



Ministère de l'Enseignement Supérieur et de la  
Recherche Scientifique  
Université de la Manouba  
Institut Supérieur des Arts Multimédia  
Manouba



# Rapport de Stage de Fins d'Etudes

Elaboré en vue de l'obtention du diplôme de

## Licence en Science Informatique

Parcours

[Informatique et Multimédia]

---

Plateforme de Test Technique pour le Recrutement de  
Stagiaires

---

Réalisé par: Nermine Bouzidi

Encadré par: Tarek Hamrouni

Au sein de l'entreprise:



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

<b>Introduction</b>	<b>1</b>
<b>1 Contexte général</b>	<b>2</b>
1.1 Organisme d'accueil . . . . .	2
1.2 Étude de l'existant . . . . .	3
1.2.1 Description de l'existant . . . . .	3
1.2.2 Critique de l'existant . . . . .	3
1.2.3 Solution proposé . . . . .	3
<b>2 Specification des besoins</b>	<b>5</b>
2.1 Méthodologie de travail . . . . .	5
2.1.1 Principe de la méthodologie Scrum . . . . .	5
2.2 L'adaptation de la méthodologie Scrum au project . . . . .	6
2.2.1 Equipe Scrum . . . . .	6
2.2.2 Carnet de produit (Product Backlog) . . . . .	6
2.2.3 Le planning des sprints . . . . .	7
2.2.4 Le plan des releases . . . . .	7
2.3 Spécification des besoins fonctionnels . . . . .	8
2.3.1 Identification des acteurs . . . . .	8
2.3.2 Identification des besoins fonctionnels . . . . .	8
2.4 Spécification des besoins non fonctionnels . . . . .	9
<b>3 Conception</b>	<b>11</b>
3.1 Choix de la méthodologie de conception . . . . .	11
3.1.1 Vue dynamique . . . . .	11
3.1.2 Vue statique . . . . .	13
<b>4 Release 1</b>	<b>14</b>
4.1 Sprint 0 : Mise en place de l'environnement . . . . .	14
4.1.1 Outils de développement . . . . .	14
4.1.2 Les technologies . . . . .	15
4.1.3 Architecture de sécurité . . . . .	16
4.1.4 Base de données . . . . .	18
4.2 Sprint 1 : Authentification . . . . .	18

4.2.1	Raffinement de cas d'utilisation "Authentification" . . . . .	19
4.2.2	Description textuelle du cas d'utilisation "Authentification" . .	20
4.2.3	Interface "Authentification" . . . . .	20
4.3	Sprint 2 : Auth . . . . .	20
4.4	Sprint 3 : Mise en place de l'environnement . . . . .	20
<b>5</b>	<b>Release 2</b>	<b>22</b>
<b>6</b>	<b>Release 3</b>	<b>23</b>
	<b>Conclusion</b>	<b>24</b>

# Liste des figures

1.1	Logo MajestEye . . . . .	2
2.1	Cycle de vie d'un produit par la methodologie Scrum . . . . .	5
2.2	Diagramme des cas d'utilisation . . . . .	10
3.1	Diagramme d'activité "inscription" . . . . .	12
4.1	VsCode logo . . . . .	14
4.2	Intellij IDEA logo . . . . .	15
4.3	Postmanlogo . . . . .	15
4.4	StarUML logo . . . . .	15
4.5	Angular 15 logo . . . . .	15
4.6	Spring boot . . . . .	16
4.7	Sonar Qube . . . . .	16
4.8	Tailwind CSS . . . . .	16
4.9	Spring boot security . . . . .	17
4.10	Spring boot security . . . . .	17
4.11	Jwt logo . . . . .	18
4.12	Représentation du jeton JWT de manière codée et décodée. . . . .	18
4.13	Diagramme du cas d'utilisation "Authentication" . . . . .	19
4.14	Interface "Login" . . . . .	20
4.15	Home page . . . . .	21

# Liste des tables

1.1	MajestEye FDS . . . . .	3
2.1	Equipe Scum . . . . .	6
2.2	Backlog de produit . . . . .	6
2.3	Planning des sprints . . . . .	7
2.4	Plan des Releases . . . . .	8
2.5	Liste des acteurs . . . . .	8
4.1	Backlog de sprint 0 . . . . .	14
4.2	Backlog de sprint 1 . . . . .	19
4.3	Description textuelle du cas d'utilisation "Authentification" . . . . .	20

# 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.

Dans le troisième chapitre, nommé **Conception** nous présentons une analyse et une conception détaillées de cas d'utilisation.

Les trois derniers chapitres constituent le corps du rapport, ils décrivent l'étape de développement des Releases de notre projet dont chacun d'entre eux est découpé en sprints selon l'ordre de réalisation en appliquant la méthodologie SCRUM.

# 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.

### 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.

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

Table 1.1: MajestEye FDS

## 1.2 Étude de l'existant

### 1.2.1 Description de l'existant

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.

### 1.2.2 Critique de l'existant

L'approche traditionnelle présente des limites, les entretiens en personne, bien que précieux, peuvent s'avérer chronophages et coûteux, autant pour les entreprises que pour les candidats. Ils exigent, une présence physique, ce qui peut nécessiter de nombreuses ressources pour l'entreprise qui recrutent.

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.

L'idée d'utiliser un logiciel de recrutement externe tel que Jobaffinity peut rationaliser certains aspects du processus, mais son installation et sa maintenance peuvent être coûteuses, notamment en termes d'achat initial et d'abonnement continu. De plus, il peut ne pas avoir la capacité d'évaluer de manière exhaustive les compétences spécifiques nécessaires pour un rôle particulier.

### 1.2.3 Solution proposé

Les problèmes de l'analyse dégagés lors de l'analyse de la situation actuelle rendent nécessaire d'élaboration d'une solution informatisée afin de faciliter le processus de recrutement des stagiaires.

Cette solution consiste à développer une plateforme interne de test technique pour le recrutement de stagiaires qui a pour objectifs principaux :

- Amélioration du processus de recrutement.
- Réduire les coûts de recrutement (temps et ressources).

- Production de tests qui se concentrent directement sur les compétences et les connaissances particulières de la société.
- Meilleure sélection des personnes compétentes et motivées.
- Expérience de candidature plus fluide

# Chapter 2: Specification 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 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

### 2.1.1 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.

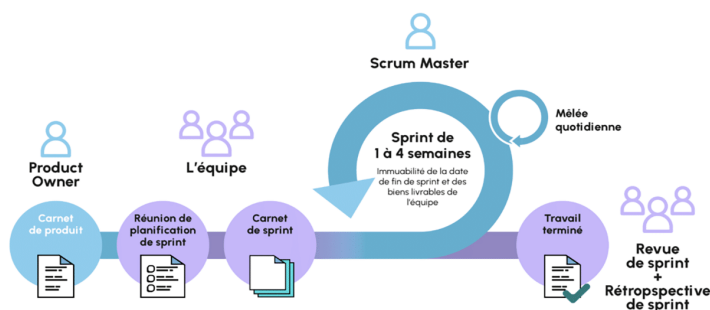


Figure 2.1: Cycle de vie d'un produit par la methodologie Scrum

## 2.2 L'adaptation de la méthodologie Scrum au project

### 2.2.1 Equipe Scrum

<b>Product Owner</b>	Mr Jihed Hannachi
<b>Scrum Master</b>	Mr Wissem Inoubli
<b>Development team</b>	Nermine Bouzidi

Table 2.1: Equipe Scum

### 2.2.2 Carnet de produit (Product Backlog)

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.

User Story	Estimation
Mise en place de l'environnement	
Inscription des candidats.	
Authentification sécurisé	
Gestion d'accès utilisateur (accepter ou refuser par l'administrateur	
Gestion des utilisateurs par l'administrateur.	
Gestion des quiz( éditer et supprimer les tests QCM).	
Gestion des problèmes (créer, éditer et supprimer les tests Problèmes).	
Intégration d'une fonctionnalité de minuterie pour limiter le temps de réponse.	
Intégration de SonarQube pour l'analyse automatique du code des questions de programmation.	
Automatisation de l'évaluation des réponses aux questions à choix multiples.	
Génération de rapports détaillés sur les performances individuelles de chaque candidat.	

Table 2.2: Backlog de produit

### 2.2.3 Le planning des sprints

Dans cette partie, nous cherchons à aménager les éléments (User Stories) du Carnet de Produit en Sprints. D'ailleurs, un sprint, bien qu'il présente une partie incomplète du projet, doit être livrable, exécutable et réutilisable. Il est donc indispensable de délimiter les sprints d'une manière à respecter cette règle.

Notre regroupement final a évoqué huit sprints à réaliser.

Nous avons considéré qu'une journée de travail est égale à huit heures.

Numéro de Sprint	User Story
Sprint 0	Mise en place de l'environnement
Sprint 1	Authentification sécurisée Inscription des candidats
Sprint 2	Gestion d'accès utilisateur (accepter ou refuser par l'administrateur Gestion des utilisateurs par l'administrateur
Sprint 3	Développement d'une interface conviviale permettant aux candidats de passer les tests.( quiz et code) Intégration d'une fonctionnalité de minuterie pour limiter le temps de réponse.
Sprint 4	Gestion des quiz( éditer et supprimer les tests QCM)
Sprint 5	Automatisation de l'évaluation des réponses aux questions à choix multiples.
Sprint 6	Gestion des problèmes (créer, éditer et supprimer les tests Problèmes)
Sprint 7	Intégration de SonarQube pour l'analyse automatique du code des questions de programmation.
Sprint 8	Génération de rapports détaillés sur les performances individuelles de chaque candidat.

Table 2.3: Planning des sprints

### 2.2.4 Le plan des releases

Dans cette partie, nous regroupons deux ou plusieurs sprints sous un seul thème dénommé 'Release'.

Numéro de Release	Nom de Release	Les Sprints
<b>1</b>	Generation des Quiz	Sprint0 Sprint1 Sprint2
<b>2</b>	Generation des Quiz	sprint3 Sprint4 Sprint5
<b>3</b>	Generation des problème de code	Sprint6 Sprint7 sprint8

Table 2.4: Plan des Releases

## 2.3 Spécification des besoins fonctionnels

### 2.3.1 Identification des acteurs

Acteur	Role
<b>Administrateur</b>	Mr Wissem Inoubli
<b>Candidats</b>	Nermine Bouzidi

Table 2.5: Liste des acteurs

### 2.3.2 Identification des 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.

- Le système doit permettre l'authentification des utilisateurs par un login et un mot de passe pour accéder aux différentes fonctionnalités.
- Permettre aux administrateurs de gérer les utilisateurs.
- Permettre aux administrateurs de gérer le contrôle d'accès des utilisateurs.
- Permettre aux administrateurs de gérer les quizzes.
- Permettre aux administrateurs de gérer les problèmes de programmation.
- Permettre aux administrateurs de consulter les résultats.



- Permettre aux administrateurs de consulter les statistiques.
- Permettre aux utilisateurs d'inscrire.
- Permettre aux utilisateurs de passer des tests.

## 2.4 Spécification des 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.

On peut décrire nos besoins non fonctionnels en deux parties:

- **Sécurité** : L'application assure la sécurité d'utilisation par la nécessité d'attribuer des rôles à chaque utilisateur pour attribuer les activités de chacun d'eux, ainsi les comptes des utilisateurs sont sécurisés par mot de passe crypté, par mécanisme de vérification d'authentification par Token JWT qui doit implémenter pour chaque tentative de connexion
- **Performance** : L'application doit être performante et doit répondre à toutes les exigences des clients d'une manière optimale et non ambiguë.
- **Convivialité** : L'application doit être facile à utiliser..

## Diagramme des cas d'utilisation

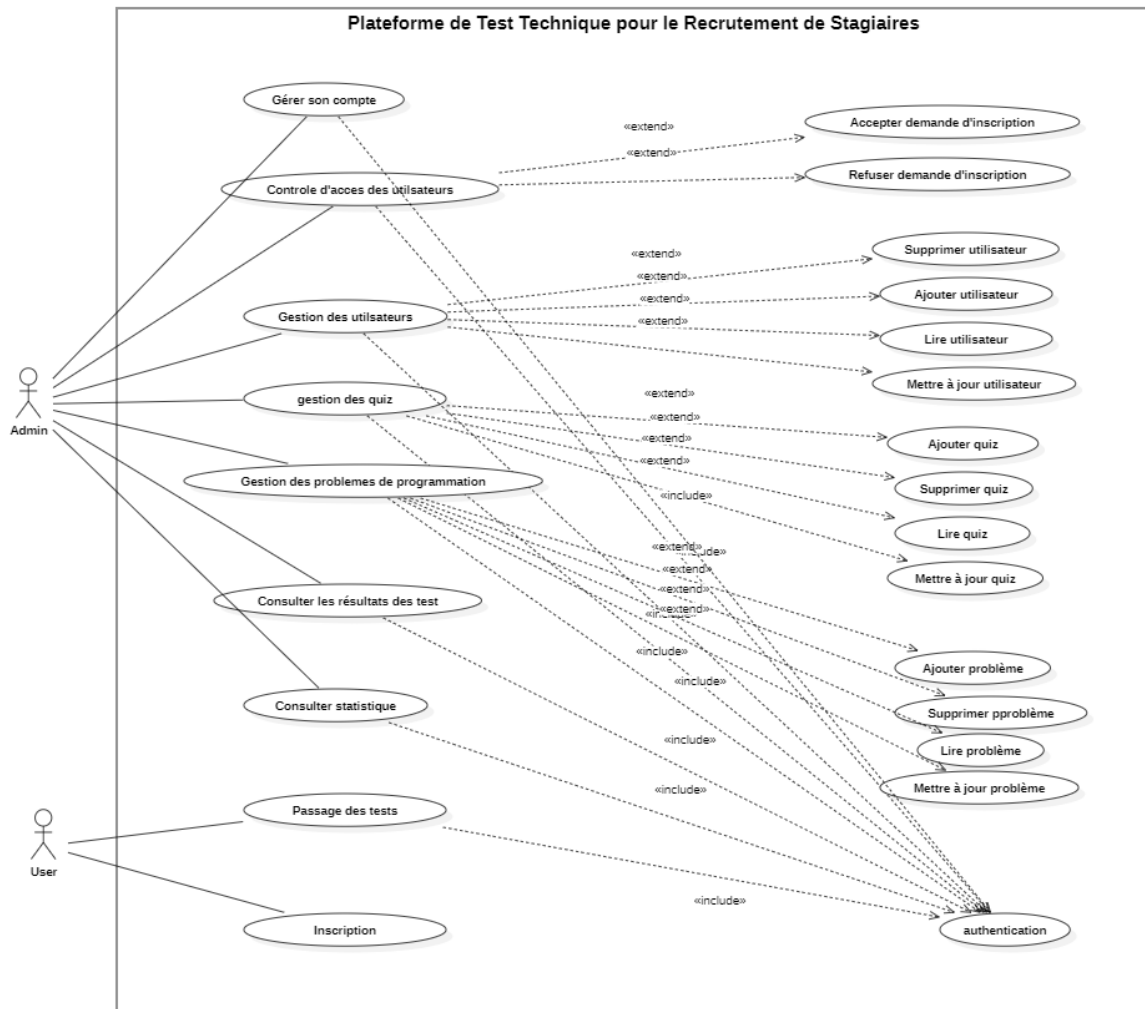


Figure 2.2: Diagramme des cas d'utilisation

# Chapter 3: Conception

## Introduction

Dans ce chapitre, nous allons donner un aperçu sur les procédures de conception et de réalisation.

### 3.1 Choix de la méthodologie de conception

Dans le cadre de notre projet, nous avons opté pour le langage UML comme un langage de conception.

Le langage de Modélisation Unifié, de l'anglais Unified Modeling Language, est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines de développement logiciel et en conception orienté objet.

#### 3.1.1 Vue dynamique

Les diagrammes de vue dynamiques permettent de mieux comprendre et de décrire le comportement des objets et leurs interactions. Ces modèles permettent une vision objective du fonctionnement du système. Ils ont pour but de mettre en évidence les relations temporelles entre les objets. Nous allons représenter les aspects dynamiques entre les classes de notre application en utilisant des diagrammes d'activités et des diagrammes de séquence.

### 3.1.1.1 Diagramme d'activité

Diagramme d'activité du cas d'utilisation inscription

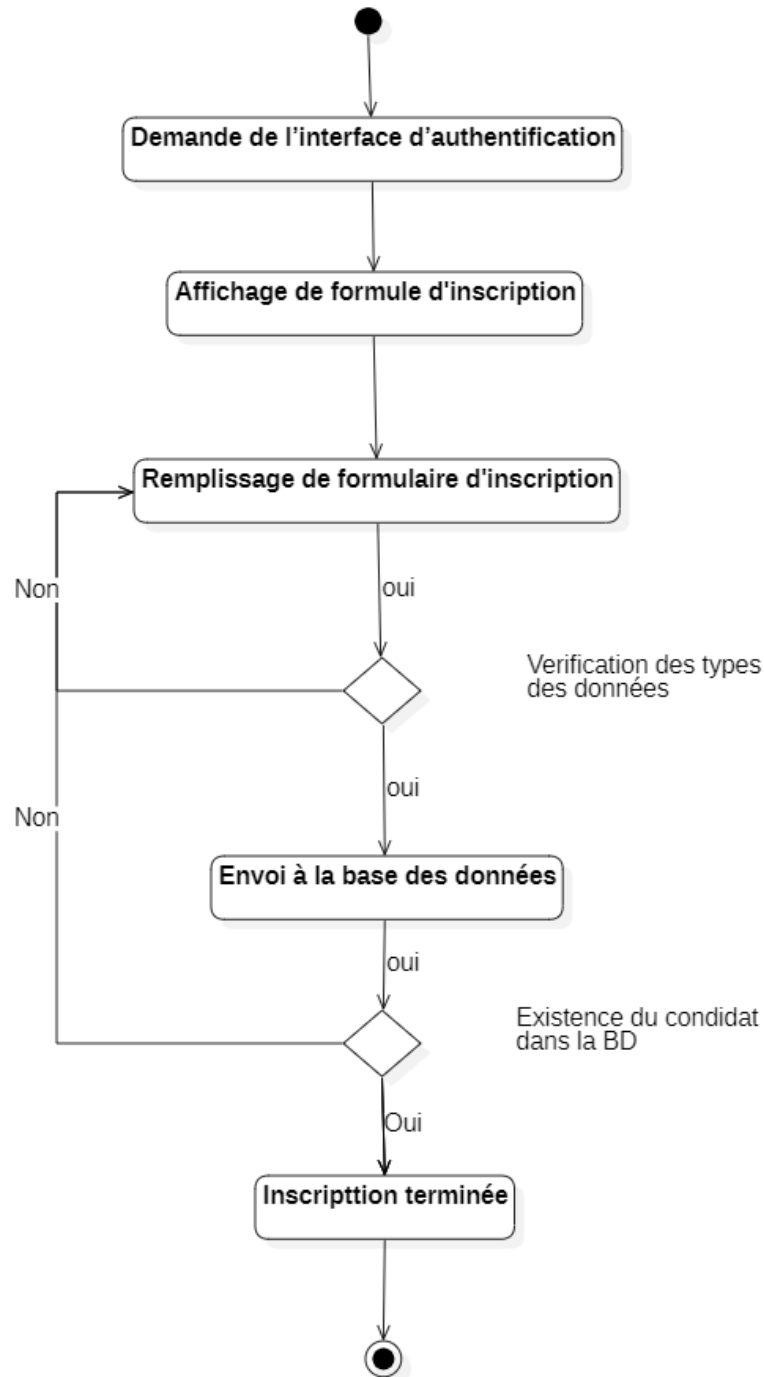


Figure 3.1: Diagramme d'activité "inscription"

Diagramme d'activité du cas d'utilisation inscription

3.1.1.2 Diagramme de séquence

3.1.2 Vue statique

3.1.2.1 Diagramme de classes

# Chapter 4: Release 1

## Introduction

Ce Release comporte trois sprints

### 4.1 Sprint 0 : Mise en place de l'environnement

Suite à la réunion du "Sprint Planning Meeting" qui est organisée avant chaque sprint, nous avons commencé par la définition de l'objectif et de l'estimation temporelle du premier sprint de cette partie.

- **Date de début de sprint : 01/02/2024**
- **Date de fin de sprint : 14/02/2024**

Élément de backlog	Tache
Conception	
Choix des outils de développement	
Choix des technologies	
Configuration du serveur et de la base de données	

Table 4.1: Backlog de sprint 0

#### 4.1.1 Outils de développement

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



Figure 4.1: VsCode logo

**IntelliJ IDEA** :est un éditeur de code source développé par Microsoft.On a utilisé comme IDE pour Spring boot tous le long de phase de développement.



Figure 4.2: IntelliJ IDEA logo

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



Figure 4.3: Postmanlogo

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



Figure 4.4: StarUML logo

#### 4.1.2 Les technologies

**Angular 15** :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 4.5: Angular 15 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 4.6: 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 4.7: Sonar Qube

**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 4.8: Tailwind CSS

### 4.1.3 Architecture de sécurité

#### 4.1.3.1 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

#### 4.1.3.2 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 4.9: 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:	
		Export:	
		Wrap Cell Content:	
	firstname	last_name	password
▶	Admin	Admin	\$2a\$10\$.9vVozpOPYxhenu.DKNDWuk0VoGJGq...
	nermine	bouzidi	\$2a\$10\$QHYKh.1P54rBYJikXebcTueZDzMnHT10...
	amina	bouzidi	\$2a\$10\$9mIv6Uq31gXWWuG73FCrXOIBWvMrj...

Figure 4.10: 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 4.11: 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.eyJzdWIiOiJib3V6aW
RpbmVybWluZTQ2QGdtYWlsLmNvbSIsIm5hbWUiO
iJuZXJtaW51IGJvdXppZGkiLCJpYXQiOiE3MDk3
ODEyODksImV4cCI6MTcwOTc5OTI4OX0.Wb1JQS2
05zaWFazk-Qh1qRU6HazaWWy3_hbi7jD2jeE
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "sub": "bouzidinermine46@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 4.12: Représentation du jeton JWT de manière codée et décodée.

#### 4.1.4 Base de données

### 4.2 Sprint 1 : Authentification

Suite à la réunion du "Sprint Planning Meeting" qui est organisée avant chaque sprint, nous avons commencé par la définition de l'objectif et de l'estimation temporelle du premier sprint de cette partie.

- Date de début de sprint : 15/02/2024
- Date de fin de sprint : 1/03/2024

Élément de backlog	Tache
Conception	
Choix des outils de développement	
Choix des technologies	
Configuration du serveur et de la base de données	

Table 4.2: Backlog de sprint 1

#### 4.2.1 Raffinement de cas d'utilisation "Authentification"

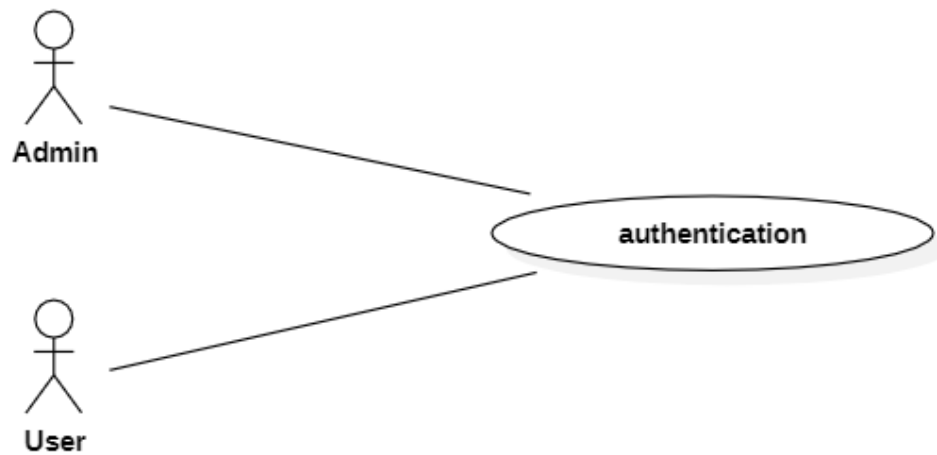


Figure 4.13: Diagramme du cas d'utilisation "Authentification"

### 4.2.2 Description textuelle du cas d'utilisation "Authentification"

<b>Acteur</b>	-Administrateur -Condidat
<b>Pré-Condition</b>	
<b>Scénario principal</b>	1)L'utilisateur demande l'interface d'authentification 2)Le système affiche l'interface d'authentification 3) L'utilisateur saisit ses informations de connexion 4)L'utilisateur valide la connexion 5)Le système vérifie les données saisies 6) le système dirige l'utilisateur vers son espace
<b>Post-Condition</b>	-L'utilisateur est authentifié
<b>Scénario alternatif</b>	si les données saisies sont invalides le système affiche un message d'erreur et le scénario reprend à partir de 3.

Table 4.3: Description textuelle du cas d'utilisation "Authentification"

### 4.2.3 Interface "Authentification"

MAJESTEYE

Sign in to your account

Email

name@company.com

Password

Remember me

Forgot password?

Sign in

Don't have an account yet? Sign up

Figure 4.14: Interface "Login"

## 4.3 Sprint 2 : Auth

## 4.4 Sprint 3 : Mise en place de l'environnement

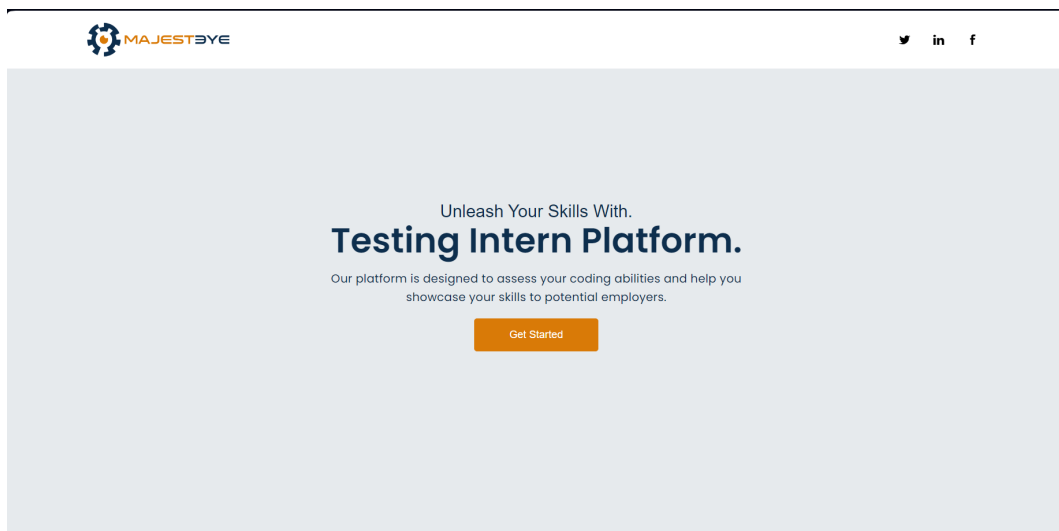


Figure 4.15: Home page

## Chapter 5: Release 2

## Chapter 6: Release 3

# Conclusion