Ministry of Higher Education
and Scientific Research

ACADEMIC YEAR
2022/2023

EPI
Private International Higher
School of Polytechnic

END OF STUDIES PROJECT

In order to obtain the National Diploma in Engineering

Field of study: Software Engineering

Entitled

Falumni: Online recruitment platform

Internship place

CleverTech

Author

Khalil Chaftar

Supervisors

Mm. Maryem Jaouadi
Mr. Dhiaeddine Harchay

Dedications

I dedicate this work
To my dear parents,
I owe you who I am today thanks to your love, your patience, your countless sacrifices,
May this modest work be for you a small compensation and recognition for what you have done incredible for me
May God, the Almighty, preserve you and grant you a healthy and long life so that I can, in turn, fulfill you.
To my dear sister and dear brother,
No dedication can express deeply enough how I feel about you.
I would say, quite simply, a big thank you, I love you.
To my dear friends,
As a testimony to the sincere friendship that bound us. I dedicate this work to you, wishing you a bright and promising future.

Thanks

At the end of this work, I would like to express my sincere thanks to everyone who has contributed, directly or indirectly, to the realization of this work.

I would like to thank my internship tutor Mr. Dhia Harchay for welcoming me warmly within the CleverTech company and for all the time he devoted to me, his precious directives, and for the quality of his follow-up throughout. the period of my internship.

I would particularly like to thank and express my gratitude to my supervisor Mrs. Maryem Jaouadi for the time she devoted to me throughout the period of my internship, for her availability, her advice and her judicious remarks and above all for her confidence and her continuous encouragement which allowed me to move forward.

Finally, I am grateful to the members of the Jury for the honor they have done me by accepting to evaluate my work.

Contents

List of Figures

List of Tables

Abreviations list

- JWT = JSON Web Token
- API = Application Programming Interface
- REST = Representational State Transfer
- JSON = JavaScript Object Notation
- MVT = Model View Template
- UML = Unified Modeling Language
- ORM = Object Relational Mapping
- UI = User Interface
- UX = User eXperience
- US = Use Case
- HR = Human Resources
- MCQ = Multiple-Choice Question
- RTK = Redux Toolkit
- MUI = Material UI

General Introduction

In an ultra-competitive world where innovation makes the difference, it is important for a company to be attentive to its performance, always in line with the expectations of its customers, while maintaining a head start on its current competitors. and potentials. It is from this observation that the process of acquiring Human Resources takes on its full importance and meaning.

The recruitment process represents the extension of a set of strategic activities whose success is an essential condition for the social and economic success of any organization. Indeed, an organization that succeeds in having qualified and motivated employees improves its chances of success in highly competitive markets and benefits from a clear advancement advantage. The competitiveness of a company depends above all on the quality of its Human Resources and the relevance of the choice of these service providers, which requires planning and implementing an effective human resources acquisition process, which will be up to daily requirements posed by the new socio-economic context.

Our graduation project falls within this context and consists of creating a Staffing platform for companies that are looking for qualified resources in the field of technology. The platform aims to bring employees of service companies closer to companies that need the skills and resources to undertake projects during a defined period. The objective of the platform is to streamline the process of searching for and identifying profiles based on an intelligent Matching algorithm between a service request and qualified and available profiles, as well as focusing on a questionnaire module to do tests for the talents who have applied for an application.

This report summarizes all the work we have done throughout this internship. It consists of four chapters as follows:

• In the first chapter, « **General context and preliminary study** », we will present the host organization and the general context of the work. We will also describe the posed problematic and the proposed solution. Finally, we will present the methodology as well as the gantt chart chosen for the realization of this project.

• The second chapter entitled « **Analysis and specification of needs** » is reserved for the specification of the actors, the functional and non-functional needs of the application and the mentioning of the product backlog at the end.

• The third chapter entitled « **Conceptual study** » focuses on the preliminary design of the application.

• The fourth and the final chapter entitled « **Implementation** » illustrates the chosen technologies, the global architecture for the application in question, the deployment process and the work carried out through some of the interfaces designed for the application in question.

This report will end with a general conclusion and outlook summarizing all the work we have done, while listing the skills and experience we have acquired during these four months of internship.

Chapter 1

General context and preliminary study

1.1 Introduction

In this chapter, we present the general context of our project. We begin with a presentation of the host organization. The second part will be devoted to our overall project context in which we will present the problem and the study of the existing situation followed by the proposed solution. And finally, the last part will be devoted to the working methodology adopted as well as the project's Gantt chart.

1.2 Presentation of the host organization

CleverTech [3] is a French digital consulting company founded in 2014. It covers the transformation and engineering needs of companies, it provides consulting around different areas of expertise grouped by subsidiaries. It offers consulting services in the identification, development and implementation of digital transformation projects that strengthen growth and accentuate the competitiveness of its clients through strategy consulting and program management, it constantly contributes to the innovation and optimization of the operational processes of several organizations. It offers a variety of digital innovation services and offers expertise that combines digital strategy consulting with user experience to develop perception, experience and interaction. It also offers application development services, it implements real technical expertise and technology monitoring to ensure the quality of each project. It has been designing and producing solutions and applications for several years. It has its own StartUp Labs dedicated to digital innovation and the development of its own IT solutions and within this department that my end-of-studies internship took place. The figure 1.1 represents the logo of the CleverTech company.

Figure 1.1: CleverTech Logo

1.3 Project presentation

1.3.1 Project context

This work is part of the end-of-study project to obtain the National Engineering Diploma in Software Engineering from the International Multidisciplinary School (EPI). This project is carried out within the company CleverTech.

1.3.2 Problem

Today, talented profiles with cutting-edge expertise are present all over the world. This learning mutation leads recruiters to find themselves faced with thousands of profiles available on the internet with unverified skills and complex acquisition procedures.

Some talented profiles are considering problems finding employment or internship opportunities abroad. Even if they find, the host company will encounter difficulties on its part in carrying out the processing operations of the necessary procedures to finalize the mobility of its target talent.

For young graduates, it is a challenge to find jobs at the stage of transition from academic life to professional life due to the lack of communication between universities and the labor market and not belonging to a network that opens the way to valuable opportunities.

Processing a lot of applications, carrying out many interviews are painful tasks for the human resources department, especially in a large company, which must carefully study the details of each application and plan interviews with different objectives which causes the consumption of several resources.

1.3.3 Project objective

The Falumni platform is structured around 4 main actors: university user, talent user, company user and administrator. In this part, the contributions of the solution for each actor will be presented.

For companies, a published job offer can be visible to all users or distributed only in private networks of alumni of prestigious schools. In order to minimize the duration of the recruitment process, the solution offers the possibility of delegating applications and their passing of specific tests in order to simplify access to quality profiles.

The Falumni solution allows prestigious universities to increase the employability of their students and alumni, thus promoting their reputation internationally. The platform offers a space for universities to bring together Alumni and students from the same university to exchange job opportunities and make them confront job offers published by partner companies that are looking for talents.

Finally, the Falumni solution gives talent and young graduates the opportunity to boost their careers in large international companies. Each talent has a very intuitive space that makes it easier for them to find a job and to load their profile and present their skills, on the platform each profile will receive offers and target content according to their professional ambitions and to have a very relevant follow-up space for monitoring the progress of applications.

1.4 Study of the existing

The study of the existing is a preponderant phase to fully understand the current system. Its objective is to study and identify the shortcomings of the existing system, to propose appropriate solutions and to define the objectives to be achieved under the heading of improvement. There are two cases when we proceed to the study of the existing :

• Either the product already exists, then it must be improved.

• Either the product does not exist, it will therefore have to be created.

1.4.1 Existing solutions

We have made a study of the existing on the national and international market.

1.4.1.1 Solutions in Tunisia

We start with the solutions in our country in order to have a better vision on the national competitors.

• **Tanitjobs:**

Tanitjobs [16] is a recruitment platform. It provides users with functionalities for managing candidacies, managing job offers, managing profiles and searching for opportunities according to several criteria.

Figure 1.2: Tanitjobs logo

• **Sintegra Consulting:**

Sintegra Consulting [15] is a recruitment firm that presents employment opportunities for Tunisians and Moroccans by ensuring recruitment campaigns. Sintegra Consulting invites representatives of partner companies to attend, come and recruit and meet their future employees directly on site.

Figure 1.3: Sintegra Consulting logo

1.4.1.2 International Solutions

In order to extend our view of the target platform and since we were aiming to create a solution on an international scale, we also did our study of the existing on the international market.

• **JobTeaser:**

JobTeaser [9] is a French company that provides recruitment and employer branding solutions for companies to hire young talents and a free career platform for the career services of higher education institutions in Europe.

Figure 1.4: JobTeaser logo

• **Cooptalis:**

Cooptalis [5] is a platform that detects, moves and develops talents for companies around the world. Cooptalis provides innovative responses to companies in their rapid search for rare talent and supports talent in their mobility and professional flexibility projects.

Figure 1.5: Cooptalis logo

1.4.2 Criticism of the existing

We have identified some requirements and criteria to consider when comparing and evaluating applications as described in Table 1.1. We will then explain the most relevant criteria for our solution.

• **Profile management**: the user can access his profile and he can modify all his personal data.

• **Application Update**: regularly provide updates to guarantee the improvement of the service of the platform and the satisfaction of the users.

• **Profile verification**: the platform verifies user profiles (identity, company name, company email, etc.).

• **Interview management**: companies can schedule or view interviews.

• **Search for profiles based on a specific job offer**: the platform makes it possible to offer the company verified profiles according to the information of a job offer.

• **Candidacy management**: companies can manage their candidacies.

• **Security and privacy management**: the user can change their security and privacy settings.

• **Space for universities**: the university can manage its network of alumni and invite its partner companies.

• **Recruitment service**: the platform takes care of the mobility process of a candidate for an offer abroad. In addition, companies can delegate the processing of applications for a given offer.

• **Realization of partnership**: the platform takes care of the functionalities of concretization of partnership of the companies with the universities.

• **Notifications**: users will receive notifications of everything that happens on the platform

• **Questionnaires and tests**: the platform offers to the companies the possibility of creating tests and questionnaires in text, video or MCQ for their potential candidates.

Table 1.1: Comparative study of existing applications

Criterion	**Tanitjob**	**Sintegra Consulting**	**Cooptalis**	**Jobteaser**	**Our platform**
Profile Management	✓	✗	✓	✓	✓
Application Update	✓	✗	✓	✓	✓
Profile Verification	✓	✓	✓	✓	✓
Candidacy Management	✓	✓	✓	✓	✓
Search for profiles based on a specific job offer	✓	✗	✗	✓	✓
Interview management	✓	✓	✓	✓	✓
Security and privacy management	✓	✓	✓	✓	✓
Space for universities	✗	✗	✗	✗	✓
Recruitment service	✓	✗	✗	✗	✓
Realization of partnership	✗	✗	✗	✗	✓
Notifications	✓	✗	✓	✓	✓
Questionnaires and tests	✗	✗	✗	✗	✓

1.4.3 Assessment of the study of the existing

The table 1.1 summarizes the result of the comparative study carried out on national and international platforms similar to the application targeted by our end-of-study project. We find that the majority of the platforms mentioned offer the features of profile management, interviews, candidacies, security and privacy management as well as application update and notifications. On the other hand, unlike Falumni that covers such criteria, some platforms do not allow the profile search functionalities according to the offers, recruitment service, realization of partnership, the university space and the questionnaires and test creation service.

1.5 Application development methodology

1.5.1 Design methodology

For the specification and design of this work, we used the modeling language UML [17], which is a modeling language that allows to describe the requirements, to document the systems and to sketch the software architecture.

UML is organized around 3 different axes, each dedicated to the presentation of a specific concept of the software system.

However, we will only show some diagrams that we found useful and sufficient to understand the project, namely use case diagrams, class diagrams, and sequence diagrams.

1.5.2 Adopted methodology

Opting for the best working method is obviously a necessary step to manage the project, which will allow you to manage the progress of the project stages.

In the next section, we will explain our adopted methodology.

1.5.2.1 Agile methodology

Agile methods are very popular project management tools among software and application developers. Recently, many communication agencies and marketing departments have been using them to redesign their products and services. One of their characteristics is that customers are at the heart of the process of developing products that meet market needs. Additionally, business leaders use it to create tools to improve employee well-being.

1.5.2.2 SCRUM method

SCRUM [14] is one of the main agile project management methods. Its main feature is to focus on people and their interactions, software functions and cooperation with customers. It is a good solution that takes into account the process, complete documentation, contract negotiations and pre-established plans. Supported by the pillars of transparency, verification and adaptation, the SCRUM approach aims to reduce difficulties such as lack of planning, changing requirements and inaccurate scope. In addition, SCRUM relies on rapid, frequent and continuous delivery of working software, continuous cooperation between the team and the company, technical excellence and simplicity.

1.5.2.3 Progress of the SCRUM method

SCRUM is based on iterations called sprints (short-term development with predefined goals). The goal is a part of the requirements repository called the product backlog, which is provided and maintained by the product owner. The repository is made up of functions that are continuously prioritized. Before each sprint, the function with the highest priority enters the sprint backlog, and therefore becomes the objective to be achieved during the sprint (iteration). The sprint always starts with the plan, starting with the discussion between the product owner and the scrum master. After the meeting, the task is confirmed and the sprint can begin.

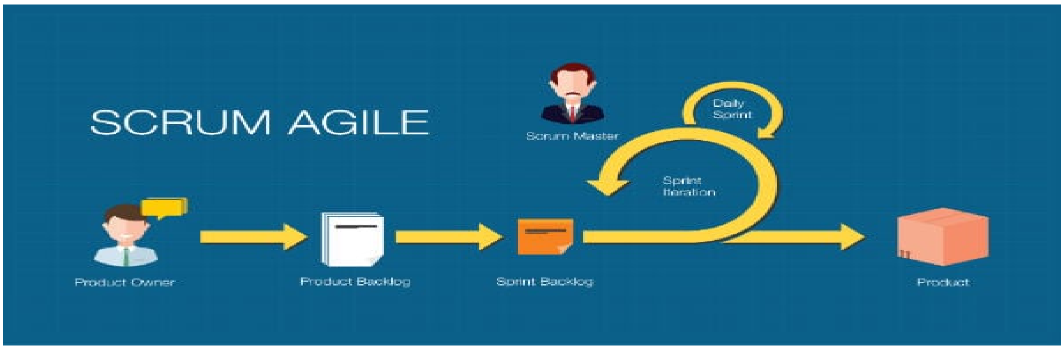


Figure 1.6: SCRUM process

The development team is led by a Scrum Master whose goal is to solve obstacles, participate in development if necessary and put everything in place. Implement to ensure the goal is achieved during the sprint.

Each sprint increases the added value of the product by adding new features that can be delivered to customers.

SCRUM roles:

- SCRUM team: a team of up to 10 people. The team is self-organized and everyone takes over the tasks of another member.
- Product owner: the business correspondent, who represents the company by managing the priority and development of the product backlog. He must allow himself to answer questions from the Scrum team.
- Scrum Master: he combines several roles: coach, project manager, coordinator... He must carry out the Scrum ritual in his team and ensure that they are not hindered by external facts while ensuring reports.

1.6 Gantt Chart

Every successful project needs a good planning. On that note, Figure 1.7 shows our Gantt chart for the entirety of our internship.

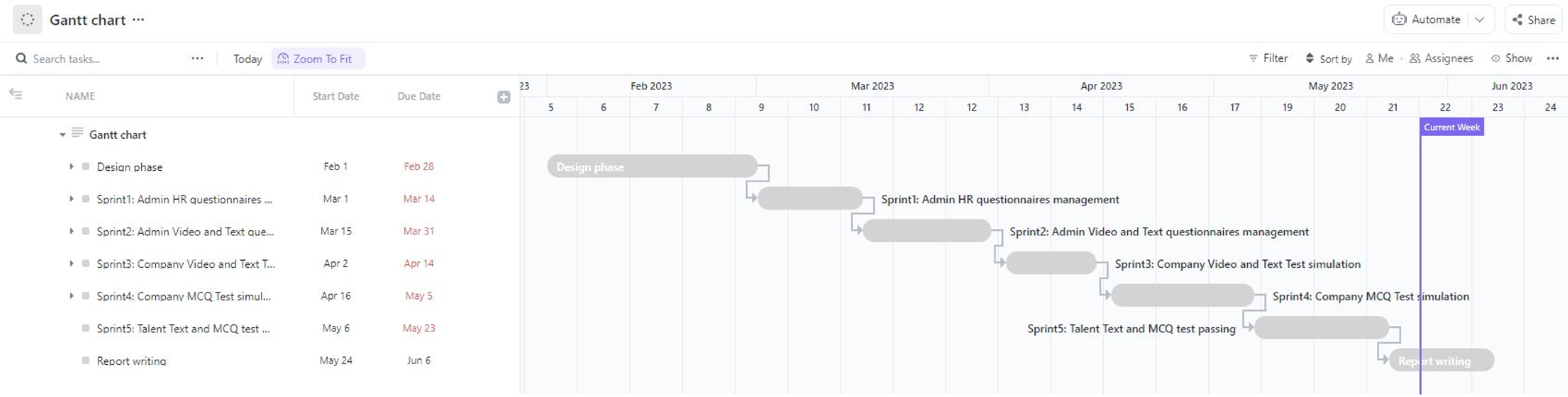


Figure 1.7: Gantt chart

1.7 Conclusion

In this chapter, we started with the description of the general framework of the project where we introduced the host company CleverTech. Then, we mentioned the problem through the study of the existing. Subsequently, we proposed the solution, following the existing tools on the market. Next, we defined the methodology chosen for the work throughout the project and finally we showed our Gantt chart to make our project plan clearer and understandable. In the next chapter, we will present the requirements of our MVP(Minimum Viable Product) solution. We will also identify the actors and the functional and non-functional needs of the platform.

Chapter 2

Analysis and specification of needs

2.1 Introduction

The analysis of requirements and the specification of needs is an essential step in the development cycle of every software. After presenting the general context, we devote this chapter to the analysis of the project. We will start by identifying the different actors who will interact with the system for the **«Questionnaires and Tests management modules»**. Then we will list the functional and non-functional needs of our scope. In addition to that, we will represent the functional requirements in a semi-formal way using use case diagrams. Finally, we will show the product Backlog since we are using SCRUM as a methodology.

2.2 Specification of needs

In this part, we are interested in the phase of analysis and specification of the needs. We will first start by identifying the actors for our scope of the project. Later, we will list and detail all the functional and non-functional requirements on which we will work to adapt them to our platform in order to satisfy the needs of our users.

2.2.1 Identification of actors

An actor is the idealization of a role played by an external person or process that interacts with a system, it can be either a main actor or a secondary actor. An actor is qualified as a principal for a use case when this case serves this actor. The other actors are then qualified as secondary. We present all the actors in Table 2.1 in the next page.

Table 2.1: Table of System actors

Actor	Role
Visitor	It is the visitor of the platform, who can create its own talent account or send a request to either create a company account or a university one.
Talent	This actor can be an alumni, a student. He is the person who seeks jobs or internships opportunities, applying for jobs applications, take tests proposed by companies if necessary.
Company Administrator	He is the person who manages a company. He can manage his company by publishing job offers, processing applications, simulating and creating all kinds of tests to recruit the right profiles.
Falumni Administrator	He is the administrator of the platform, he has all the permissions and all the access rights. He can manage companies, universities, accounts, requests, etc. Also, he can fill the questions database to help companies choose the right questions for their tests

2.2.2 Functional needs

The application to be created must offer a set of functionalities which must be related to a set of user needs. These needs define the services that will be provided by this application. We will present the functional needs by actor as follows:

2.2.2.1 Functional needs of the visitor

The visitor can:

- **Request a company account creation** : it is the manager of a company who can send a request to create his own account in order to benefit from the functionalities provided by the platform concerning recruitment.

- **Create a talent account** : the visitor can create his own talent account on the platform to benefit from the provided features.

Note: each of the following actors must authenticate in order to access their space and use the provided operations.

2.2.2.2 Functional needs of the talent

To make the Questionnaire and tests module work, the platform should allow the talent to:

- **Apply for a job offer** : the talent can apply and send the necessary documents and elements according to the offer.
- **Manage account settings**: everyone can consult and modify their configurations: those of notifications, security and confidentiality and job offer preferences.
- **Pass tests**: if a talent is to be accepted in a job or internship offer he should pass a test(technical, video and MCQ) proposed to him by the company which he applied to.
- **Test devices**: before starting the video test, the talent should be able to test his peripherals(camera and microphone) and choose the right peripheral if he has more than one.

2.2.2.3 Functional needs of the company administrator

The company administrator, who plays the role of its manager, has the right to access the following functionalities:

- **Manage questionnaires**: the company administrator can edit all data about the questionnaires he made for a specific test(add questions, consult the list of questions, delete questions ect.).
- **Simulate test**: the company manager can create a test that contains technical, language and human resources questions.
- **Consult the questions list proposed by FALUMNI**: while simulating a test, the company administrator can consult the questions list chosen by our platform and use them to create his test and questionnaires.

- **Check the list of questions in a questionnaire**: before submitting the test that he is about to create, the company administrator can check his list of questions by questionnaire type and the total duration of the test.
- **Consult the score of a completed test**: when the talent pass the test given to him by the company, the company manager can see his score during this test and decides whether to recruit him or not.

2.2.2.4 Functional needs of the Falumni administrator

This administrator, who manages the back office, has full access rights to all features. He also has the possibility of:

- **Manage questions**: he can create, consult, modify and delete any question he created whether it is a human resources, a technical or a language one.
- **Manage propositions**: for the multiple choices questions case, the administrator can modify the propositions and whether or not that proposition is the right one or not.
- **Manage questionnaires**: like the company administrator, the Falumni administrator can manage his questionnaires that will be proposed to the company administrator at a later time.
- **Manage skills**: the Falumni administrator can create technical skills and language skill with mentioning their level and create questionnaires based on a specific skill and level.
- **Manage requests**: the Falumni administrator has the power to browse the requests, sort them by date etc. He can decide whether to accept or deny the company account or university account requests.

2.2.3 Non-functional needs

A non-functional need is a requirement that characterizes a desired property (quality) of the system. Our solution must meet the following criteria:

- **Validity**: the produced product must satisfy its functionalities; that exactly meets the needs of users.
- **Security and confidentiality**: this property is one of the most important criteria that we will ensure by assigning the list of permissions and access rights to the different spaces according to the type of account that the user has. Also, the

confidentiality of our users' data is ensured by protecting the visibility of each subscriber's sensitive information.

- **Performance**: our solution must be efficient in terms of response time (when loading data, browsing between interfaces, updating information and data in real time, etc.). We will guarantee it by creating a single page application (SPA: Single Page Application).

- **Maintainability and extensibility**: the decomposition that we are going to apply to our application can help us to quickly adapt our application to new requirements. This will allow us to identify and easily solve the problems of the modules.

- **Reuse**: thanks to the technologies used and the architecture of our application, which we present in the following chapters, we can use and apply any module, component, style, functions, etc. on other projects or on the same project.

- **Evolution**: the platform team must listen to new demands of its users' needs in order to satisfy them.

- **Ease of use**: consists of offering the end user a good UX (User eXperience). Clear interfaces, easy to understand and user-friendly to navigate between the pages so that the latter can easily navigate and use the functionalities of our platform.

2.3 Use case diagrams

2.3.1 Global use case diagram

At this level, we can represent the different use cases in a general way by the users of the application. The different use cases are shown in Figure 2.1 in the next page.

Figure 2.1: Global use case diagram

2.3.2 Use case refinement

In this part, we will describe in detail the use cases that we have previously identified in order to obtain a precise expression of the needs.

Use case refinement : Pass tests

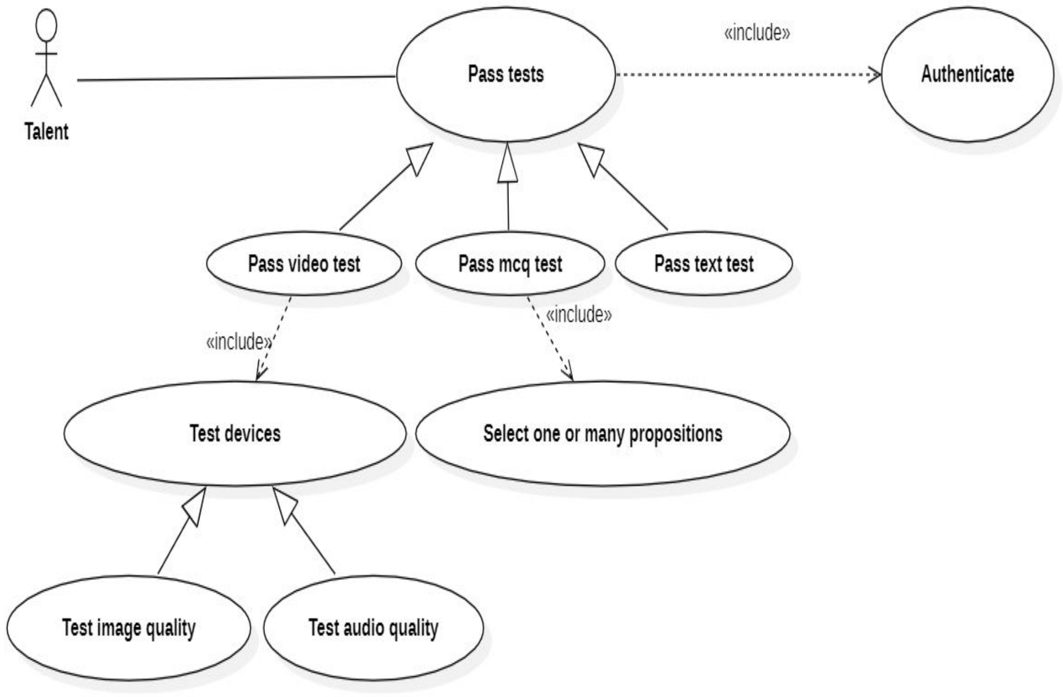


Figure 2.2: Pass tests use case diagram

Table 2.2: Use Case : Pass tests

Use case : Pass tests
Brief description : Pass the tests proposed by the company administrator.
Actor : Talent
Pre-conditions : Talent should be authenticated and a test should exist.
Post-conditions : The post-conditions are given according to what test the talent has at the moment: - Pass video test : Prepare the peripherals and answer in video. - Pass mcq test : Pick the right answer or answers for a question. - Pass text test : Write the response for a question.

Nominal scenario : 1. After authentication, the talent navigates to his profile and clicks on tests. 2. A list of tests prepared by the companies he applied to shows up. 3. The talent selects the wanted test and clicks on "Take test" button. 4. A confirmation pop-up shows up for the talent. 5. Talent clicks the "Confirm" button located in the pop-up. 6. Once the button clicked, the pop-up closes and the talent gets redirected to the test interface. 7. The talent begins his test and starts by answering the questions he has with respecting the time limit for each question. 8. In case he has a video test, the talent can test his peripherals and the platform offers him a mock question so that he knows the process of the test. 9. The talent finishes all the questions within the time limit. 10. The talent gets redirected back to his profile and gets notified in a later time by the response of the company manager.
Alternative scenario : After the 1st step: 1. There is a possibility that no companies prepared a test for the talent yet, so he should just wait for one. In the 7th step, the Talent can: 1. Click the "Leave Test" button and gets redirected to his profile with showing an alert telling him that he failed the test and can't take it another time.

Use case refinement : Manage questionnaires

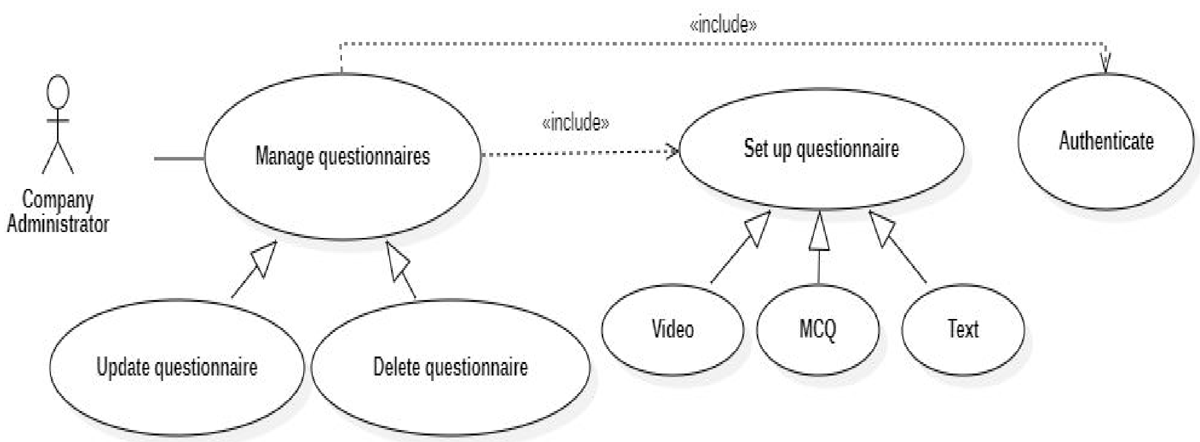


Figure 2.3: Manage questionnaires use case diagram

Table 2.3: Use Case : Manage questionnaires

Use case : Manage questionnaires
Brief description : Prepare the questionnaires for the test given to the Talent.
Actor : Company Administrator
Pre-conditions : Company administrator should be authenticated, a job offer should exist and a talent should apply for that job offer.
Post-conditions : The post-conditions are given according to the choice (the use case) of the user: - Video questionnaire : Test contains only video questions. - MCQ questionnaire : Test contains only mcq questions. - Text questionnaire : Test contains only text questions.
Nominal scenario : 1. After authentication, the company administrator navigates to a specific job offer. 2. A list of candidates appears. 3. The company administrator selects the wanted candidate and clicks on "Simulate test" button. 4. The company administrator gets redirected to an interface where he can choose the type of questionnaires present in his test(Video, MCQ, Text or all of them) 5. Once he made his choice, the company administrator clicks on the "Start" button and gets redirected to the test simulation interface which contains a stepper based on what questionnaire type he chose. 6. The company administrator then has the choice to start filling his questions based on the questionnaire type. 7. The company administrator can pick one or more questions from the questions list proposed by the Falumni platform based on the questionnaire type. 8. The company administrator clicks on "Create question" if he wants to make his own question. 9. For the technical and language questions for the video and text questionnaires, the company administrator can pick a question from the Falumni database only when he filters the questions based on their skill and level.

10. For the MCQ questionnaire case, the company administrator can't add his own question therefore he can only choose from the questions provided by the platform. 11. Once the company administrator fills his questionnaires with questions, he clicks on the "Verify" button. 12. A pop-up shows up containing how many questions he made based on the questionnaire type and the total duration of the test he made. 13. The company administrator then clicks on "Create test" button in the bottom of the pop-up. 14. The pop-up closes and he gets redirected to his job offer again and waits for the talent to take the proposed test. 15. When the talent finishes his test, the company administrator can download the score file and makes the decision about accepting or denying the candidate.
Alternative scenario : After the 12th step, the company administrator can: 1. Click the cross mark button located on the top right section of the pop-up to add, delete or even update one or more questions and then repeat the 11th step.

Use case refinement : Create question

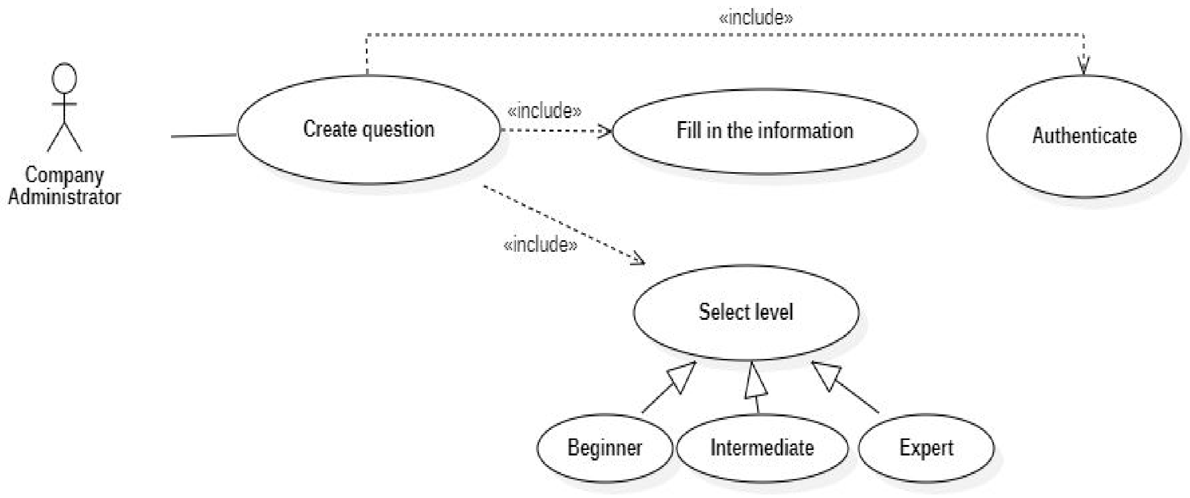


Figure 2.4: Create question use case diagram

Table 2.4: Use Case : Create question

Use case : Create question
Brief description : Create a question for a questionnaire in the test simulation phase.
Actor : Company Administrator

Pre-conditions : Company administrator should be in the Test simulation phase.
Post-conditions : Question created and assigned to a questionnaire.
Nominal scenario : 1. As mentioned before in Table 2.3, in the test simulation interface the company administrator can create his own technical, language or human resources question except for the MCQ type. 2. The company administrator clicks on the "Create question button". 3. A pop-up shows up containing the fields that the company administrator should fill. 4. If it is a human resources question then the company administrator mentions the question content and the duration for the question. 5. If it is a technical or language question, aside from the content and duration, the company administrator should select the skill and its level. 6. The company administrator clicks the "Create" button. 7. The pop-up closes and the question gets added to the right questionnaire.
Alternative scenario : /

Use case refinement : Manage questions

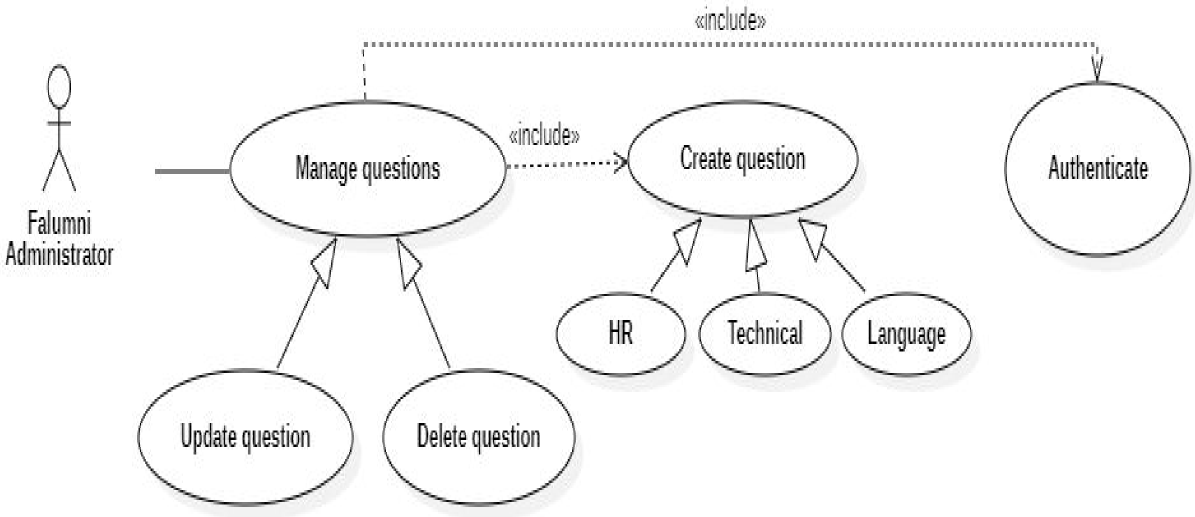


Figure 2.5: Manage questions use case diagram

Table 2.5: Use Case : Manage questions

Use case : Manage questions
Brief description : Prepare questions for the company administrator to choose from.
Actor : Falumni Administrator

Pre-conditions : Falumni Administrator should be authenticated.
Post-conditions : The post-conditions are given according to the choice (the use case) of the user: - HR : Human Resources question created. - Technical : Technical question created. - Language : Language question created.
Nominal scenario : **Human resources questions** 1. After authenticating, the Falumni administrator clicks on "Questions" located in the navigation bar. 2. The questions interface shows up and the Falumni administrator clicks on "Human Resources" located in the sub navigation bar. 3. A list of all the human resources appears for the Falumni administrator to consult. 4. The Falumni administrator clicks the "Add your question" button. 5. The Falumni administrator finds a toggle button for the in bulk question creation. 6. The Falumni administrator clicks the toggle button and has the option to download a csv template, fill it with the questions he wants and then upload it to the platform. 7. Once the file is uploaded, the questions newly added appear. 8. The Falumni administrator clicks the "Save" button and all the questions he has added got saved successfully.
Alternative scenario : After the 5th step, the Falumni administrator can: 1. Instead of clicking the toggle button for the in bulk creation, he chooses not to and therefore he is gonna create just one question. 2. Fill in the available field(question content) and click the "Save" button and the question is gonna be added to the human resources questions list.

Use case refinement : Manage skills

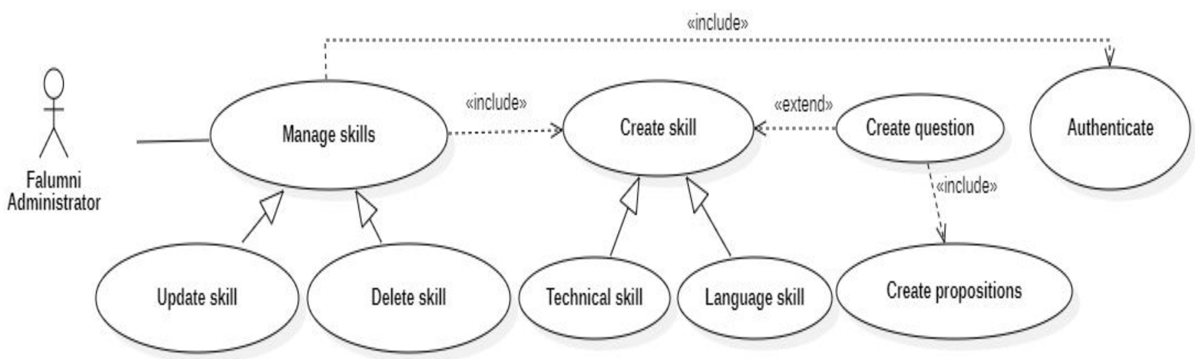


Figure 2.6: Manage skills use case diagram

Table 2.6: Use Case : Manage skills

Use case : Manage skills
Brief description : Create a language or technical skill with its assigned questions.
Actor : Falumni Administrator
Pre-conditions : Falumni Administrator should be authenticated.
Post-conditions : The post-conditions are given according to the choice (the use case) of the user: - Technical skill : Technical skill with its specific level created. - Language skill : Language skill with its level created.
Nominal scenario : **Technical skills for mcq questions** 1. After authenticating, the Falumni administrator clicks on "Questions" located in the navigation bar. 2. The questions interface shows up and the Falumni administrator clicks on "Technical" located in the sub navigation bar. 3. The Falumni administrator clicks on the mcq icon button. 4. A list of all the technical skills and levels that are for the multiple choices questions type appear for the Falumni administrator to consult. 5. The Falumni administrator clicks the "Add your skill" button. 6. A pop-up shows up before the Falumni administrator. He fills in the skill name and level and clicks the "Save" button. 7. The pop-up closes and a new skill has been created and listed like the others. 8. The Falumni administrator clicks the "show questionnaire" button that is located in every skills table cell that redirects you to the questionnaire assigned to that specific skill. 9. The Falumni administrator gets redirected to the questionnaire interface linked to the skill he just added. 10. The Falumni administrator clicks the "Add question" button. 11. A pop-up shows up. Since it is a mcq question, the Falumni administrator fills in the question content and 6 propositions with checking the right answers for that question and clicks the "Save" button. 12. The pop-up closes and a new question gets added. The administrator can consult the question with its propositions. **Note:** this process or scenario is the same for the case of a Language skill for mcq questions.

Alternative scenario :
After the 6th step, if the Falumni administrator pressed the "Save" button with an existing skill name and level then an alert will appear telling him that the skill already exists

2.4 Product Backlog

The table below shows the Product Backlog of our project.

Table 2.7: Product Backlog

Product Backlog		
Team	Khalil Chaftar	
Date	01/02/2022	
Release	Back office(Admin Space)	
Sprint	Features	User story
1	Consult HR questionnaires	• As an administrator, I can see the list of all HR questions created.
	Manage HR questionnaires	• As an administrator, I can add my own question. • As an administrator, I can edit and/or delete existing questions. • As an administrator, I can download an Excel template to write my questions. • As an administrator, I can add a list of questions through an Excel document.

2	Manage technical and language skills	• As an administrator, I can see the list of skills as well as the level and number of questions in a questionnaire of each skill. • As an administrator, I can create a skill to assign it to a questionnaire. • As an administrator, I can delete a skill.
	Manage the questionnaires assigned to skills	• As an administrator, I can see the questions of a questionnaire. • As an administrator, I can create a text, video or an mcq question depending on the skill created. • As an administrator, I can modify and/or delete questions from a specific questionnaire.
	Manage the MCQ questions and propositions	• As an administrator, in the case of adding an mcq question, I can write the question and 6 propositions with checking the right proposition. • As an administrator, I can see the propositions for an mcq question. • As an administrator, I can modify and/or delete propositions.
Release	Company	
Sprint	Features	User story
1	Choose questionnaires types	• As a company administrator, I can choose to create one or all types of questionnaires (video, multiple choice, text).

	Simulate video and text test	• As a company administrator, I can add my own question by choosing a skill and a level as well as the duration of the answer of the proposed question. • As a company administrator, I can modify the details and/or delete a question that I have already created. • As a company administrator, I can choose one or more questions offered from the FALUMNI database according to their skill and level. • As a company administrator, I can see the skill and level assigned to each question in my questionnaire. • As company administrator, I can see the total number of questions and the total duration of the test.
2	Simulate MCQ test	• As company administrator, I can see the total number of skills and the total number of questions. • As a company administrator, I don't have to create MCQ questions, I can just choose 10,20 or 30 questions offered by FALUMNI for each technical or language skill.
Release	Talent	
Sprint	Features	User story
1	Pass the MCQ and text test	• As a talent, I must fill in the fields corresponding to the answer for the question asked. • As a talent, I must respect the given time for a question. • As a talent, I need to know the number of tests I will take as well as the number of questions for each test. • As a talent, I must choose one or the correct answers from the MCQs to pass the test. • As a talent, I can leave the questionnaire.

2	Test peripherals before starting the video test	• As a talent, I need to check that my peripherals are working properly. • As a talent, I must test my equipment on the platform before starting the test.
	Pass the mock question	• As a talent, I have to pass the mock video question to know how the test will be. • As a talent, I need to know the steps needed to finalize the test.
	Pass the video test	• As a talent, I have to answer the offered test questions in video format. • As a talent, I can leave the test. • As a talent, I cannot apply for a specific offer again if I voluntarily leave the registration.

2.5 Conclusion

This chapter allowed us to present the project well and have a clearer vision on the subject. We started by identifying the actors and enumerating the functional and non-functional that needed to be present in our platform so that our project scope works perfectly. Then, we presented the global use case diagram and the detailed use case diagrams. Finally, we mentioned our Product Backlog to give a better explanation about what we did in our internship period. In the next chapter, we will identify the concept study of the project.

Chapter 3

Concept study

3.1 Introduction

We took a lot of time to carry out the analysis and specify the needs in the previous chapter. In fact, this allowed us to start the design phase with a fairly clear vision of the main functions to be implemented. In this chapter, we will present the design method used as well as the related diagrams. This phase will allow us to set up a model on which we will rely on during the implementation of the solution.

3.2 Static view

A static view makes it possible to represent the model structure without taking into account the evolution over time. This view is represented by diagrams, which define the static design opinion (use case diagrams, class diagrams and object diagrams). In the next sub section we will present the global class diagram as well as the classes' description.

3.2.1 Class diagram

The class diagram makes it possible to identify the different structures of a system with the different attributes and methods of each class. While the use case diagram shows a system of actors' point of view, the class diagram shows the internal structure. It provides an abstract representation of the system objects that will interact with each other in order to achieve the wanted use cases. It is important to note that the same object could very well intervene in the realization of several use cases. Therefore, the use cases do not realize a partition of the classes of the class diagram. A class diagram is therefore not suitable (except in a particular case) to detail, break down, or illustrate the realization of a particular use case. Figure 3.1 represents the global class diagram of our application.

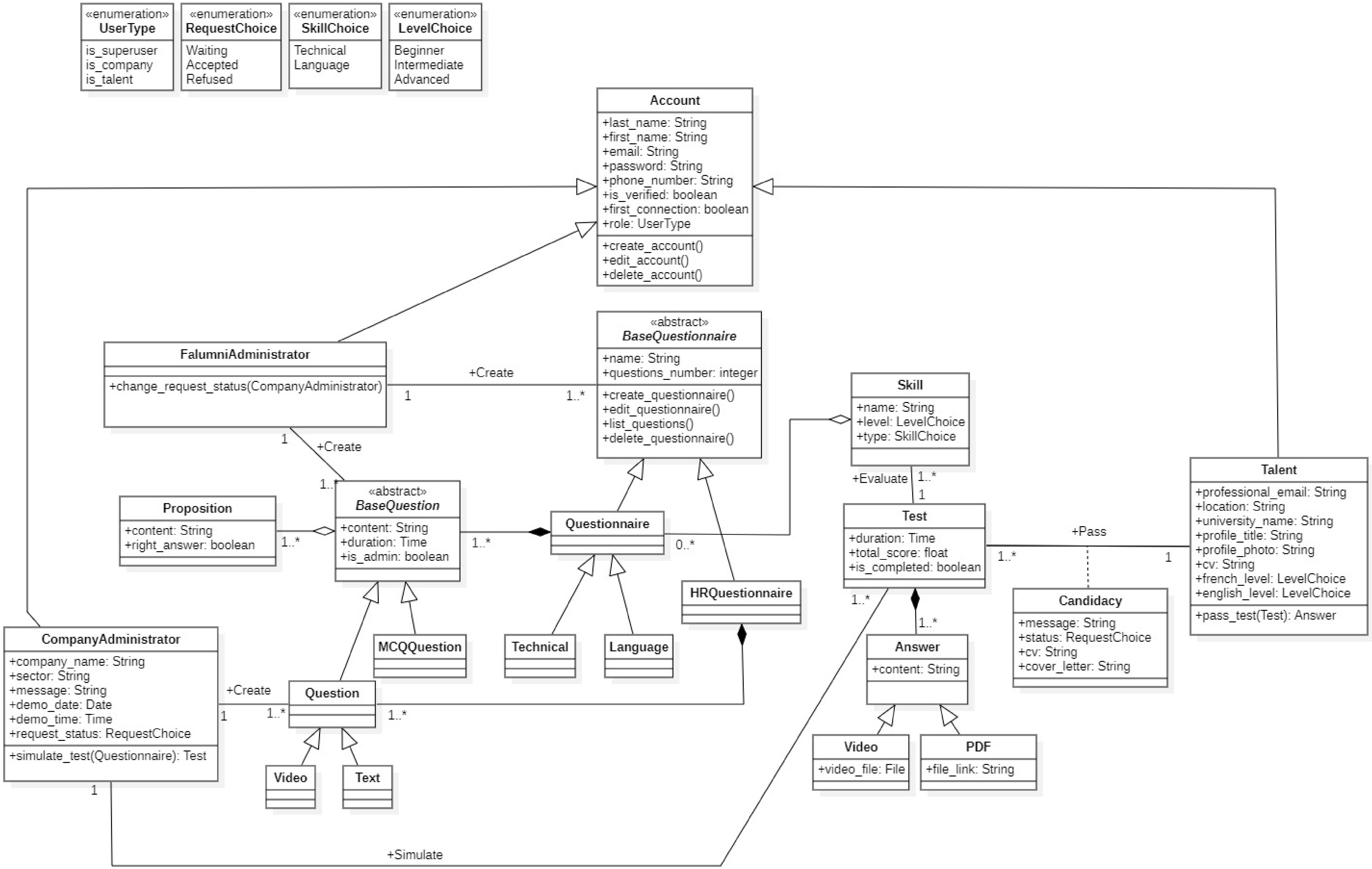


Figure 3.1: Global class diagram

3.2.2 Classes description

A description of each class is identified in Table 3.1 containing the name of the class with a brief description that shows the general aspect of this class and its position in relation to the others.

Table 3.1: Classes description

Class	Description
Account	The Account class presents the created entity by the users so that they can access the platform.
CompanyAdministrator	This class inherits from Account and presents the company administrator's account.
FalumniAdministrator	This class inherits from Account and presents the platform administrator's account.
Talent	This class inherits from Account and presents the talent's account.
Candidacy	This class presents the participation of a talent in a job offer posted by a certain company.
BaseQuestionnaire	This class is an abstract class and is the class mother of all our questionnaires types.
Questionnaire	This class inherits from BaseQuestionnaire and presents the skeleton for the technical and language questionnaires.

HRQuestionnaire	This class inherits from BaseQuestionnaire and presents the human resources questionnaire.
BaseQuestion	This class is an abstract class and is the class mother of all our questions types.
MCQQuestion	This class inherits from BaseQuestion and presents the multiple choices question. We made this a seperate class from the Question class because we don't want the MCQQuestion to be in a HRQuestionnaire.
Question	This class inherits from BaseQuestion and presents the skeleton for the video and text questions.
Proposition	As its name suggests, this class presents the choice or answer for a certain question.
Skill	This class presents the skills(technical or language) to be evaluated in a test.
Test	This class presents the test simulated and created by the company administrator for a talent to pass.
Answer	This class presents the answers the talent will make when passing a test.
Video	This class inherits from Answer and presents the video answer made by the talent.

PDF	This class inherits from Answer and presents the mcq and text answers made by the talent in a PDF format.

3.3 Dynamic view

A dynamic view models the dynamic aspect of the system. Generally, an interaction diagram corresponds to a use case. The essential elements of an interaction diagram are objects, actors and messages. The actions between objects and actors are mainly: sending messages, creating and destroying objects. In the next sub section, we will present the essential sequence diagrams of our application.

3.3.1 Sequence diagram

The sequence diagram is the graphical representation of the interactions between the actors and the system according to a chronological order in the UML formulation.

Sequence diagram : Company account creation

The sequence diagram presented in Figure 3.2 shows the steps of creating a company account. The visitor fills in the form by following the steps. The user controller then verifies the entered data. If the entered data are valid, the controller will save the account in the user entity. The user entity sends a request to the platform administrator to finalize the account creation. If the administrator accepts the account creation request then a company account will be created and send an acceptance email containing the first connection link.

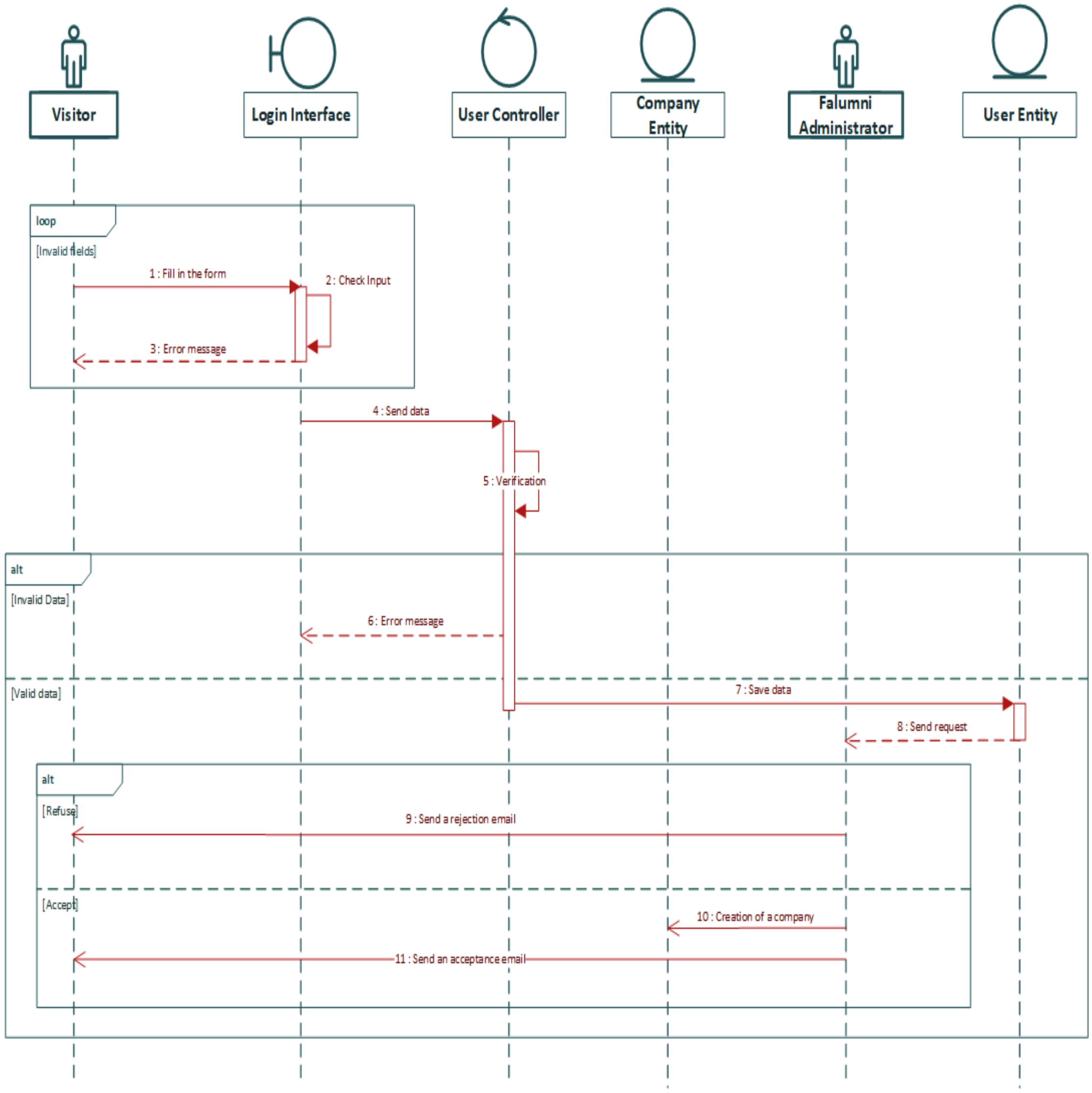


Figure 3.2: Sequence diagram : Company account creation

Sequence diagram : Authentication

The sequence diagram presented in Figure 3.3 shows the user authentication in steps. The visitor enters the email and password. The user controller will therefore check the data entered: if the data is compliant with the data stored in the database then the controller will request the creation of a JWT service token and then a redirection will happen based on the user type.

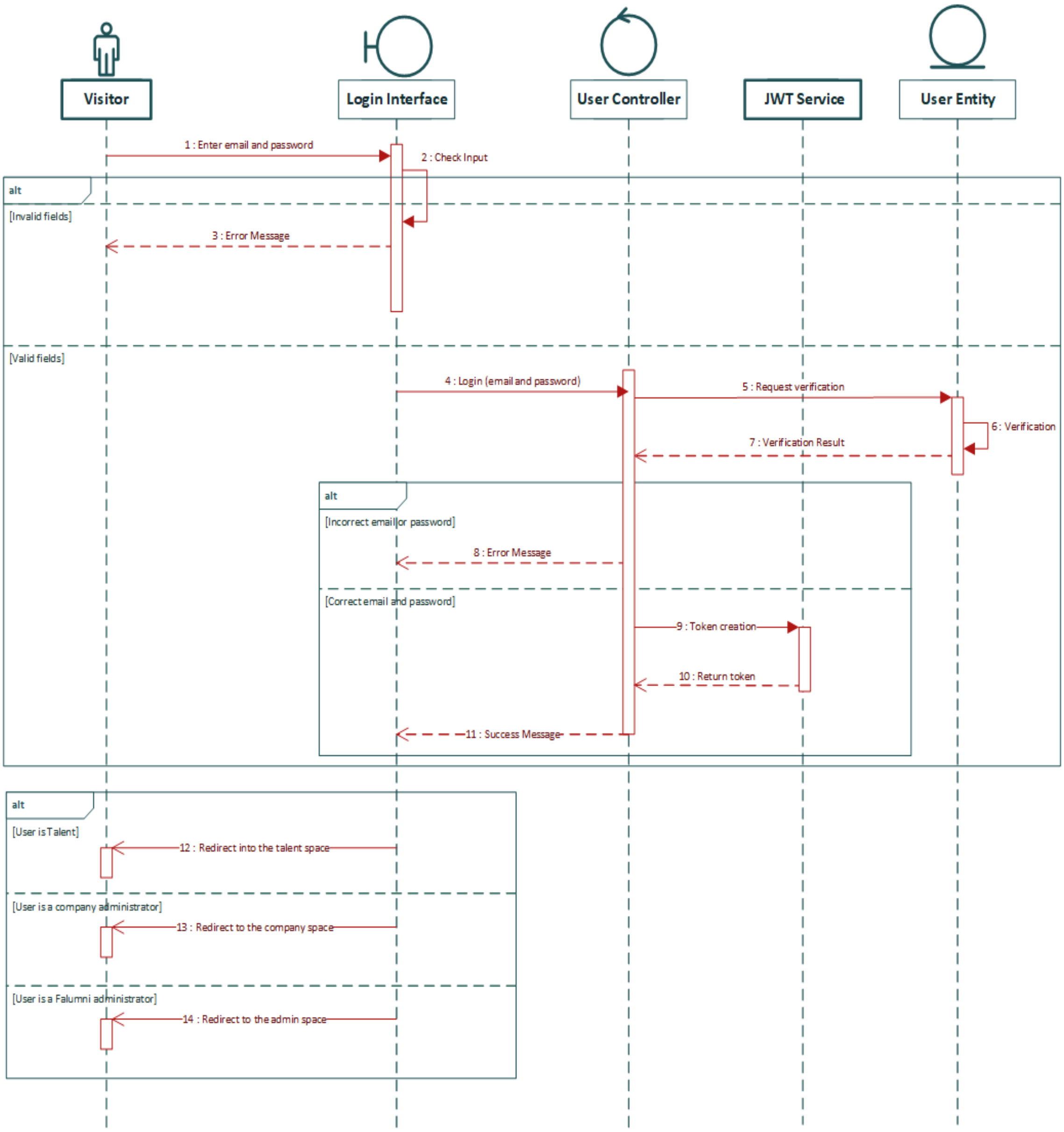


Figure 3.3: Sequence diagram : Authentication

Sequence diagram : Human resources question creation

Adding a HR question to our application follows a specific sequence of events shown in Figure 3.4. This sequence begins with the authentication, the Falumni administrator will be redirected to his space. By clicking on “Questions” button in the navigation bar and then clicking the "Human Resources" in the sub navigation bar, he can see his questions list. When he clicks on “Add your question”, an interface containing a toggle button appears. If he selects "Add by question" a form field appears for him to write his HR question content with respecting the character number limit before clicking the "Save" button or else he will receive an error message. Once the condition is respected and the "Save" button is pressed, the data will be sent to the Question controller which will assign the data to the human resources questionnaire and save it and the Falumni administrator then can see then added question in the questions list. In the case where he chooses the "Mass creation"(question creation in bulk), the administrator can download an Excel template to fill all the questions he wants and then upload them to the platform. the data will be sent to the controller to be assigned to questionnaire and saved with a success message received by the administrator.

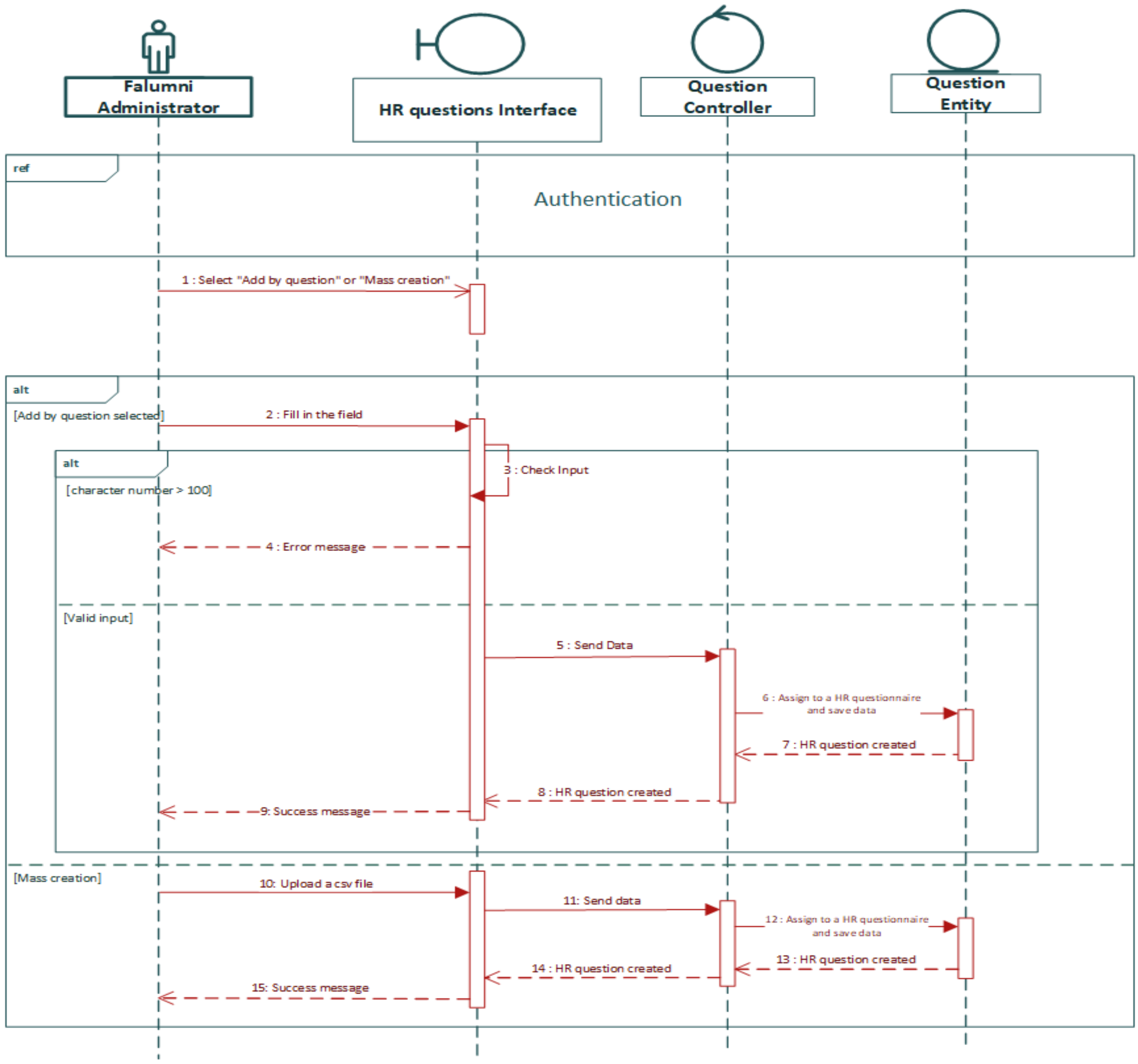


Figure 3.4: Sequence diagram : Human resources question creation

Sequence diagram : Technical skill creation

Figure 3.5 presents the sequence diagram that describes the “Create a technical skill” scenario. The Falumni administrator should be authenticated and redirected to his space. By clicking on "Questions" located in the navigation bar and then selecting "Technical" located in the sub navigation bar, the admin can choose on the top left section whether he wants the skill to be created in a video, a text or an mcq test. Once he chose that, he then clicks the "Create your skill" button and a pop-up shows up containing a form to be filled. He writes the skill name and chooses the skill level and then click "Save". The data then gets sent to the Skill controller, the controller will then create the skill and the admin can see his newly added skill in the skills list.

Note: the language skill creation follows the same process.

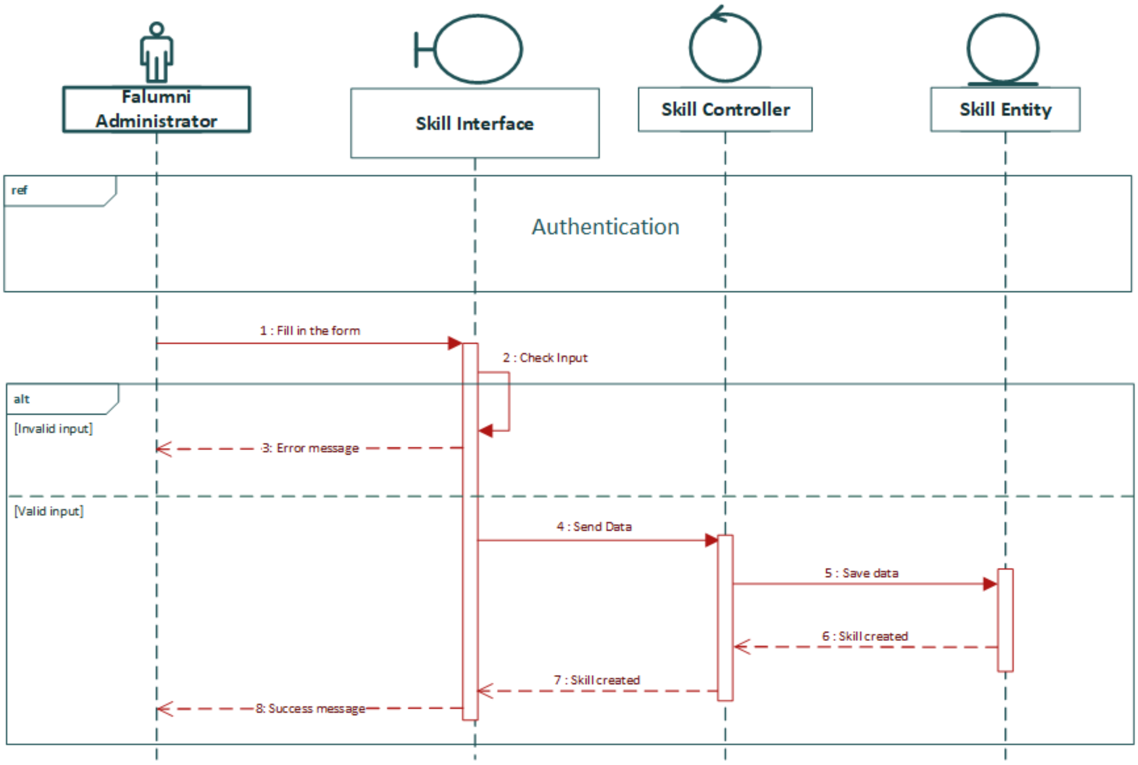


Figure 3.5: Sequence diagram : Technical Skill creation

Sequence diagram : MCQ question creation

Creating an mcq question will require a chain of steps as shown in Figure 3.6. The Falumni administrator should be authenticated. If he wants to create an mcq question for a technical skill then a skill should exist. If no skills exist, he should create one. The same thing goes for the case where he wants to create a question for a language skill. Once that done, the admin clicks on the "View questionnaire" button located in the skill table cell, he gets redirected to the MCQ questions interface. He then clicks on "Add your question", a pop-up shows up for him to fill the question field and the six propositions for that question. Then he clicks the "save" button. If the data is valid, it gets sent to the controller which will assign it to the targeted questionnaire and creates the question. And by then, the admin receives a success message and can see his question and propositions in his questionnaire.

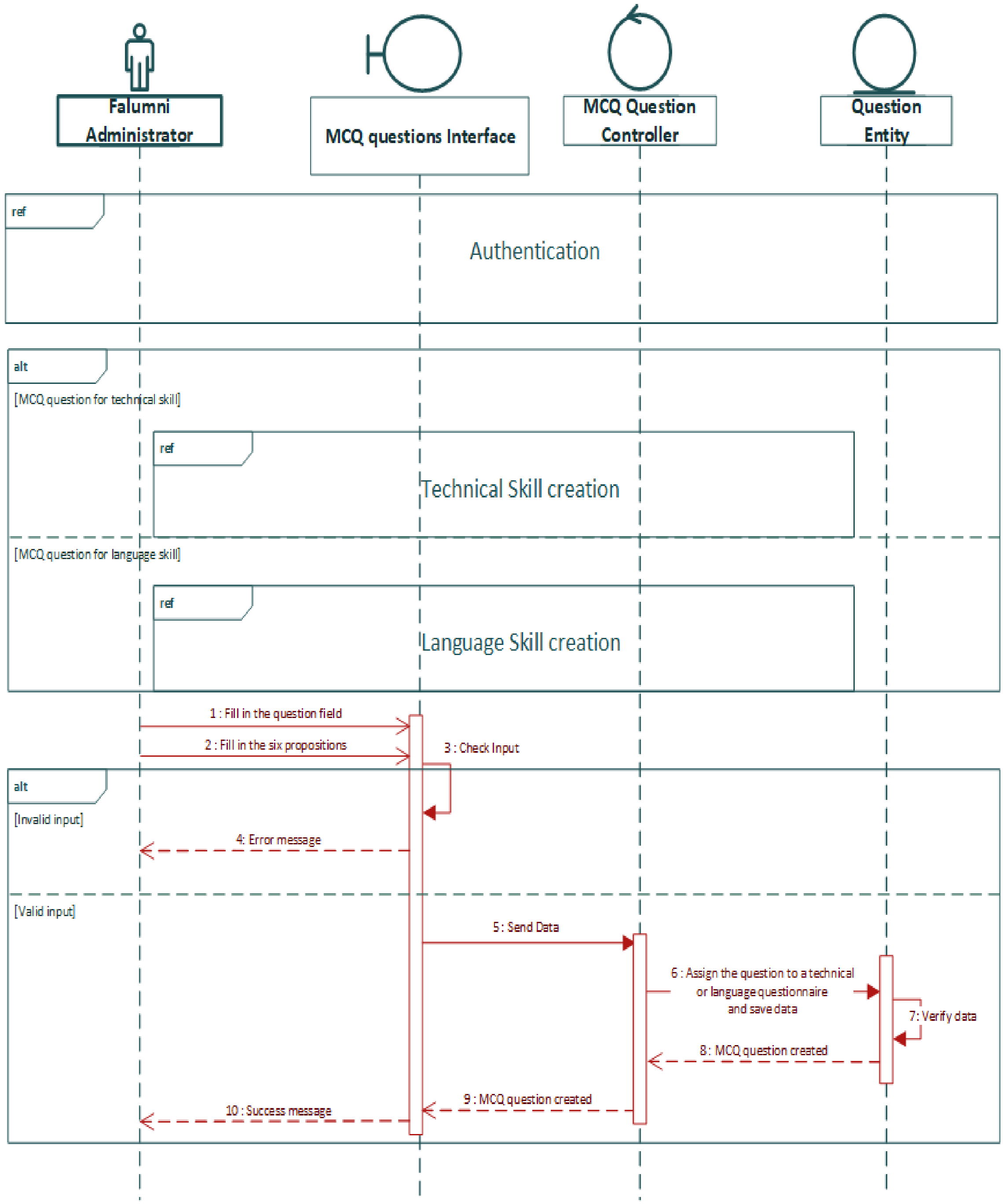


Figure 3.6: Sequence diagram : MCQ question creation

Sequence diagram : Test simulation

Simulating a test for the talent requires the steps shown in Figure 3.7. The company administrator should be authenticated and redirected to his space. A published job offer that he made appears for him to check if there are any candidates that applied for that offer. If a candidacy exist then he has the choice to click on the "Simulate test" button. Once the button pressed he gets redirected to the test simulation interface where he should choose a test type to simulate(video, text, MCQ or all of them). Once the choice is made, he clicks the "Begin" button and gets redirected to a slider interface based on the choice he made. Let's take the example of him choosing all the options. In that case, he can pick what questionnaire type he wants to fill first, if he started by the video questionnaire he then picks some HR questions from the Falumni database and create his own questions, the same thing goes for the technical and language questions with the soul difference being that he has to select the skill and skill level for the question he wants to make. And the same process gets repeated for the text questionnaire type, he fills the human resources, the technical and then language questions. For the MCQ questions case and as mentioned in the class description table 3.1, MCQ questions can't be created for HR questionnaires. Also, the company administrator can't create his own MCQ question, he can only choose from the questions created from the Falumni administrator. Once he completed the job of filling and picking his questions, the company administrator then presses the "Finish" button. A pop-up shows up containing the total number of questions he made and the total duration of the test in general. He can either press "Confirm" or closes the pop-up to add more questions. If he pressed "Confirm", the data gets sent to the Test controller that will create the test. Once that done, the company administrator gets redirected back to the job offer interface and wait for the candidate to finish the test.

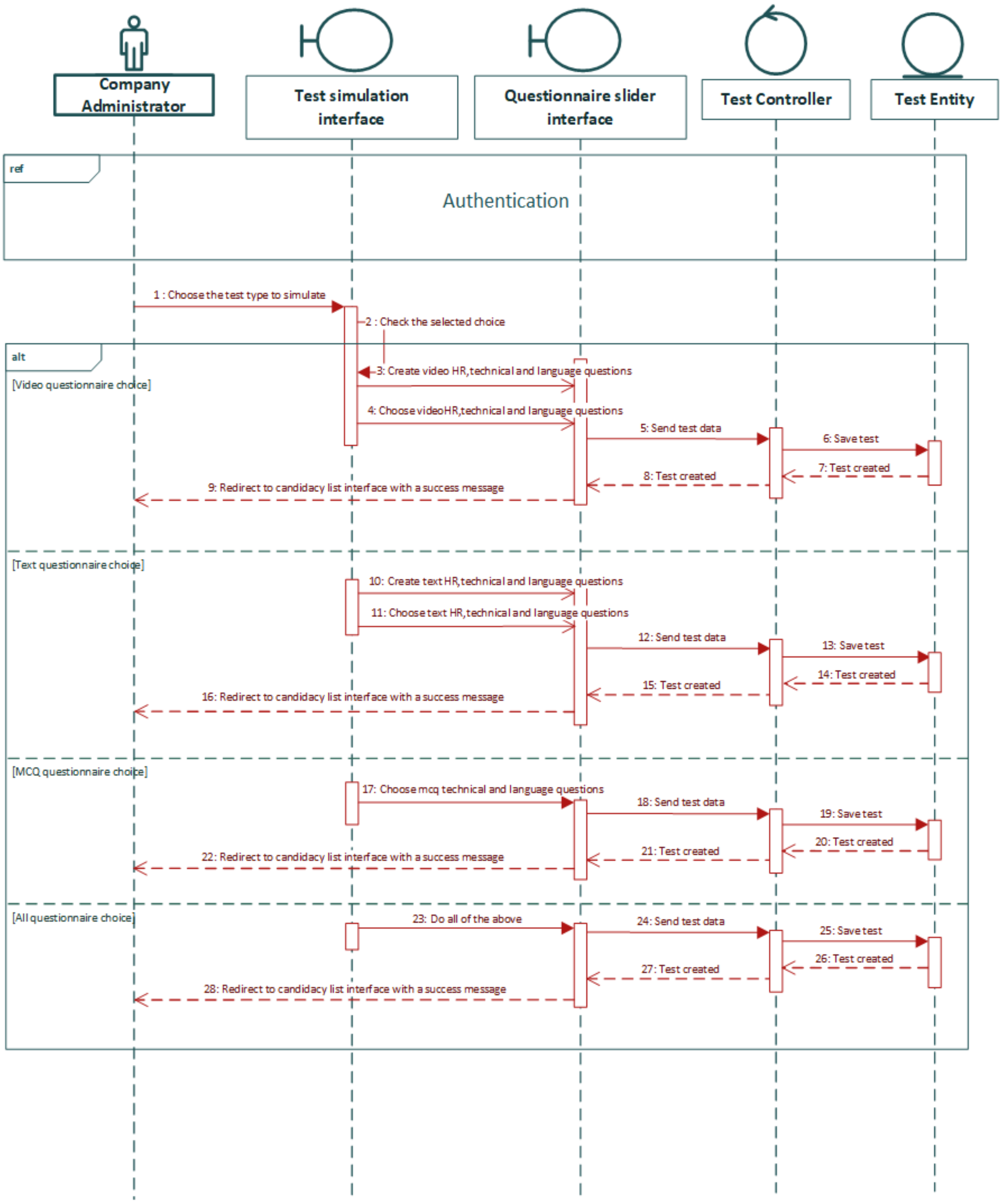


Figure 3.7: Sequence diagram : Test simulation

Sequence diagram : Pass MCQ and text test

After explaining the test simulation scenario, it is time now to mention how the talent pass an MCQ and text test. The steps of this process are mentioned in Figure 3.8. The talent should obviously be authenticated and in his space. He then navigates to his profile and clicks the "Tests" button located on the side bar. A list of tests simulated by companies for job offers that he applied to appears. He clicks the "Pass test" button for a test he chose. He then gets redirected to the test interface to begin his test. Once he starts the test, the talent will encounter 2 types of question, the text question and the mcq question. For the text question he is gonna write a written answer and for the mcq question he is going to pick the right answer. The talent should also respects the time limit or he will fail the test. When he completes the test, the data will be sent to the answers controller to save the answers and generate a PDF file containing the total score for the company admin to check. The talent then gets redirected to his profile with being told to wait for the company admin to check his answers to the test.

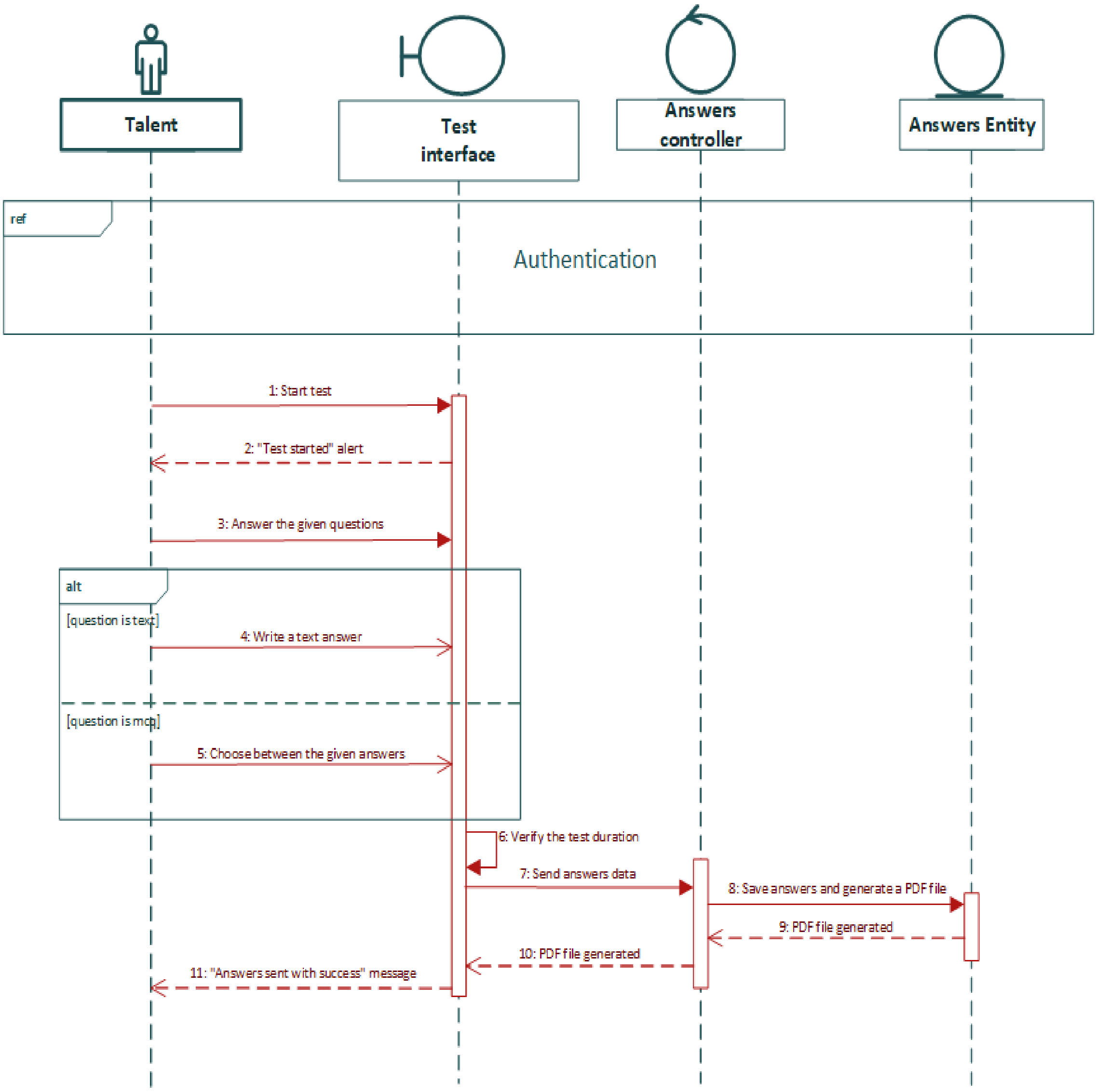


Figure 3.8: Sequence diagram : Test simulation

Sequence diagram : Pass video test

The video test process is explained in Figure 3.9. As mentioned in «Pass text and MCQ test», when the talent clicks the "Pass test" button, he gets redirected to the test interface to take his test. For the case of passing the video test, the platform offers the talent the opportunity to verify his video and sound peripherals. After choosing the right equipment, the platform offers him a mock question in form of a tutorial so that he can understand the test process. When he finishes the tutorial, the talent proceeds then to answering the given questions in a video format while respecting the time limit. When he finishes the test, the data gets sent to the answers controller which will generate a video for the company administrator to see. By then, the talent gets redirected back to his profile and wait for the company's answer.

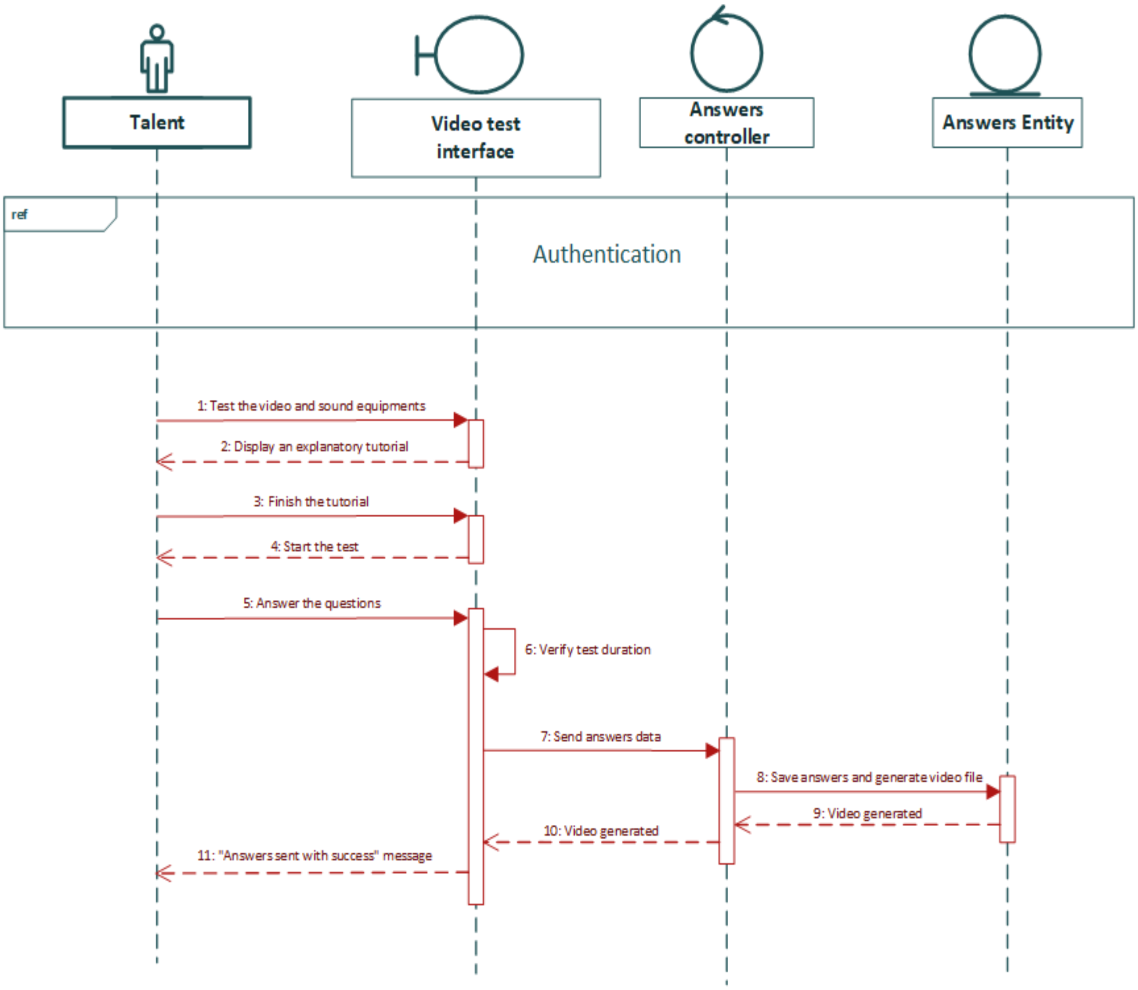


Figure 3.9: Sequence diagram : Pass video test

3.4 Conclusion

In this chapter, we were able through the UML diagrams and the methodological approach to identify the elements and the conceptual modules that will allow us to facilitate the realization and the development of our solution, which will be the subject of the next chapter.

Chapter 4

Implementation

4.1 Introduction

After the design stage of the application, we will describe in this chapter the implementation phase. We will present, first, the work environment created for the development of the application, mentioning the steps for our deployment phase and finally, we will give an overview of the work done through screenshots.

4.2 Development environment

In this section, we specify the tools and technologies used for the design and realization of our project as follows in Table 4.1 :

Table 4.1: Used tools and technologies

Logo	Technology
	For the front-end part, we chose React [12]. React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like django.
	For the back-end, we opted to use the django [6] web framework. django is a Python-based free and open-source framework that follows the model–view–template (MVT) architectural pattern that follows the MVC pattern. In other words, in django, views are called templates and controllers are called views. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.
	Our project uses React in the front end, so it needs to consume APIs for our functionalities to work perfectly. django REST framework [7] is a powerful and flexible toolkit for building Web APIs using django.

PostgreSQL	As a Relational Database Management System, we chose PostgreSQL. It is a highly stable database management system, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL [11] is used as the primary data store or data warehouse for many web, mobile, geo-spatial and analytic applications.
	To make the components development and styling easier, we picked MaterialUI(**MUI**) [10]. It offers a comprehensive suite of UI tools to help shipping new features faster. Material-UI is an open-source project that features React components that implement Google's Material Design.
Redux Toolkit	For the state management of our React application, we used Redux Toolkit(RTK) [13]. It is an open-source JavaScript library for managing and centralizing application state. It includes several utility functions that simplify the most common Redux use cases, including store setup, defining reducers, immutable update logic, and even creating entire "slices" of state at once without writing any action creators or action types by hand.
Xd	For the user experience and interface design part, we used AdobeXD [1] which is a tool used for designing interfaces and User eXperiences(UX) for both web and mobile applications.
ClickUp	For our project management, we have used ClickUp [4]. It is a cloud-based collaboration and project management tool suitable for businesses of all sizes and industries. It allowed us to store our sprints and their status to make the work more agile.

GitHub	For the follow-up control and project versioning management, we chose to work with GitHub [8]. It is an open-source code hosting platform for version control and collaboration based on Git functionalities. GitHub lets you and others work together on projects from anywhere. It revolutionizes the development, collaboration between teams and security management and the CI/CD approach.
Azure DevOps	For the CI/CD approach, we used Azure DevOps [2]. It provides an integrated set of services and tools to manage your software projects, from planning and development through testing and deployment. Azure DevOps delivers services through a client/server model. Azure DevOps Server supports integration with GitHub Enterprise Server repositories.

4.3 Choice of architecture

The architectures used in the context of web applications are multiple. They depend on factors such as application usage and deployment. We will start by mentioning the back-end architecture first and then move to the front-end one.

4.3.1 Back-end architecture

As mentioned in Table 4.1, we chose django as a framework so the architecture we are opting at the level of our solution is the MVT architecture. In fact, the MVT architecture is inspired by the MVC architecture that is widely used in web frameworks. its purpose is to separate the responsibilities of each pole so that every pole focuses on its tasks. It represents an architecture oriented around three poles.

- **Model**: interacts with a database via the ORM.
- **Vue**: receives an HTTP request and returns an appropriate HTTP response (for example if the request is an interaction with a database, the view calls a model to retrieve the requested items).
- **Template** : is an HTML file retrieved by the view and sent to the visitor with the model data.

4.3.1.1 MVT architecture advantages:

- Better code organization.
- A clear and efficient design thanks to the separation of the tasks carried out by the different elements(poles).
- Greater flexibility to organize the development of the website between the different developers.

The Figure 4.1 below shows how the different components of django's MVT architecture interact to respond to a user's request.

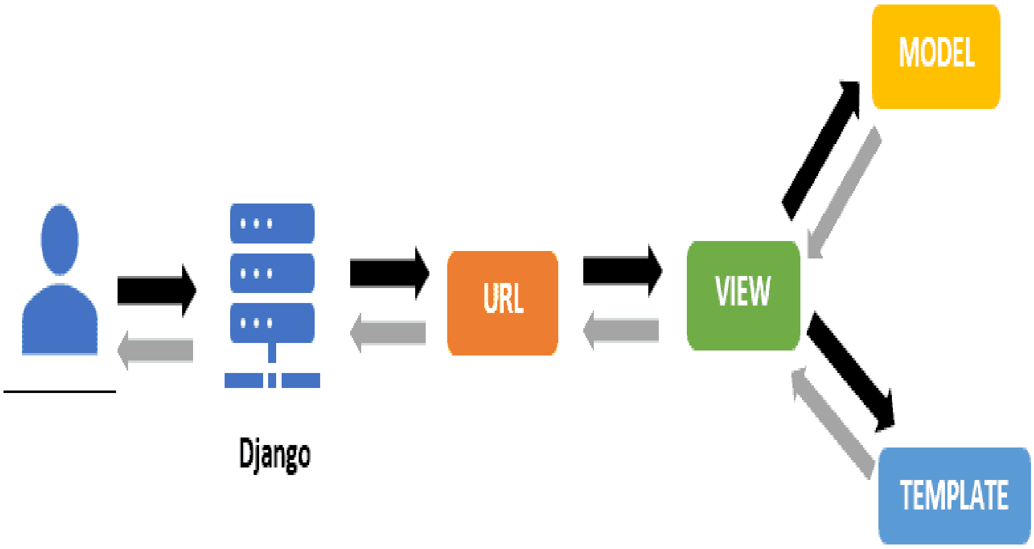


Figure 4.1: django MVT architecture

4.3.2 Front-end architecture

For the Frontend part, we used the Redux architecture which is a JavaScript library that allows us to reduce the complexity of the code by applying the restriction on the way and time of the state update. It can occur as shown in Figure 4.2

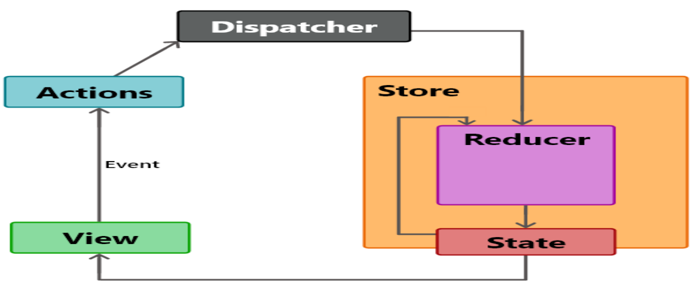


Figure 4.2: Redux architecture

- **Views**: they are JavaScript or TypeScript classes or functions called components in which we define the logic of the user interface, they accept other properties (props) coming from other components and return the shape of our user interface. There is a root component that wraps all the other components as children named App.js.

- **Redux store**: It is a global state management container for our React application. Any component connected to the Redux store can send an action call, after the components pass properties to the reducers, each reducer has a defined action and a given state. The role of the reducers is to provide the new state to the components.

4.4 Solution architecture

Our application is based on a 3-tier architecture presented by figure 4.3. It is a logical model of application architecture which aims to clearly separate three layers within the same system.

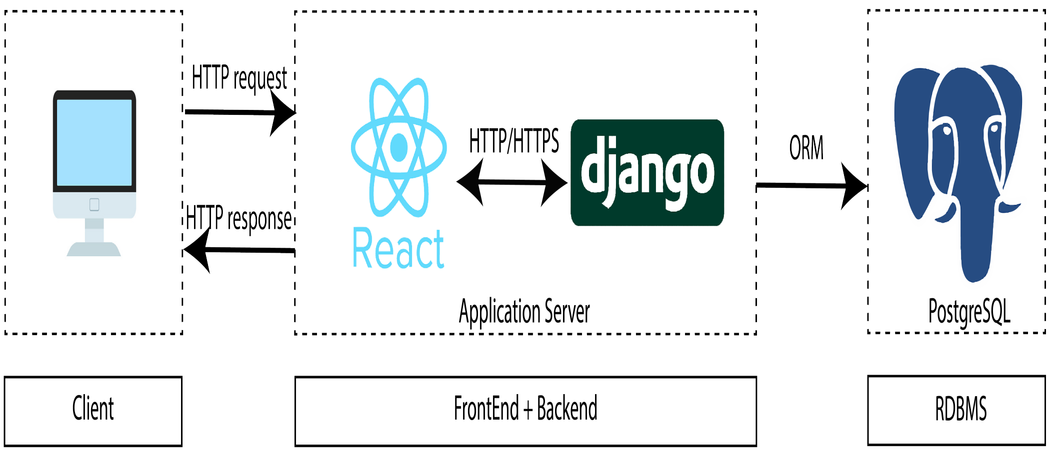


Figure 4.3: Solution architecture

The figure shows the architecture of our application. It consists of three components:

• A web application that represents the front-end of our application. We used React js for its flexibility and performance.

• A server which represents the application logic on the server side, it is connected to our database and our front-end application. We chose the django framework for the back-end development for its rapidity and real time response.

• For data persistence we chose PostgreSQL because of its features like: autonomy, consistency, durability etc. PostgreSQL also uses a variety of backup and recovery strategies to ensure data is not lost in the event of a system crash or unintentional delete.

In addition to that, we used Swagger for the api testing before letting React js consume them.

4.5 Deployment

As mentioned in Table 4.1, we picked Azure for the CI/CD approach. In the following section, we will describe the steps done to deploy our project on a deployment server.

4.5.1 Deployment steps

Each time we finish a sprint, we deploy our application on Azure following these steps.

• We push our back-end and front-end work on their respective main branches in our GitHub repository after unit testing the main features.

• We then create a pull request to the branch linked to the Azure server and see if we are able to merge the two branches or not.

• Finally, GitHub will start merging the pull request by doing the following jobs: build the application by downloading the necessary dependencies and verifying that the application can be deployed and doesn't contain any error then deploy it to the Azure Web Application as shown in the next figure.

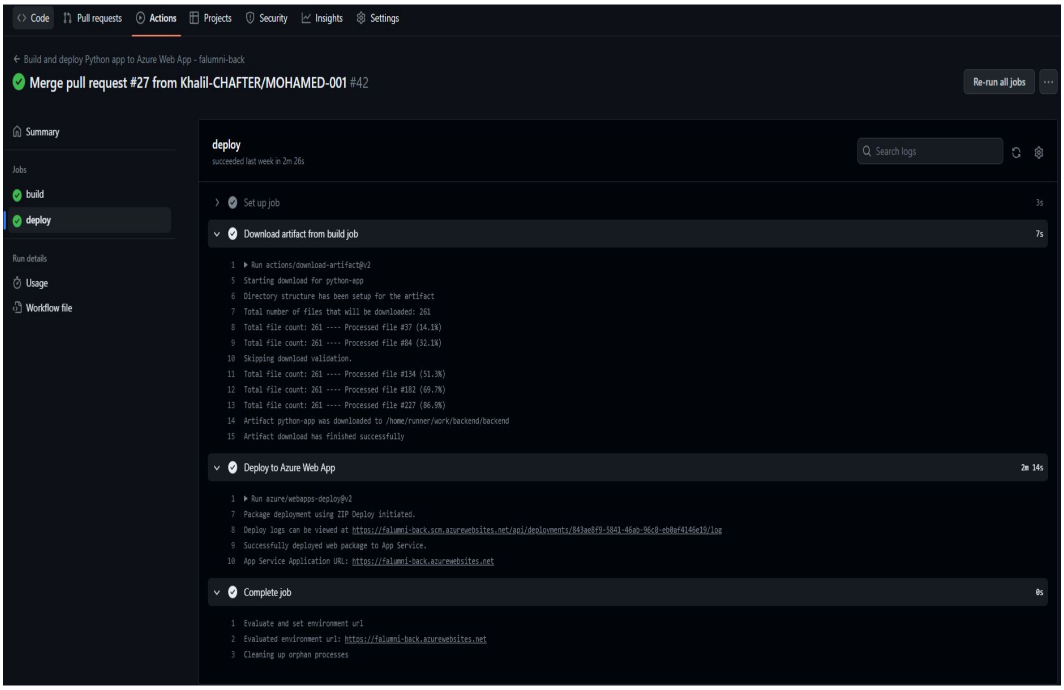


Figure 4.4: Deployment job

4.6 Presentation of the interfaces

In this part, we expose the realized interfaces of our application by describing the usefulness of each one. Our application is composed of several interfaces such as:

4.6.1 Back-office interfaces(Admin space)

Falumni administrator Login interface

Figure 4.5: Falumni administrator Login interface

The login interface contains the email field and the password with a check on the email field if it's not valid. In addition to that, an error alert will show up if you entered one of the fields wrong and pressed the "Login" button or if you typed everything in the right format and the user doesn't exist.

Account requests interface

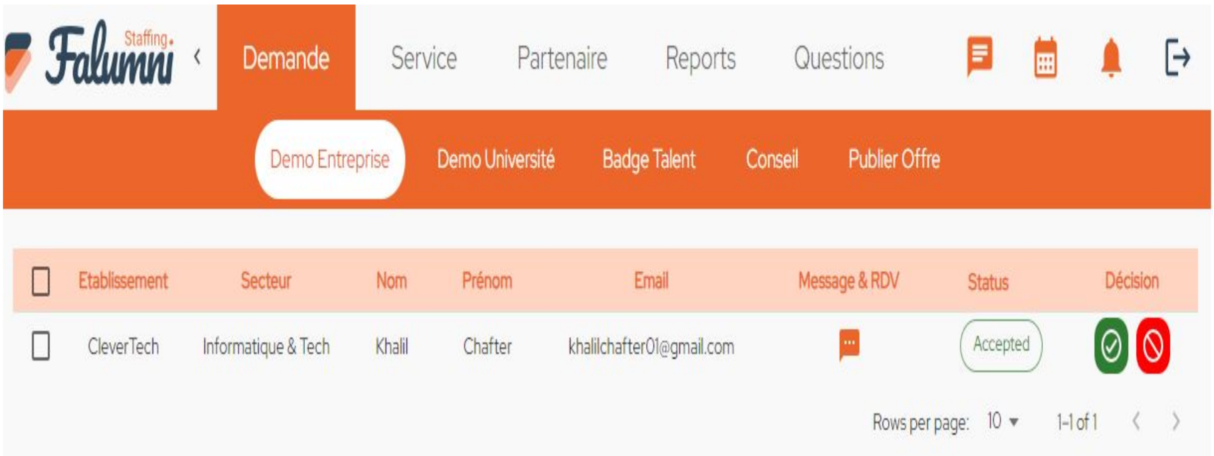


Figure 4.6: Account requests interface

This interface shows the requests made by the company administrator while trying to create an account. So, the Falumni administrator can either deny or accept the account creation. When he accepts the request, an email containing the link to the first connection will be sent to the company administrator to finish setting up his account.

HR questions interface

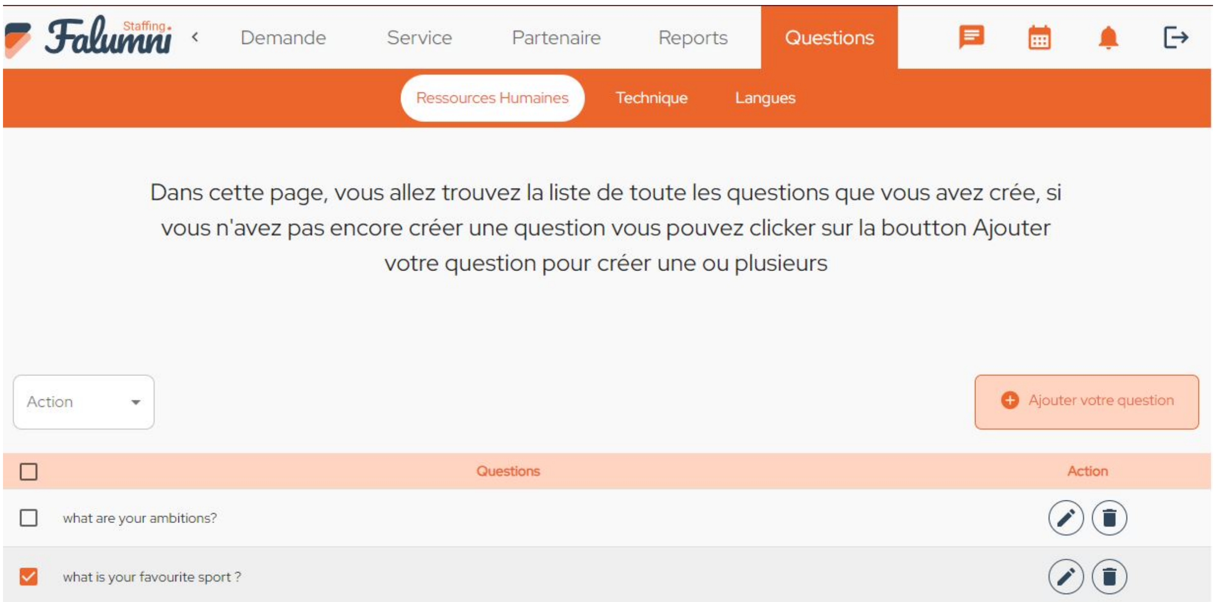


Figure 4.7: HR questions interface

In this interface, the Falumni administrator has the access to consult his already created human resources questions, he can either edit, delete one or select many using the checkbox and pressing the "Action" select field and then selecting "delete" to delete many questions at the same time. If he clicks the "Add your question" button the creation interface will show up as shown in Figure 4.8.

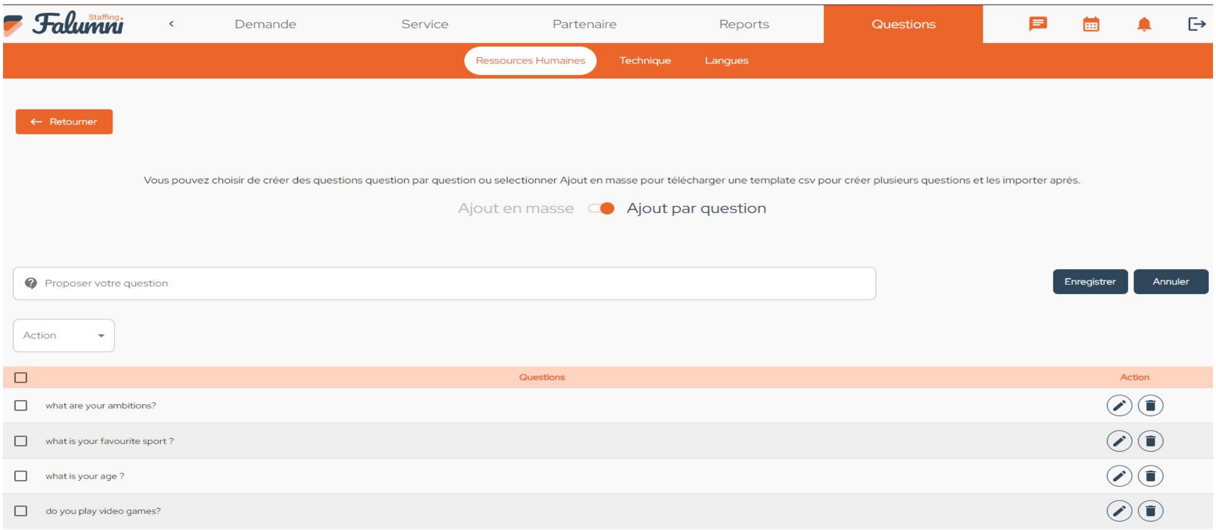


Figure 4.8: HR questions creation interface

In the interface above, the Falumni administrator can still do the same things he can do from the previous interface with the addition of creating a question. He can fill in the question field if he wants to create a question or select the "Add bulk of questions" if he wants to make it easier for himself as we will show in the next interface.

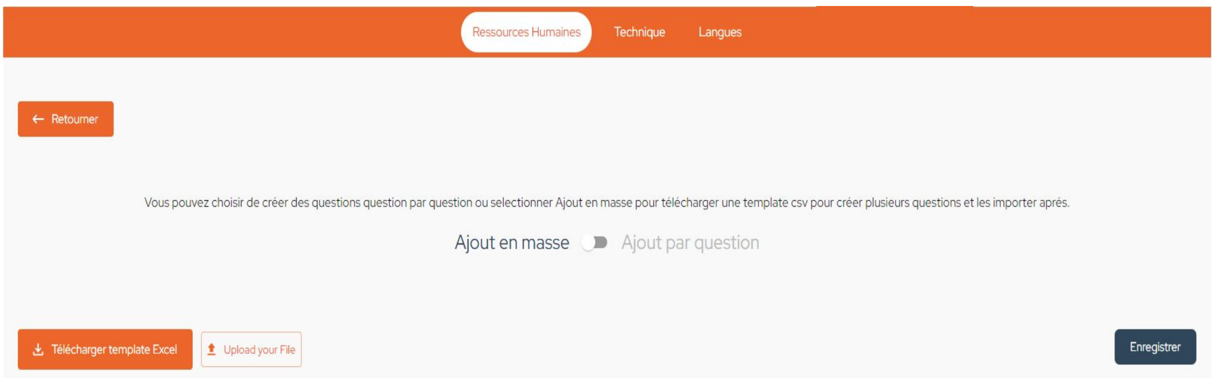


Figure 4.9: HR questions creation in bulk interface

At this point, the Falumni administrator can download the csv template to fill in his questions. Once he finishes, he can upload the csv file into the platform and a table will show up for him to check if those are the questions he wants to add and then press the "Save" button to create all of them at once.

Technical skills interface

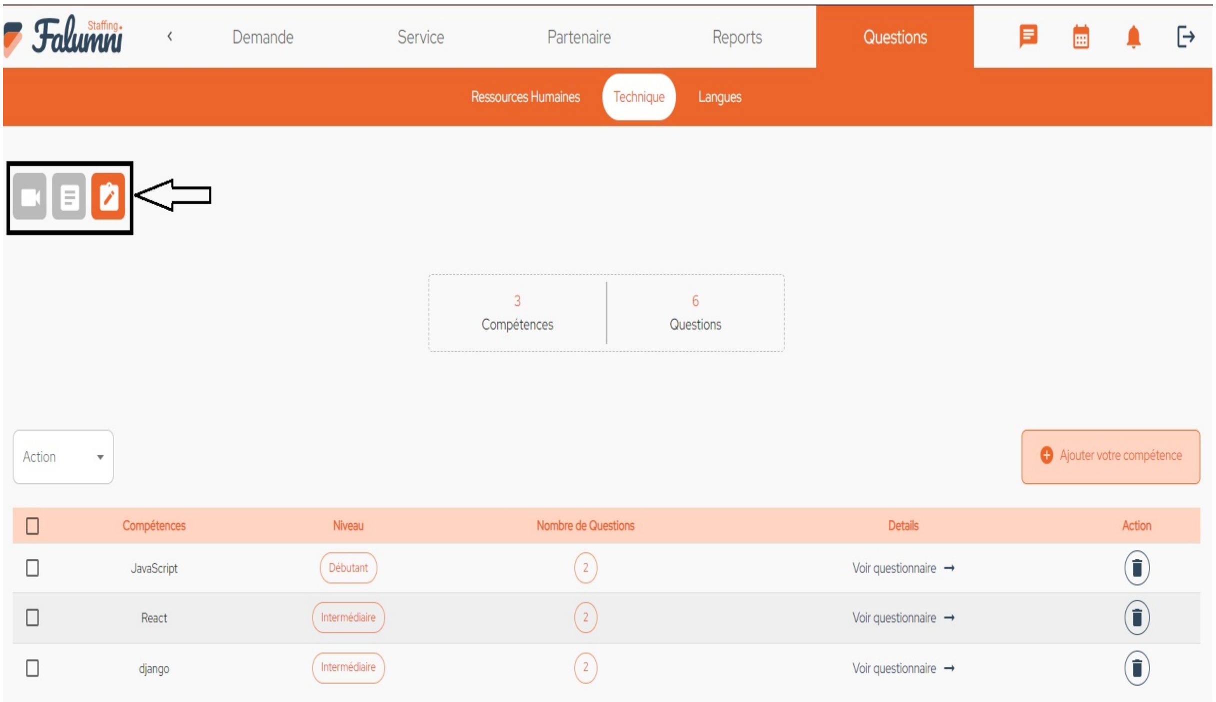


Figure 4.10: Technical skills interface

This interface shows the technical skills created for a certain type of questions. As shown at the top left corner of the interface, the Falumni administrator has the choice between a video, text or MCQ questions to assign with the created skills. So this interface represents the technical skills for the MCQ questions. He can create a new skill by pressing the "Add your skill" button which brings up this pop-up.

Figure 4.11: Add skill pop-up

He then fills the necessary fields which are the skill name, select the level and choose a name for the questionnaire that will be created alongside the skill. If he wants to add an MCQ question he needs to press "Show questionnaire" in the table cell to redirect him to the MCQ question creation interface for that specific skill. The language skills has the same scenario for its creation.

MCQ questions interface

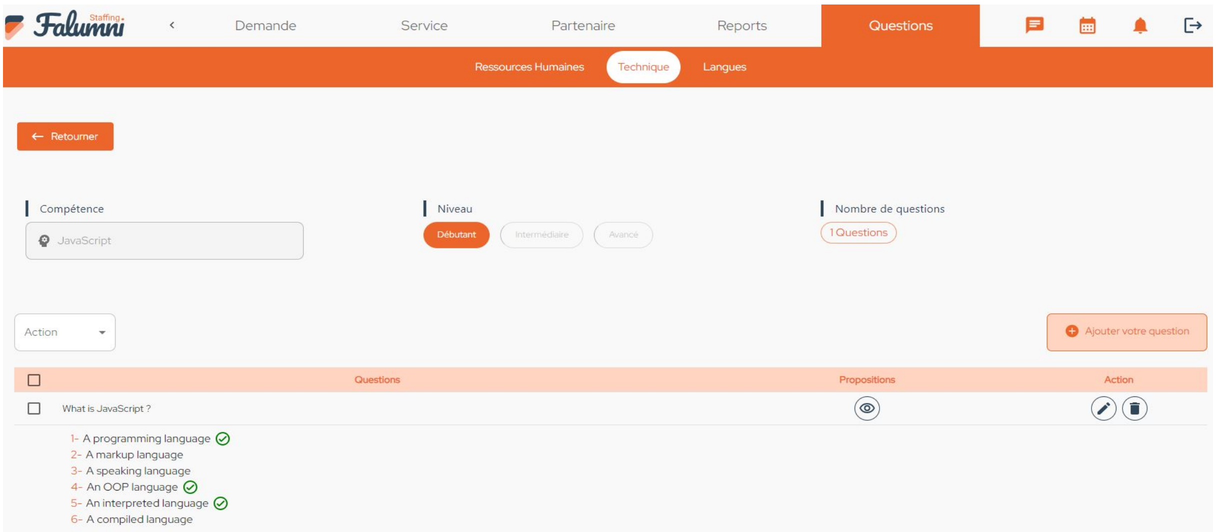


Figure 4.12: MCQ questions interface

The Falumni administrator can consult the MCQ questions he created for a certain skill as it shows in Figure 4.12. By clicking on the eye icon he can see the propositions for that questions as well as if that is a right answer or not. By clicking the "Add your questions" button, the creation pop-up opens up.

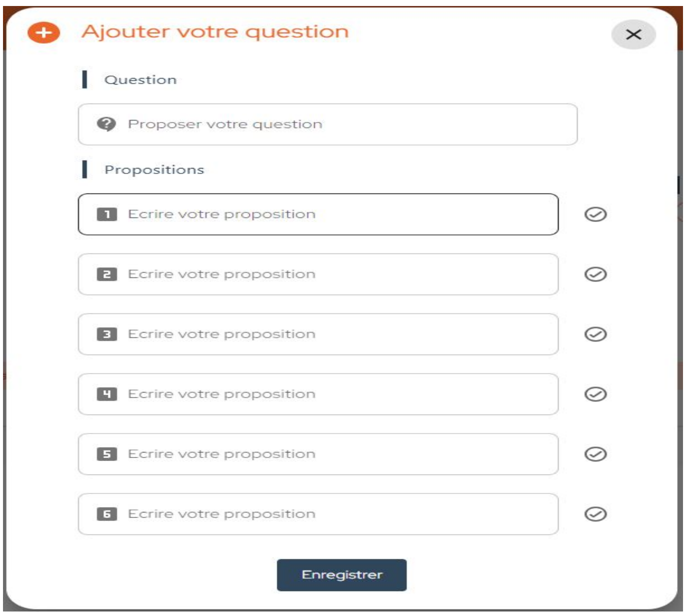


Figure 4.13: Add MCQ question pop-up

He then fills the question field and the six propositions along side picking the right proposition for that answer.
The MCQ creation process for a language skill is the same. For the video and text questions case, they follow the same process as the Human resources questions creation.

4.6.2 Company space interfaces

Company account creation interface

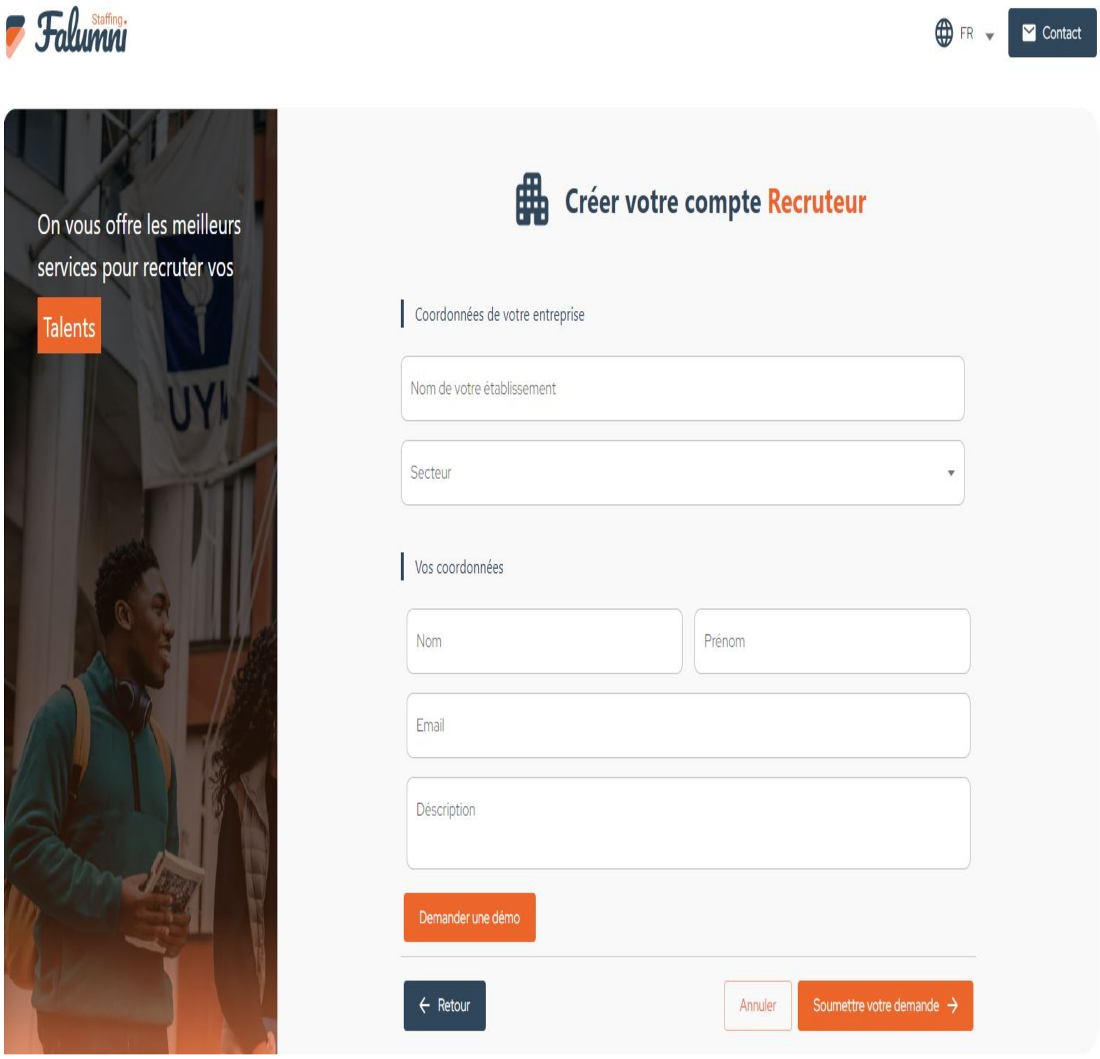


Figure 4.14: Company account creation interface

In this interface, the company administrator is invited to fill his company name and sector as well as his information and clicks the "ask for demo" button. Once he clicks the button, the following pop-up shows up.

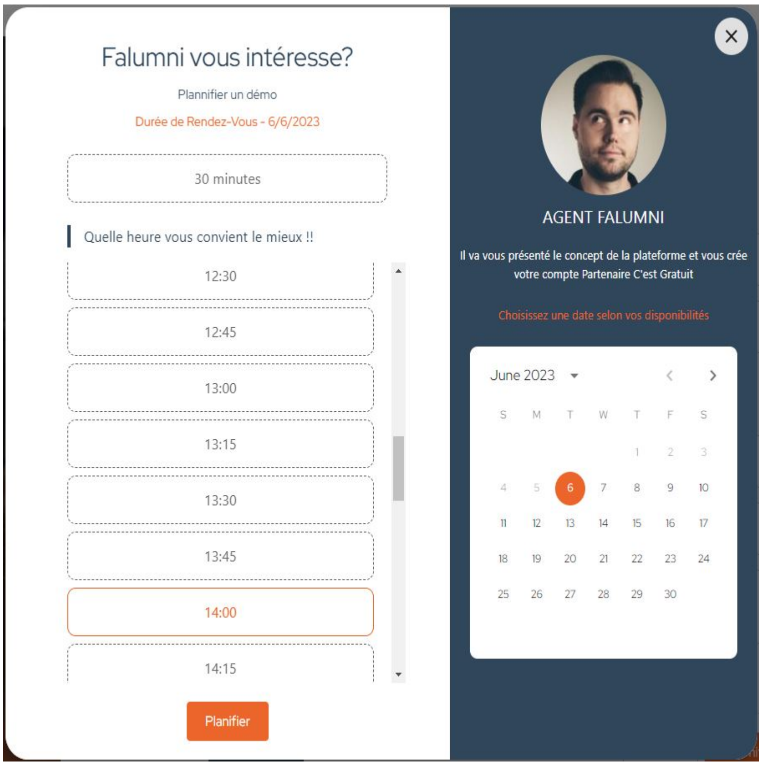


Figure 4.15: Demo pop-up

He proceeds then to pick the date and time for the online meeting with the Falumni administrator and clicks the "Plan" button. Then he clicks the "Submit your request" button to send a request to the Falumni administrator to create his account as shown in Fig.4.6. If the Falumni administrator accepts the request, the company administrator will receive an email containing a link to the first connection for him to finish setting up his account.

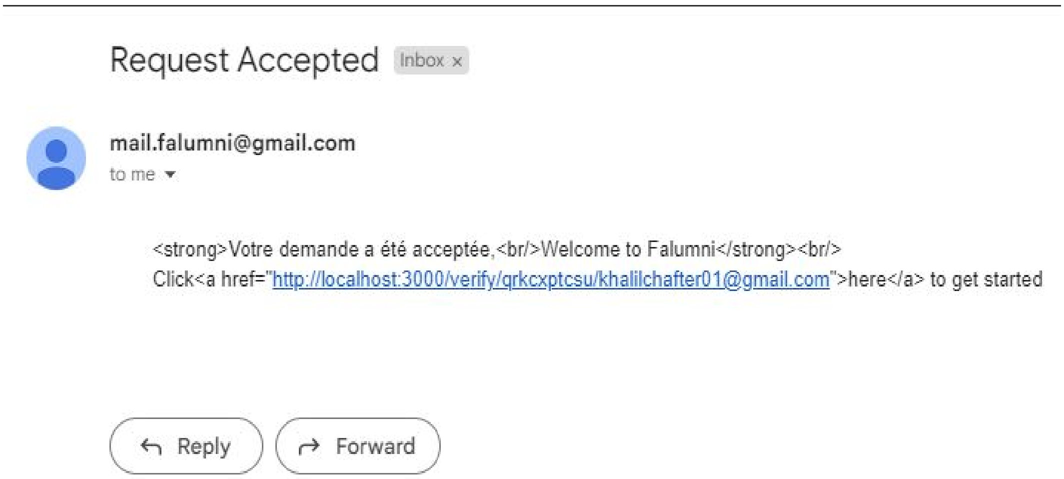


Figure 4.16: Request accepted Email

Company login interface

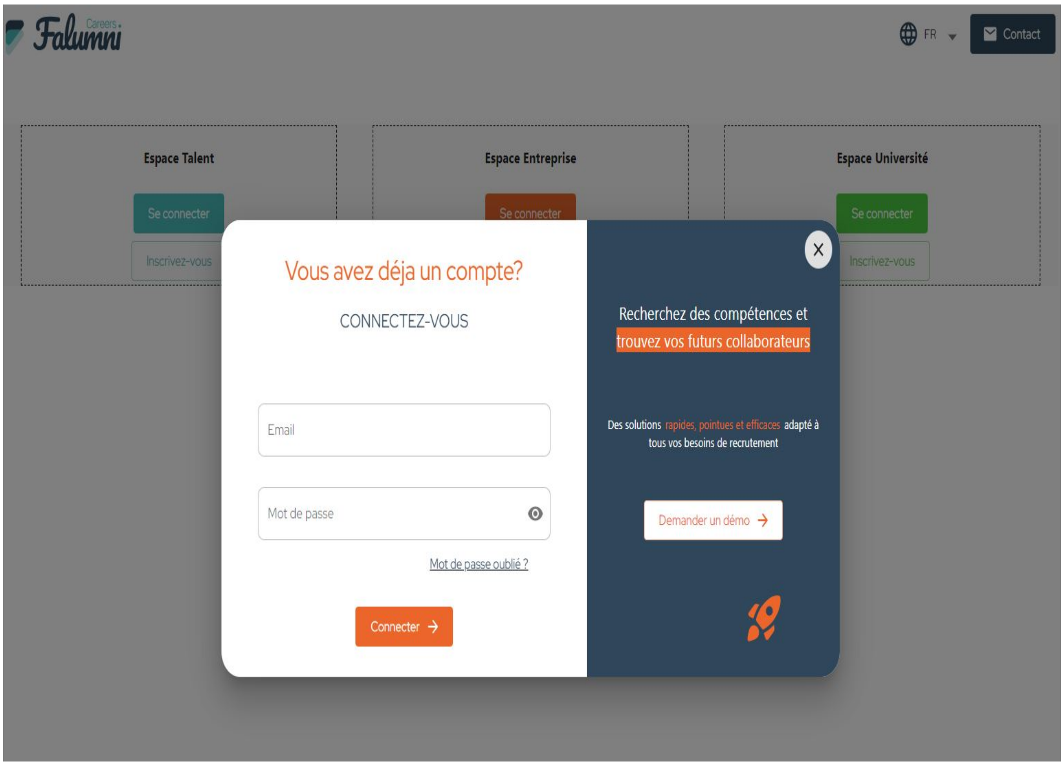


Figure 4.17: Company login interface

As mentioned before for the Falumni administrator's login interface, the company administrator needs to fill in his email and password and then clicks the "Login" button to get redirected to his space. If the user doesn't exist or the credentials he submitted are invalid an error alert will show up.

Job offer interface

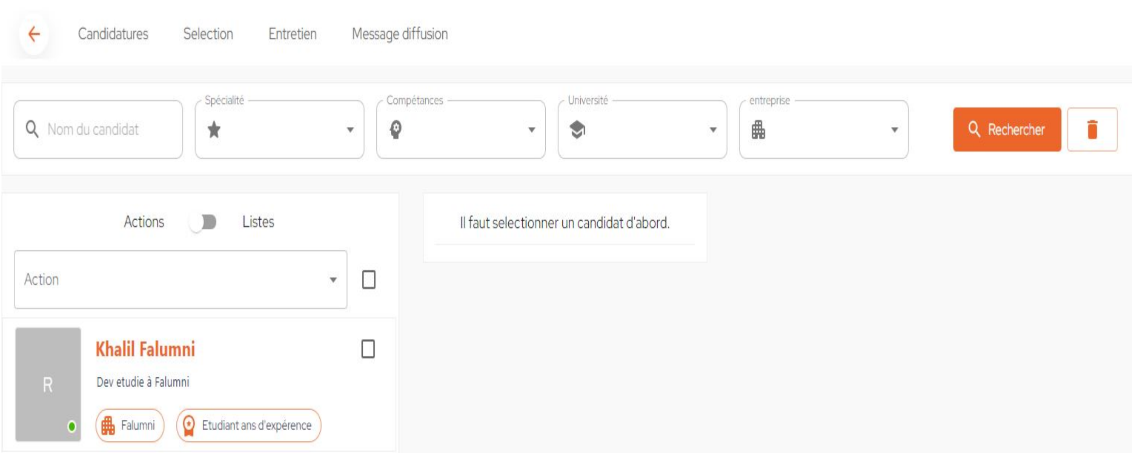


Figure 4.18: Job offer interface

On the left section of the job offer interface we find the candidates that participated in this job offer and waiting for a response from the company administrator. If the company administrator clicks on one of the candidates on the list the middle and right sections will change.

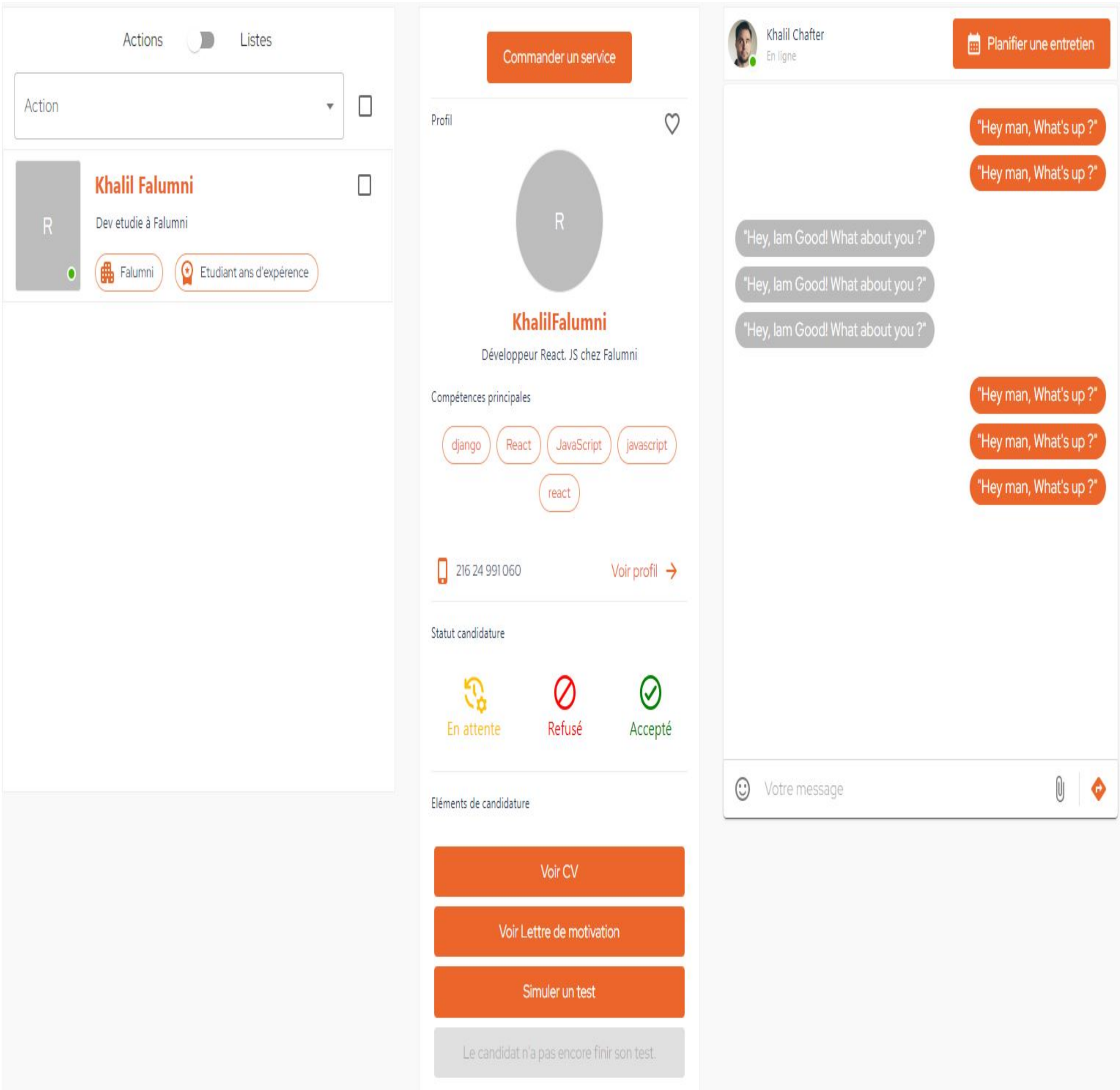


Figure 4.19: Job offer selected candidacy interface

As we can see when the company administrator select a candidate, the middle and the right sections change. the right section which contains the conversation between the company administrator and the talent is still under development. If the company administrator clicks the "Simulate test" button, he gets redirected to the test simulation interface to make a test for the talent. The grey button is disabled unless the talent finishes up his test.

Test choice interface

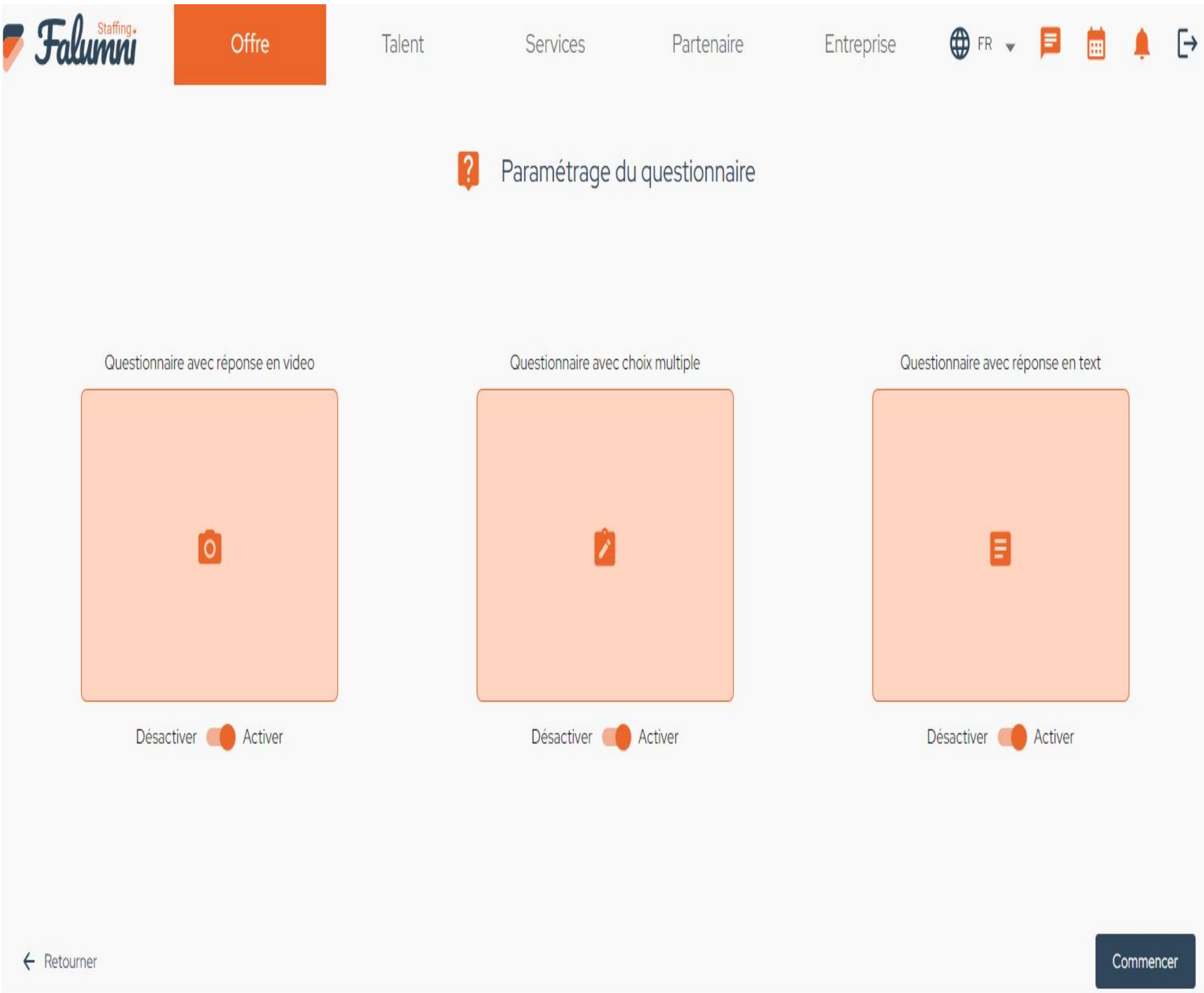


Figure 4.20: Test choice interface

In the test choice interface, the company administrator has to choose at least one of the 3 types of questionnaires whether it is video, text or MCQ. He has the option to choose all of them as well. Once he made his choice, he then proceeds to click the "Start" button to start the test simulation as shows the figure in the next page.

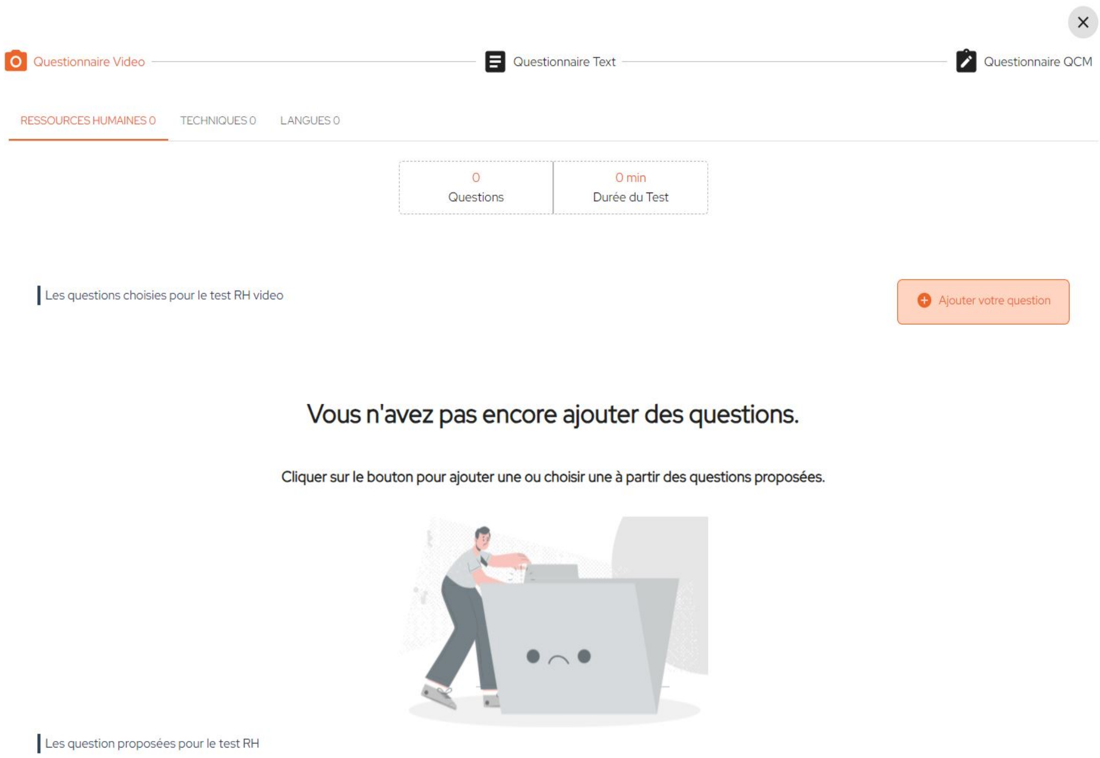


Figure 4.21: Test Simulation interface

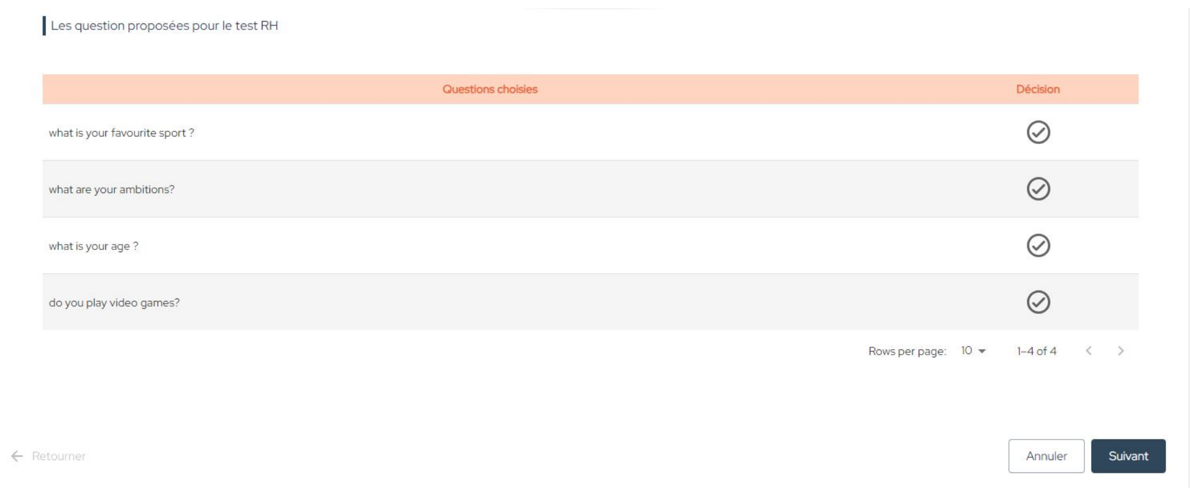


Figure 4.22: Test Simulation interface

The company administrator has the option to start filling any questionnaire he wants. He can choose from the questions proposed by the Falumni administrator by clicking on the "Check" icon or he can create his own question by clicking the "Add your question" button to show a pop-up.

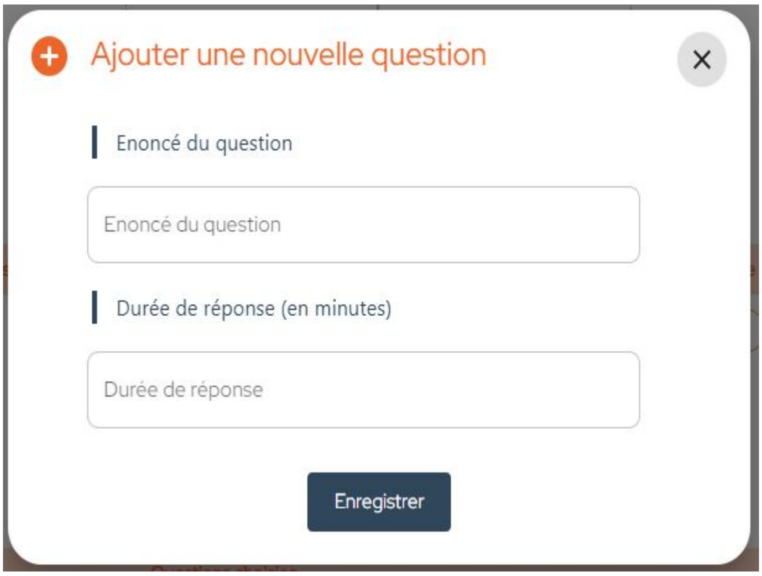


Figure 4.23: Test question creation pop-up

He mentions the question content and its duration. And by clicking the "Save" button the question gets created and the previous interface will look like this.

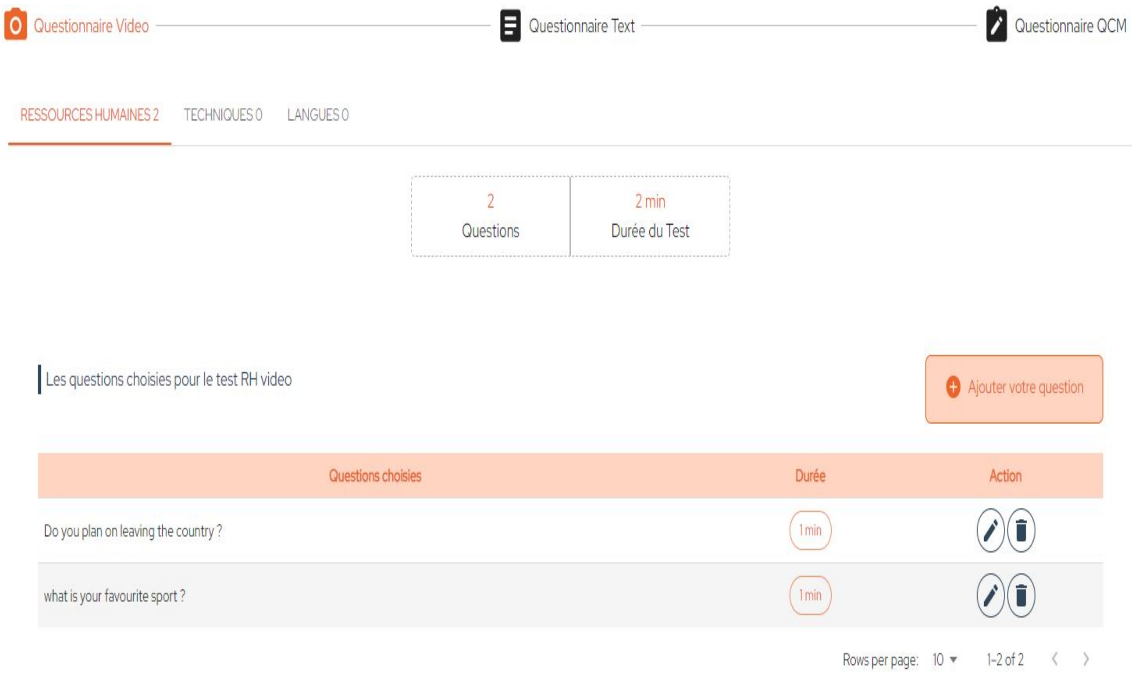


Figure 4.24: Test HR questions list

Once he finishes filling up the first section, he then moves to filling the technical and language questions as shows the next figure.

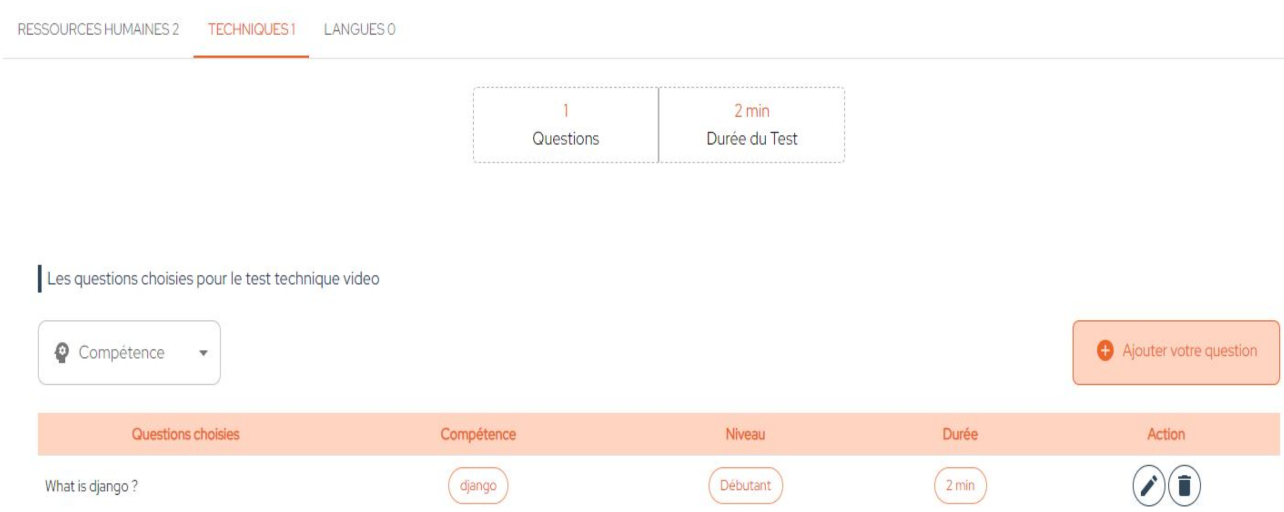


Figure 4.25: Test technical questions list

The select field contains the skills mentioned in the job offer and the company administrator can even choose from the Falumni platform based on those skills to make it easier for him. He can also create his own question by clicking the "Add your question" button.

Figure 4.26: Test technical questions creation pop-up

He should mention the targeted skill as well as its corresponding level. Once he finishes with the technical questions he can move to the language questions which is basically repeating the same process he had already done. After that he moves to the

MCQ section and picks from the skills available with their respective levels to finish simulating the test.

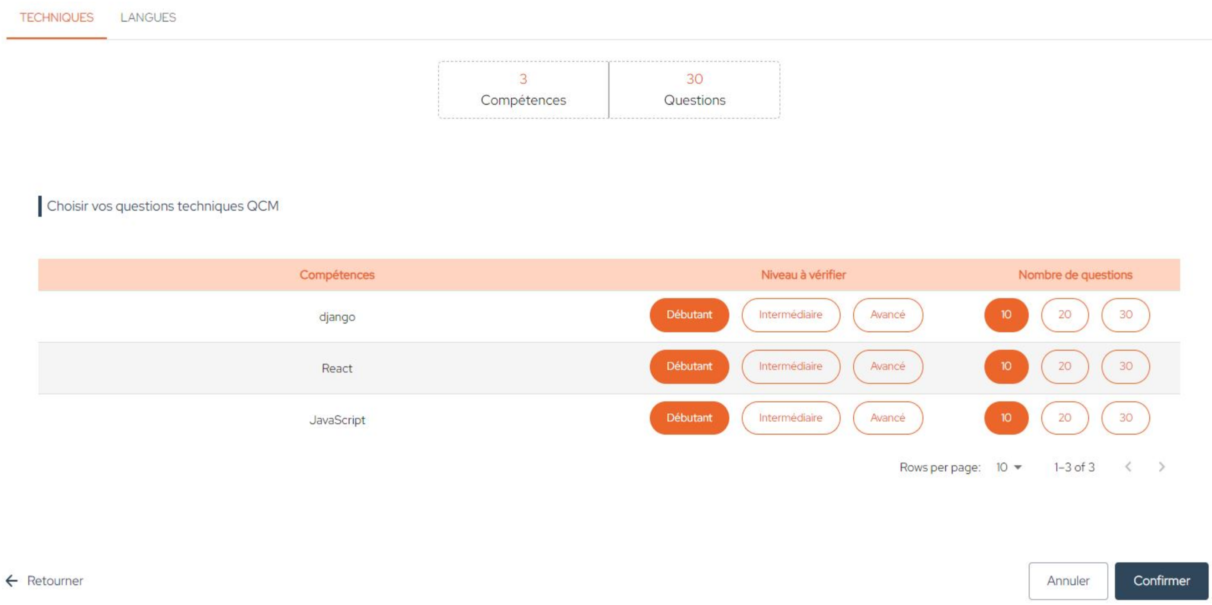


Figure 4.27: Test MCQ questions interface

Once he finished picking, the company administrator then proceeds to press the "Confirm" button that will open the verification pop-up as shown in the next page.

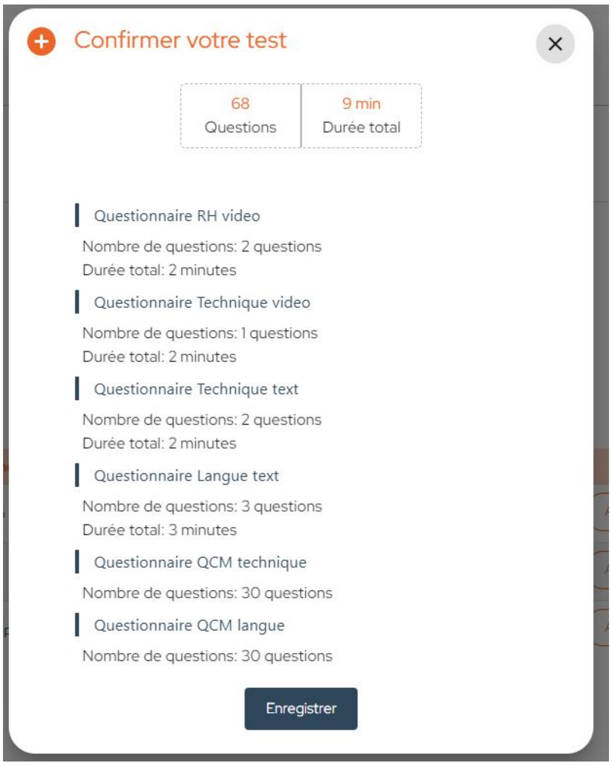


Figure 4.28: Test verification pop-up

By clicking the "Save" button, he creates the test he already simulated and got redirected to the job offer interface. He can click the cross mark button if he wants to add some other questions.

4.6.3 Talent space interfaces

Talent account creation interface

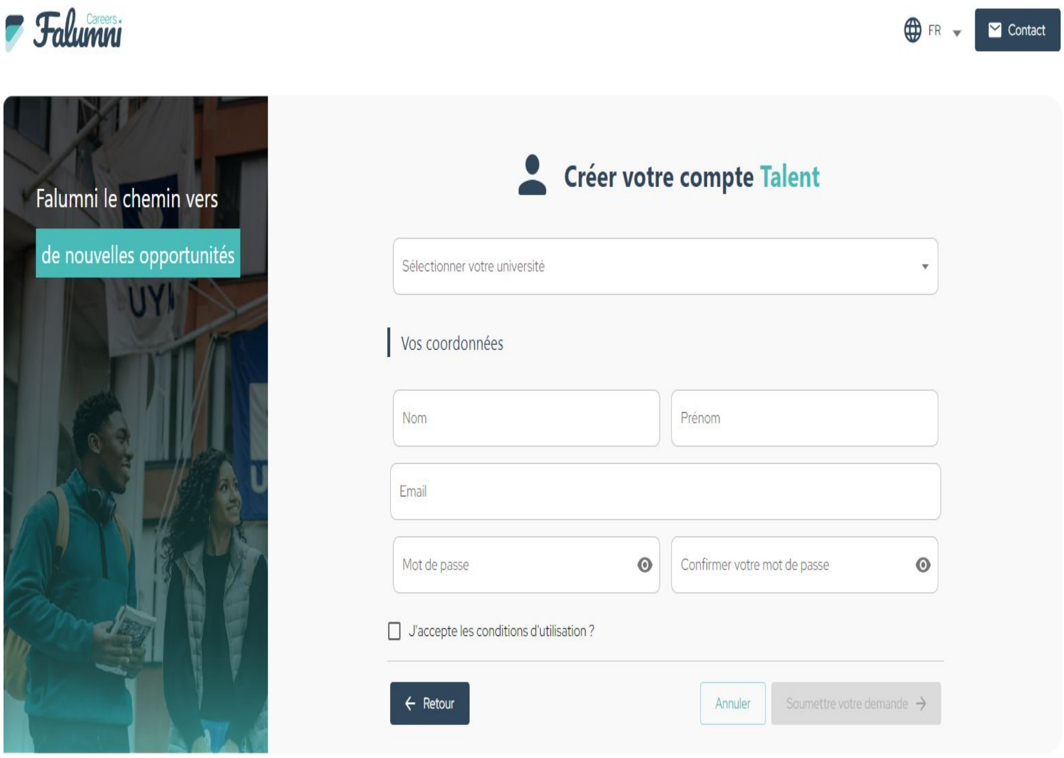


Figure 4.29: Talent account creation interface

To create his account, the talent starts by selecting his university as well as filling his personal information alongside his password. Once that done, he clicks on the "create account" button to create his account.

Talent login interface

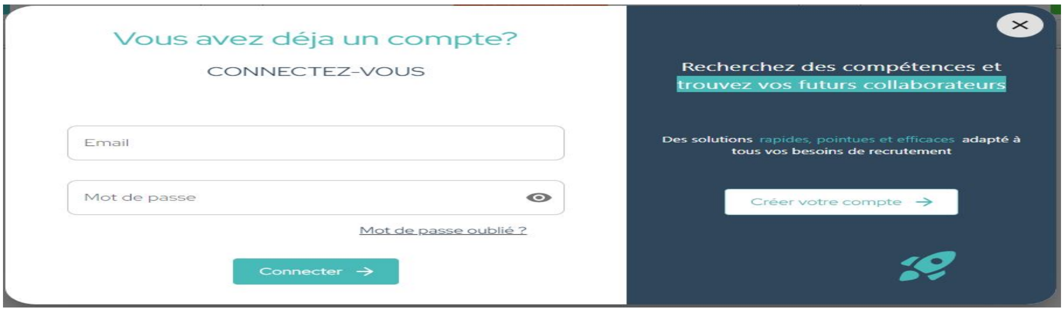


Figure 4.30: Talent login interface

The talent login process is the same as the Falumni administrator and the company administrator. He just needs to fill in the fields correctly and then clicks the "Login" button to be redirected to his space.

Talent profile interface

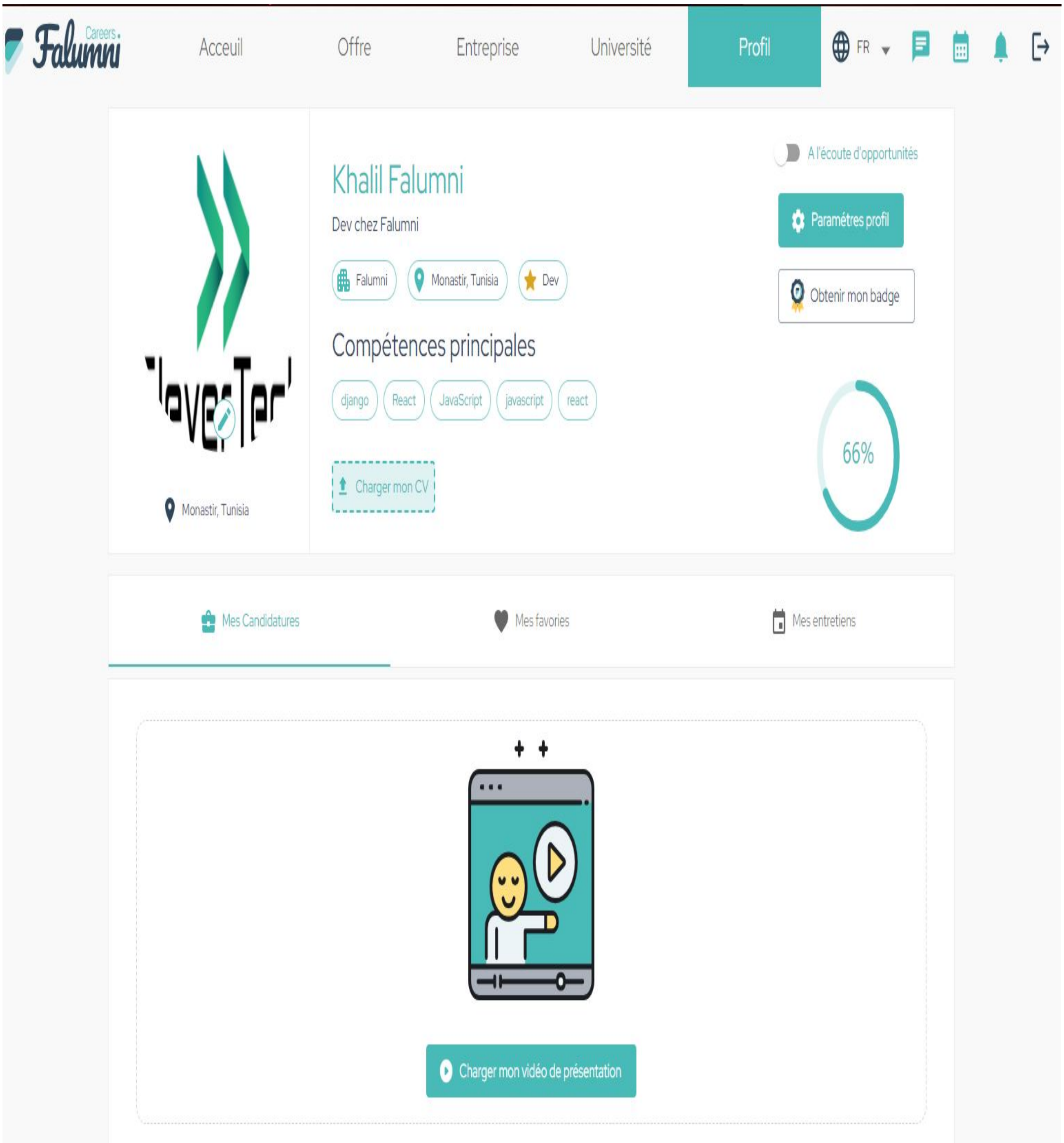


Figure 4.31: Talent profile interface

The talent profile interface is where he finds his information that he put the first time he logged in into his account. There he can add his cv ect. To access his tests, the talent should click on the "Profile settings" button location on the top right of the interface.

Talent profile settings interface

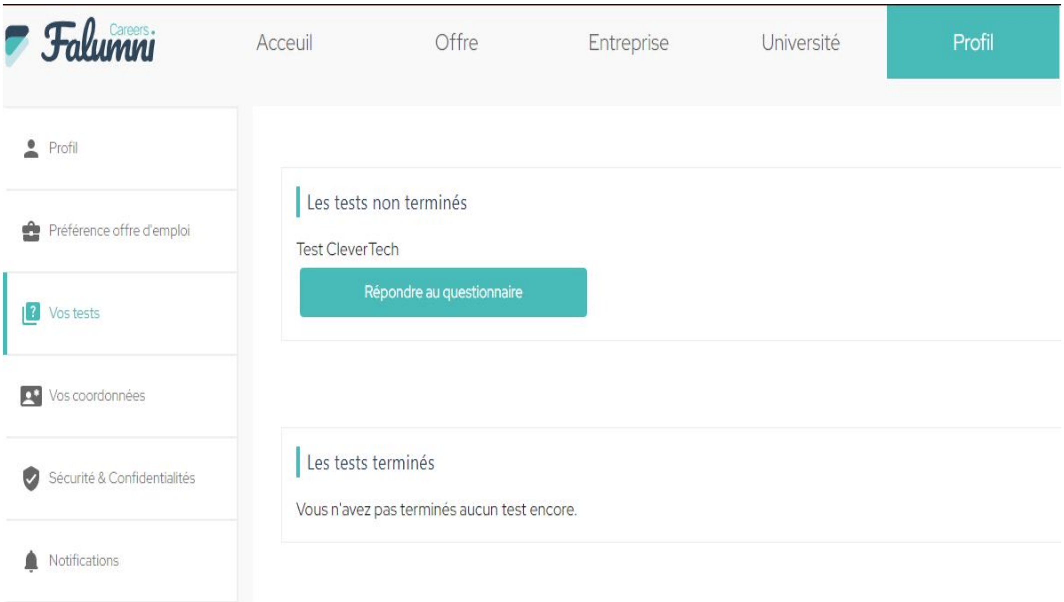


Figure 4.32: Talent profile settings interface

The talent can access his personal information through this interface. When he clicks on the "Your tests" button located in the left side bar the interface shows his finished tests and non finished ones. If he wants to pass a test he can just click the "Respond to questionnaires" button.

Pass text questions interface

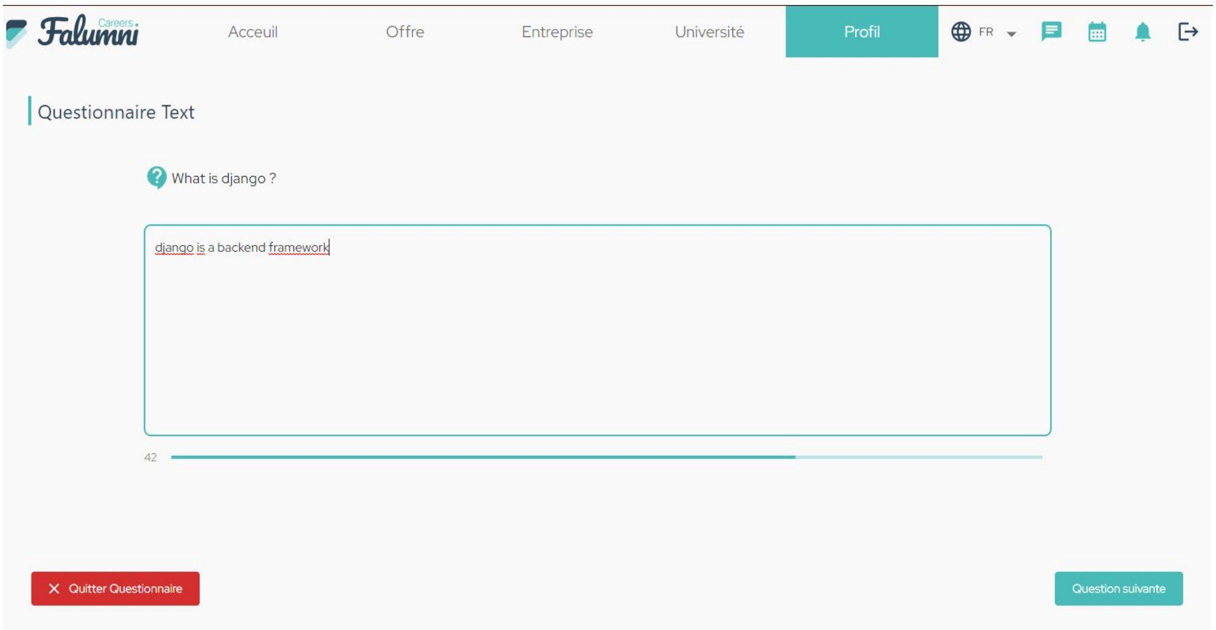


Figure 4.33: Pass text questions interface

Once he gets redirected to the test interface, the talent will begin his exam. He always starts by the text questionnaire first if it is available. He should answer the given questions properly and within the time limit or else the next question will pop automatically without him finishing his answer. When he finishes answering a question within the time limit, he clicks the "Next question" button to answer the next question until the last one. He also has the option of quitting the questionnaire. When he quits, he can't pass the test again.

Pass MCQ questions interface

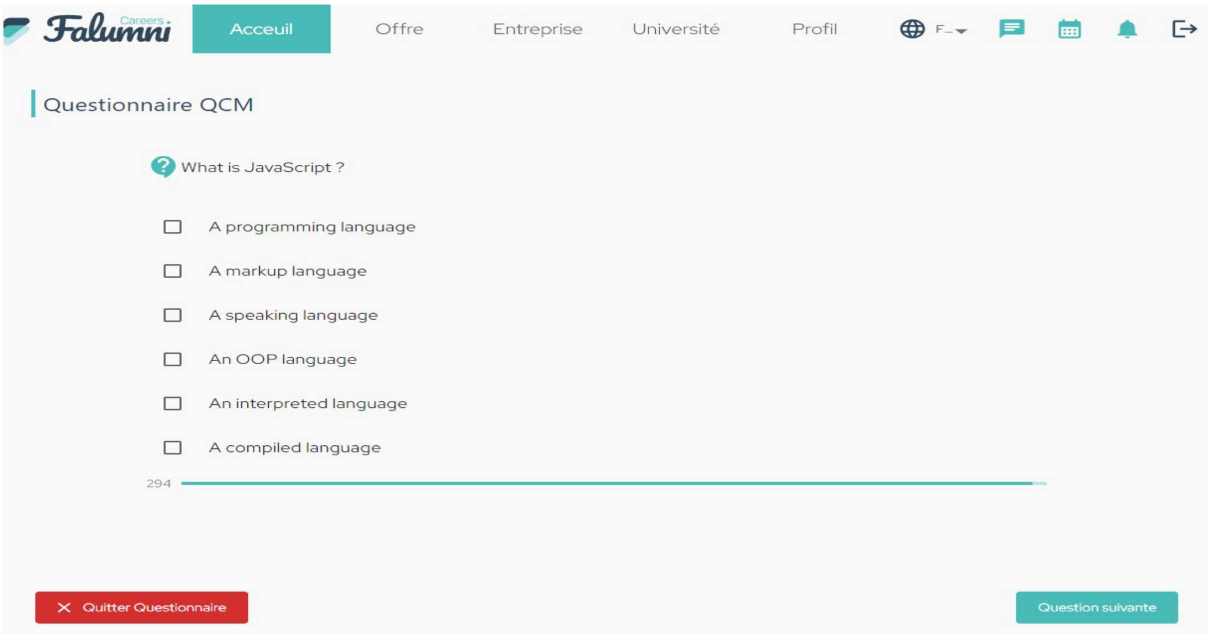


Figure 4.34: Pass MCQ questions interface

After completing his answers for the text test, the talent gets redirected to pass the MCQ test immediately if it exists. The process is pretty simple, he needs to pick the right choices for the test. If one choice is wrong, the whole question will not be validated. After finishing the questions with respecting each question duration, he then gets redirected to his profile and the test will move to the finished tests. For the company administrator, now he can visualize the answers the talent made for his test as shows the next two figures.

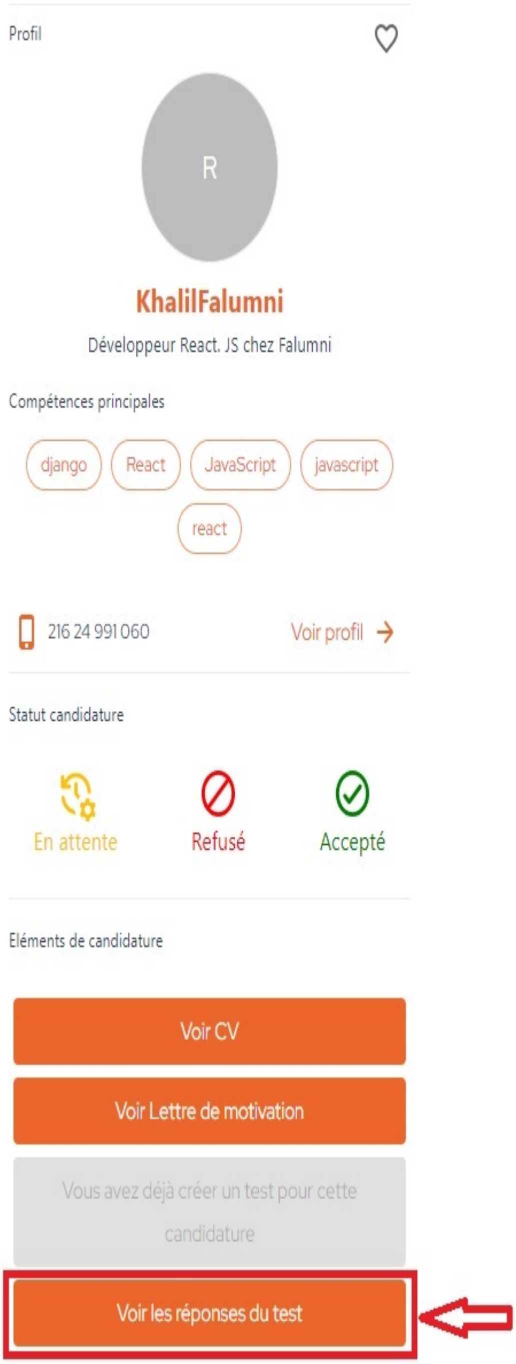
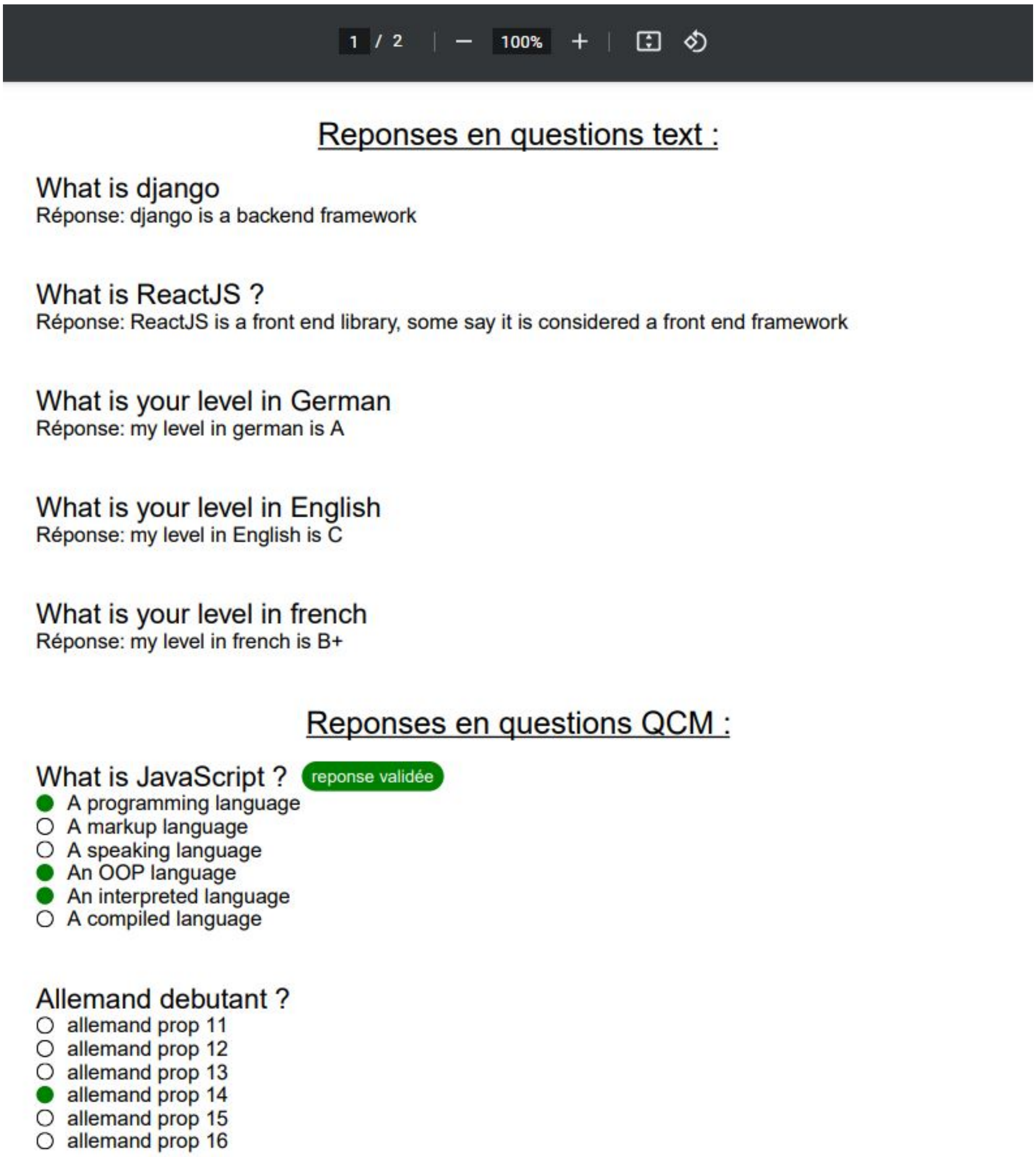


Figure 4.35: Job offer interface



1 / 2 100%

Reponses en questions text :

What is django
Réponse: django is a backend framework

What is ReactJS ?
Réponse: ReactJS is a front end library, some say it is considered a front end framework

What is your level in German
Réponse: my level in german is A

What is your level in English
Réponse: my level in English is C

What is your level in french
Réponse: my level in french is B+

Reponses en questions QCM :

What is JavaScript ? reponse validée
- A programming language
- A markup language
- A speaking language
- An OOP language
- An interpreted language
- A compiled language

Allemand debutant ?
- allemand prop 11
- allemand prop 12
- allemand prop 13
- allemand prop 14
- allemand prop 15
- allemand prop 16

Figure 4.36: Generated Answers PDF

After clicking the "Show test answers" button, a PDF will generate based on the answers given by the talent. For the MCQ answers, if the company administrator finds "Validated response" in front of the question then the answer is correct. If he didn't find that then it means that the talent failed to answer the question correctly.

4.7 Conclusion

This chapter represents the last part of this report. Its purpose was to present the software environment, the physical and logical architectures of our application as well as the steps to deploy each sprint on a deployment server. As for its last part, it was reserved for the navigation between the different functionalities of the platform which are represented in the form of screenshots.

General Conclusion

As a conclusion, we can mention that the creation of this new platform will facilitate the companies managers lives considering the fact that they are the ones who are going to make the tests for talents that applied to their job offers, through the presence of various useful and indispensable services in the contemporary world. In this sense, our report is composed of 4 chapters :

• The first chapter focuses its attention on the characteristics of the working environment, the presentation of the host organization, without forgetting the motivations, the problem, the analysis of the existing as well as the proposed solution and the working methodology with specifying the Gantt chart of this project.

• The second chapter shows the originality of the actors, the functional and non-functional needs of the application and its technical requirements.

• The third chapter presents the preliminary design of the application.

• Finally, the fourth chapter explains the chosen tools to develop the platform as well as the required architecture with mentioning the deployment server and adding at the end some of the interfaces designed for the application in question.

Our study has borne fruit thanks to our current position, as IT developers, within the company CleverTech. This experience allowed us to enrich our knowledge and to master more of the technologies out there especially ReactJS, django, and the automated tests as well as participating in practically all the phases of the software life cycle.

We plan to expand our platform where we will improve the user experience more so it does feel smoother in the eyes of our users. We want to add a messaging service so that the company administrator and talent can talk about the offer and requirements after passing the test. In addition to that, we will make sure that passing the video test will work as intended since it requires a lot of research. Last but not least, we want to add a report feature in our platform so that all of our users feel secured about using it.

References

[1] *AdobeXD*. [Consulted on 30/05/2023]. URL: https://www.adobe.com/products/xd/learn/get-started/what-is-adobe-xd-used-for.html.

[2] *Azure DevOps*. [Consulted on 30/05/2023]. URL: https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops.

[3] *Clevertech*. [Consulted on 28/05/2023]. URL: http://www.clevertech-france.fr.

[4] *ClickUP*. [Consulted on 30/05/2023]. URL: https://www.softwareadvice.com/project-management/clickup-profile/.

[5] *Cooptalis*. [Consulted on 28/05/2023]. URL: https://www.cooptalis.com.

[6] *django*. [Consulted on 30/05/2023]. URL: https://www.djangoproject.com/.

[7] *django Rest Framework*. [Consulted on 30/05/2023]. URL: https://www.django-rest-framework.org/.

[8] *GitHub*. [Consulted on 30/05/2023]. URL: https://kinsta.com/knowledgebase/what-is-github/.

[9] *Jobteaser*. [Consulted on 28/05/2023]. URL: https://www.jobteaser.com/fr.

[10] *MUI*. [Consulted on 30/05/2023]. URL: https://mui.com/material-ui/getting-started/overview/.

[11] *PostgreSQL*. [Consulted on 30/05/2023]. URL: https://www.postgresql.org/.

[12] *ReactJS*. [Consulted on 30/05/2023]. URL: https://legacy.reactjs.org/.

[13] *Redux Toolkit*. [Consulted on 30/05/2023]. URL: https://redux.js.org/redux-toolkit/overview.

[14] *SCRUM*. [Consulted on 30/05/2023]. URL: https://www.scrum.org/resources/what-is-scrum/.

[15] *Sintegra consulting*. [Consulted on 28/05/2023]. URL: https://sintegraconsulting.com.

[16] *Tanitjobs*. [Consulted on 28/05/2023]. URL: https://www.tanitjobs.com.

[17] *UML*. [Consulted on 30/05/2023]. URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/.