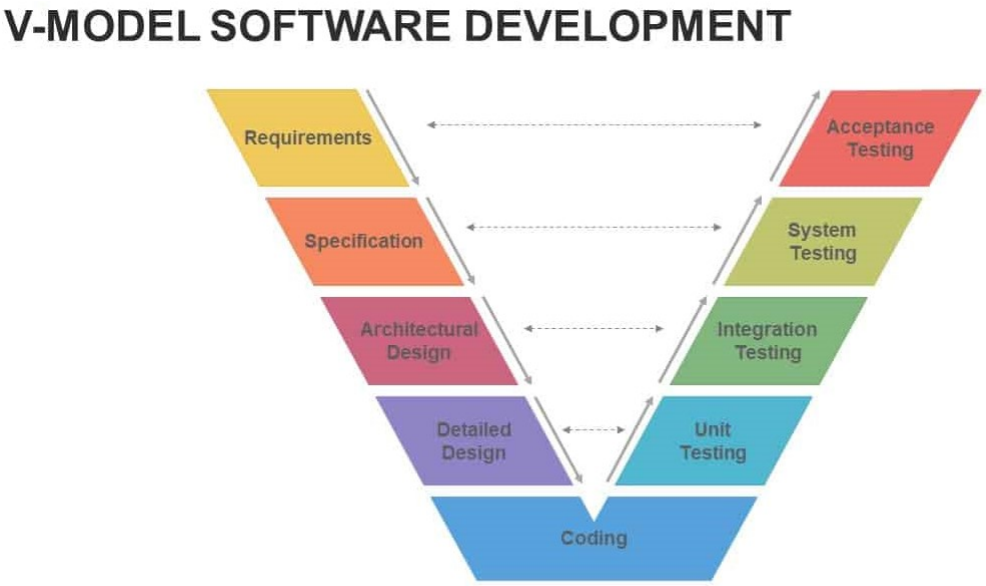


Figure 1.2: V-Shaped Methodology
[4]

Agile: Agile is a project management methodology that emphasizes flexibility, collaboration, and continuous improvement. It is based on the Agile Manifesto, a set of principles for software development that values individuals and interactions, working software, customer collaboration, and responding to change. [5]

Advantages:

- Flexibility: Agile values the ability to respond to change and encourages teams to adapt to new information and changing requirements. This makes it an ideal approach for projects that are subject to frequent changes or have uncertain requirements.
- Continuous Improvement: Agile encourages the team to continuously reflect on their processes and identify ways to improve, which leads to a culture of continuous improvement and a better product in the long run.
- Faster Time-to-Market: By working in short iterations and delivering usable portions of the product early and often, Agile allows organizations to get their products to market more quickly.
- Better Quality: By incorporating frequent feedback from stakeholders and emphasizing testing and quality assurance, Agile helps to ensure that the final product meets the needs of the customers and is of high quality.
- Increased Customer Satisfaction: Agile encourages regular engagement with customers and stakeholders, which helps to build a relationship of trust and improve customer satisfaction.

Disadvantages:

- Emphasis on documentation: Agile development methodology tends to prioritize working software over comprehensive documentation. While this can be an advantage in some cases, it can also lead to incomplete documentation, which can create difficulties in maintenance and future development.
- Requires Strong Leadership: Agile requires strong leadership and a supportive organizational culture to be successful. Without a clear understanding of Agile principles and a commitment to continuous improvement, organizations may struggle to adopt this methodology effectively.

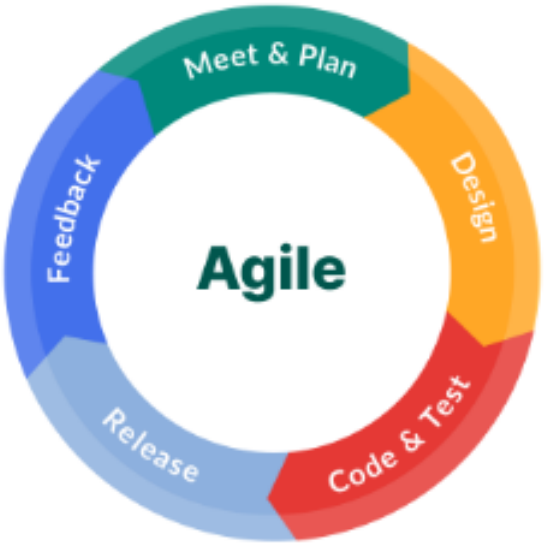


Figure 1.3: Agile Methodology
[6]

2. Why scrum

I have chosen Scrum as the methodology for my project due to its strong focus on communication and adaptability. Scrum's foundation is based on Agile methodology, as a result, it includes transparency, inspection, and adaptation and makes it ideal for delivering high-quality software in a dynamic environment. [7]

- flexibility: My project comprises numerous features, which allow me to both add new ones and make modifications to existing ones.
- high-quality software: Scrum based on regular reviews and tests throughout the software development process to identify and resolve issues early on, ultimately resulting in higher quality software.

Scrum Framework Events:

1. Product Backlog Refinement: During this stage, the product backlog is reviewed and refined to ensure that it is up-to-date and that the team has a clear understanding of the work that needs to be done.

2. Sprint Planning: In this stage, the team selects a set of items from the product backlog to be completed during the next sprint, and creates a plan for how to complete them.

3. Daily Scrum: This is a brief, daily meeting in which the team members synchronize their work, discuss progress, and identify any obstacles that need to be addressed.

4. Sprint Review: At the end of each sprint, the team demonstrates the completed work to stakeholders and receives feedback.

5. Sprint Retrospective: In this stage, the team reflects on the completed sprint and identifies areas for improvement. The team then creates a plan for implementing these improvements in the next sprint.

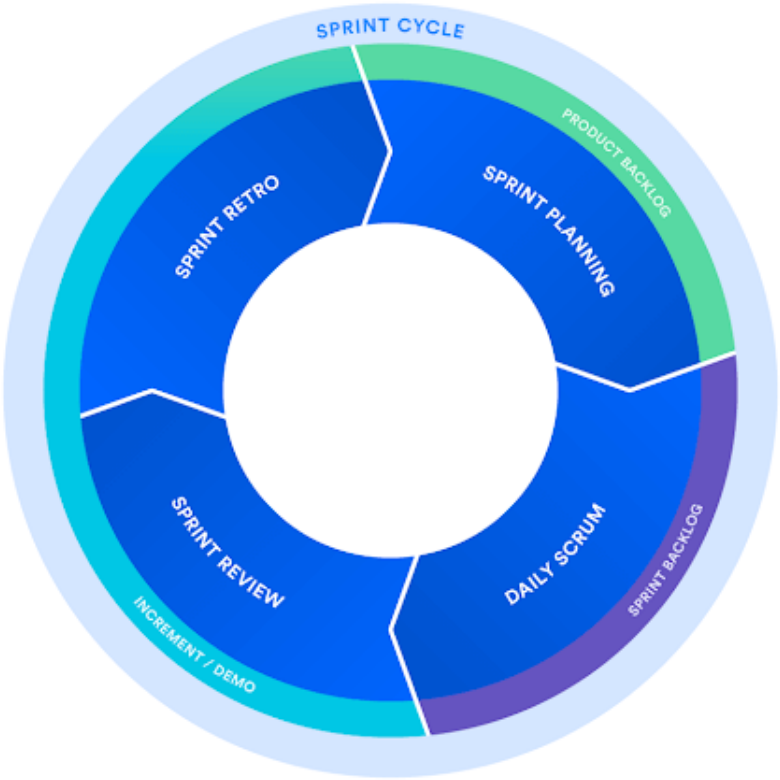


Figure 1.4: Sprint cycle
[8]

3. Modeling language (UML)

We have chosen UML as the modeling language for my project for many reasons: first UML is a powerful tool for software development projects that provides numerous advantages for software designers, and developers. One of the main benefits of UML is its ability to effectively communicate complex systems, making it easier for everyone involved to understand and identify potential design problems. Moreover, UML provides a structured way to identify potential design problems and provides a systematic approach to resolving those issues. This will be beneficial for my project as it helps ensure that my software is properly designed and optimized for performance. Additionally, UML is widely accepted as a standard for software modeling and has been widely adopted by the software development industry. By using UML, I will be aligning with industry best practices and standards, making it easier for me to integrate with other systems and technologies. In conclusion, the use of UML is a wise choice as it provides a structured approach to software modeling, helps ensure proper design, and aligns with industry standards. [8]

V. Conclusion

In conclusion, the preliminary study of the project has provided a comprehensive overview of the hosting organization (ETC) and analyzed the existing situation. The choice of methodology has been a crucial aspect of this study as it lays the foundation for the successful completion of the project. The selected methodology will ensure that the project is completed efficiently and effectively, delivering high-quality results that align with the objectives of the hosting organization. This preliminary study serves as a valuable reference for the next steps in the project and sets the stage for its successful implementation.

Chapter 2

Sprint 0 Planning and Architecture

I. Introduction

Chapter 2 is devoted to the key initial phase of software development, often referred to as Sprint 0 planning and architecture. The focus of this stage is the planning and design of the project architecture and the determination of requirements. This chapter provides an in-depth discussion of key aspects of Sprint 0, such as functional and non-functional requirements, project planning techniques, and product backlog organization. By emphasizing these key factors during Sprint 0, the project can be successful and meet customer needs.

II. Capture and Analysis of Requirements

1. Identification of actors

Actors are the people, organizations, or external systems that interact with the software system being developed. Identifying the actors is important because it helps to understand the system's requirements, define the use cases, and determine how the actors interact with the system.

Actors	
Candidate	Individuals who are looking for employment opportunities create a profile on the job platform to search and apply for jobs.
Recruiter	Companies or organizations that post job openings on the platform and review and select candidates for their positions.
Administrator	The third-party service that connects job seekers with employers and facilitates the application process. The job platform is responsible for providing a user-friendly interface for both job seekers and employers, as well as ensuring the security and confidentiality of the information shared on the platform.

Table 2.1: Identification of actors

2. Identification of functional requirements

Functional requirements are the specific features and capabilities that a software system must possess to meet the expectation of the Client. Identifying and documenting these

requirements is critical to ensure that the software system delivers the desired functionality and meets its objectives.

So the various features we want to implement as part of this project Can be divided into the following points:

- The platform should allow candidates and recruiters to manage their accounts and profiles easily.
- The platform should provide job search and filtering functionality for candidates to find relevant job openings.
- The platform should enable recruiters to search for qualified candidates based on various criteria such as job title, location, and experience.
- The platform should allow candidates to submit their applications to job openings easily and seamlessly.
- The platform should provide recruiters with tools to manage job postings, and candidate applications.
- The platform should offer a comprehensive dashboard that displays statistics and analytics related to job postings, candidate applications, and other recruitment metrics.
- The platform should offer recruiters the capability to assess the skills and knowledge of candidates through the inclusion of pre-defined quizzes or assessments within the job application process.
- The platform should provide a feature that allows recruiters to send feedback to candidates on why their application was not accepted for a position.

2. Identification of non-functional requirements

Non-functional requirements play a crucial role in enhancing the overall quality of a software system. They serve as important constraints to be considered while implementing a solution that meets the expected standards and avoids any potential inconsistencies in the system.

Maintainability: The system should be easy to maintain and update, with clear documentation and coding standards.

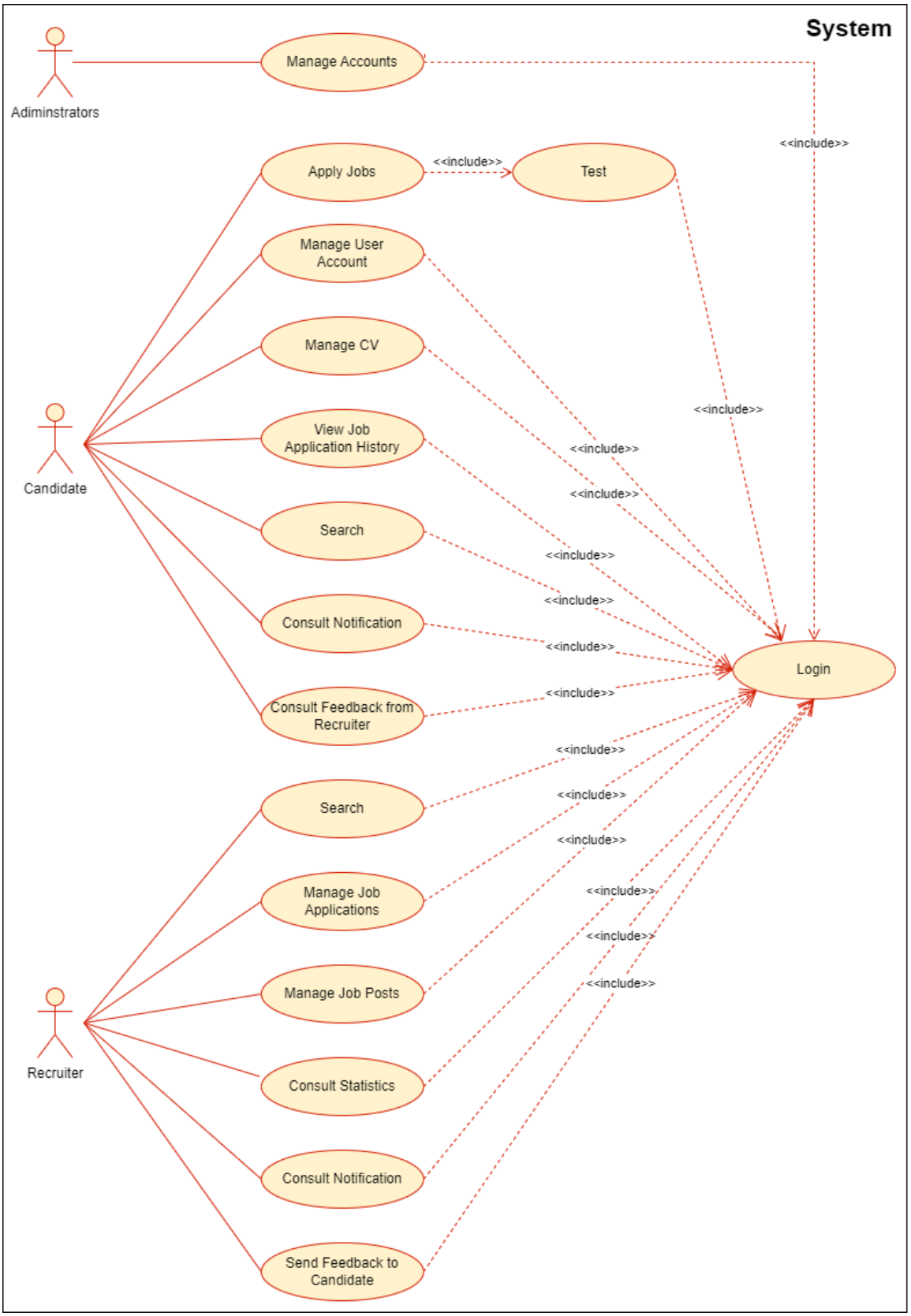
Security: The system should be secure and protected against unauthorized access, data breaches, and other security risks.

Scalability: The system should be able to scale up or down based on user demand or data volume.

3. Modeling of requirements

Global Use Case Diagram

The diagram displayed in figure 2.2 represents the interaction between system users and the system itself. It helps to illustrate the different use cases or scenarios in which a user interacts with the system, along with the actors or external entities involved in those interactions.


System
Adiminstrators
Manage Accounts
Apply Jobs
<<include>>
Test
Manage User Account
Manage CV
View Job Application History
Search
Consult Notification
Consult Feedback from Recruiter
Candidate
Login
Search
Manage Job Applications
Manage Job Posts
Consult Statistics
Consult Notification
Send Feedback to Candidate
Recruiter

4. Product Backlog

In Scrum, a product backlog is a crucial tool that helps prioritize the user stories or requirements that define the desired functionality of the software product. It is continuously updated and refined based on feedback and changing market needs and used as a powerful tool for planning and managing the work to be done during each sprint. This ensures that the highest priority items are addressed first and facilitates the delivery of a successful software product.

- **M**: « Must have », All the functionalities that are essential and necessary.
- **S**: « Should have », The tasks that are essential but not mandatory will be developed once the ones in the Must category have been delivered.
- **C**: « Could have », These are tasks that will only be done if there is enough time remaining.
- **W**: « Won't have this time but would like in the future », These are additional tasks that are considered to be of low priority.

ID	Theme	ID	User story	Priority	Estimation
1	login	1.1	As a User, I would like to authenticate myself to access the platform.	M	8
2	Apply for Jobs	2.1	As a candidate, I want to be able to apply to job postings directly through the platform.	S	3
		2.2	As a candidate, I want to be able to include my resume and cover letter with the job submission.	S	2
3	Manage User Account	3.1	As a candidate, I want to be able to change my profile photo	M	2
		3.2	As a candidate, I want to be able to change my cover letter.	S	2
		3.3	As a candidate, I want to be able to change my work designation.	M	5

		3.4	As a candidate, I want to be able to upload my resume for potential employers to review.	M	5
		3.5	As a candidate, I want to be able to save job postings that interest me for later review.	M	3
4	View Job Application History	4.1	As a candidate, I want to be able to view my job application history.	S	5
5	Search	5.1	As a candidate, I want to be able to search and filter job postings based on keywords.	C	3
		5.2	As a candidate, I want to be able to search and filter job postings based on location.	M	5
		5.3	As a candidate, I want to be able to search and filter job postings based on job type.	M	5
		5.4	As a Recruiter, I want to be able to search and filter candidates based on keywords.	M	5
		5.5	As a Recruiter, I want to be able to search and filter candidates based on job type	M	5
6	Consult Notification	6.1	As a candidate, I want to be able to receive notifications when new job postings have been added that match my search criteria.	C	3
		6.2	As a candidate, I want to be able to receive notifications when someone views my profile.	M	5
7	Consult Feedback from Recruiter	7.1	As a candidate, I want to be able to consult feedback from the recruiter on why I got rejected from the job post.	C	3

8	Manage Job Application	8.1	As a Recruiter, I want to be able to manage and review job applications submitted through the platform.	C	3
		8.2	As a Recruiter, I want to be able to sort job applications submitted by candidate qualifications.	M	5
		8.3	As a Recruiter, I want to be able to filter job applications submitted by candidate qualifications.	M	5
9	Manage Job Posts	9.1	As a Recruiter, I want to be able to add job posts.	C	3
		9.2	As a Recruiter, I want to be able to delete the job post.	M	5
		9.3	As a Recruiter, I want to be able to update job posts.	M	5
		9.4	As a Recruiter, I want to be able to include the job description with the job posts.	M	5
		9.5	As a Recruiter, I want to be able to add the required qualifications to the job posts.	M	5
		9.6	As a Recruiter, I want to be able to include quizzes with the job application submitted.	M	5
10	Consult Statistics	10.1	As a Recruiter, I want to be able to consult statistics and analytics related to job postings, candidate applications, and other recruitment metrics.	C	3
11	Send Feedback to Candidate	11.1	As a Recruiter, I want to have the option to send feedback to candidates through the platform.	C	3

12	Manage Accounts	12.1	AS an administrator, I want to be able to add a user account.	C	3
		12.2	As a Recruiter, AS an administrator, I want to be able to delete a user account.	M	5
		12.3	AS an administrator, I want to be able to modify a user account.	M	5
		12.4	AS an administrator, I want to be able to filter the list of user accounts.	M	5
		12.5	AS an administrator, I want to be able to search for a user account.	M	5

Table 2.2: Project Backlog

5. Sprint Planning:

The Decomposition of the sprint is an essential process in agile project management that involves breaking down a larger project or sprint goal into smaller, more manageable tasks. This process is highly beneficial as it helps to enhance our understanding of the work that needs to be accomplished and enables us to plan and execute tasks more efficiently. Ultimately, this ensures that the sprint goal is achieved within the expected scope and timeframe.

Figure 2.1: Theoretical sprint estimation

- **sprint 1:**
  Set up the project with the necessary tools and frameworks.
  Create a basic user interface for job seekers to search and apply for jobs.
  Implement a basic database schema to store job listings and user profiles.

Implement user authentication and authorization to secure user data.

- **sprint 2:**
  Implement a job listing management interface for recruiters to post job openings.
  Allow recruiters to manage their job listings, including editing and deleting.
  Implement a search and filtering system to help candidates find relevant job listings.
  Implement a search and filtering system to help recruiters find relevant candidates.
  Allow candidates to save job listings for later reference.

- **sprint 3:**
  Implement an application management system for recruiters to manage job applications.
  Allow Recruiters to view, sort, and filter job applications.
  Send automated email notifications to candidates when a relevant job post is available.

- **sprint 4:**
  Implement a recommendation system to suggest relevant job listings to candidates.
  Allow candidates to upload resumes and cover letters.
  Implement an analytics dashboard for recruiters to track job postings, candidate applications, and other recruitment metrics.

III. Work environment

1. Hardware environment

We used a computer with the following specifications to carry out this project.

- **Processor:**AMD Ryzen™ 5 4600H
- **Memory:**16 GB DDR4-3200 MHz
- **Storage:**512GB PCIe® 3.0 NVMe™ M.2 SSD
- **Graphics:**NVIDIA® GeForce® GTX 1660Ti, 6GB GDDR6
- **Operating System:**Windows 11, 64-bit version

2. Software environment

2.1 Development environment

VS Code:

VS Code is a free and open-source code editor developed by Microsoft. It offers a broad range of programming languages and features. It's compatible with Windows, macOS, and Linux operating systems and is highly favored by developers. [9]

Draw.io:

Draw.io is a web-based diagramming tool that enables users to create various visual representations. It offers a drag-and-drop interface and a variety of templates and shapes. [10]

PostMan:

Postman is a collaboration platform for API development that allows developers to design, test, and document APIs. It provides a user-friendly interface for sending HTTP requests, viewing responses, and organizing API endpoints. [11]

Overleaf:

Overleaf is an online LaTeX editor that allows users to create and collaborate on LaTeX documents in real time. It offers a rich text editor, various templates, and an intuitive interface for creating and organizing LaTeX documents. Overleaf is widely used in academia and research for writing scientific papers, theses, and other technical documents. [12]

2.2 Technology

Angular:

Angular is a front-end web application framework used for building dynamic and interactive web applications. It is based on TypeScript and offers features such as data binding, dependency injection, and component-based architecture. Angular provides robust tools and libraries for building complex and scalable applications, making it a popular choice for web developers. [13]

Node.js:

Node.js is an open-source, cross-platform JavaScript runtime environment that enables developers to run JavaScript code outside of a web browser. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient for building scalable and high-performance applications. [14]

PostgreSQL:

PostgreSQL is a powerful open-source object-relational database management system that offers advanced features such as transactional integrity, concurrency control, and support for complex queries. It is highly scalable and flexible, making it suitable for a wide range of applications from small applications to large, enterprise-level systems. [15]

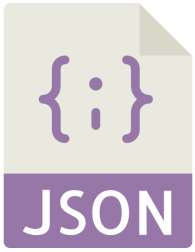
HTML:

HTML (Hypertext Markup Language) is the standard markup language used for creating web pages and other types of electronic documents that are displayed on the World Wide Web. [16]

CSS:

CSS (Cascading Style Sheets) is a styling language used for describing the presentation of HTML and XML documents. It provides a range of features and tools for designing and styling web pages, such as layout, color, font, and animation. [17]

JSON:



JSON (JavaScript Object Notation) is a lightweight data-interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It uses a simple and flexible syntax to represent data in key-value pairs or ordered lists. [18]

2.3 Software architecture

The design of logical architecture is an important aspect of software development where we break down the system into smaller, more manageable components. It helps us to understand the relationships and dependencies between different components and how they work together to achieve the desired functionality. For this project, we decided to use the Model-View-Controller (MVC) design pattern, which is widely adopted in web development and other application development frameworks. One of the main advantages of the MVC pattern is that it promotes a clear separation of concerns, allowing for more manageable and maintainable code. This separation makes it easier to develop, test, and modify individual components of the application without affecting others.

- **Model:** This component represents the data and business logic of the application. Its main responsibility is to oversee how data can be accessed and modified by applying relevant rules.
- **View:** This component represents the user interface of the application. Its core responsibility is to present the data to users and gather input from them.
- **Controller:** This component acts as an intermediary between the Model and the View components. It is responsible for handling user input and subsequently updating both the Model and View components as needed.

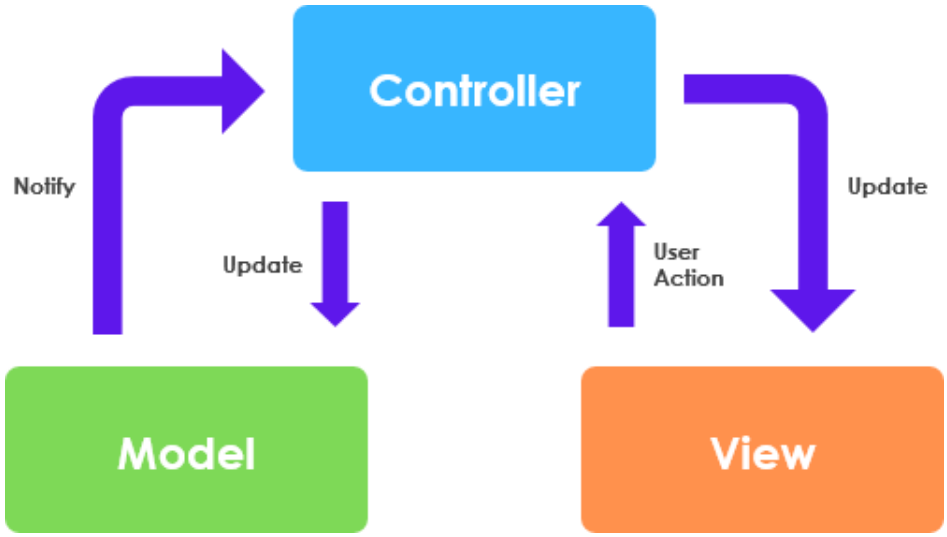


Figure 2.2: MVC architecture
[19]

2.4 Hardware architecture

The physical architecture of a system is a critical factor in determining its performance, reliability, and scalability. In our project, we have chosen to implement a three-layer approach to the hardware architecture. This approach involves a presentation layer, a business logic layer, and a data storage layer, which provides an array of advantages for our project.

One of the advantages of the three-layer approach is the clear separation of concerns, making it easier to manage and maintain the different system components. On top of that, it is an extremely flexible and scalable system since each layer can adjust independently based on the ever-shifting demands of the system. What's more, this approach encourages code reuse and an organized modularity system, giving each layer the chance to be individually developed and assessed before it's integrated into the system.

- **Presentation layer:** This is the topmost layer of the application, which is responsible for presenting data and information to users in a user-friendly format. It handles user input and output and communicates with the other layers via well-defined interfaces.
- **Business logic layer:** This is the middle layer of the application, which contains the core business logic and application processing. It handles application-specific processing and communicates with the presentation layer and data access layer.
- **Data access layer:** This is the bottommost layer of the application, which is responsible for accessing data from the underlying data store, such as a database or file system. It communicates with the business logic layer via a well-defined interface and provides the necessary data for processing.

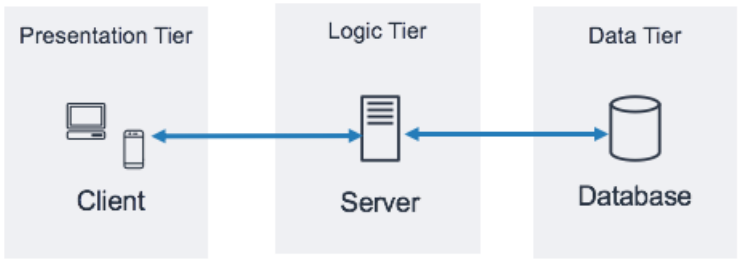


Figure 2.3: three-layer architecture
[20]

VI. Conclusion

In conclusion, Chapter 2 (Sprint 0) has provided valuable insights into the initial planning stages of agile software development. The concept of Sprint 0 has been introduced as a preparatory phase that focuses on gathering requirements, defining goals, and establishing the architectural framework for the project. The use of use case diagrams and product backlog has also been highlighted as key tools in the planning process. Finally, the decomposition of the sprint has been discussed as an essential process in ensuring the successful completion of the sprint goal. By following the principles outlined in this chapter, We can effectively plan and execute our projects, resulting in successful and high-quality software products.

Chapter 3

Sprint 1

3.1 Introduction

This chapter is dedicated to the first sprint of developing an application, which involves setting up the project and establishing basic functionality. This sprint comprises three fundamental steps:

- Implementing a basic database schema to store job listings and user profiles.
- Implementing user authentication and authorization to secure user data.

3.2 Sprint Backlog

The main focus of this sprint is to implement user authentication and basic database functionality for storing user profiles and job listings. The implementation of user authentication and authorization is crucial in ensuring that the system is secure and users' sensitive information is protected. The basic database schema will enable candidates to store and manage their profile information, as well as allow recruiters to post job listings. The database will be designed with scalability in mind to accommodate future growth and changes to the system. The implementation of these features will provide a solid foundation for the rest of the project, allowing for the development of more advanced features and functionality in later sprints. We present in Table 4.1 the backlog for this sprint, along with the estimated effort per day for each functionality to be implemented.

ID	Theme	ID	User story	Priority	Estimation
1	login	1.1	As a User, I would like to authenticate myself to access the platform.	M	8
2	Manage Profile	2.1	As a Person, I want to be able to change my profile image	M	2
		2.2	As a Person, I want to be able to change my personal information.	S	2
		2.3	As a Person, I want to be able to upload my resume for potential employers to review.	M	5
3	Manage Job Application	3.1	As a Candidate, I want to be able to apply for job posts	M	2

		3.2	As a Candidate, I want to be able to consult job posts.	S	2
		3.3	As a Candidate, I want to be able to abandon the job application.	M	5
		3.4	As a Recruiter or Candidate, I want to be able to consult job applications.	M	5
4	Manage Job Posts	4.1	As a Recruiter, I want to be able to add job posts	M	2
		4.2	As a Candidate, I want to be able to delete job posts.	S	2
		4.3	As a Candidate, I want to be able to update job posts.	M	5
5	Manage Accounts	5.1	As an Administrator, I want to be able to Delete user	M	2
		5.2	As an administrator, I want to be able to consult user profiles.	S	2
		5.3	As an administrator, I want to be able to search for users.	M	5

Table 3.1: Project Backlog

3.3 Analysis

Now that the requirements for our first sprint have been defined, we move on to the presentation of the general use case diagrams. The purpose of these diagrams is to provide an overview of all the functionalities provided by the application, along with textual descriptions that outline the scenarios for each use case. In this chapter, we will discuss the general use case diagrams developed for Sprint 1 of the project, providing insights into the design decisions and how they relate to the requirements.

3.3.1 General use case for Sprint 1

The needs to be fulfilled in our first sprint have been specified. We now move on to the presentation of the use case diagrams, which aim to provide an overview of all the features provided by the application, as well as textual descriptions that describe the scenarios of

each case.

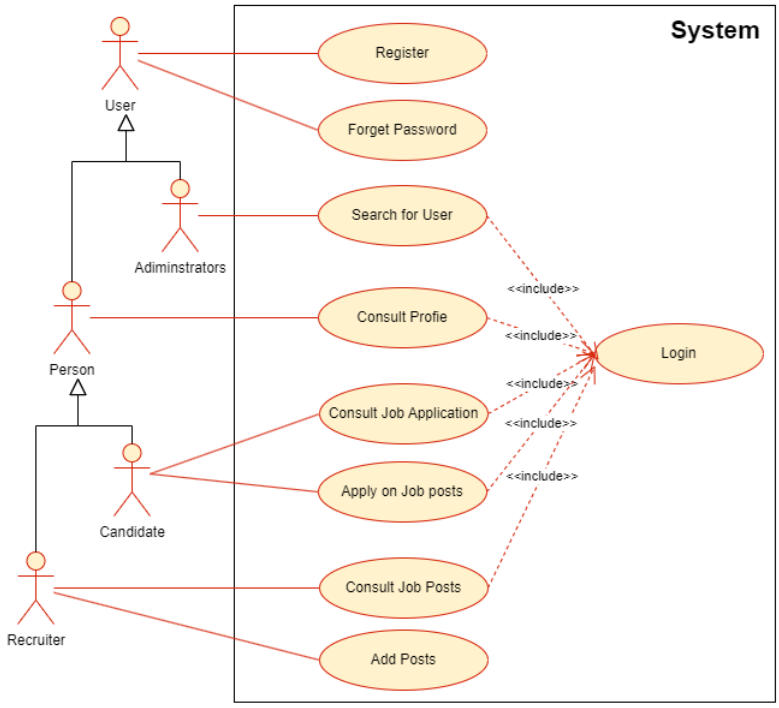


Figure 3.1: Caption

3.3.1.1 Refinement of general use case «register»

To connect to the platform, the user needs to create an account with his email **??** below.

Figure 3.2: Caption

Textual description of the use case "register":

Title	Register
Actors	User
Preconditions	The user does not have an account in the system. The system is operational and available.
Main Flow	1. The user accesses the register page. 2. The system presents the register form. 3. The user enters their name, email address,..., and password. 4. The user confirms their password. 5. The system verifies that the email address is unique. 6. The system creates a new account for the user. 7. The system redirects the user to the login page.
Postconditions	The user's account has been created in the system. The user is redirected to the login page.
Alternative Flow	A1. If the user enters an email address that is already registered in the system, the system displays an error message and asks the user to try again. 1. The user enters a different email address. 2. The flow continues from step 3 of the main flow.

Table 3.2: Sign up Use Case

3.3.1.2 Refinement of use case «login»

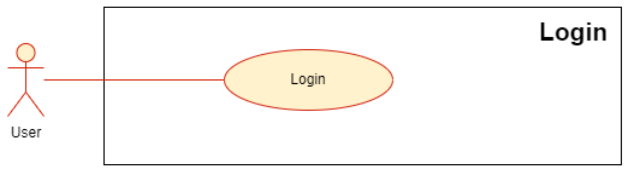


Figure 3.3: Caption

Textual description of the use case "login":

Title	Login
Actors	User
Preconditions	The user has an active account in the system. The system is operational and available.
Main Flow	1. The user accesses the login page. 2. The system presents the login form. 3. The user enters their email address and password. 4. The user clicks the "Login" button. 5. The system verifies the user's credentials. 6. The system logs in the user and redirects them to their account home page.
Postconditions	The user is logged in to the system. The user is redirected to their account home page.
Alternative Flow	A1. If the user enters incorrect credentials, the system displays an error message and asks the user to try again. 1. The user enters their email address and password again. 2. The flow continues from step 4 of the main flow.

Table 3.3: Sign up Use Case

3.3.1.3 Refinement of use case «Forget Password»

To manage job posts, a Recruiter has the ability to perform the functionalities modeled in figure 3.4 below.

Figure 3.4: Caption

Textual description of the use case "Forget Password":

Title	Forget Password
Actors	User
Preconditions	The user has an active account in the system. The user has forgotten their password. The system is operational and available.
Main Flow	1. The user clicks on the "Forgot Password" link on the login page. 2. The system presents a form for the user to enter their email address. 3. The user enters their email address and clicks the "next" button. 4. The system verifies that the email address is associated with an active account. 5. The system generates a unique verification code and sends it to the user's email address. 6. The user receives the verification code. 7. The system presents a form for the user to enter the verification code, and the new password and confirm it. 8. The user enters their verification code, and new password and confirms it. 9. The system updates the user's password in the system and logs them in. 10. The system redirects the user to their account home page.
Postconditions	The user's password has been reset and updated in the system. The user is logged in to the system. The user is redirected to their account home page.
Alternative Flow	A1. If the user enters an email address that is not associated with an active account, the system displays an error message and asks the user to try again. 1. The user enters a different email address.

Table 3.4: Sign up Use Case

3.3.1.4 Refinement of use case «Consult Profile»

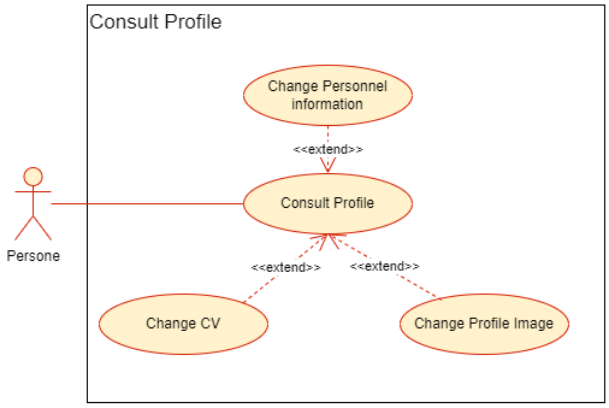


Figure 3.5: Caption

Textual description of the use case "Consult Profile":

Title	Consult Profile
Actors	Person
Preconditions	The Person is logged in to the system. The Person has a profile in the system.
Main Flow	1. The Person clicks on the "My Profile" link in the navigation menu. 2. The system displays the user's profile page. 3. The Person views their profile information, including their name, email address, and any other details they provided during registration. 4. The Person can edit their profile information by clicking on the "Edit" button. 5. The system presents a form for the user to edit their profile information. 6. The Person edits their information and clicks the "Save" button. 7. The system saves the updated information and displays the user's profile page with the new changes.
Postconditions	The user's profile information has been updated in the system (if applicable). The user can view their profile information on their profile page.
Alternative Flow	A1. If the user clicks on the "My Profile" link and they are not logged in, the system redirects them to the login page. 1. The user logs in to the system. 2. The system redirects it to the home page.

Table 3.5: Sign up Use Case

3.3.1.5 Refinement of use case «Search User»

To manage a job application, a Candidate has the ability to perform the functionalities modeled in Figure 3.5 below.

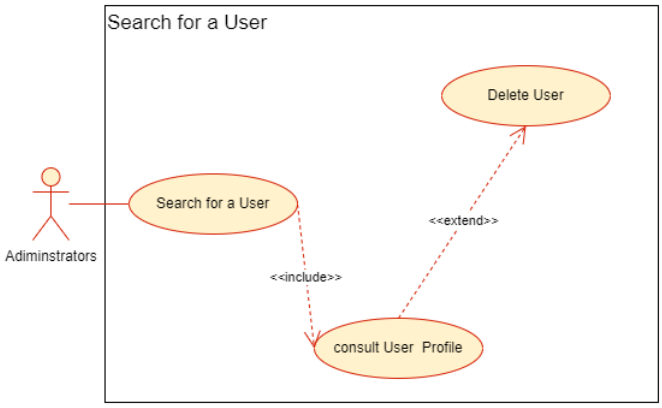


Figure 3.6: Caption

Textual description of the use case "Search User":

Title	Search User
Actors	Administrator
Preconditions	The administrator is logged in to the system. The system has user records to search from.
Main Flow	1. The administrator clicks on the "Search Users" input box. 2. The administrator enters the search criteria, such as name, email address, or username. 3. The system searches the user records for matching entries based on the search criteria. 4. The system displays a list of matching user records. 5. The administrator can view the details of any user record by clicking on the user's name or username. 6. The administrator can delete a user account.
Postconditions	The administrator can search for and view user records based on search criteria. The administrator can delete a user account if desired.
Alternative Flow	A1. If no matching user records are found for the search criteria, the system displays a message indicating that no users were found. 2. The flow ends

Table 3.6: Sign up Use Case

3.3.1.6 Refinement of use case «Add Job Post»

Figure 3.7: Caption

Textual description of the use case "Add Job Post":

Title	Add Job Post
Actors	Recruiter
Preconditions	The Recruiter is logged into the system. The Recruiter has permission to add job posts.
Main Flow	1. The Recruiter clicks on the "Add Job Post" link in the navigation menu. 2. The system presents a form for the Recruiter to fill in the job post details, including job title, category, location, job description, requirements, and application instructions. 3. The Recruiter fills in the required fields and submits the form. 4. The system displays a confirmation message indicating that the job post was successfully created.
Postconditions	A new job post is created in the system with the details provided by the Recruiter.
Alternative Flow	A1. If the Recruiter leaves any required fields blank or provides invalid input, the system displays an error message indicating which fields need to be corrected. 2. The Recruiter corrects the errors and resubmits the form. 3. The flow returns to step 3 of the main flow.

Table 3.7: Sign up Use Case

3.3.1.7 Refinement of use case «Consult Job Posts»

To manage users, an administrator has the ability to perform the functionalities modeled in figure **??** below.

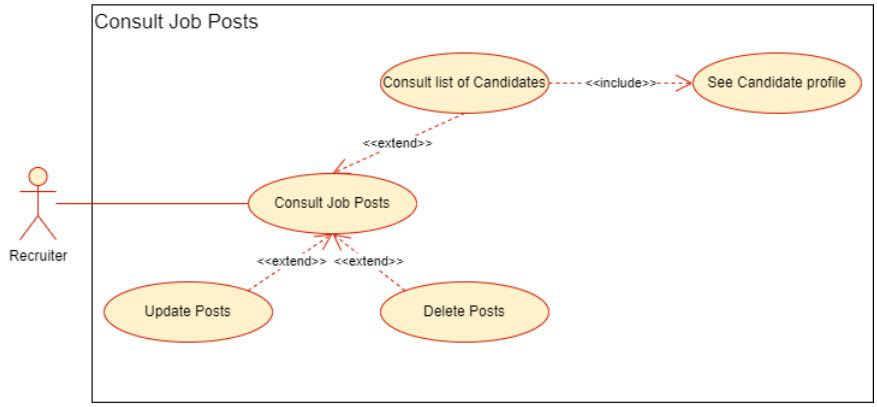


Figure 3.8: Caption

Textual description of the use case "Consult Job Posts":

Table 3.8: Sign up Use Case

Title	Consult Job Posts
Actors	Recruiter
Preconditions	The Recruiter is logged into the system.
Main Flow	1. The Recruiter clicks on the "Job Posts" link in the navigation menu. 2. The system presents a list of all available job posts. 3. The Recruiter can filter the job posts by criteria such as location, job type, and category. 4. The Recruiter selects a job post from the list. 5. The system presents the details of the job post, including the job description, requirements, and application instructions. 6. The Recruiter can edit or delete the job post if necessary.
Postconditions	The Recruiter can view the details of the job posts. The Recruiter can edit or delete the job post if necessary.
Alternative Flow	A1. If there are no job posts available to consult, the system displays a message indicating that there are no job posts available. 2. The flow ends

3.3.1.8 Refinement of use case «Apply Job Post»

Figure 3.9: Caption

Textual description of the use case "Apply Job Post":

Title	Apply Job Post
Actors	Candidate
Preconditions	The Candidate is logged in to the system. A job post is available in the system.
Main Flow	1. The Candidate navigates to the job post they want to apply to. 2. The Candidate clicks on the "Apply" button. 3. The system presents a form for the user to submit their application, including their resume and a cover letter. 4. The Candidate fills out the application form and uploads their resume and cover letter. 5. The Candidate clicks the "Submit" button to send their application. 6. The system confirms that the application has been submitted successfully.
Postconditions	The Candidate's job application has been submitted successfully. The job poster has been notified of the new application.
Alternative Flow	A1. If the Candidate clicks on the "Apply" button and they are not logged in, the system redirects them to the login page. 1. The Candidate logs in to the system. 2. The system redirects it to the home page.

Table 3.9: Sign up Use Case

3.3.1.9 Refinement of use case «Consult Job Application»

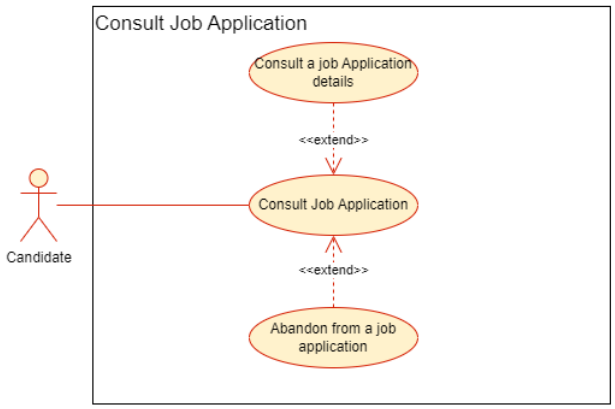


Figure 3.10: Caption

Textual description of the use case "Consult Job Application":

Title	Consult Job Application
Actors	Candidate
Preconditions	The Candidate is logged in to the system. The Candidate has applied for a job in the system.
Main Flow	1. The Candidate clicks on the "Job Applications" link in the navigation menu. 2. The system displays a list of the user's job applications. 3. The Candidate selects the job application they want to consult. 4. The system presents the job application details, including the job post details and the user's application information. 5. The Candidate can delete their job application if desired.
Postconditions	The Candidate can view the details of their job application. The Candidate can delete their job application if desired.
Alternative Flow	A1. If the user clicks on the "Apply" button and they are not logged in, the system redirects them to the login page. 1. The user logs in to the system. 2. The system redirects it to the home page.

Table 3.10: Sign up Use Case

3.4 Design

3.4.1 Sequence diagram

For a good understanding of the components of the first sprint, it is essential to present sequence diagrams of the main use cases. These diagrams describe the scenarios of each case by emphasizing the chronological factor, as well as the interaction between the different objects that make it up.

3.4.1.1 Sequence diagram «login»

Figure 3.11: Caption

3.4.1.2 Sequence diagram «delete user»

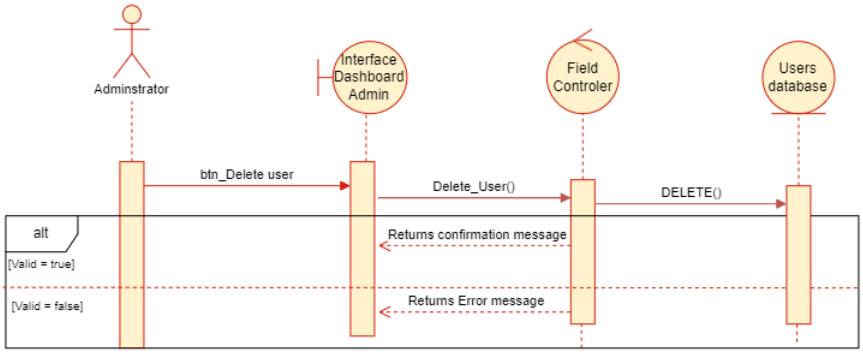


Figure 3.12: Caption

3.4.1.3 Sequence diagram «consult user»

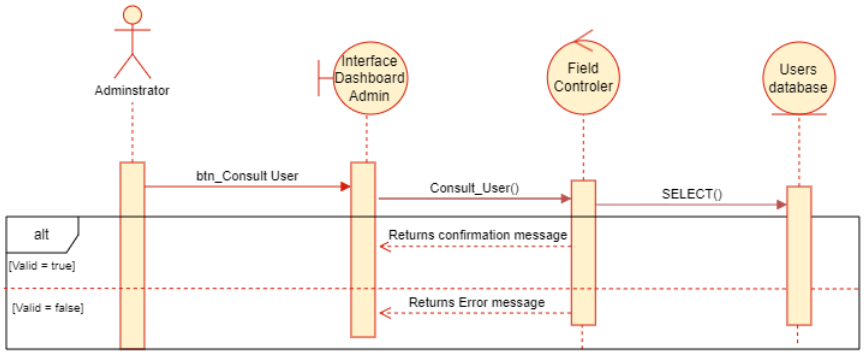


Figure 3.13: Caption

3.4.1.4 Sequence diagram «search user»

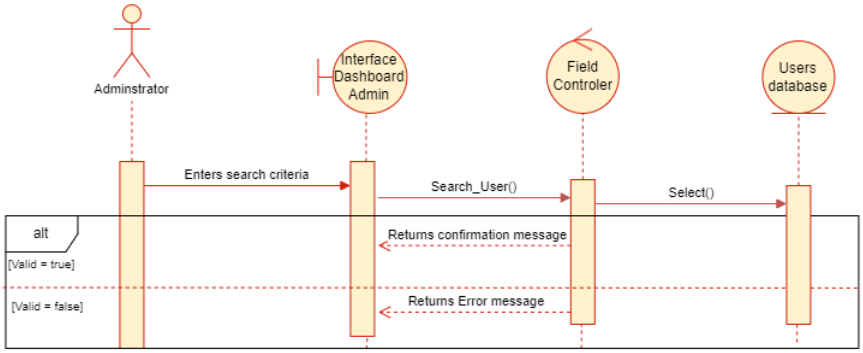


Figure 3.14: Caption

3.4.1.5 Sequence diagram «Add job Post»

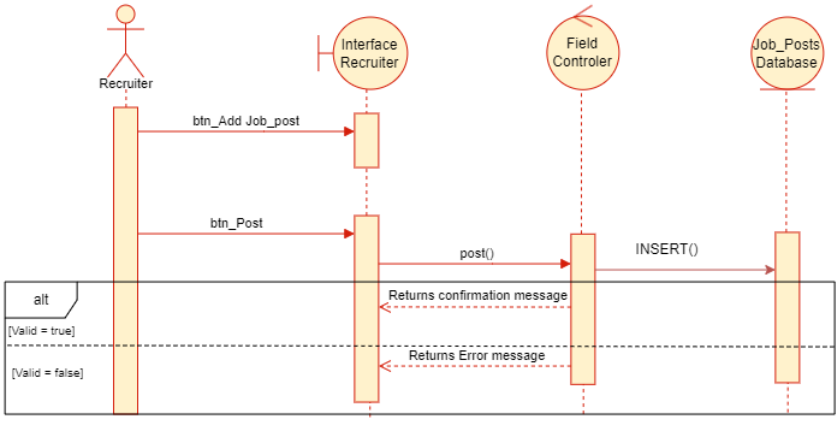


Figure 3.15: Caption

3.4.1.6 Sequence diagram «delete job Post»

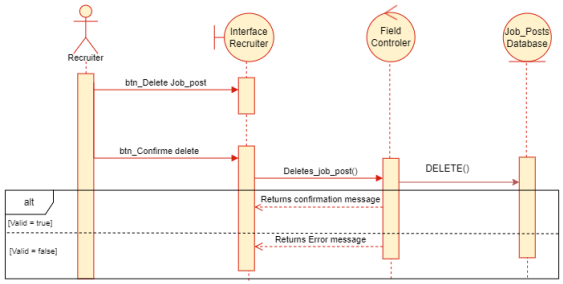


Figure 3.16: Caption

3.4.1.7 Sequence diagram «edit job post»

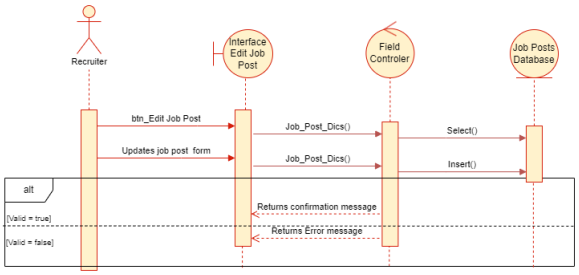


Figure 3.17: Caption

3.4.2 Diagram du class

The sequence diagram presented in the previous section allowed us to see a dynamic view of our system. In order to present the different entities that make up our system along with their various relationships in a static manner, we have dedicated this section to defining a diagram that is part of the structural diagrams of UML, which is the class diagram illustrated in figure 4.6 below.

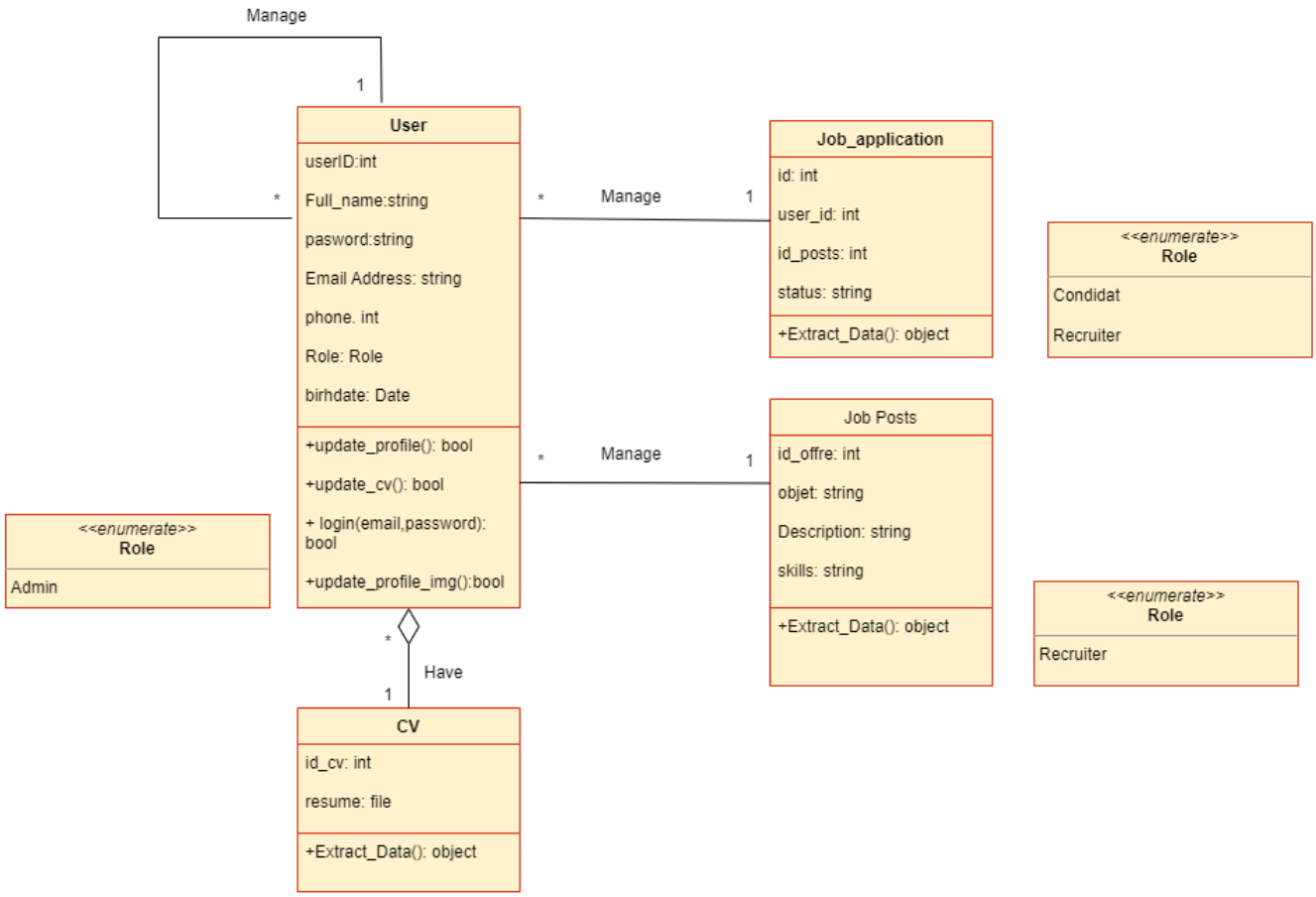


Figure 3.18: Caption

3.5 Implementation

After presenting the conceptual solution, we will now showcase the fruit of our work by presenting the different interfaces that are most significant for this first sprint.

3.5.1 Register Interface

The 3.19 illustrates the interface corresponding to the user story "Register". The user must fill out the required fields to create their account.

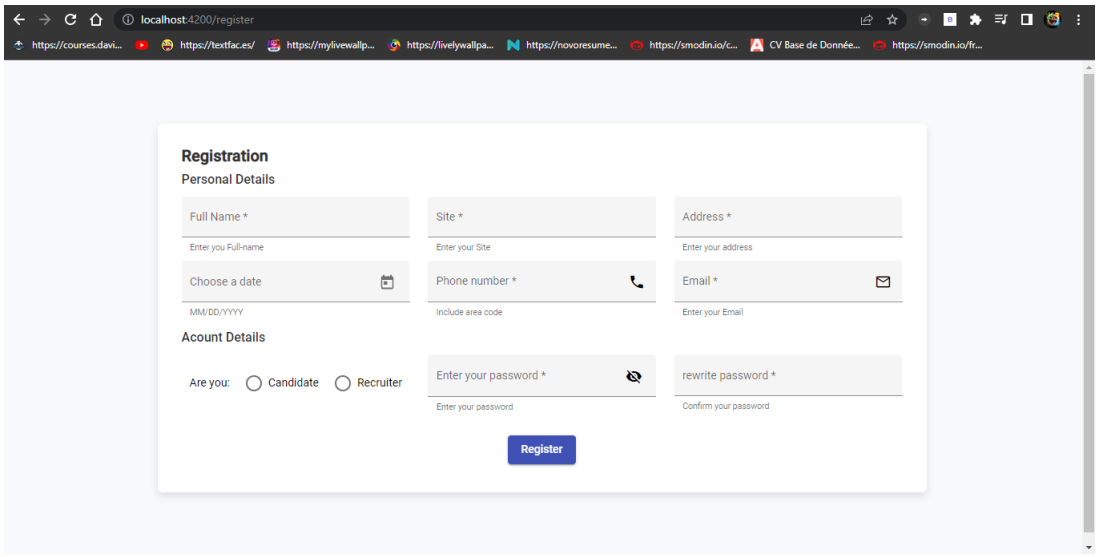


Figure 3.19: Interface «Register»

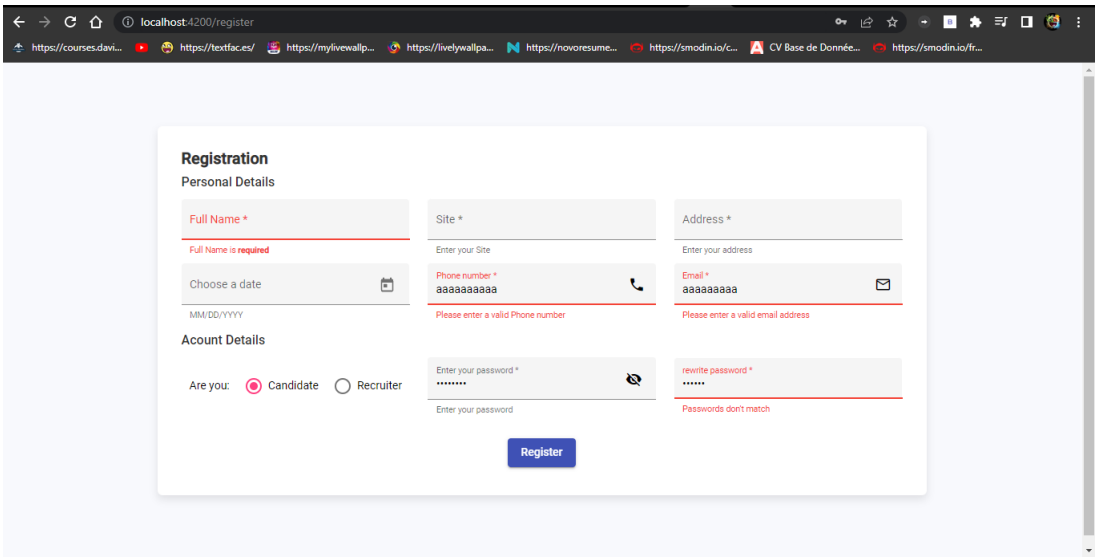


Figure 3.20: Wrong Input Interface

3.5.2 Login Interface

When a user wants to log in to their account, they must therefore authenticate themselves. This interface allows the application user to access the system after validating their email and password.

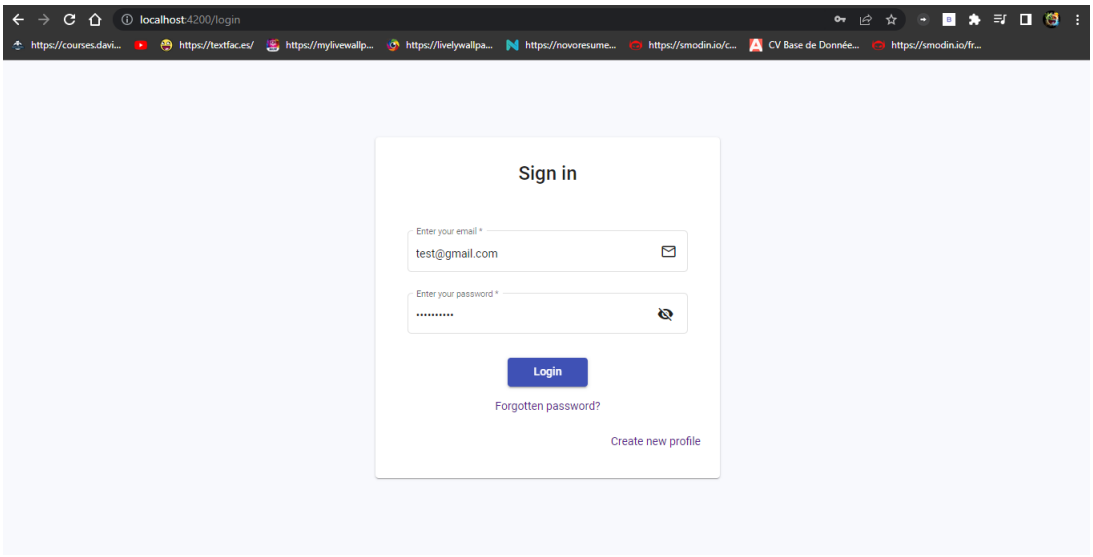


Figure 3.21: Interface «Login»

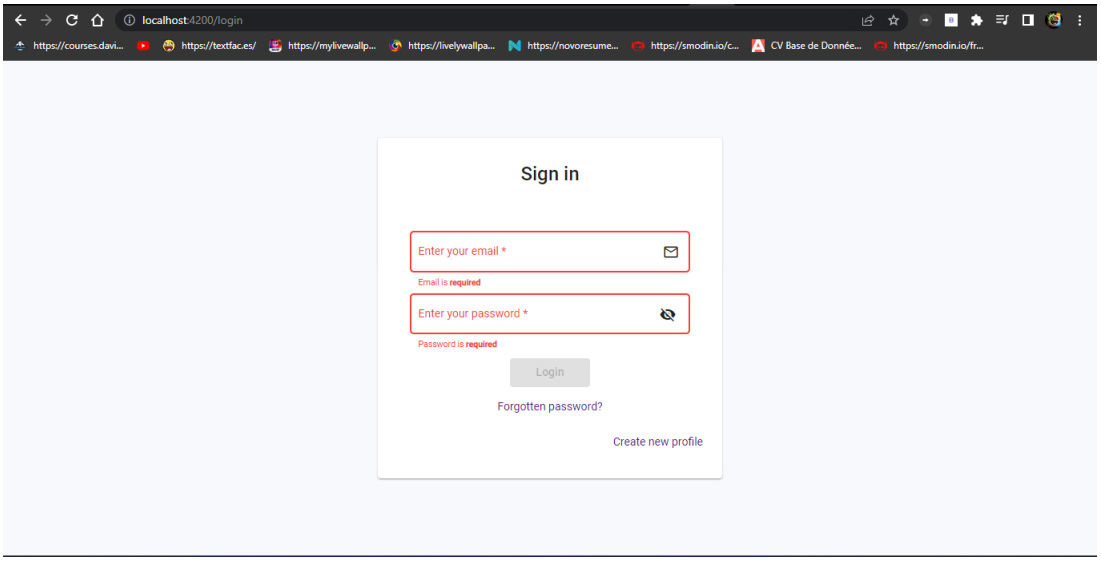


Figure 3.22: Wrong Input Interface

3.5.3 Reset Password Interface

The 3.23 illustrates the interface corresponding to the user story "Forgot Password". The user must enter their email to receive a code on their email and then re-enter it into the system to reset their account password.

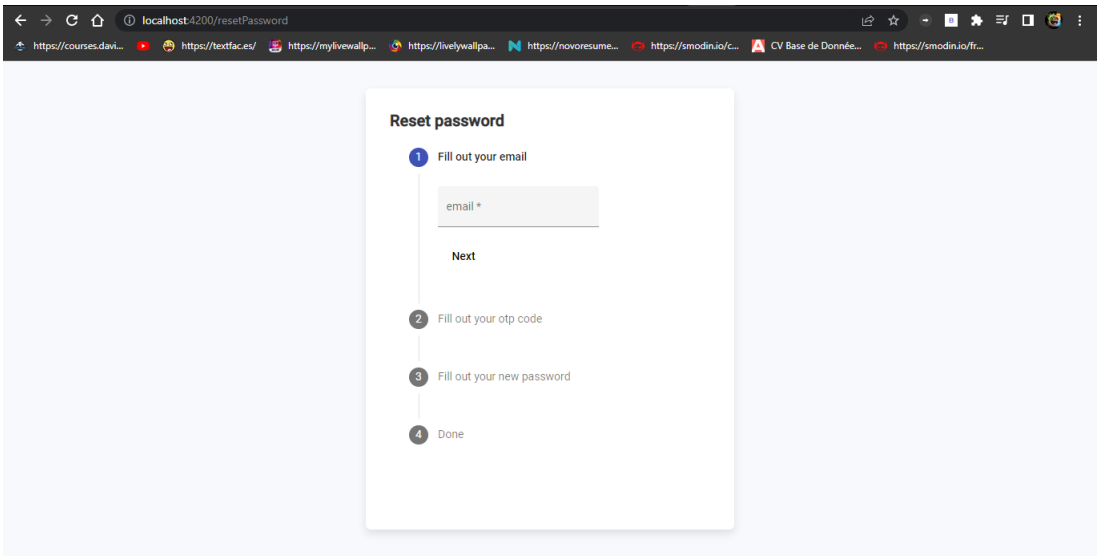


Figure 3.23: Inerface «Reset Password»

3.6 Testing

We will perform some test cases to test the web services using Postman and we will test the application itself

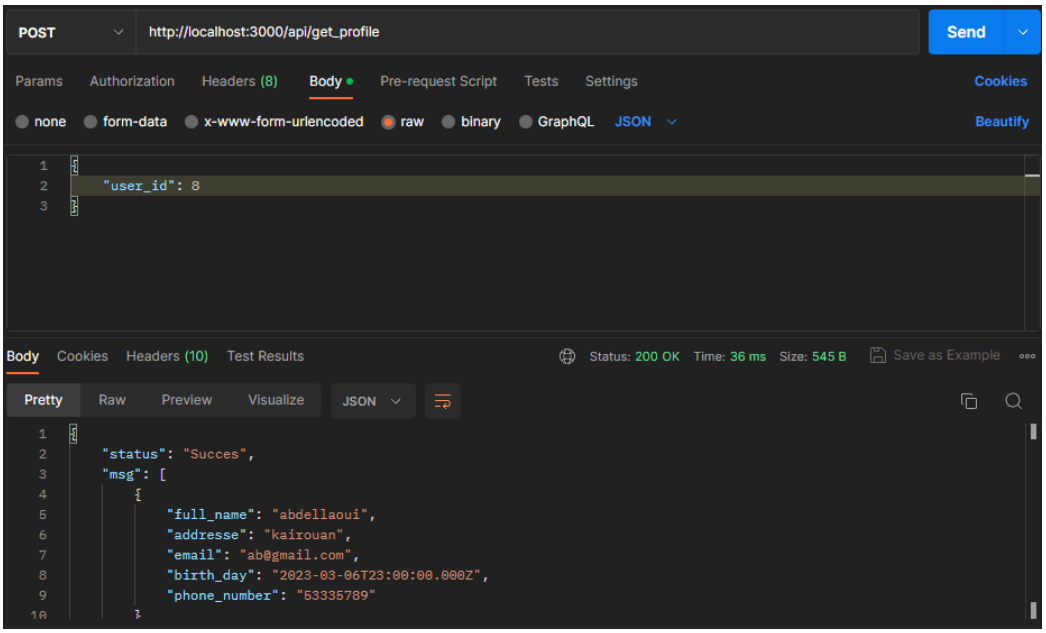


Figure 3.24: Testing the Get Profile Data API

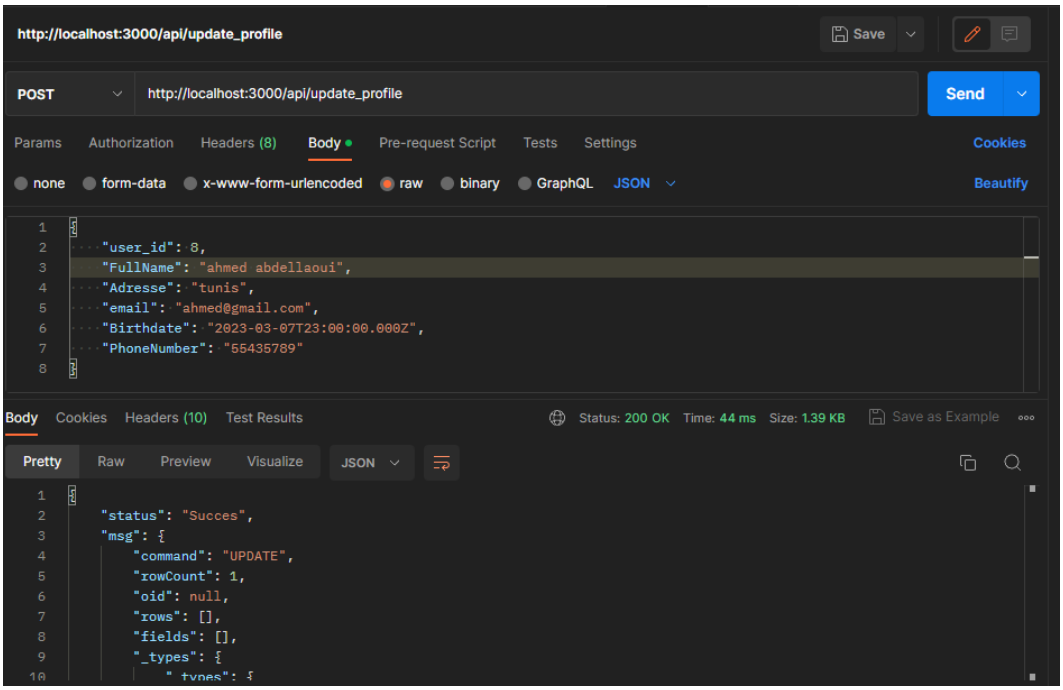


Figure 3.25: Testing the Update Profile API

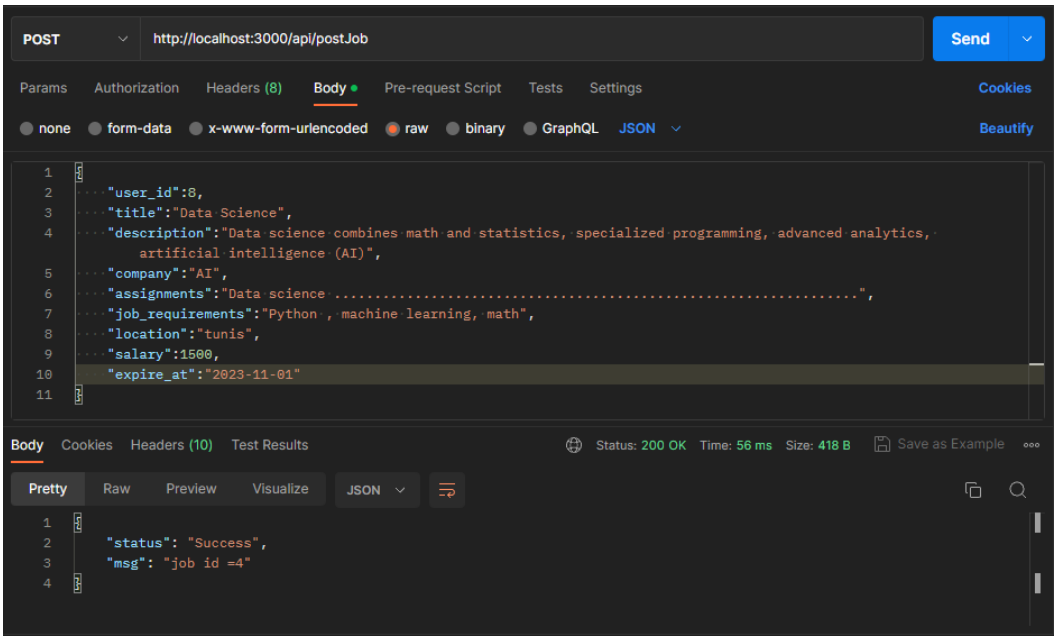


Figure 3.26: Testing the job Post API

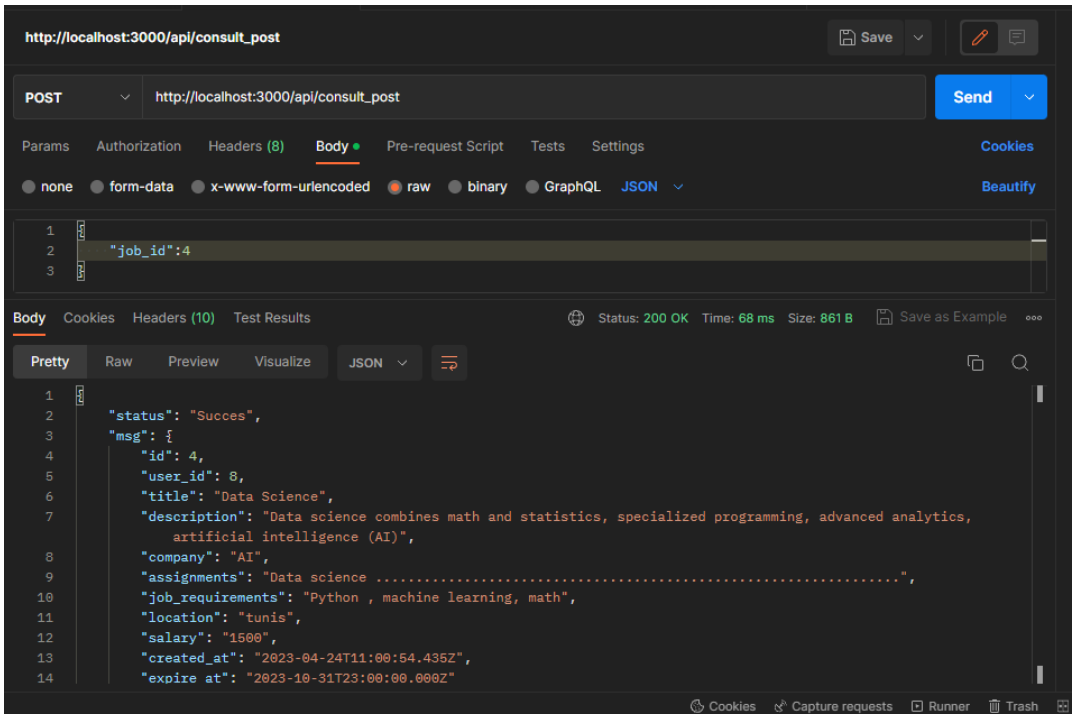


Figure 3.27: Testing the Retrieve Job Post Information

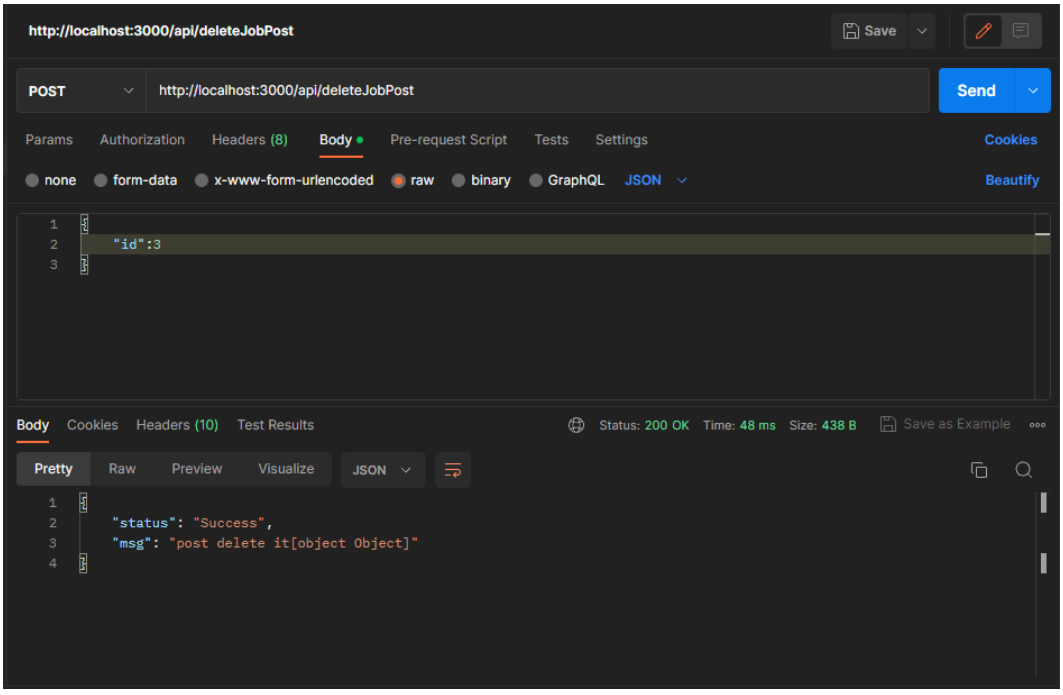


Figure 3.28: Testing the delete of job Post API

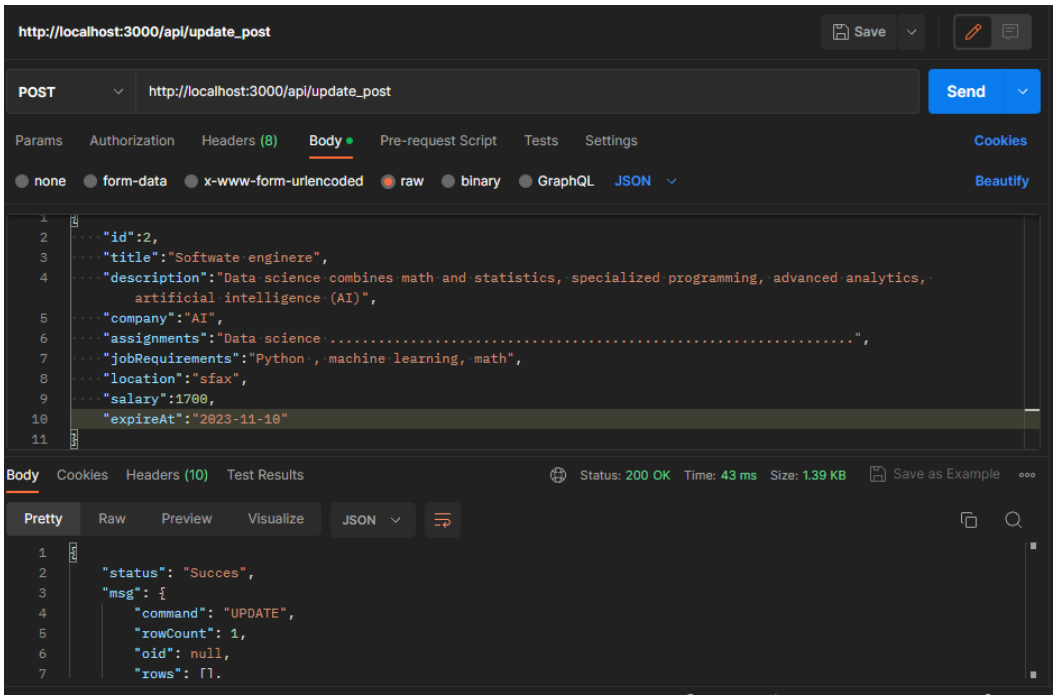


Figure 3.29: Testing the update post information API

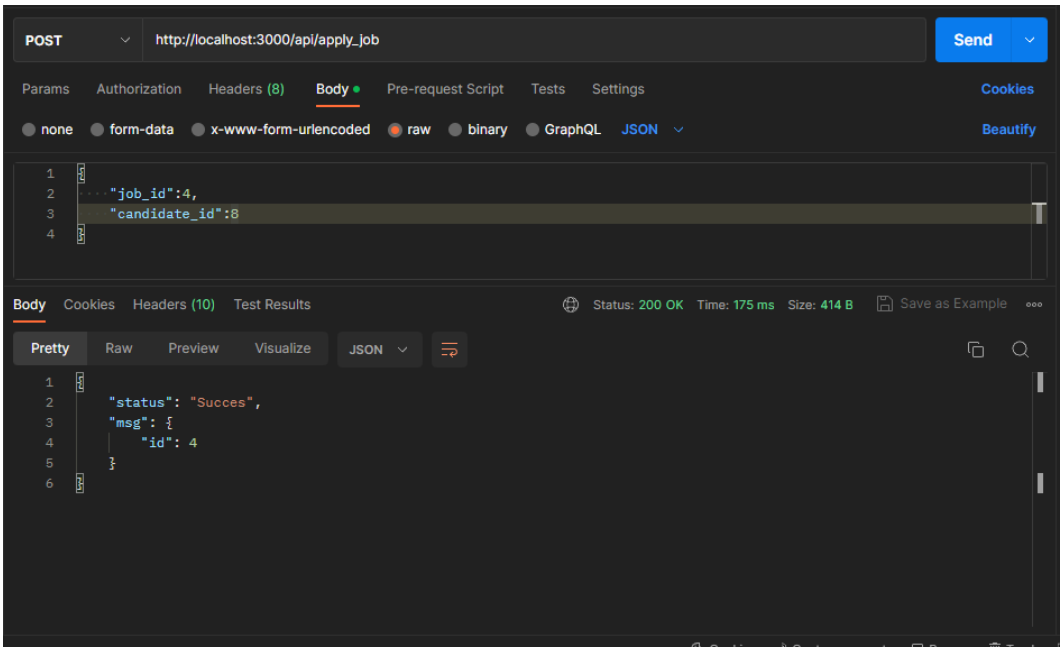


Figure 3.30: Testing the application on Job Post API

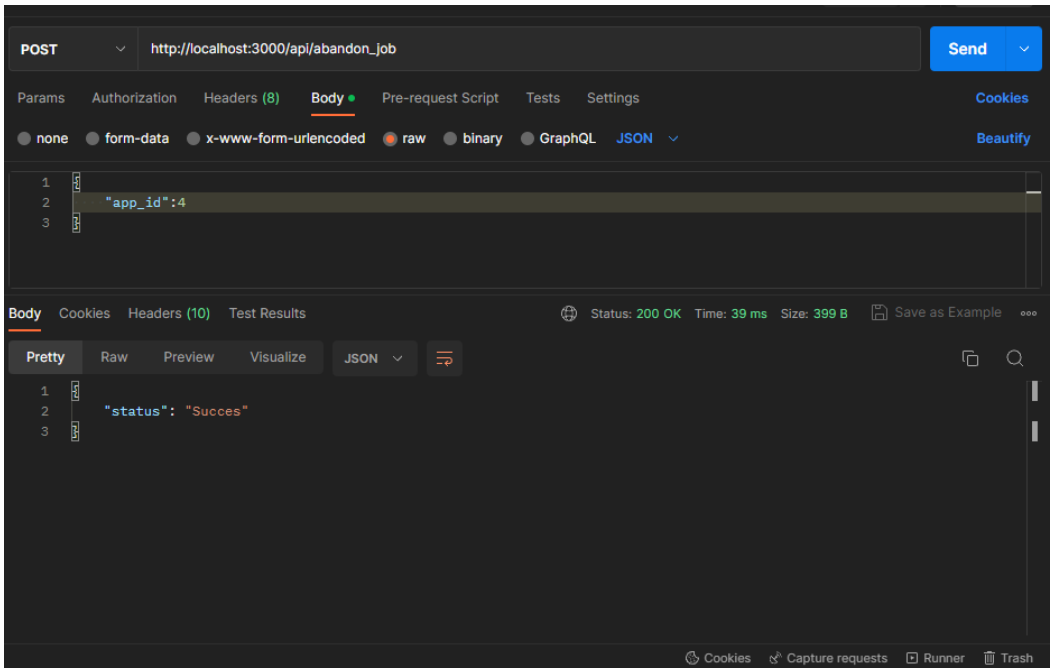


Figure 3.31: Testing the abandoned on job application

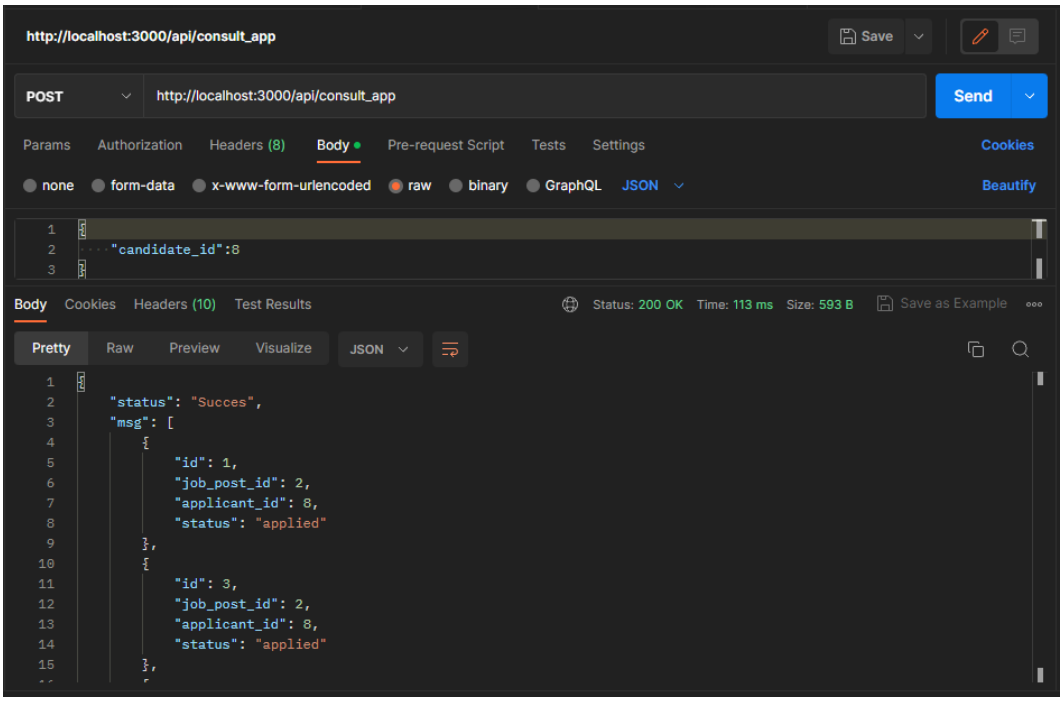


Figure 3.32: Testing the Retrieve of all job application History

Conclusion

Through this chapter, we have presented the specification of needs, the design illustrated by detailed sequence diagrams and a class diagram, as well as the implementation of user stories for the first sprint. After testing and validating the features with the Scrum Master and the Product Owner during the sprint review, we will now move on to the next step: sprint 2.

Chapter 4

****

Conclusion générale