

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de la Manouba

Institut supérieur des Arts Multimédias



Mémoire de fin d'études

Préparé en vue de l'obtention du diplôme en licence Fondamentale en informatique
et multimédia

Plateforme de Test Technique pour le Recrutement de Stagiaires

Réalisé par

Nermine Bouzidi

Encadrant ISAMM

M. Tarek HAMROUNI

Encadrant Entreprise

M. Wissem INOUBLI

Année universitaire 2023/2024

Dédicaces

Remerciement

Au terme de ce travail, je tiens à remercier profondément mon encadreur académique à l'ISAMM Monsieur Tarek Hamrouni pour l'encadrement qui m'a apporté et pour m'avoir écouté et conseillé ainsi pour son dévouement profond.

Je tiens à remercier Monsieur Wissem Inoubli mon encadreur professionnel à MajestEYe pour m'avoir offert l'opportunité de réaliser ce projet et pour la qualité du sujet proposé.

Enfin, je suis honoré par la présence des membres du jury et j'exprime d'avances mes sincères remerciements à tous les membres qui ont bien accepté de juger mon travail rentrant dans le cadre du stage de fin d'études.

Table des matières

Introduction	1
1 Contexte général	2
1.1 Organisme d'accueil	2
1.2 Étude de l'existant	3
1.2.1 Description de l'existant et critique de l'existant	3
1.2.2 Solution proposé	4
1.3 Choix technologiques et architecture logicielle	4
1.3.1 Framework et langages	4
1.3.2 Environnement logiciel	6
1.3.3 L'architecture d'API adoptée :	7
1.3.4 Architecture logicielle	9
1.3.5 Patron de conception	10
1.4 Méthodologie de travail	11
1.4.1 Principe de la méthodologie Scrum	11
2 Spécification des besoins	13
2.1 Spécification des besoins	13
2.1.1 Identification des acteurs	13
2.1.2 Besoins fonctionnels	13
2.1.3 Besoins non fonctionnels	14
2.2 Diagramme de cas d'utilisation	15
2.3 Pilotage du projet par Scrum	15
2.3.1 Équipe Scrum	15
2.3.2 Backlog du produit	15
2.3.3 Planification des sprints	18
2.4 Conclusion	18

Liste des figures

1.1	Logo MajestEye	2
1.2	Processus de recrutement de MajestEye	4
1.3	Angular logo	5
1.4	Spring boot	5
1.5	Sonar Qube	5
1.6	UML	5
1.7	Tailwind CSS	6
1.8	VsCode logo	6
1.9	Intellij IDEA logo	6
1.10	Postmanlogo	6
1.11	StarUML logo	6
1.12	MySQL logo	7
1.13	Spring boot security	7
1.14	Spring boot security	8
1.15	Jwt logo	8
1.16	Représentation du jeton JWT de manière codée et décodée.	9
1.17	Architecture global	10
1.18	Caption	11
1.19	Cycle de vie d'un produit par la methodologie Scrum	12
2.1	Diagramme des cas d'utilisation global	15
2.2	La priorisation de MOSCOW	16

Liste des tables

1.1	MajestEye FDS	3
2.1	Equipe Scum	15
2.3	Planification des sprints	18

Liste des acronymes

UML Unified Modeling Language

SOAP Simple Object Access Protocol

REST Representational State Transfer

Introduction

Le monde professionnel est en constante évolution, avec de nouvelles technologies et de nouveaux défis émergeant chaque jour. Face à cette évolution, les entreprises sont constamment à la recherche de talents prometteurs qui répondent aux besoins de la société. Le recrutement efficace des stagiaires revêt une importance cruciale. Cependant, identifier les candidats les plus adaptés sur le plan technique peut s'avérer être un défi de taille pour les recruteurs.

Dans cette perspective, l'entreprise MajestEye est dite prêt à abandonner la manière archaïque de recrutement des stagiaires et à s'investir à fin d'avoir une solution informatique permettant de simplifier ce processus.

C'est dans ce cadre que vient mon projet de fin d'étude en vue de l'obtention de la licence en science informatique qui consiste à développer une plateforme de test technique en ligne dédiée au recrutement de stagiaires.

Le présent rapport synthétise tout le travail que nous avons effectué. Il s'articule autour de quatre grands chapitres désignés comme suit :

Le premier chapitre sera dédié **Contexte général** c'est un aperçu sur l'entreprise d'accueil Majest Eye ainsi qu'une étude et critique de l'existant pour proposer enfin une solution adéquate

Le deuxième chapitre intitulé **Spécification des besoins** est consacré à l'étude de la méthodologie de travail, l'analyse des besoins fonctionnels et non fonctionnels.

Chapter 1: Contexte général

Introduction

Avant d'aborder l'étude approfondie de notre stage de fin d'études, nous allons décrire brièvement, à travers ce premier chapitre, l'organisme au sein duquel se déroule notre stage. Ensuite, nous présenterons une critique de l'existant afin de dévoiler notre solution proposée. Enfin on va étudier et déterminer les différents choix méthodologiques et technologiques afin de répondre aux objectifs du projet.

1.1 Organisme d'accueil

MajestEYE est un fournisseur de solutions en science des données. Sa mission est de faciliter, d'assister et de réussir la transition vers un modèle commercial axé sur les données. Il propose une technologie analytique avancée exclusive, conçue pour dynamiser toute initiative en analyse avancée, permettant ainsi la transition vers un modèle commercial axé sur les données.



Figure 1.1: Logo MajestEye

Domaine d'activité : Services financiers, gouvernement, santé et pharmacie.

Produits et activités : Grâce à son positionnement, à sa technologie d'Intelligence Artificielle de pointe et à ses data scientists, ingénieurs de données et experts en la matière de haut niveau, MajestEYE est en mesure de fournir des solutions pour un large éventail de défis dans divers secteurs tels que la santé et la pharmacie, les services financiers, etc.

Nom de la société	MajestEYE
Domaine d'activité	fournisseur de solution en science des données
Adresse	Cyber park, Jendouba – 8100 Tunisia
Site web	https://www.majesteye.com

Table 1.1: MajestEye FDS

1.2 Étude de l'existant

1.2.1 Description de l'existant et critique de l'existant

1.2.1.1 Le processus actuel de recrutement au sein de MajestEye:

Le processus de recrutement pour les stages au sein de MajestEye se fait actuellement d'une manière traditionnelle, basé sur l'examen des CV et la réalisation d'entretiens techniques en personne.

L'approche traditionnelle du recrutement, qui repose largement sur des entretiens en personne, présente des limites évidentes. Bien que ces entretiens soient souvent considérés comme précieux pour évaluer les candidats, ils peuvent s'avérer chronophages et coûteux à la fois pour les entreprises et pour les candidats. En effet, la nécessité d'une présence physique lors de ces entretiens peut entraîner une consommation importante de ressources pour les entreprises recruteuses.

Organiser des entretiens en personne implique non seulement la réservation d'un espace approprié pour mener les entretiens, mais aussi la mobilisation des ressources humaines nécessaires pour les coordonner. Cela peut inclure la participation d'un panel d'intervieweurs, la gestion des horaires des candidats et des intervieweurs.

De même, les candidats peuvent faire face à des contraintes géographiques, en particulier lorsqu'ils doivent se déplacer pour des tests en personne. Cela peut restreindre le vivier de candidats et poser des obstacles pour ceux qui ne peuvent pas être présents physiquement.

1.2.1.2 Plateformes d'évaluation technique similaires:

Il existe de nombreuses plateformes dédiées à l'évaluation des compétences techniques.

1.2.1.3 Codingame by CoderPad

CodinGame¹ est une plateforme en ligne utilisée par de nombreuses entreprises pour évaluer les compétences techniques des candidats, offrant une sélection de défis techniques préétablis conçus pour évaluer les capacités des développeurs.

¹site web :<https://www.codingame.com/work/>

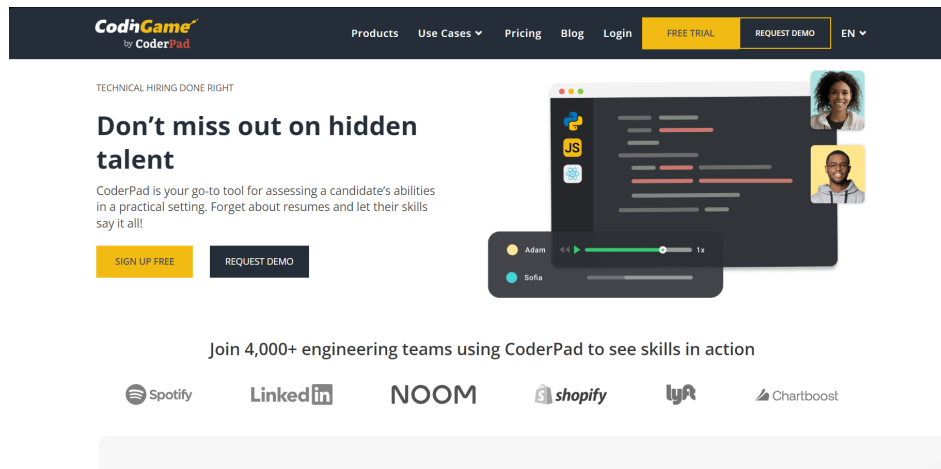


Figure 1.2: Processus de recrutement de MajestEye

Bien que Codingame offre une gamme diversifiée de défis techniques et une plateforme conviviale, elle présente toutefois des limites importantes; Les modèles d'abonnement à paliers et les coûts cachés peuvent rendre son utilisation coûteuse, en particulier pour les petites entreprises avec des budgets limités. Les abonnements à différents niveaux offrent un nombre limité de tests par mois ou par an, ce qui peut rapidement devenir insuffisant. De plus, le manque de personnalisation des tests peut compliquer l'évaluation des compétences spécifiques nécessaires.

1.2.2 Solution proposé

Face aux limites de l'approche traditionnelle du recrutement et des plateformes externes et après avoir examiné les besoins et les exigences de l'entreprise MajestEye, nous proposons la mise en place d'une plateforme interne de test technique pour le recrutement de stagiaires qui permet de rationaliser le processus de recrutement en éliminant la nécessité d'organiser des entretiens en personne. Cela permettra d'économiser du temps et des ressources, car il ne sera plus nécessaire de réserver un espace approprié pour les entretiens ou de mobiliser un panel d'intervieweurs. De plus, l'entreprise peut également personnaliser les tests en fonction des compétences spécifiques requises pour les postes de stagiaires. Cela permettra d'évaluer de manière plus précise les capacités des candidats et de garantir qu'ils possèdent les compétences nécessaires pour réussir dans l'entreprise.

1.3 Choix technologiques et architecture logicielle

1.3.1 Framework et langages

Angular : Il s'agit d'un framework web open source basé sur TypeScript qui permet la création d'applications web structurées et faciles à maintenir.



Figure 1.3: Angular logo

Spring Boot :est un framework Java open-source basé sur les microservices proposé par Spring, particulièrement utile pour le développement des applications web et des microservices.



Figure 1.4: Spring boot

SonarQube :est un outil d'assurance qualité du code qui effectue une analyse approfondie du code et génère un rapport d'analyse pour garantir la fiabilité du code. SonarQube combine l'analyse statique et dynamique du code pour permettre des pratiques continues de qualité du code.



Figure 1.5: Sonar Qube

Langage de Conception UML StarUML est un outil de modélisation de logiciels open source qui permet de créer des diagrammes UML (Unified Modeling Language) pour la conception de logiciels.



Figure 1.6: UML

Tailwind CSS :est un framework CSS qui fournit un ensemble de classes utilitaires préconçues. Ces classes permettent un stylisme rapide et efficace sans avoir à écrire des CSS personnalisés, ce qui favorise la cohérence et l'évolutivité.



Figure 1.7: Tailwind CSS

1.3.2 Environnement logiciel

VsCode : est un éditeur de code source développé par Microsoft. On l'a utilisé comme IDE pour Angular 15 tout au long de la phase de développement.



Figure 1.8: VsCode logo

IntelliJ IDEA :est un éditeur de code source développé par Microsoft. On l'a utilisé comme IDE pour Spring Boot tout au long de la phase de développement.



Figure 1.9: IntelliJ IDEA logo

Postman :est un éditeur de code source développé par Microsoft. On l'a utilisé pour le test d'API, pour construire et tester rapidement des requêtes http.



Figure 1.10: Postmanlogo

StarUML :Logiciel de modélisation UML. Utilisé pour la création des diagrammes.



Figure 1.11: StarUML logo

MySQL MySQL est un système open source de gestion de base de données relationnelle (SGBDR).



Figure 1.12: MySQL logo

1.3.3 L'architecture d'API adoptée :

L'API est un moyen efficace de faire communiquer deux programmes informatiques. Pour lui permettre d'évoluer plus facilement, pour séparer encore plus distinctement back-end et front-end et pour faciliter le développement d'applications web ou pour laisser les membres.

Il existe deux grands protocoles de communications sur lesquels s'adossent les API:

- SOAP
- REST

Si nous choisissons le type d'API celle qui s'adapte le mieux à nos besoins de notre application, nous choisirons l'API REST pour plusieurs raisons:

- simple à utiliser
- flexible
- interopérable

1.3.3.1 Sécurisation d'Api Restful :

Spring Security

C'est un cadre d'authentification et de contrôle d'accès puissant et hautement personnalisable. C'est la norme de facto pour la sécurisation des applications basées sur Spring Boot. C'est une Framework qui fournit à la fois l'authentification et l'autorisation aux applications Java.



Figure 1.13: Spring boot security

Parmi les fonctionnalités offertes par Sprint Security, le cryptage de mot de passe. Comme l'indique la figure ci-dessous.

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="button" value="Export"/>			
Wrap Cell Content: <input type="checkbox"/>			
	firstname	last_name	password
▶	Admin	Admin	\$2a\$10\$.9vVozpOPYxhenu.DKNDWuk0VoGJGq...
	nermine	bouzidi	\$2a\$10\$QHYKh.1P54rBYJikXebcTueZDzMnHT10...
	amina	bouzidi	\$2a\$10\$9mIv6Uq31gXWWuG73FCrXOIBWvMrj...

Figure 1.14: Spring boot security

Protocole d'authentification JWT

JSON Web Token est un standard (RFC 7519) qui définit une solution compacte et autonome pour transmettre de manière sécurisée des informations entre les applications en tant qu'objet structuré au format JSON (JavaScript Objet Notation). Ces informations peuvent être vérifiées et approuvées, car elles sont signées de manière numérisée.



Figure 1.15: Jwt logo

Un JWT est simplement constitué de trois parties séparées par un point.

- Header
- Payload
- La signature

La figure ci-dessous montre la structure de JWT qui à le Header et le Payload et il est signé avec un secret.

Algorithm HS256

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJib3V6aWRpbmVybWluZTQ2QGdtYWlsLmNvbSIsIm5hbWUiOiJuZXJtaW5lIGJvdXppZGkiLCJpYXQiOiJlbnQ3MDk3ODEyODksImV4cCI6MTcwOTc5OTI0X00.Wb1JQS2O5zaWFazk-Qh1qRU6HazaWWy3_hbi7jD2jeE
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256"
}
```

PAYLOAD: DATA

```
{
  "sub": "bouzidnermine46@gmail.com",
  "name": "nermine bouzidi",
  "iat": 1709781289,
  "exp": 1709799289
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☒ secret base64 encoded
```

Signature Verified

SHARE JWT

Figure 1.16: Représentation du jeton JWT de manière codée et décodée.

1.3.4 Architecture logicielle

L'architecture logique d'un système s'appuie sur l'organisation conceptuelle du système logiciel, en définissant les composants, les relations et les propriétés .

Nous avons utilisé l'architecture trois tiers, l'architecture logique du système est divisée en trois couches :

- **Couche présentation** : Elle agit comme interface entre l'utilisateur et le système, gérant l'affichage des informations et la saisie des données.
- **Couche métier** : C'est le cœur du système, où se concentre la logique métier et les règles de traitement des données.
- **Couche des données** : Elle s'occupe du stockage, de la récupération et de la gestion des données persistantes du système.

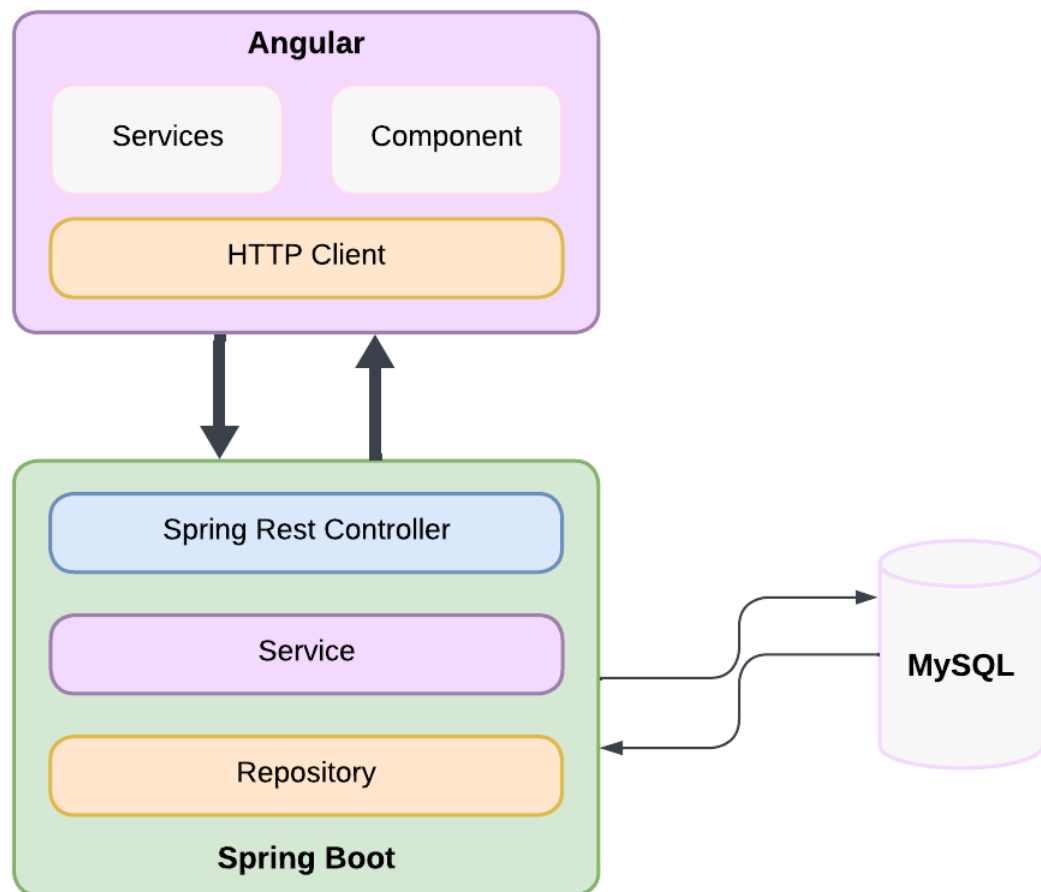


Figure 1.17: Architecture global

Cette figure explique montre les trois couches principales dans notre architecture:

- **frontend** : Application angulaire responsable de l'interaction avec l'utilisateur final,
- **backend** : Serveur Spring Boot en tant que module qui communiquera avec la base de données et l'application angulaire,
- **database** Base de données MySQL pour les entités persistantes

1.3.5 Patron de conception

MVC (Modèle-Vue-Contrôleur) est un patron de conception très répandu. Ce patron de conception a pour but de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

- **Modèle** : C'est le niveau le plus bas du modèle responsable de la gestion des données. Le modèle est responsable de la gestion des données d'application. Il répond à la demande de vue et aux instructions du contrôleur afin de se mettre à jour.
- **Vue** : Elle est responsable de l'affichage partiel ou total des données à l'utilisateur dans un format particulier.
- **Contrôleur** : C'est un code logiciel qui contrôle les interactions entre le modèle et la vue. Le contrôleur répond aux entrées de l'utilisateur et effectue des interactions sur les objets du modèle de données. Le contrôleur reçoit les entrées, les valide, puis effectue des opérations qui modifient l'état du modèle de données.

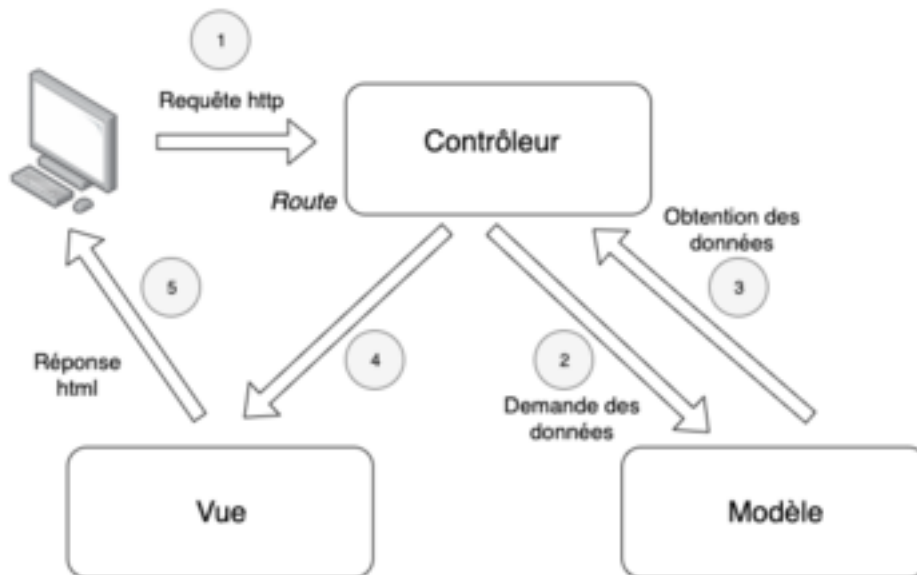


Figure 1.18: Caption

1.4 Méthodologie de travail

Afin de garantir un déroulement fluide de notre projet, nous avons opté pour la méthodologie agile Scrum, qui ajoute de la flexibilité et de l'organisation au projet en termes de temps et de qualité

1.4.1 Principe de la méthodologie Scrum

Scrum est une méthode agile dédiée à la gestion de projet. Cette méthode de gestion a pour objectif d'améliorer la productivité de son équipe. Elle met en avant la rentabilité et surtout la satisfaction du client. L'équipe de développement s'organise elle-même pour trouver la meilleure façon de produire les étapes les plus privilégiées. En d'autres

termes, Scrum consiste à réaliser les tâches petit à petit par ordre de priorité avec des phases de contrôle et d'échange avec le client.

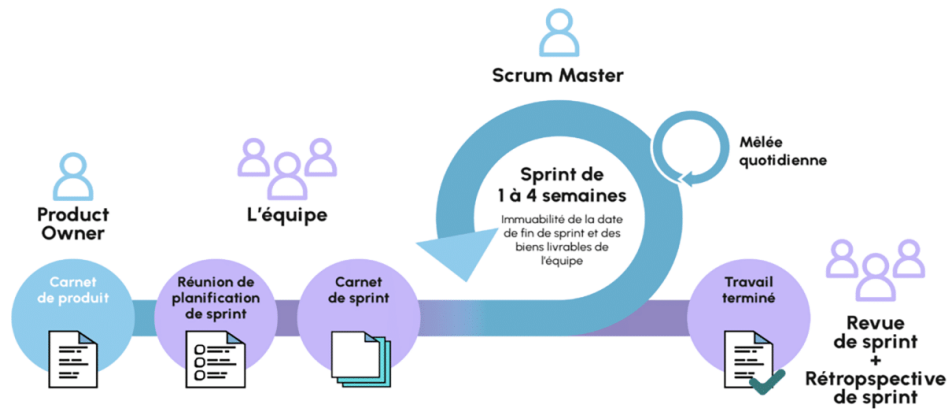


Figure 1.19: Cycle de vie d'un produit par la methodologie Scrum

Conclusion

En somme, ce chapitre a exposé le contexte de travail du projet et a précisé la problématique de la société d'accueil ainsi que les solutions envisagées pour y remédier. Le chapitre suivant se concentrera sur les détails des spécifications , étape essentielle pour garantir sa réussite.

Chapter 2: Spécification des besoins

Introduction

Dans ce chapitre, nous allons étudier et déterminer les différents choix méthodologiques et technologiques afin de répondre aux objectifs du projet.

2.1 Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents et de préciser les risques les plus critiques.

2.1.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. L'identification des acteurs est une étape importante dans un projet de développement. Elle permet d'apprécier les interrelations et de mieux mesurer l'influence de chacun des groupes d'acteurs sur l'action à mener.

Après avoir mené une étude du système, trois acteurs interviennent dans notre plateforme :

- **Administrateur** : Un utilisateur autorisé à gérer et administrer la plateforme
- **Évaluateur** : Un utilisateur chargé de l'évaluation des candidats et des tests
- **Candidat** : Un utilisateur capable d'inscrire et de passer un test.

2.1.2 Besoins fonctionnels

Les besoins fonctionnels représentent les exigences du futur système. Donc, les actes du système sont définis par les besoins fonctionnels. De ce fait, le projet proposé doit satisfaire d'une façon informelle les fonctionnalités suivantes.

- **Administrateur**
 - **Authentification** : permet à l'administrateur de s'authentifier.

- **Gestion des utilisateurs** : permet à l’administrateur de gérer les utilisateurs.
- **Gestion compte** : permet à l’administrateur de modifier ses données personnelles de son compte.
- **Evaluateur**
 - **Authentification** : permet à l’administrateur de s’authentifier.
 - **Gestion compte** : permet à l’évaluateur de modifier ses données personnelles de son compte.
 - **Traitement des candidatures** : permet à l’administrateur de gérer les utilisateurs.
 - **Gestion des QCMs** : Permet à l’évaluateur de gérer les questions à choix multiples.
 - **Gestion des problèmes de codage** : Permet à l’évaluateur de gérer les problèmes de codage.
 - **Gestion des tests** : Permet à l’évaluateur de créer, modifier et supprimer des tests.
 - **Évaluation des tests** : Permet à l’évaluateur d’évaluer les tests soumis par les candidats.
- **Candidat**
 - **Authentification** : permet au candidat de s’authentifier.
 - **Inscription**: permet au candidat de gérer les utilisateurs.
 - **Passage de test** :Permet au candidat de s’inscrire.”

2.1.3 Besoins non fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d’objectifs que doit atteindre le système.

- **Ergonomie** :Il est important pour les utilisateurs de naviguer facilement dans l’application et de trouver rapidement ce qu’ils recherchent, en utilisant des interfaces simples et des menus clairs.
- **Performance** :Lorsqu’une page contient une grande quantité de contenu, cela peut ralentir le temps de chargement de la page, ce qui peut frustrer les utilisateurs. Ainsi il est important d’utiliser la pagination qui permet d’améliorer les performances d’un site web en réduisant le temps de chargement des pages.
- **Modularité** : Le code doit être facile à maintenir pour des raisons de réutilisation et de modification.

2.2 Diagramme de cas d'utilisation

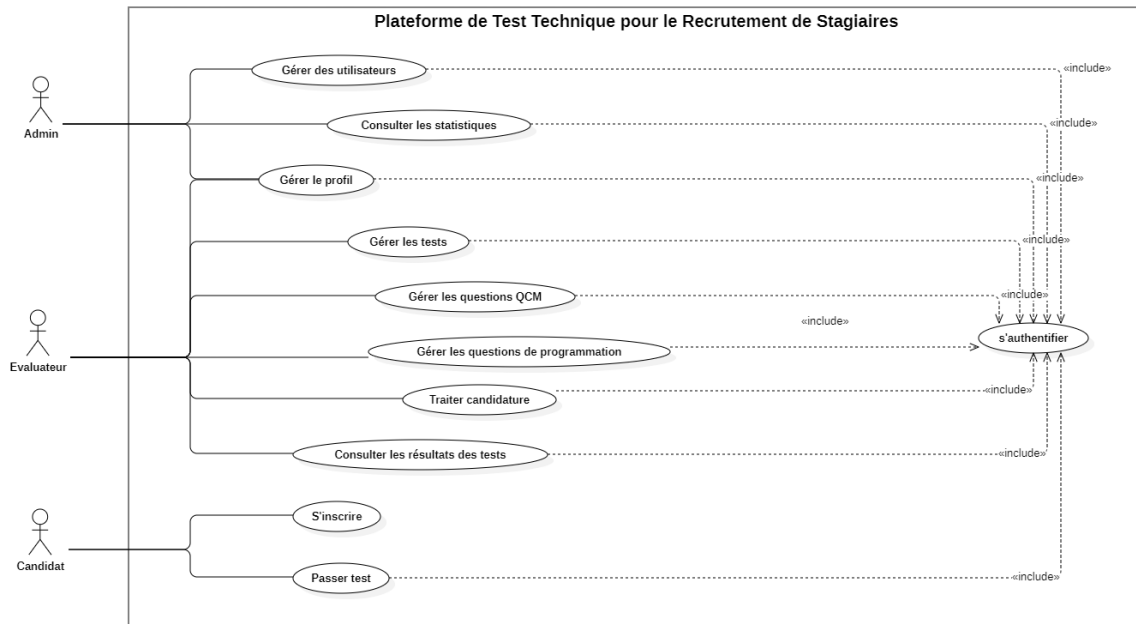


Figure 2.1: Diagramme des cas d'utilisation global

2.3 Pilotage du projet par Scrum

2.3.1 Équipe Scrum

La méthodologie Scrum repose sur la collaboration étroite d'une équipe pour mener une bonne gestion du projet.

Voici un aperçu de notre équipe Scrum.

Product Owner	Mr Jihed Hannachi
Scrum Master	Mr Wissem Inoubli
Development team	Nermine Bouzidi

Table 2.1: Equipe Scum

2.3.2 Backlog du produit

Le carnet de produit consiste en une liste ordonnée regroupant toutes les exigences du projet. Il constitue en effet l'unique source des besoins pour tous les changements à effectuer sur le produit. Il s'agit d'un document qui évolue constamment au cours de la vie de ce dernier.

Avant de commencer à réaliser notre backlog du produit, il est préférable d'adopter une méthode de priorisation des stories, conformément à la méthodologie SCRUM.

Pour ce faire, nous avons opté pour la méthode de priorisation MOSCOW.
 Cette figure ci-dessous explique la perspective de priorisation MOSCOW :

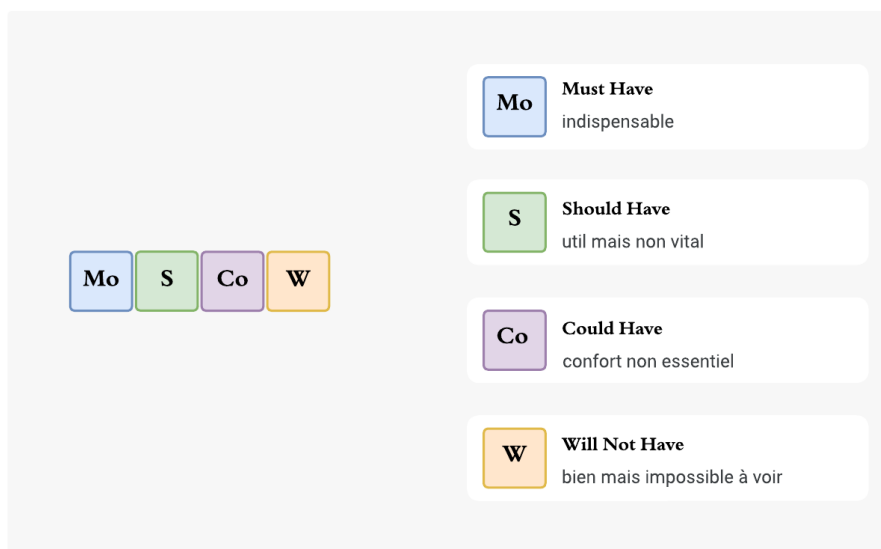


Figure 2.2: La priorisation de MOSCOW

ID	Fonctionnalité	User Story ID	User Story		Priorité
			Entant que	Je souhaite	
1	Authentification	1.1	Utilisateur	M'authentifier pour accéder à l'application.	M
		1.2	Candidat	m'inscrire	M
2	Gestion compte	2.1	Administrateur	Gérer mon compte	M
		2.2	Evaluateur	Gérer mon compte	M
3	Gérer les utilisateurs	3.1	Administrateur	Ajouter un utilisateur.	M
		3.2	Administrateur	Modifier un utilisateur.	M

		3.3	Administrateur	Supprimer un utilisateur.	M
		3.4	Administrateur	Consulter profil d'un utilisateur.	M
4	Traitement des candidatures	4.1	Evaluateur	Consulter profil d'un candidat.	
		4.2	Evaluateur	Accepter demande d'inscription d'un candidat	
		4.3	Evaluateur	Refuser demande d'inscription d'un candidat	
5	Gestion des QCMs	5.1	Evaluateur	Ajouter un QCM	
		5.2	Evaluateur	Modifier un QCM	M
		5.3	Evaluateur	Supprimer un QCM	M
		5.4	Evaluateur	Consulter un QCM	M
6	Gestion des problèmes de codage	6.1	Evaluateur	Ajouter un problème	
		6.2	Evaluateur	Modifier un problème	M
		6.3	Evaluateur	Supprimer un problème	M
		6.4	Evaluateur	Consulter un problème	M
7	Gestion des tests	7.1	Evaluateur	Créer un test	
		7.2	Evaluateur	Modifier un test	M
		7.3	Evaluateur	Supprimer un test	M

		7.4	Evaluateur	Consulter un test	M
8	Passage de test	8.1	Candidat	Passer un test	
9	Évaluation des tests	9.1	Evaluateur	Consulter les résultats des test	M
		9.2	Evaluateur	Évaluer manuellement les réponses des questions de programmation	
10	Visualisation des statistiques sur le tableau de bord	10.1	Administrateur	consulter les statistiques sur le tableau de bord	

2.3.3 Planification des sprints

	Sprint1 de (01/02/2024 à 0/0/0000)	Sprint2 de (0/0/0000 à 0/0/0000)	Sprint3 de (0/0/0000 à 24/05/2024)
ID	1-2-3	4-5-6	7-8-9-10

Table 2.3: Planification des sprints

2.4 Conclusion

Ce chapitre nous permet de présenter divers besoins fonctionnels et non fonctionnels ainsi que les acteurs de notre système.