

Table des matières

1	Sprint 0	1
1.1	Introduction	2
1.2	Analyse des besoins	2
1.2.1	Identification des acteurs	2
1.2.2	Identification des besoins	3
1.3	Pilotage du projet avec Scrum	4
1.3.1	Les fonctionnalités du backlog	4
1.3.2	Diagramme des cas d'utilisation global	8
1.3.3	Diagramme de classes global	9
1.3.4	Planification des sprints	10
1.4	Spécification architecturale	12
1.4.1	Architecture logique : Architecture MVC	12
1.4.2	Architecture physique	13
1.5	Environnement de travail	14
1.5.1	Environnement matériel	14
1.5.2	Environnement de développement	14
1.5.3	Environnement logiciel	14
1.5.4	Choix de technologies	16
1.6	Conclusion	17

Table des figures

1.1	Diagramme des cas d'utilisation global	9
1.2	Diagramme de classes	10
1.3	Diagramme de gantt	11
1.4	Sprint 1	11
1.5	Sprint 2	11
1.6	Sprint 3	12
1.7	Sprint 4	12
1.8	Architecture logique de notre système	13
1.9	Architecture physique de notre système	13
1.10	Logo du Visual studio code	14
1.11	Logo du My SQL Server	14
1.12	Logo du MySQL	14
1.13	Logo du VueJs	15
1.14	Logo du Axios	15
1.15	Logo du Django-Python	15
1.16	Logo du Django Rest Framework	15
1.17	Logo du Figma	15
1.18	Logo du Jira	15
1.19	Logo du Github	16

Liste des tableaux

1.1	Les acteurs dans le système et leurs rôles	2
1.2	Pilotage du projet avec scrum	4
1.3	Les fonctionnalités du backlog	8
1.4	Les configurations de PCs du travail	14
1.5	Nombre de requêtes par minute pour des systèmes d'intelligence artificielle . .	16

Liste des sigles et acronymes

CV	<i>Curriculum Vitae</i>
TPE	<i>Très Petite Entreprise</i>
PME	<i>Petite ou Moyenne Entreprise</i>
CRM	<i>Customer Relationship Management</i>
SEO	<i>Search Engine Optimization</i>
MVC	<i>Model View Controller</i>
API	<i>Application Programming Interface</i>

Chapitre 1

Sprint 0

1.1 Introduction

Dans ce chapitre, nous allons identifier les acteurs de notre projet et leurs rôles, ainsi que les besoins fonctionnels et non fonctionnels. Nous détaillons le travail par la méthodologie choisie Scrum en précisant le matériel de base , les technologies et les langages de programmation utilisés pour la mise en place de l'environnement de travail.

1.2 Analyse des besoins

L'analyse et la spécification des besoins représentent la première phase du cycle de développement. Elle sert à identifier les acteurs réactifs du système et leur associer chacun l'ensemble de rôles avec lesquelles il intervient dans l'objectif de satisfaire les exigences de la totalité des utilisateurs. Nous exposons dans ce qui suit leurs besoins fonctionnels ainsi que les besoins non fonctionnels communs à tous les acteurs .

1.2.1 Identification des acteurs

Un acteur représente une entité externe qui interagit avec le système. En réponse à l'action d'un acteur, le système fournit un service qui répond à ses besoins. Notre plateforme présente 3 acteurs identifiés dans le tableau 2.1.

Acteurs	Roles
Administrateur du système	Il consulte les comptes des utilisateurs, bloque un utilisateur en cas de comportement inapproprié et suivre l'utilisation de la plateforme.
Etudiant	Il peut créer un compte et gérer son profil afin de consulter, filtrer les offres de stage et postuler sur ceux qui l'intéressent, saisir son CV pour générer un portfolio, communiquer avec les entreprises.
Représentant de l'entreprise	Il peut créer un compte et gérer son profil dans le but de publier et gérer des offres de stage et PFE books ainsi que générer un PFE book et filtrer les profils des étudiants selon leurs préférences et communiquer avec eux.

Tab. 1.1 : Les acteurs dans le système et leurs rôles

1.2.2 Identification des besoins

Besoins fonctionnels

Les besoins fonctionnels représentent ce que le système offre comme services aux utilisateurs. La finalité de notre projet est de concevoir une plateforme nommée SW Connect qui se divise en 3 parties :

- Partie Administrateur du système :

- Gestion des comptes : Consulter et bloquer les comptes des utilisateurs.
- Suivi de l'utilisation de la plateforme : consulter dashboard qui contient des informations précises et détaillées sur la performance de la plateforme.

- Partie Etudiant :

- Inscription et gestion d'un compte.
- Génération automatique des portfolios personnalisables : Importer un CV dans le format PDF, choisir un modèle de portfolio et le partager.
- Consulter, filtrer et postuler sur des offres et les ajouter aux favoris.
- Communication avec l'entreprise via la boîte de messagerie.
- Recevoir des notifications de refus ou acception de demandes.
- Consulter l'historique de son compte.

- Partie Représentant de l'entreprise :

- Inscription et gestion d'un compte.
- Publication des offres de stage et pfe book.
- Génération automatique des pfe books personnalisables : sélectionner les offres publiées, choisir un modèle de pfe book et le partager.
- Consulter, filtrer des profils des étudiants et les ajouter aux favoris.
- Communication avec les candidats via la boîte de messagerie.
- Recevoir des notifications de nouvelles demandes de stage.
- Consulter l'historique de son compte.

Les besoins non fonctionnels

• Interactivité : Offrir une interface utilisateur claire et simple pour faciliter la navigation, la recherche de stages, la candidature et la communication entre les étudiants et les entreprises.

• Concurrence : Notre système est capable de gérer d'une manière effective les interactions simultanées entre plusieurs utilisateurs, en évitant les conflits et en assurant la cohérence des données.

• Sécurité : Pour pouvoir accéder aux interfaces du système, chaque utilisateur doit saisir son login et son mot de passe.

• Disponibilité : Une bonne gestion et configuration de la base de données et du serveur.

1.3 Pilotage du projet avec Scrum

Nous présentons dans cette section l'équipe de développement qui est constituée, dans notre cas, de deux personnes chargées de la réalisation du projet, le Product Owner et le Scrum Master. Dans le tableau 2.2, nous présentons les rôles, les acteurs associés et leurs missions.

Role	Mission	Acteur
Scrum team	Conception Développement Tests et Validation Déploiement	DIMASSI Khaoula KITAR Chiraz
Product Owner Scrum Master	Approbation du projet	DIMASSI Aymen

Tab. 1.2 : Pilotage du projet avec scrum

1.3.1 Les fonctionnalités du backlog

Le backlog du produit est l'artéfact le plus important de Scrum, c'est l'ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité.[1]

Le tableau 2.3 présente le backlog de notre système :

- ID qui représente l'identifiant de la use story.
- Thème pour mieux ordonner les user stories.
- Scenario ou User Story comporte la description des user story suivant le forme « En tant que . . . Je veux . . . ».
- La priorité de la user story selon la valeur métier et l'ordre de réalisation.

ID	Thème	User Story	Priorité
1	Inscription	En tant qu'utilisateur, je souhaite pouvoir saisir mes données, recevoir un message d'erreur clair en cas de saisie incorrecte des identifiants ou un champ vide, afin de créer un compte de manière sécurisée et ainsi accéder aux fonctionnalités de la plateforme	1
2	Authentification	En tant qu'utilisateur, je veux pouvoir saisir mes identifiants, recevoir un message d'erreur clair en cas de saisie incorrecte afin de pouvoir corriger mes erreurs et accéder à mon compte de manière sécurisée.	1
3	Gestion des utilisateurs	En tant qu'administrateur, je veux consulter tous les utilisateurs et je veux bloquer n'importe quel compte étudiant en cas d'un comportement inapproprié.	1
4	Gestion de compte	En tant qu'utilisateur, je souhaite pouvoir consulter mon profil, modifier les informations personnelles et supprimer mon compte.	1
5	Overview	En tant qu'entreprise, je souhaite accéder à ma page overview pour voir les offres les plus visitées, la similitude entre les offres publiées et les candidatures reçues.	3
6	Overview	En tant qu'étudiant, je veux pouvoir accéder à ma page overview pour consulter les meilleures offres d'emploi publiées, ainsi que d'autres informations utiles.	3
7	Overview	En tant qu'administrateur, je souhaite accéder à ma page overview pour suivre la performance du site, l'activité des utilisateurs, les indicateurs de performance clés, ainsi que d'autres statistiques importantes.	3
8	Drive	En tant qu'étudiant, je veux pouvoir accéder à mon interface drive pour télécharger mes documents académiques tels que mes rapports, mes CV, mes attestations afin de les utiliser dans mes candidatures à des stages.	4

9	Drive	En tant qu'entreprise, je veux pouvoir accéder à mon interface drive pour télécharger mes documents professionnels tels que les PFE Books afin de les partager avec les étudiants.	4
10	Génération de portfolio	En tant qu'étudiant, je veux pouvoir téléverser mon CV dans une interface de génération de portfolio pour obtenir un portfolio personnalisé mettant en valeur mes compétences, réalisations et expériences afin de le partager avec les entreprises.	2
11	Offre de stage	En tant qu'entreprise, je veux pouvoir publier des offres à partir d'un formulaire détaillé et facile à interpréter, le modifier et le supprimer.	1
12	Offre de stage	En tant qu'étudiant, je souhaite pouvoir consulter les offre de stage afin d'être informé sur leurs détails.	1
13	Pfe book	En tant qu'entreprise, je veux pouvoir publier des pfe books, les modifier et les supprimer.	2
14	Pfe book	En tant qu'étudiant, je veux pouvoir consulter des pfe books publiés par les entreprises ou générés par le système.	2
15	Recherche et filtrage	En tant qu'étudiant, je souhaite pouvoir parcourir une liste organisée et filtrable des offres de stage, afin de trouver rapidement celles qui correspondent à mes intérêts et compétences.	2
16	Recherche et filtrage	En tant qu'entreprise, je souhaite pouvoir parcourir une liste organisée et filtrable des profils des étudiants, afin de trouver rapidement ceux qui correspondent à mes intérêts et besoins.	2
17	Demande de stage	En tant qu'étudiant, je souhaite pouvoir postuler à les offres de stage disponibles qui m'intéressent, afin de trouver des opportunités de stage correspondantes à mes intérêts et compétences.	2

18	Demande de stage	En tant qu'étudiant, je veux pouvoir accéder à la page de demandes de stage pour supprimer les demandes que j'ai postulées.	2
19	Demande de stage	En tant qu'entreprise, je veux pouvoir consulter les demandes de stage reçues pour accepter ou refuser les candidatures.	2
20	Génération de pfe book	En tant qu'entreprise, je souhaite accéder à la page de génération de PFE Book, afin de choisir un des modèles de PFE Book créé à partir des offres publiées sélectionnées.	2
21	Historique	En tant qu'utilisateur, je souhaite accéder à ma page d'historique afin de consulter les activités passées effectuées sur la plateforme.	4
22	Communication	En tant qu'étudiant, je veux pouvoir accéder à une boîte de messagerie pour contacter les entreprises et y déposer mon CV ou mon portfolio.	3
23	Communication	En tant qu'entreprise, je souhaite communiquer avec les étudiants pour discuter de plus de détails concernant sa candidature ou autres informations sur les offres disponibles.	3
24	Notification	En tant qu'étudiant, je veux pouvoir recevoir des notifications instantanées lorsqu'une entreprise accepte ou refuse ma candidature.	3
25	Notification	En tant qu'entreprise, je veux recevoir des notifications instantanées lorsqu'il y a une nouvelle candidature sur une offre de stage, afin de rester informé en temps réel des interactions des candidats avec les offres plus tard.	3

30	Favoris	En tant qu'étudiant, je souhaite avoir la possibilité d'ajouter des offres de stage à une liste de favoris, la modifier et la visualiser, afin d'accéder facilement les offres de la liste.	4
31	Favoris	En tant qu'entreprise, je souhaite avoir la possibilité d'ajouter des profils des étudiants à mes favoris, la modifier et la visualiser, afin d'accéder facilement les profils plus tard.	4

Tab. 1.3 : Les fonctionnalités du backlog

1.3.2 Diagramme des cas d'utilisation global

Dans la figure 2.1, nous présentons une vue globale concernant le comportement fonctionnel du système qui représente les interactions entre les acteurs et les cas d'utilisation .

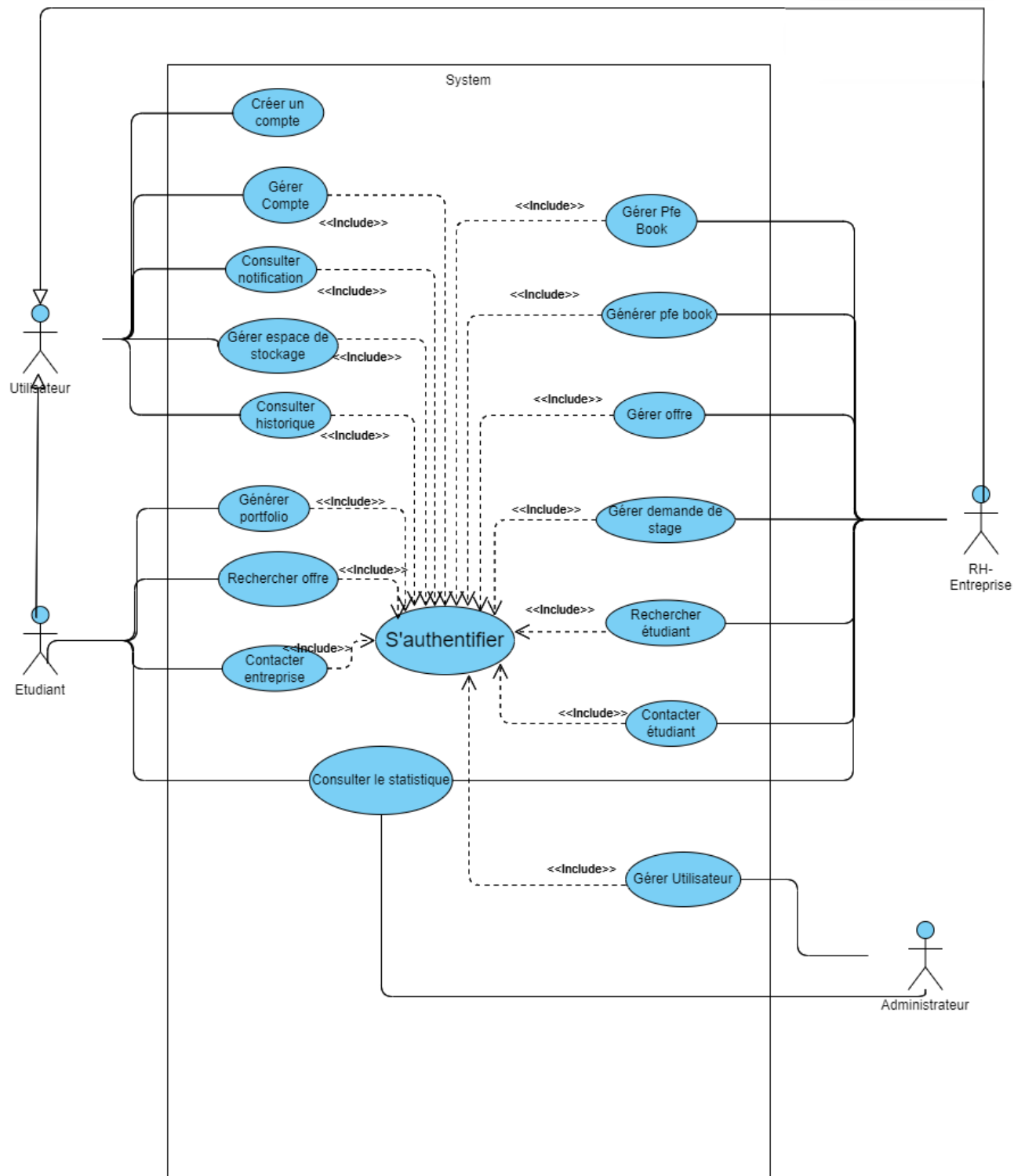


Fig. 1.1 : Diagramme des cas d'utilisation global

1.3.3 Diagramme de classes global

Dans la figure 2.2, nous présentons les classes intervenantes dans le système et les relations entre eux.

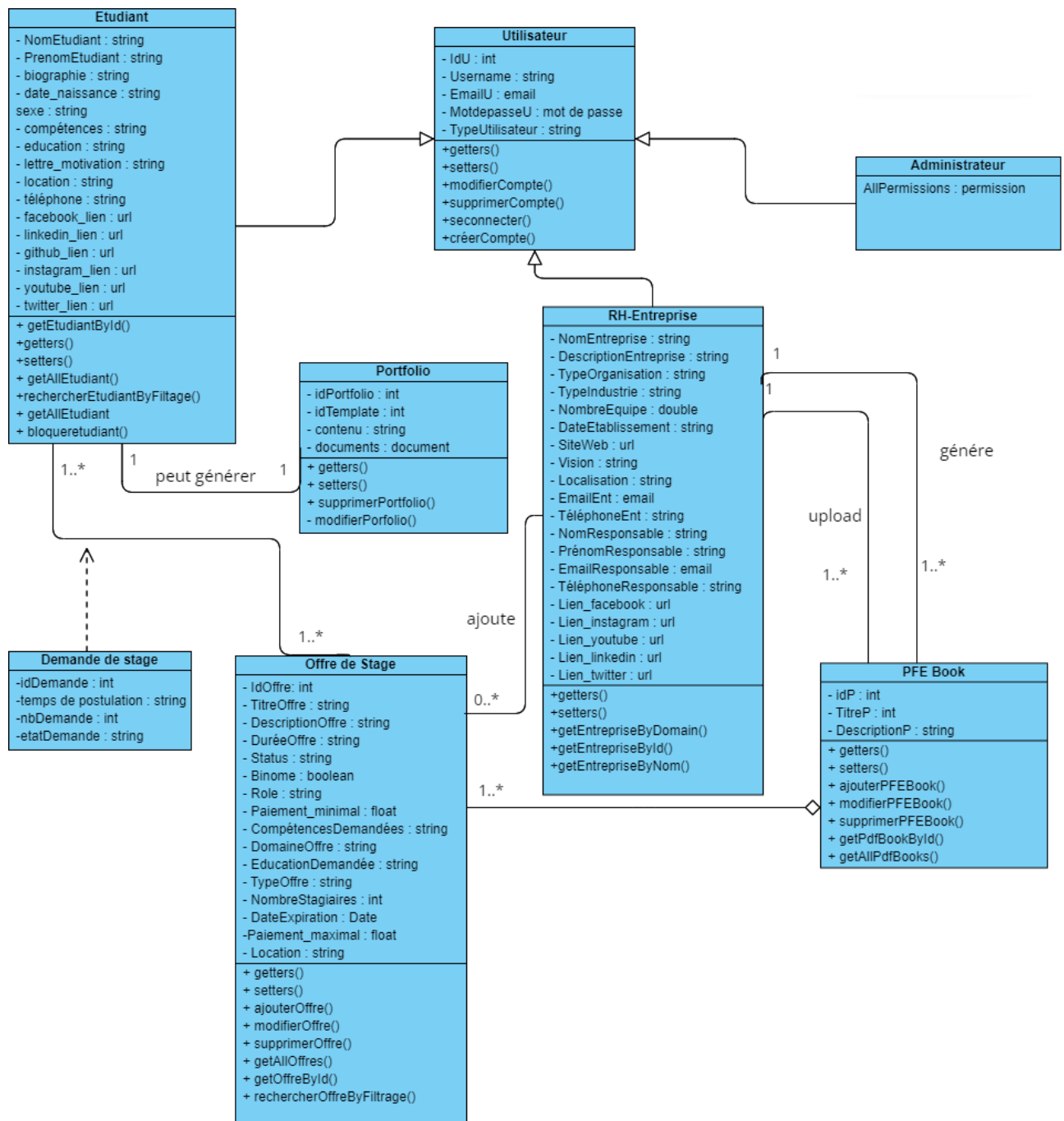


Fig. 1.2 : Diagramme de classes

1.3.4 Planification des sprints

Une fois nous avons terminé le backlog du produit, nous avons passé à la phase de la planification. Le but de cette phase est de construire le backlog de sprint en se basant sur le backlog de produit.

Après avoir faire une réunion avec le product owner, nous avons identifié les durées prévisionnelles du travail à effectuer durant chaque sprint.

La figure 2.3 montre la répartition des tâches de sprints relative à notre système dans un diagramme de gantt.

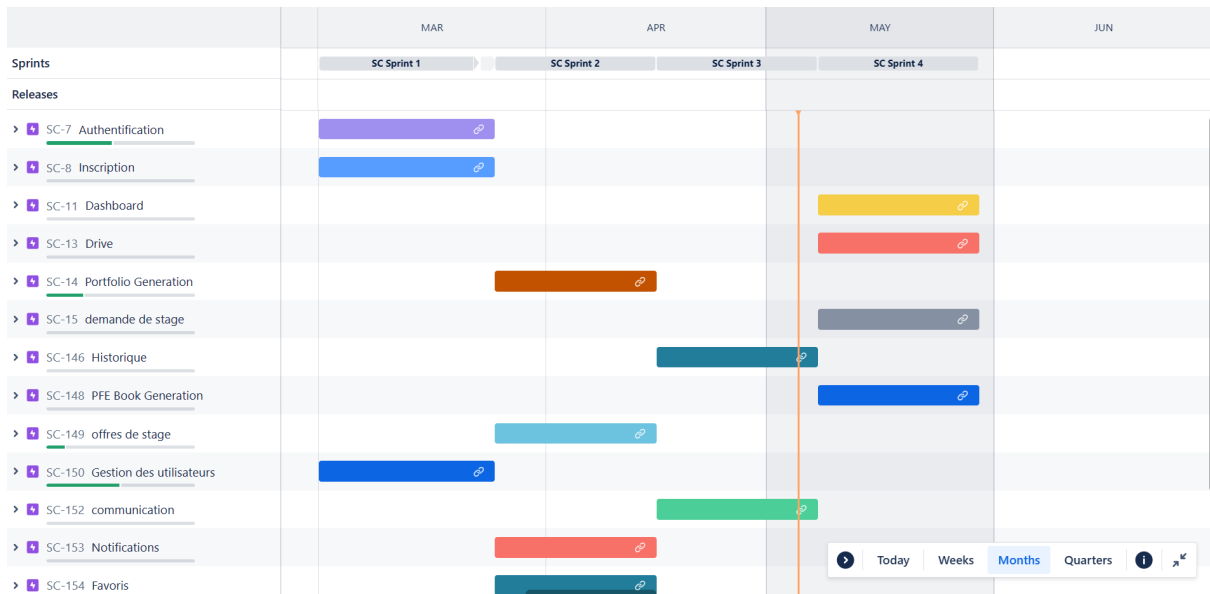


Fig. 1.3 : Diagramme de gantt

Pour notre projet nous avons divisé le travail sur quatre sprints de 3 semaines.

Nous réliserons dans le sprint 1 comme indiqué dans la figure 2.4 les tâches de l'inscription, l'authentification, la gestion de comptes et la gestion des utilisateurs.



Fig. 1.4 : Sprint 1

Après terminer la réalisation du sprint 1, nous avons passé au sprint 2 comme présenté dans la figure 2.5 durant lequel nous avons implémenté les fonctionnalités des offres de stage, des notifications, la liste des offres de stage favoris, la génération de portfolio et la gestion de pfe book.

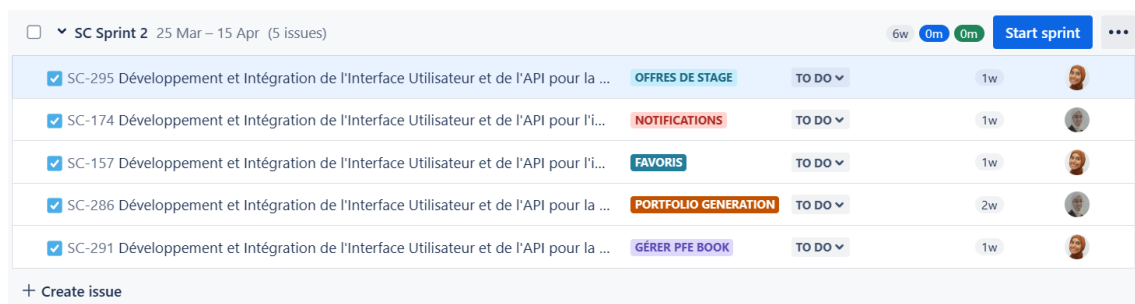


Fig. 1.5 : Sprint 2

La figure 2.6 indique la réalisation des tâches de sprint 3 qui sont la recherche filtrée des offres de stage et profils des étudiants, la communication et l'historique.



Fig. 1.6 : Sprint 3

Dans le dernier sprint, nous avons terminé notre projet par l'implémentation des fonctionnalités de l'overview du dashboard qui représente des statistiques personnalisées selon chaque utilisateur, le drive où l'entreprise et l'étudiant gèrent leurs espaces de stockage, la génération de pfe book et les demandes de stage.

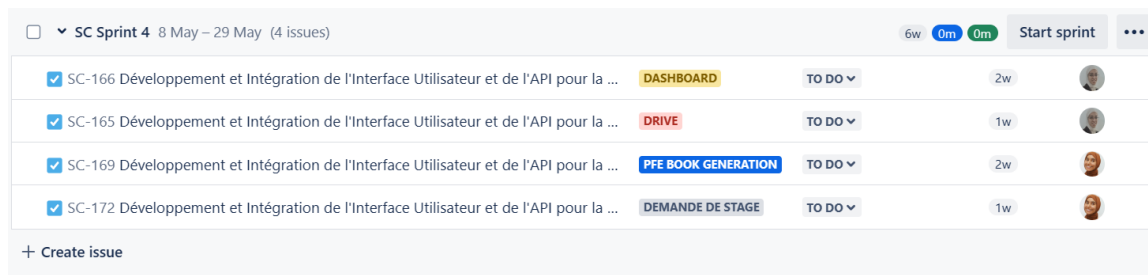


Fig. 1.7 : Sprint 4

1.4 Spécification architecturale

1.4.1 Architecture logique : Architecture MVC

Afin de bien organiser notre code et de réussir notre projet, nous avons adopté le modèle MVC (Modèle-Vue-Contrôleur) qui est un modèle de conception logicielle. Il permet de structurer le système en trois composants logiques principaux comme présenté dans la figure 2.8:

- Un modèle : contenant les données traitées par le programme.
- Une vue : contenant la présentation de l'interface graphique de l'utilisateur.
- Un contrôleur : contenant la logique concernant les actions effectuées par l'utilisateur.

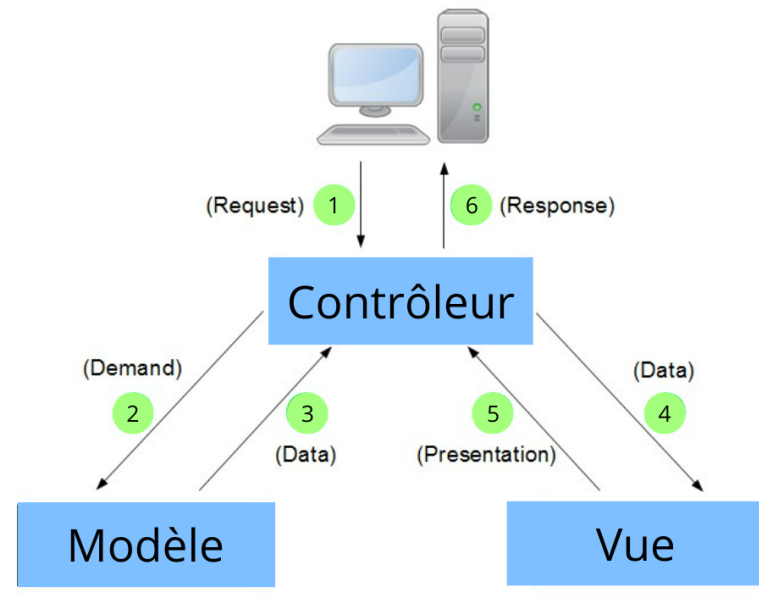


Fig. 1.8 : Architecture logique de notre système

1.4.2 Architecture physique

L'ensemble des composants matériels représentant l'architecture physique est indispensable pour faire fonctionner notre application.

Dans notre cas, l'architecture physique choisie, comme présentée dans la figure 2.9 :

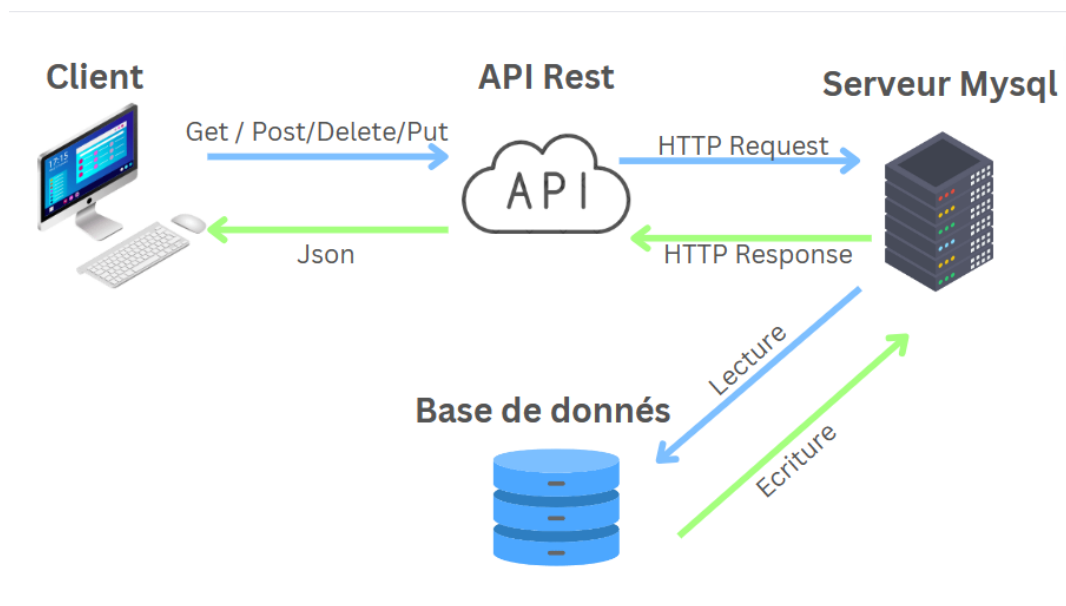


Fig. 1.9 : Architecture physique de notre système

1.5 Environnement de travail

1.5.1 Environnement matériel

Durant la réalisation de notre projet, nous avons utilisé deux ordinateurs portables qui disposent de les configuration suivantes comme indique le tableau 2.4 :

	PC 1	PC 2
Processeur	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz	Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz 2.49 GHz
Système d'exploitation	Windows 11 Professionnel	Windows 10 Famille
Mémoire RAM	8,00 Go	8.00 Go

Tab. 1.4 : Les configurations de PCs du travail

1.5.2 Environnement de développement



Visual Studio Code : un éditeur de code source développé par Microsoft facilitant le développement d'applications pour n'importe quelle plateforme et dans n'importe quel domaine.

Fig. 1.10 : Logo du Visual studio code



Fig. 1.11 : Logo du My SQL Server

MySQL Server : est un système de gestion de base de données fournissant l'interface serveur qui permet aux clients (applications, sites web, etc.) d'accéder et de manipuler les données stockées dans une base de données MySQL.



Fig. 1.12 : Logo du MySQL

MySQL Workbench : est un outil client graphique utilisé pour administrer et travailler avec des bases de données MySQL. Il permet aux utilisateurs de travailler plus facilement avec MySQL Server en fournissant des fonctionnalités visuelles et intuitives.

1.5.3 Environnement logiciel



Fig. 1.13 : Logo du VueJs



Fig. 1.14 : Logo du Axios



Fig. 1.15 : Logo du Django-Python



Fig. 1.16 : Logo du Django Rest Framework



Fig. 1.17 : Logo du Figma



Fig. 1.18 : Logo du Jira

Vue.js : est un framework JavaScript open-source progressif utilisé pour construire des interfaces utilisateur interactives et dynamiques côté client et des applications web monopages. Ce framework s'appuie sur des technologies complémentaires telles que HTML5, CSS et JS.

Axios : une bibliothèque JavaScript populaire, offre un ensemble complet de fonctionnalités pour effectuer des requêtes HTTP. L'une de ses fonctionnalités est celle des intercepteurs, qui permettent d'intercepter et de transformer les requêtes et réponses. Ils permettent donc de centraliser la logique de gestion des appels HTTP, rendant le code plus clair et plus facile à maintenir.[2]

Django : Le framework Web Django est un framework gratuit et open source qui peut accélérer le développement d'une application Web créée dans le langage de programmation Python.[3]

Django Rest Framework : Django Rest Framework est une librairie permettant de mettre en place une API sur un projet Django.[4]

Figma : Figma est un logiciel de conception UI qui permet aux UX/UI designers de créer facilement des interfaces graphiques pour les applications mobiles et les sites web.[5]

Jira : Jira est un système de gestion de projets collaboratif qui permet de planifier et suivre le déroulement du travail afin de répondre aux exigences des fonctionnalités et atteindre un objectif.



Fig. 1.19 : Logo du Github

Github : une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels.[6]

1.5.4 Choix de technologies

Vue Js :

Vue.js est facile à intégrer des templates HTML et simple à utiliser et à apprendre. Son point fort est son écosystème composé de nombreuses bibliothèques et les outils nécessaires pour développer des applications web complexes tels que Vue Router pour la navigation et Vuex pour la gestion de l'état.

Django :

On a choisi Django comme un framework de développement backend pour son approche de développement rapide en fournissant un ensemble complet de fonctionnalités prêtes à l'emploi, son architecture MVC qui rend le code plus facile à maintenir, à tester et à évoluer au fil du temps, son architecture modulaire et extensible permet d'ajouter de nouvelles fonctionnalités ou en améliorer les performances au besoin. Il permet aussi de créer des API REST simples et efficaces qui peuvent être utilisées pour interagir avec des modèles d'IA depuis d'autres applications.

Api gemini vs API Chatgpt vs API Copilot

Lors de la réalisation de la tâche de la génération de portfolio à partir d'un cv numérique de format pdf, nous avons eu besoins d'extraire les données existantes dans le cv saisi sous des données JSON à récupérer et afficher dans les modèles du portfolio généré.

Pour atteindre notre objectif, nous avons utilisé la bibliothèque PyPDF2 du python qui nous a permis d'extraire les données sous format texte mais le problème reste toujours à convertir ces données en format JSON.

Nous avons établi une recherche en comparant le nombre de requêtes envoyées par minute pour la version gratuite des systèmes d'intelligence artificielle.

API	Requêtes par minute
Api gemini	15
API Copilot	5
API Chatgpt	3

Tab. 1.5 : Nombre de requêtes par minute pour des systèmes d'intelligence artificielle

Depuis le tableau 2.5, nous résolvons que le système gemini permet le plus grand nombre de requêtes gratuites par minute ce qui nous a fait choisir de consommer son api pour analyser et transformer le texte extrait vers des données sous format JSON pour les intégrer dans le portfolio généré.

1.6 Conclusion

Au cours de ce chapitre, on a analysé les besoins tout en précisant le backlog et le comportement fonctionnel de notre système en s'appuyant sur une architecture spécifique pour réaliser notre projet dans un environnement de travail pertinent. Dans le chapitre suivant, nous commençons la réalisation du sprint 1.

Webographie

[1] NAKKACH Houda,HACHANI Nadia/ University Lib, Penser Agile, SCRUM, pour la Gestion de Projet : «Améliorer les méthodes d'enseignement»/ 7 Mars 2020 (en ligne) / <https://www.universitylib.com/penser-agile-scrum-pour-la-gestion-de-projet-ameliorer-les-methodes-denseignement/> consulté le 08 Avril 2024.

[2] sfeir.dev / Les intercepteurs Axios / 28 juin 2023 (en ligne) / <https://www.sfeir.dev/back/les-intercepteurs-axios/> :.text=Axios / consulté le 15 Avril 2024.

[3] IBM / Qu'est-ce que Django ? / <https://www.ibm.com/fr-fr/topics/django> / consulté le 15 Avril 2024.

[4] Open Classrooms / Mettez en place une API avec Django REST Framework / 28 Novembre 2023 (en ligne) / <https://openclassrooms.com/fr/courses/7192416-mettez-en-place-une-api-avec-django-rest-framework/7424482-decouvrez-django-rest-framework> / consulté le 15 Avril 2024.

[5] La grande ourse / Les possibilités de Figma / 25 Novembre 2020 (en ligne) / <https://la-grandeourse.design/blog/outils/figma/> consulté le 15 Avril 2024.

[6] Lemagit / GitHub / août 2016 (en ligne) / <https://www.lemagit.fr/definition/GitHub> / consulté le 8 Mai 2024.