



Année Universitaire 2021-2022

RAPPORT TECHNIQUE

IMPLÉMENTATION D'UN INTRANET ET DE FONCTIONNALITÉS DE GESTION POUR LE
SITE INTERNET D'HORUS, JUNIOR ENTREPRISE RATTACHÉE À ISIS



Présenté par
Hamza MAKRI (chef de projet)
Ikhlass HAJJAM
Anaïs TAYLOR

Tuteur école
Nicolas SINGER

En collaboration avec :
Tristan FRÉDÉRICK (Président Horus)
Killian MARY (Ancien Responsable SI Horus)
Hamza MAKRI (Responsable SI Horus) Ikhlass HAJJAM (Chargé SI Horus)

Remerciement:

Nous tenons à remercier l'ensemble de l'équipe enseignante pour leurs formations et conseils utiles et avisés, notamment M.Bastide, M. et Mme. Pecatte pour leurs enseignements au cours de l'année dans le cadre du module de Technologies Web, ainsi que M. Pingaud et M. Eshane pour leurs enseignements des fondamentaux nécessaires à la gestion de projet.

SOMMAIRE

I- Cadrage:	6
1.1 Définition du contexte	6
1.2 Expression des besoins	6
1.3 Analyse de l'existant	7
1.3.1 Existant interne	7
1.3.2 Existant externe	7
II-Analyse et faisabilité	8
1.2 Définition des objectifs	8
2.2 Contraintes et risques	8
2.2.1 Contraintes	8
2.2.2 Risques	9
2.2.3 Tableau de criticité des risques et mitigation	9
2.2.4 Matrice de criticité des risques	9
III-Conception	10
3.1 Diagrammes de classe et d'instance	10
3.2 Organigramme des tâches	10
3.3 Diagramme de Gantt	11
3.4 La matrice RACI	11
3.5 Maquettage	11
IV-Réalisation	12
4.1 Choix des moyens techniques - Cohérence et pertinence	12
4.2 Présentation des solutions techniques	13
4.2.1 Architecture logiciel	13
4.2.2 Système d'authentification et sécurité	14
4.3 Difficultés rencontrées	15
4.3.1 Retard sur le planning initial	15
4.3.2. Difficultés techniques	16
4.3.3. Solutions apportés	16
V. Etat d'avancement du projet	17
5.1 Objectifs atteints	17
5.2 Améliorations	17
5.3 Suite du projet	18
VI. Conclusion	18
VII- Bibliographie	19
VIII - Annexes	20
Annexe 1 : Note de cadrage	20
Annexe 2 : RACI prévisionnel	21
Annexe 3 : Organigrammes des tâches (WBS)	24
Annexe 4 : Diagramme de Gantt prévisionnel	25

Annexe 5 : Diagramme de Gantt réel:	26
Annexe 6 : Diagramme de classe et diagramme d'objet	27
Annexe 7 : Tableau de criticité	29
Annexe 8 : Matrice de criticité	30
Annexe 9 : Maquette - Page d'Accueil	30
Annexe 10 : Maquette - Page de connexion	31
Annexe 11 : Maquette - "Qui sommes-nous?"	32
Annexe 12 : Matrice RACI réelle :	33
Annexe 13 : Site Web de la junior entreprise de L'ESCadrille :	35
Annexe 14 : Site Web de la junior entreprise de HEC :	36
Annexe 15 : Système de connexion de la Junior Entreprise de l'Insa Toulouse:	36
Annexe 16 : Schéma des requêtes échangées entre Client et Serveur pour le système d'authentification (avant personnalisation) :	37
Annexe 17 : Schéma de l'architecture logiciel du système d'authentification :	37

Résumé

Dans le cadre de notre première année du cycle d'ingénieurs à ISIS, nous avons réalisé un projet de fin d'année d'une période de deux mois, débutant le 21 février 2022 et aboutissant le 15 avril 2022. L'objectif du projet était de mettre en application les compétences acquises au cours de l'année à travers différents modules. Une série de sujets ont été proposés par M.Bastide traitant de divers aspects de la gestion administrative et pédagogique d'ISIS. Néanmoins, nous avons pris l'initiative de contracter un partenariat avec l'association **Horus HealthCare Systems**, la Junior Entreprise de l'école, afin de proposer notre propre sujet. Celui-ci consistait en la réalisation d'une **application web plus dynamique et fonctionnelle** dans le but de proposer à la Junior Entreprise des outils de gestion de l'association (**ERP**) par l'utilisation des frameworks **Spring Boot** et **VueJS**. Afin de mener à bien ce projet, nous avons élu Hamza Makri en tant que chef de projet.

Notre groupe se constitue de 3 membres issus de formations très variées pour que chacune d'entre elles puisse apporter son savoir-faire dans le domaine qui lui avait été confié:

- Hamza Makri : Chef de projet et responsable développement web Backend et Frontend.
- Ikhlass Hajjam: Chargée de la gestion de projet et développeur Frontend.
- Anaïs Taylor : Chargée de la gestion de projet.

Afin de fixer nos objectifs, il a été important de contextualiser dans un premier temps notre projet et d'y poser un cadre. Nous avons par la suite établi la conduite de gestion de projet en réalisant une **étude des besoins** de l'association, puis grâce à une analyse de l'existant, nous avons pu nous rendre compte des ressources que possède l'association et celles des autres Juniors Entreprises. Ainsi, une solution adaptée a pu être imaginée et nos objectifs fixés. Elle consiste à mettre en place un système d'authentification permettant aux membres d'accéder à diverses fonctionnalités de gestion en fonction des pôles les concernant et d'améliorer l'aspect esthétique du site internet.

Par la suite, nous avons étudié l'**analyse et la faisabilité** du projet avant d'entamer sa **réalisation** afin d'évaluer les conditions nécessaires à la réussite du projet (planification, matériel, compétence, ...). Nous avons tout d'abord identifié les **contraintes** s'imposant à notre projet, puis les **risques** pouvant mettre en péril sa bonne réalisation. Ceux-ci ont été quantifiés en fonction de leur probabilité et de leur impact, pour être ensuite listés dans un tableau de criticité afin d'établir des stratégies de **mitigation** de ces risques. Par exemple, une des principales contraintes étant le temps imparti à la réalisation du projet, le projet risquait donc de ne pas voir le jour. En conséquence, nous avons établi une hiérarchisation des objectifs par ordre d'importance. Nous avons également eu recours à différents outils de **planification** comme un diagramme de Gantt.

Ainsi, une **conception** plus approfondie nous a permis d'apporter une vision claire du travail à réaliser de manière temporelle et conceptuelle. Ainsi ordonnées, les tâches ont pu être réparties en fonction des compétences de chacun. La responsabilité des éléments de gestion de projet incombent à Ikhlass Hajjam et Anaïs Taylor et celle du développement à Hamza Makri.

En fin de projet, des points sur les différentes **difficultés rencontrées** lors de la réalisation de ce projet et sur l'état d'avancement des objectifs ont été réalisés afin de pouvoir donner suite au projet dans les années à venir.

Mots clés: Horus, application web, Spring Boot, VueJS, ERP

I- Cadrage:

1.1 Définition du contexte

Nous réalisons notre projet dans le cadre de la junior entreprise d'ISIS, Horus, créée en 2011. Celle-ci offre ses services dans le domaine de la e-santé et de l'informatique. Elle est entièrement gérée par des étudiants ingénieurs de l'école souhaitant monter en compétence et acquérir de l'expérience dans le monde professionnel et entrepreneurial, en plus des enseignements que propose l'établissement. Les membres de l'association travaillent dans un but pédagogique et non lucratif. Il existe plusieurs pôles, chacun responsable d'un aspect de la junior entreprise (finance, Ressources-humaines, Systèmes d'information ...). Lorsque Horus réalise un projet pour une entreprise tierce, elle recherche ses intervenants parmi les étudiants d'ISIS afin qu'ils réalisent les différentes missions du projet. Son siège social se situe à Castres dans le Tarn (81) et ses locaux se trouvent au sein de l'école d'ingénieurs ISIS. Ses domaines d'activités sont les suivants:

- Technologie Web
- E-santé
- Big Data
- Applications mobiles
- Assistance à la maîtrise d'ouvrage et rédaction
- Développement logiciel

On y compte actuellement environ 25 adhérents et, en quête d'évolution, celle-ci souhaite la facilité et s'y préparer en réorganisant son SI.

1.2 Expression des besoins

Horus HealthCare Systems possède déjà un site web, mais celui-ci n'a pas été mis à jour depuis quelque temps (<https://www.horus-hcs.fr/index.html>). De plus, celui-ci est purement statique et ne possède aucun aspect dynamique permettant de pousser son utilité au-delà d'une simple vitrine. Dans l'optique de faciliter et mieux cadrer le travail de ses administrateurs, Horus a donc lancé un projet de refonte de son site web, afin de le transformer en ERP, permettant ainsi de mettre à disposition des membres du conseil d'administration de l'association, à travers les mandats, des outils de gestion spécifiques aux différents pôles. Afin de débiter rapidement ce projet et de permettre aux étudiants de monter en compétence, les membres du pôle SI d'Horus ont souhaité le léguer aux étudiants de 3ème année dans le cadre du projet tutoré. Un cahier des charges déjà établi au préalable par le précédent mandat.

Les besoins sont donc les suivants :

- Avoir des outils de développement
- Dynamiser le site web Site web plus attractif et professionnel
- Centraliser les informations au sein d'Horus HCS selon le profil de l'étudiant
- Mise en place d'un ERP, suivi des adhésions, suivi d'études
- Les administrateurs d'Horus se retrouvent constamment en déplacement au cours de leur mandat en raison des différentes mobilités et stages, une solution portable est donc nécessaire.

1.3 Analyse de l'existant

1.3.1 Existant interne

Comme expliqué dans les parties précédentes, Horus Healthcare Systems est la Junior Initiative de l'école ISIS. La cible recherchée par le site internet d'Horus se compose d'une part des étudiants d'ISIS, afin de donner envie à ces derniers de rejoindre l'association. D'autre part, le site internet vise à donner de la visibilité à l'association auprès des entreprises et potentiels clients.

Le site internet actuel d'Horus décrit la junior entreprise, présente les actualités avec des interviews, des articles techniques, des annonces... ainsi un espace dédié pour découvrir comment se déroule un projet chez Horus HealthCare Systems et effectuer par la suite une demande d'étude. Au niveau de la communication entre les adhérents et le pôle administratif, les administrateurs communiquent avec les adhérents sur discord ou messenger (par exemple, pour fixer des rendez-vous, des conseils d'administration, les assemblées générales ordinaires et extraordinaires) et les membres du conseil d'administration communiquent par slack.

Supports :

- Un site statique (HTML/CSS/JS)
- Un contenu actif sur les réseaux sociaux (Facebook, LinkedIn, Instagram) pour partager les avancées des projets, les nouveautés de l'entreprise.

Le site web d'Horus a une fonction informative, il n'a ni base de données, ni fonctionnalités pour gérer les différents aspects de l'association.

Nous avons donc établi une note de cadrage ([Annexes 1](#)) afin d'établir les objectifs de notre partenariat.

1.3.2 Existant externe

Dans le but d'améliorer le site internet d'Horus, nous avons analysé des sites internet d'autres junior entreprise offrant les mêmes services que Horus HealthCare Systems. Vous retrouverez en bibliographie les différents sites que nous avons pu analyser. On remarque qu'ils ont tous une interface dynamique et facile d'utilisation pour le lecteur. Avec des informations claires, et les sites ne sont pas surchargés d'informations.

Vous pouvez observer le site Web de la junior entreprise ESCadrille ([Annexe 13](#)) de Toulouse, qui propose une interface dynamique et attractive qui est agréable pour le visiteur.

Le site Web de la Junior Entreprise d'HEC ([Annexe 14](#)) est lui aussi très structuré et qui met bien en valeur les choses les plus importantes.

Le site Web d'Horus ne possède pas d'intranet, on peut voir un exemple de système de connexion de la Junior Entreprise INSA ([Annexe 15](#)).

En parcourant, plusieurs sites Web de junior entreprise nous pouvons voir qu'elles ne sont pas nombreuses à posséder un intranet. On peut donc penser que Horus HealthCare pourra se démarquer des autres, grâce à notre intranet qui pourra faciliter la transmission d'informations.

II-Analyse et faisabilité

1.2 Définition des objectifs

Voici donc les objectifs que nous avons pu nous fixer pour ce projet après étude des besoins et des existants

Le site web se composerait de deux parties. Une partie à but informatif accessible à tous les visiteurs du site et une seconde partie accessible uniquement aux adhérents qui contiendrait des fonctionnalités de gestion de la JE

Côté visiteur :

- Ajout d'un système d'authentification sécurisé pour les adhérents leur donnant accès à un tableau de bord personnalisé en fonction de leur rôle dans la JE.
- Page "Qui sommes-nous?" afin de présenter les membres du conseil d'administration d'Horus aux clients.
- Page "Articles" qui affichera une liste d'articles disponibles à la lecture.
- Page Postes permettant de présenter les différents pôles au sein de la Junior et de lister et expliquer les postes qu'ils contiennent, mais également ceux disponibles.

Côté tableau de bord :

- Onglet CRUD des utilisateurs, des rôles, disponible uniquement pour le président, permettant de créer, supprimer et modifier
- Ajouter de la rédaction d'articles
- Ajouter un onglet avec la liste des recherches de postes et le descriptif des postes.
- Avoir une page avec les retours des clients sur notre gestion, ressenti sur notre capacité à gérer leur projet.

Nous avons plusieurs objectifs et peu de temps pour les réaliser, nous allons donc dans un premier temps nous concentrer sur le système de connexion, les fonctionnalités "Article" et "Qui Sommes Nous ?" en accès réservé aux personnes autorisées et nous verrons par la suite si nous pouvons en ajouter d'autres.

2.2 Contraintes et risques

2.2.1 Contraintes

Des contraintes s'imposant dans le cadre du projet limitent la liberté de la maîtrise d'œuvre.

- La contrainte la plus forte est la contrainte temporelle. En effet, le rapport est à rendre pour le 15 Avril 2022, ce qui nous donne très peu de temps pour implémenter nos nouvelles fonctionnalités sur le site web et constituer notre rapport technique avec la partie de gestion de projet.
- Une contrainte de moyen technique s'impose également. En effet, l'objectif de ce projet étant de mettre en application nos compétences acquises au cours de l'année, utiliserons les framework Spring Boot et VueJS

- Nous avons une contrainte de confidentialité due à la collecte de données personnelles (mots de passe, nom utilisateur...), il faut que nos données soient protégées et à accès restreint.

2.2.2 Risques

Les risques qui découlent ou non des contraintes peuvent impacter l'atteinte des objectifs du projet (qualité/ ressources/ délai). Ils doivent être gérés par une identification de leur criticité et la mise en place d'indicateurs et de plans de mitigation.

Nous avons des risques découlant:

- Des contraintes temporelles; nous avons une grande quantité de travail et peu de temps. Avec un projet aussi grand, le risque est d'accumuler du retard et de ne pas pouvoir finir le projet.
- Du facteur humain : l'équipe peut impacter le projet par l'absence d'un ou plusieurs membres.
- Des contraintes techniques : L'accessibilité demeurant un point important (faut préciser ce que vous entendez par "accessibilité" pasque vous en avez pas parlé dans vos objectifs), il est nécessaire d'apprendre à maîtriser ce point essentiel à notre projet, mais notre manque d'expérience dans ce domaine peut impacter le projet. On nous demande par ailleurs de mobiliser des connaissances techniques que nous venons juste de voir, ce qui peut avoir des conséquences sur notre efficacité due à notre manque d'expérience.

2.2.3 Tableau de criticité des risques et mitigation

Après avoir défini tous les risques pouvant survenir au cours du projet, nous avons construit un tableau de criticité ([Annexe 7](#)) contenant tous les risques possibles, leur niveau de criticité ainsi que les solutions possibles pour éviter ces risques ou y remédier s'ils surviennent.

2.2.4 Matrice de criticité des risques

Nous avons par la suite fait une matrice de criticité ([Annexe 8](#)) pour donner une vision graphique de la situation des risques dans le projet.

Le calcul de la criticité des risques consiste à identifier les risques et dangers pouvant survenir lors de la réalisation du projet (et même après), et de les classer, de les quantifier, afin de savoir lesquels nécessitent le plus d'attention et établir un ordre de priorité en ce qui concerne la gestion de ces risques et contraintes.

Criticité = (probabilité d'apparition) x (gravité).

Les autres paramètres à prendre en compte lors de ce calcul sont :

- L'effectif exposé.
- Le niveau de maîtrise.

Afin de réaliser cette évaluation nous nous sommes entre autres aidé de la matrice de criticité suivante :

		Gravité			
		1 Faible	2 Moyen	3 Grave	4 Très grave
Probabilité	4 Très probable				
	3 Probable				
	2 Improbable				
	1 Très improbable				

III-Conception

3.1 Diagrammes de classe et d'instance

Il est essentiel avant de commencer la phase de réalisation de concevoir notre modèle relationnel de données. Ainsi, nous avons modélisé notre base de données grâce à un diagramme de classe ([Annexe 6](#)).

Sur ce diagramme, nous constatons qu'il existe des relations "many to many" entre les classes User & Role, Role & Permission et Role & Pole :

User & Role : Ici la classe Role représente un poste au sein d'Horus (Chargé de communication, président, secrétaire générale, etc). En effet, au sein d'Horus, en raison du nombre limité de candidature à chaque mandat, il est commun qu'une personne cumule plusieurs postes à la fois, c'est pour cela qu'un utilisateur peut posséder plusieurs rôles/postes. Inversement, un rôle possède plusieurs utilisateurs car il est possible d'avoir plusieurs personnes occupant le même poste (en général, c'est le cas pour les postes de chargés).

Role & Permission : Il est possible, à travers les différents mandats, que de nouveaux postes soient créés. Les postes étant représentés par la classe "Rôle" dans notre base de donnée, il sera alors facilement possible de créer un nouveau poste/rôle, personnalisé, en choisissant les permissions qui lui seront accordées. En accord avec notre modèle de donnée, un rôle possède donc plusieurs permissions et une permission sera accordée à plusieurs rôles.

Role & Pole : La classe Pole permet de catégoriser plusieurs rôles. En effet un pôle regroupe plusieurs rôles/postes. Par exemple, les postes de trésorière et de chargé de comptabilité font partie du pôle "Trésorerie". De l'autre sens, il peut paraître étrange qu'un rôle ai plusieurs pôles. Mais en effet, nous pouvons donner l'exemple du poste de secrétaire général qui fait à la fois partie du pôle "Bureau" mais également "Ressources Humaines". Cette relation many to many est donc justifiée

Enfin, une relation "one to many" est présente entre User et Article car un utilisateur peut écrire plusieurs articles mais un article est uniquement écrit par un utilisateur.

Afin de vous illustrer ces exemples, nous avons réalisé un diagramme d'instance/d'objet ([Annexe 6](#))

3.2 Organigramme des tâches

Afin de veiller à la bonne réalisation du projet, nous avons mis en place un organigramme de tâches (WBS). Ceci nous a permis d'élaborer un planning, nous donnant également une visibilité sur toutes les tâches que nous avons à réaliser et les personnes à qui revenait la responsabilité de chaque tâche.

Notre organigramme ([Annexe 3](#)) comporte quatre parties majeurs :
Une première partie dans laquelle nous avons décrit le contexte du projet et ciblé les besoins que nous voulions traiter. Dans un second temps, nous avons analysé la faisabilité du projet

en nous intéressant aux risques possibles ainsi qu'aux différentes manières de les mitiger. Nous avons ensuite réfléchi à la conception du projet, en réalisant différents diagrammes, dont nous parlerons plus en détails dans les parties suivantes. Enfin nous avons planifié la réalisation du projet en mettant en place une liste de tâches (développement, gestion de projet, livrables ...) qui nous permettront finalement de proposer une solution aux besoins évoqués dans la première partie.

3.3 Diagramme de Gantt

Etant donné la taille du projet importante et le peu temps dont nous disposons, nous avons réalisé un diagramme de Gantt ([Annexe 4](#)) à l'aide du logiciel "GanttProject" afin d'avoir une vue globale sur l'ensemble des tâches du projet ainsi que le niveau d'avancement de chaque tâche, et surtout indiquer les dépendances des tâches

Au fur et à mesure de l'avancée du projet nous avons dû ajouter ou supprimer certaines tâches et modifier les délais initiaux. Nous avons donc réalisé un nouveau diagramme de Gantt ([Annexe 5](#)).

3.4 La matrice RACI

Afin de partager les tâches que nous avons définies au préalable, nous avons choisi la méthode RACI ([Annexe 2](#)) afin de préciser le rôle de chacun et l'étendue de sa mission afin d'éviter les risques de chevauchement ou de conflit de ressources

La répartition des tâches a été faite en fonction des compétences de chacun en réalisant la matrice du RACI prévisionnel ([Annexe 2](#)).

Enfin, afin de suivre en temps réel la réalisation de chacune des tâches, nous les avons séparé en quatre catégories sur le logiciel "Trello" (disponible en lecture en suivant [ce lien : https://trello.com/b/oyg2rA9K/projetptut](https://trello.com/b/oyg2rA9K/projetptut)) "À faire", "En cours", "À valider" (par le chef de projet) et "Terminée" ainsi la matrice de RACI réelle ([Annexe 12](#)).

3.5 Maquettage

Les maquettes du site Web c'est l'étape qui précède la réalisation du projet, il est judicieux alors de réaliser des maquettes pour nous aider à construire l'arborescence de nos pages pour se rapprocher au plus de l'allure qu'aura le site à sa sortie, et finalement gagner du temps. La réalisation des maquettes a été faite à l'aide de l'éditeur graphique Figma. Vous pouvez retrouver l'intégralité de nos maquettes en annexes. Nous avons la maquette de la page d'accueil ([Annexe 9](#)), la page de connexion ([Annexe 10](#)) ainsi que la maquette qui représente l'onglet " Qui sommes nous?" ([Annexe 11](#)).

IV-Réalisation

4.1 Choix des moyens techniques - Cohérence et pertinence

Bien que le développement d'une application web reste une contrainte imposée par l'établissement, un site internet reste la solution la plus adaptée aux besoins de l'association. Ainsi, un site internet serait disponible de partout sans nécessité d'installation particulière du côté de l'utilisateur.

Afin de débiter la phase de réalisation des solutions, la question du choix des outils techniques doit se poser. L'objectif de ce projet étant l'application des compétences acquises en cours. Nous avons donc opté pour les langages et frameworks web étudiés en cours, c'est-à-dire Spring Boot pour la partie backend et Vue Js 3 pour la partie frontend.



Spring Boot est un framework open-source en Java. Dans le cadre de notre projet, Spring est utilisé afin de développer une API Web sécurisée grâce aux dépendances Spring Security et Spring Validation, sourçant ses données d'une base de données MySQL. La connexion et l'interaction avec la base est rendu possible grâce aux dépendances Spring Data JPA et MySQL Driver



VueJS, également open-source, est un framework permettant de développer des applications web monopages. Il comprend une multitude de dépendances diverses et variées, indépendantes et compatibles entre elles, permettant de simplifier l'implémentation de fonctionnalités standards, tout en accordant une bonne marge de manœuvre à la personnalisation de celles-ci. Le framework permet également un rendu dynamique, très pratique au développement.



Nous avons choisi une base de données MySQL. Ce choix se justifie tout simplement par la popularité de ce système de gestion de base de données performant et par l'abondance de ressources en ligne l'utilisant. MySQL comprend également un logiciel de gestion et d'administration de bases de données grâce à une interface graphique, MySQL Workbench.

Enfin, les IDE choisis sont :



IntelliJ pour la partie backend car étant spécifique au développement Java, il permet un développement plus efficace et une résolution des erreurs plus simple, parfois même automatique



Visual Studio Code pour la partie frontend. Cet IDE permet l'ajout d'extensions spécifiques au framework Vue JS et ses dépendances et permet donc un développement plus aisé et efficace.



Postman est un logiciel permettant de tester notre API Web sans avoir à développer notre frontend. Il s'agit d'un client HTTP qui teste des requêtes paramétrées manuellement à travers une interface utilisateur graphique et permettant de visualiser leurs réponses.

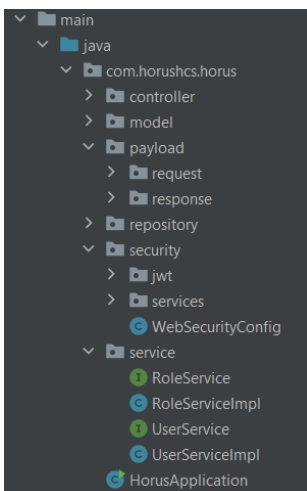
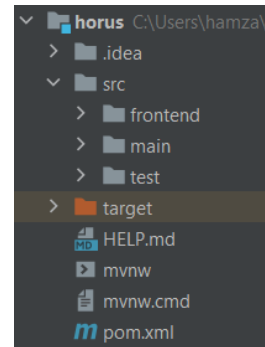
4.2 Présentation des solutions techniques

4.2.1 Architecture logiciel

Comme appris en cours, Spring Boot fonctionne selon le model MVC, celui-ci permet non seulement une plus grande lisibilité et une meilleure compréhension du code mais également une facilité de modification, car les vue et le code fonctionnel sont séparés, ainsi la modification ou l'ajout de l'un n'impacte pas l'autre.

La structure globale du projet est la suivante :

Le code est composé de deux parties. Une partie frontend et une backend (située dans le dossier "main"). Cela permet notamment de développer simultanément et de manière indépendante les deux parties.



Côté backend :

Comme nous pouvons le voir, Spring Boot possède une architecture MVC. Celui-ci permet non seulement une plus grande lisibilité et une meilleure compréhension du code mais également une facilité de modification, car les vue et le code fonctionnel sont séparés, ainsi la modification ou l'ajout de l'un n'impactera pas l'autre.

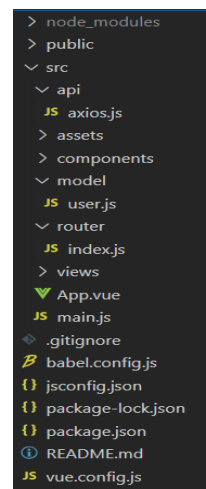
La partie vue correspond donc au frontend, et le backend réuni l'aspect Controller dans le package "controller" et l'aspect Model dans "model"

Enfin, les packages "security" et "payload" permettent la gestion des requêtes et la sécurisation de l'API avec Spring Security.

Le Repository quant à lui, fait l'intermédiaire entre le Model et le Controller.

Côté frontend :

De ce côté, nous avons un projet Vue JS avec la structure que nous connaissons. Un dossier views avec toutes les vues qui implémentent des composants du dossier components. Un fichier App.vue qui implémente les views. Et enfin, des packages additionnels contenant des fichiers JS afin d'implémenter les packages que nous utilisons.



4.2.2 Système d'authentification et sécurité

Afin de développer un système d'authentification sécurisé et de contrôle d'accès basé sur les permissions des utilisateurs, nous nous sommes servis de la dépendance Spring Security.

Il est important de sécuriser l'échange de données entre client et serveur afin d'éviter la fuite de données sensibles ou l'injection SQL.

Pour cela, côté frontend, l'affichage des outils de gestion s'effectue en fonction des rôles et permissions de chaque utilisateur. Mais cela ne suffit pas à sécuriser l'application. Un utilisateur mal intentionné pourra facilement utiliser des moyens détournés afin d'envoyer des requêtes vers le serveur que ne permet pas normalement le frontend, afin de récupérer des données sensibles. Spring Security nous permet donc de sécuriser notre backend et de contrôler les requêtes entrantes. Afin d'expliquer son fonctionnement, nous prendrons en exemple notre système d'authentification ([Annexe 16](#)), personnalisé et adapté à nos besoins :

Tout d'abord, voici comment le processus de connexion fonctionne : Le serveur backend réceptionne la requête contenant le mail et mot de passe entrés dans le formulaire de connexion. La requête est redirigée vers le contrôleur correspondant à l'adresse vers laquelle celle-ci est envoyée et le programme va ensuite vérifier que l'email existe bien et que le mot de passe lui correspond. Les données correspondantes à cet utilisateur seront ainsi facilement retrouvées par une requête simple vers la base de données (de type Select Where email = "X") et renvoyées au client, accompagnées d'un token généré qui permettra par la suite de confirmer l'authentification pour les prochaines requêtes. Le frontend réceptionne ces données, stocke le token dans une variable locale et génère l'affichage nécessaire, c'est-à-dire un message d'erreur si les informations ne sont pas correctes, ou vers le tableau de bord correspondant aux rôles et permissions de l'utilisateur si les informations sont correctes.

Ensuite, la création d'un nouvel utilisateur est disponible uniquement sur la page "new user" par l'utilisateur connecté est conditionnée par l'acquisition des permissions nécessaires. Ainsi, lorsque celui-ci va envoyer une requête vers le serveur backend avec les informations nécessaires, c'est-à-dire nom, prénom, mail, mot de passe et le token de l'utilisateur connecté, celle-ci va être analysée et traitée. Tout d'abord, le token que transmet l'utilisateur va être contrôlé. Celui-ci est en effet généré lors du processus de connexion vu précédemment lorsque la connexion a été validée, ainsi, le programme va pouvoir comparer ce token avec celui enregistré sur la session côté serveur, et certifier, ou non, que l'utilisateur est bien connecté. Si le token transmis est donc valide, le programme sait que la requête provient du même utilisateur ayant établi une connexion précédemment. Il pourra donc consulter ses informations et notamment les permissions accordées par les rôles qu'il détient et s'assurer qu'il possède donc bien les bonnes pour cette requête en particulier.

→ *En pratique, par manque de temps et en raison de la complexité du fonctionnement de Spring Security et du système d'authentification ([Annexe 17](#)), nous n'avons pas réussi à adapter le système de connexion afin que cette vérification se fasse sur les permissions. Nous avons alors décidé de baser les conditions simplement sur les rôles détenus par l'utilisateur connecté.*

Cela se traduit dans le code par le paramétrage du fichier Web SecurityConfig, dans la fonction configure afin de restreindre l'accès à un certain type de "routes", comme ceci par exemple:

```
.antMatchers( ...antPatterns: "/api/test/**").hasRole("president")  
.antMatchers( ...antPatterns: "/api/role/**").permitAll()
```

Ou bien en ajoutant la balise "@PreAuthorize()" au dessus de la déclaration du controller afin de restreindre l'accès uniquement sur ce contrôleur en particulier comme ceci :

```
@DeleteMapping("/delete/{id}")  
@PreAuthorize("hasAuthority('president')")  
@Transactional  
public String deleteUser(@PathVariable("id") Long id) {...}
```

Une fois les conditions remplies, le programme va vérifier qu'aucun utilisateur n'existe avec ce mail. Ensuite, les informations (qui ont été mises sous forme d'un Signup Request) sont enregistrées dans un objet User, en cryptant au passage le mot de passe avec un PasswordEncoder que fournit Spring Security, qui est ensuite enregistré par appel de la fonction save du repository "userRepository". Si aucun rôle n'a été envoyé, alors par défaut, le nouvel utilisateur aura le rôle d'adhérent. Enfin un message de confirmation de création de l'utilisateur est créé. Si jamais au cours de ce processus, une erreur était levée, un message précisant l'erreur sera envoyé au client.

Enfin, pour toutes autres requêtes spécifiques à une certaine permission, le principe de vérification des droits est le même. On contrôle le token certifiant une connexion en amont et vérifie que l'utilisateur possède bel et bien les permissions nécessaires.

Tout ce système permet alors de traiter de manière sécurisée les requêtes entrantes et de renvoyer au client les données demandées. Or, il est toutefois possible qu'un utilisateur mal intentionné se positionne entre le client et le serveur afin d'intercepter leurs échanges et récupérer les informations sensibles, comme notamment le mail et mot de passe qu'envoie le client au serveur pour l'authentification. Ce type d'attaque se nomme "Man In The Middle". Heureusement, le paramétrage du serveur d'hébergement lors du déploiement du site web afin que celui-ci fonctionne sur la base du protocole HTTPS nous permettra d'éviter ce genre d'attaque, et bien plus encore (meilleur référencement, confiance des visiteurs et clients, performance, etc)

4.3 Difficultés rencontrées

4.3.1 Retard sur le planning initial

Nous avons eu certaines difficultés à respecter le planning initialement prévu (diagramme de Gantt prévisionnel Vs diagramme de Gantt réel) ([Annexe 4 /Annexe 5](#)) notamment à cause du peu de temps dont nous disposions. Les raisons de ces retards sont diverses. Tout d'abord, un facteur décisif de ce retard fut le manque d'efficacité de certains membres dû à la maladie ou à un simple manque d'investissement. De plus, "l'écart" la différence de compétences au sein du groupe était un risque que l'on avait probablement sous-estimé. Les charges de travail ont ainsi été très déséquilibrées, imposant des charges

beaucoup trop lourdes à certains membres et l'avancée du projet n'a donc pas pu être efficace.

Par ailleurs, à cause de l'état de santé compliqué de l'un des membres du projet, nous avons failli accumuler un retard conséquent. Nous avons essayé de rattraper celui-ci au retour de notre membre, avec succès, grâce à l'aide d'un autre membre du groupe.

4.3.2. Difficultés techniques

Le développement fut une étape assez longue et fastidieuse.

- Tout d'abord, l'installation même du projet fut assez compliquée. En effet, nous devions partir de zéro et ne savions pas encore quelles dépendances nous utiliserons. Nous avons donc mis un certain temps à installer notre projet, en particulier le backend.
- Une des réelles difficultés rencontrées fut l'implémentation d'un système d'authentification sécurisé. Une partie des problèmes rencontrés étaient dus à notre manque de connaissances théoriques quant au fonctionnement de ce système. N'ayant jamais eu de cours de sécurité web à ISIS ou dans nos formations antérieures, il nous était difficile d'identifier les différentes manières dont un site pouvait être piraté et par conséquent, comment résoudre de manière pratique ces problèmes. Dans l'optique d'acquérir de nouvelles connaissances, objectif principal de ce projet, nous avons pris notre mal en patience et essayé tant bien que mal de comprendre les rouages de ce mécanisme. Une des premières versions de notre backend avec Spring Security fut un échec et nous n'arrivons pas à obtenir un système de connexion adapté à nos besoins. Nous avons alors décidé de mettre de côté Spring Security et d'y revenir par la suite. C'est donc plus tard que nous avons trouvé un [tutoriel](#) fournissant tous les éléments nécessaires à l'implémentation de Spring Security au Projet. Toutefois, la compréhension de la logique du code fut assez longue et difficile. De plus, il était nécessaire de modifier un bon nombre de ces éléments afin de l'adapter à notre projet sans causer d'erreurs irréparables. Bien que délicat, à force de persévérance, l'opération fut un succès, même si ce système ne répond pas à 100% à nos exigences spécifiques.

4.3.3. Solutions apportés

- Solutions au "retard sur le planning initial"

Nous avons établi différentes propositions et nous en avons longuement discuté ensemble, nous avons en premier lieu hiérarchiser les tâches par ordre de priorité et en fonction de nos compétences techniques, nous avons par la suite créer la matrice RACI pour la répartition des tâches et le diagramme de Gantt pour identifier les dépendances des tâches et avoir un calendrier global des tâches.

- Solutions aux difficultés techniques

Notre principal problème étant notre manque de connaissances sur la sécurisation d'un site web, nous nous sommes accordé du temps afin de pouvoir nous renseigner sur le sujet. Cela s'est fait par la recherche de ressources sur internet, mais aussi par la sollicitation de proches dans le domaine, mais aussi de M.Pecatte que nous remercions.

V. Etat d'avancement du projet

5.1 Objectifs atteints

Voici, un tableau récapitulant les objectifs techniques de notre projet ainsi que leur état d'accomplissement.

Objectifs		Etat d'avancement
Squelette du site web		Atteint
Création d'une page d'accueil		Atteint
Côté visiteur	Système d'authentification sécurisé	Atteint
	Page "Qui sommes-nous?" (Backend)	En cours
	Page "Qui sommes-nous?" (Frontend)	En cours
	Consultation des fiches de poste + postes à pourvoir	Non atteint
	Formulaire "Retour client"	Non atteint
	Page "Articles"	Non atteint
Côté tableau de bord	CRUD User	En cours / Partiellement atteint
	CRUD Rôles	En cours
	Rédaction d'articles	Non atteint
	Page "Edition d'articles"	Non atteint
	Consultation des retours client	Non atteint

5.2 Améliorations

En conséquence des événements ayant causé du retard dans notre planification des tâches à accomplir, des concessions, en particulier sur la partie réalisation, ont dû être faites. En effet, face à la complexité du système d'authentification, malgré le fait d'être parvenu à adapter certains aspects de ce processus, nous ne sommes pas parvenu à faire en sorte d'imposer les contraintes d'accès aux données sur les permissions. Cela correspondrait mieux aux besoins de l'association car en effet, il est possible, à travers les différents mandats, que de nouveaux postes soient créés. Les postes étant représentés par la classe "Rôle" dans notre base de donnée, il sera alors facilement possible de créer un nouveau poste, et donc un

nouveau rôle, personnalisé en choisissant les permissions qui lui seront accordées. En accord avec notre modèle de donnée, un rôle possède donc plusieurs permissions et une permission sera accordée à plusieurs rôles.

L'aspect visuel a également été mis de côté afin de nous concentrer plus sur l'aspect fonctionnel. Celui-ci pourrait donc être très fortement amélioré.

Enfin, lors de la réalisation du système d'authentification et après avoir discuté avec monsieur Singer, nous avons tenté de nous renseigner sur l'utilisation des services de Firebase afin d'y héberger notre base de données et ainsi augmenter considérablement la sécurité de notre application web. Malheureusement, après de longues tentatives, nous ne sommes pas parvenus à résoudre nos soucis de compatibilité avec les frameworks frontend et backend que nous utilisons. Dans le cadre d'une potentielle reprise du projet, il serait donc intéressant de réussir à utiliser ces services.

5.3 Suite du projet

Nous n'avons pas pu atteindre tous les objectifs souhaités, il reste quelques fonctionnalités à implémenter sur le site Web de la Junior Entreprise. Dans les années futures, notre projet pourrait être repris afin de compléter le cahier des charges, de rendre le site web plus dynamique (voir [site web ESCadrille \(Annexe 14 \)](#)) et d'ajouter de nouvelles fonctionnalités dont pourrait avoir besoin la JE.

Avec les bases que nous avons construites et les fonctionnalités que nous avons déjà implémentées, l'ajout de fonctionnalités additionnelles pourra être fait plus efficacement et rapidement. Il suffirait de prendre en exemple les fonctionnalités existantes, reprendre le rapport et les ressources que nous avons utilisé et la reprise du projet devrait se faire de manière fluide.

VI. Conclusion

Nous avons comme ambition de rendre le site Web d'Horus plus dynamique et attrayant. Dans un premier temps, nous nous sommes servi de nos connaissances en gestion de projet pour planifier au mieux l'avancement des tâches. Nous nous sommes ensuite appuyés sur les cours de Technologie Web pour nous permettre d'implémenter de nouvelles fonctionnalités sur le site.

Ce projet nous a permis d'acquérir une plus grande rigueur dans notre suivi de projet, une plus grande connaissance des étapes que l'on retrouve derrière un projet, et un enseignement plus pertinent de la gestion des risques que l'on peut rencontrer.

Ce fut un projet stimulant qui nous a beaucoup appris, tant au niveau des connaissances techniques qu'au niveau de la gestion et planification. Chacun de nous retire de ce projet un enseignement qui pourra être utilisé dans le futur.

Dû à un manque de temps, tous les objectifs n'ont pas été atteints mais le principal est accompli. Nous laissons aux futurs développeurs un code compréhensible et rigoureux qui représente une base solide à laquelle ils pourront ajouter facilement de nouvelles fonctionnalités.

VII- Bibliographie

Nom	Lien(s)
Cahier des charges d'Horus:	Cahier des charges Intranet_ERP.pdf
Sites Web d'autres Junior Entreprises	https://emlyonjuniorconseil.com/expertises/ https://www.hec-junior-conseil.fr https://junior-consulting-sciencespo.com/blog/ https://www.juniorinsaservices.fr https://www.escadrille.org
Spring Rest	https://spring.io/guides/tutorials/rest/
Spring Security	https://spring.io/projects/spring-security https://medium.com/@gustavo.ponce.ch/spring-boot-spring-mvc-spring-security-mysql-a5d8545d837d
Explication du fonctionnement de la couche Service	https://blog1.westagilelabs.com/why-to-use-service-layer-in-spring-mvc-5f4fc52643c0
Fonctionnement combinées de Spring Boot + VueJs :	https://www.danvega.dev/blog/2021/01/22/full-stack-java-vue/ https://simplesolution.dev/spring-boot-restful-api-vuejs-frontend/
VueJS - Router	https://www.digitalocean.com/community/tutorials/how-to-navigate-between-views-with-vue-router
VueJS - Axios	https://5balloons.info/post-form-data-to-api-using-axios-in-vuejs/
VueJS CRUD - Router + Axios	https://www.bezkoder.com/vue-3-crud/
Git	https://www.nobledesktop.com/learn/git/git-branches
Pour la compréhension de la logique d'authentification	https://medium.com/@albert.kim/understanding-user-authentication-in-your-web-app-and-how-to-implement-it-part-1-the-high-level-ab91336ab77c
Système d'authentification	https://www.bezkoder.com/vue-3-authentication-jwt/ https://www.bezkoder.com/spring-boot-jwt-authentication/
Firebase :	https://learnvue.co/2021/06/a-vue-firebase-authentication-tutorial-vue-3-and-firebase/

VIII - Annexes

Annexe 1 : Note de cadrage

Implémentation d'un intranet et de fonctionnalités sur le site web d'Horus HCS

Horus HealthCare Systems est une junior entreprise partenaire de la CNJE, créée en 2012. Au cours du mandat 2022-2023, elle comporte 7 membres au conseil d'administration. Il s'agit d'une association en plein essor qui cherche à se dynamiser et à gagner en visibilité grâce à son site internet. La demande nous a été délivrée le 20 Février 2022.

Afin de bien analyser le besoin et définir les objectifs du projet, nous travaillerons en partenariat avec le responsable des systèmes d'informations de la Junior, ainsi qu'avec son président. Ayant déjà un site internet, ce projet vise à mettre à jour celui-ci en y implémentant des fonctionnalités de gestion et améliorer la partie vitrine. L'objectif est donc également d'apporter un aspect plus pratique au site web et gagner en visibilité. Pour cela, il a été convenu d'implémenter un intranet ainsi qu'un outil de gestion des utilisateurs et d'édition d'articles.

Un premier rendu d'un rapport technique nous est demandé pour le 13 avril 2022. Enfin, nous préparerons une soutenance pour le 22 avril 2022. Cela nous laisse donc la courte période de 2 mois afin de mener à bien ce projet, période bien trop courte en raison de la charge de travail à réaliser et des cours, projets et partiels que nous devons suivre en parallèle.

Nous avons donc décidé de répartir le travail en deux parties qui se réaliseront de manière parallèle. Une première partie gestion de projet effectuée par Ikhlass HAJJAM et Anaïs TAYLOR. Une deuxième de développement de l'application web réalisée par Hamza MAKRI. Si la charge de travail s'avère être bien trop grande par rapport au temps qui nous est alloué, nous effectuerons une mitigation des objectifs.

Certaines étapes du cycle de vie d'un projet devront également être sautées.

Nous avons un budget de 0 euro pour mettre à bien ce projet.

Le comité de pilotage constitué de :

Hamza Makri, chef de projet et responsable des SI à Horus

Ikhlass Hajjam, chargée des systèmes d'information à Horus

Anais Taylor, étudiante à ISIS

Nous pouvons de plus nous adresser à notre enseignant référent, M. Singer, afin d'obtenir quelques conseils tout au long du projet.

Annexe 2 : RACI prévisionnel

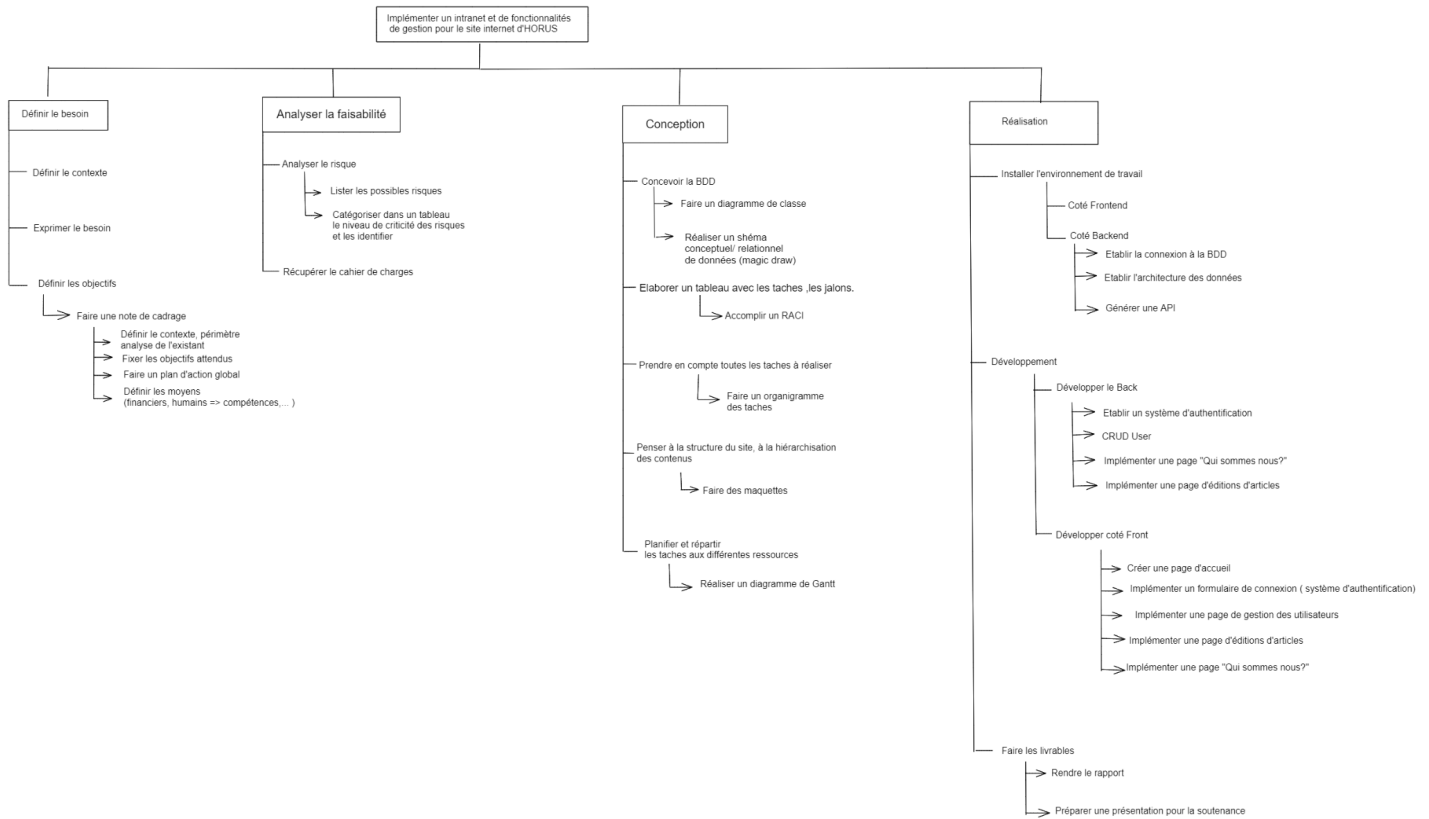
Les tâches			Ikhlass	Anaïs	Hamza
Définir le besoin	Définir le contexte		A	R	A
	Exprimer le Besoin		R	A	A
	Définir les objectifs⇒ Faire une note de cadrage	Définir le contexte, périmètre, analyse de l'existant	A	R	A
		Fixer les objectifs attendus	A	A	
		Faire un plan d'action global	R	R	R
		Définir les moyens (financiers, humains selon les compétences par exemple,...)	R	R	
Analyse et faisabilité	Faire un cahier de charges		C	C	C
	Analyser le risque	Lister les possibles risques	A	R	I
		Catégoriser dans un tableau le niveau de criticité des risques et les identifier	A	R	I
		Faire la matrice de criticité des risques	R	A	I
Conception	Concevoir la BDD	Faire un diagramme de classe	R	C	
		Réaliser un schéma conceptuel/relationnel de données	R	C	

	Réaliser un diagramme de Gantt 1 prévisionnel		R	R	A	
	Accomplir un RACI		R	I	C	
	Faire un organigramme des tâches		R	C	A	
	Faire des maquettes pour la structure du site		R	A	A	
Réalisation	Installer l'environnement de travail	Côté Frontend		I	I	R
		Côté Backend	Création de la BDD			
			Construction du MVC			
			Paramétrage de l'API auto-généré			
	Développement	Côté Frontend	Créer une page d'accueil	I	I	R
			Système d'authentification	I	I	R
			CRUD User	I	I	R

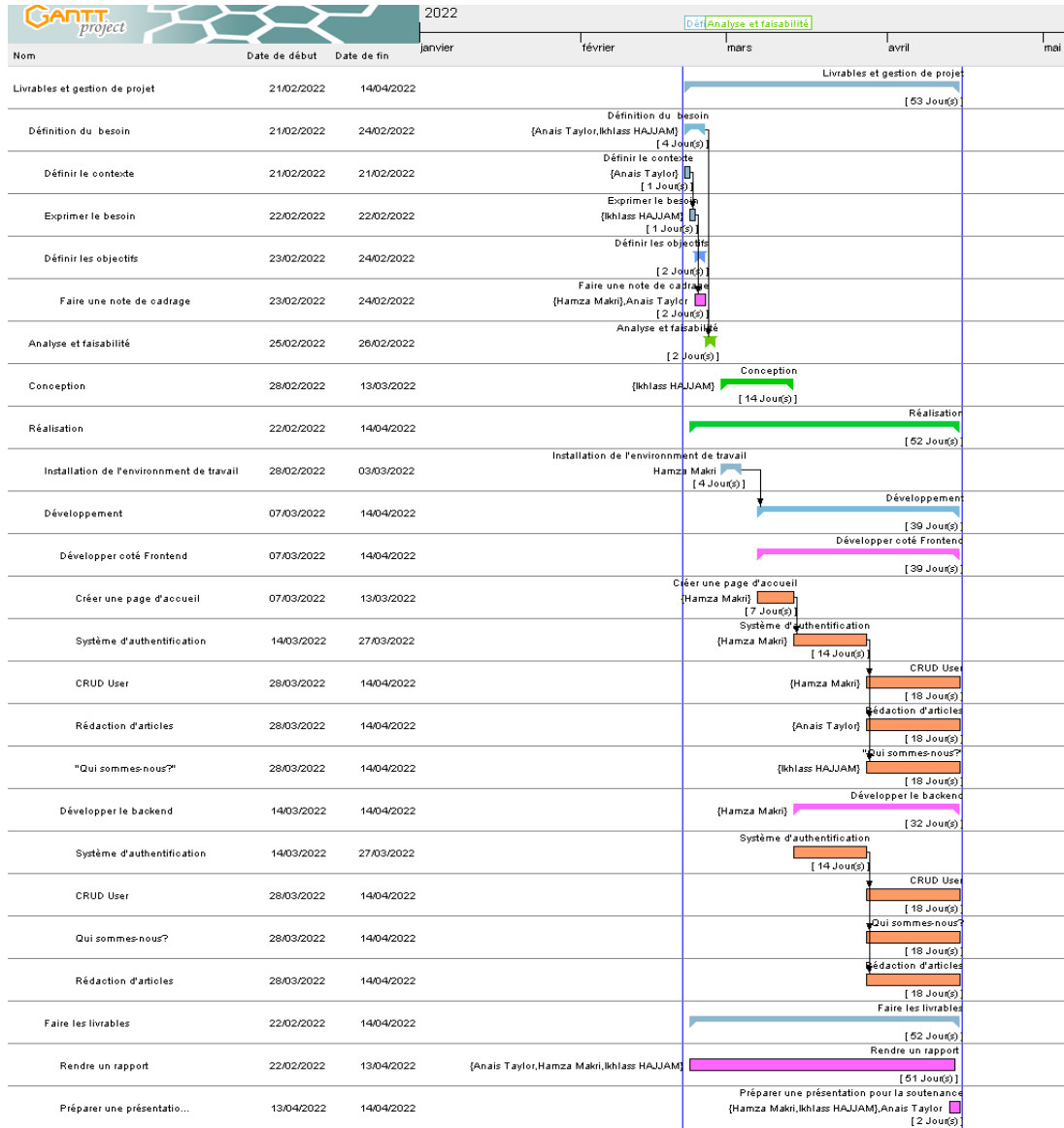
			la page de la rédaction d'articles	I	R	A
			la page "Qui Sommes-nous?"	R	I	A
		Côté Backend	Système d'authentification	I	I	R
			CRUD User	I	I	R
			La page "Qui Sommes-nous?"	C	I	R
			Page "Articles"	I	A	R
	Faire les livrables	Rendre le rapport		R	R	R
		Préparer une présentation pour la soutenance		R	R	R

R	Réalisateur
A	Approbateur
C	Consultant
I	Informé

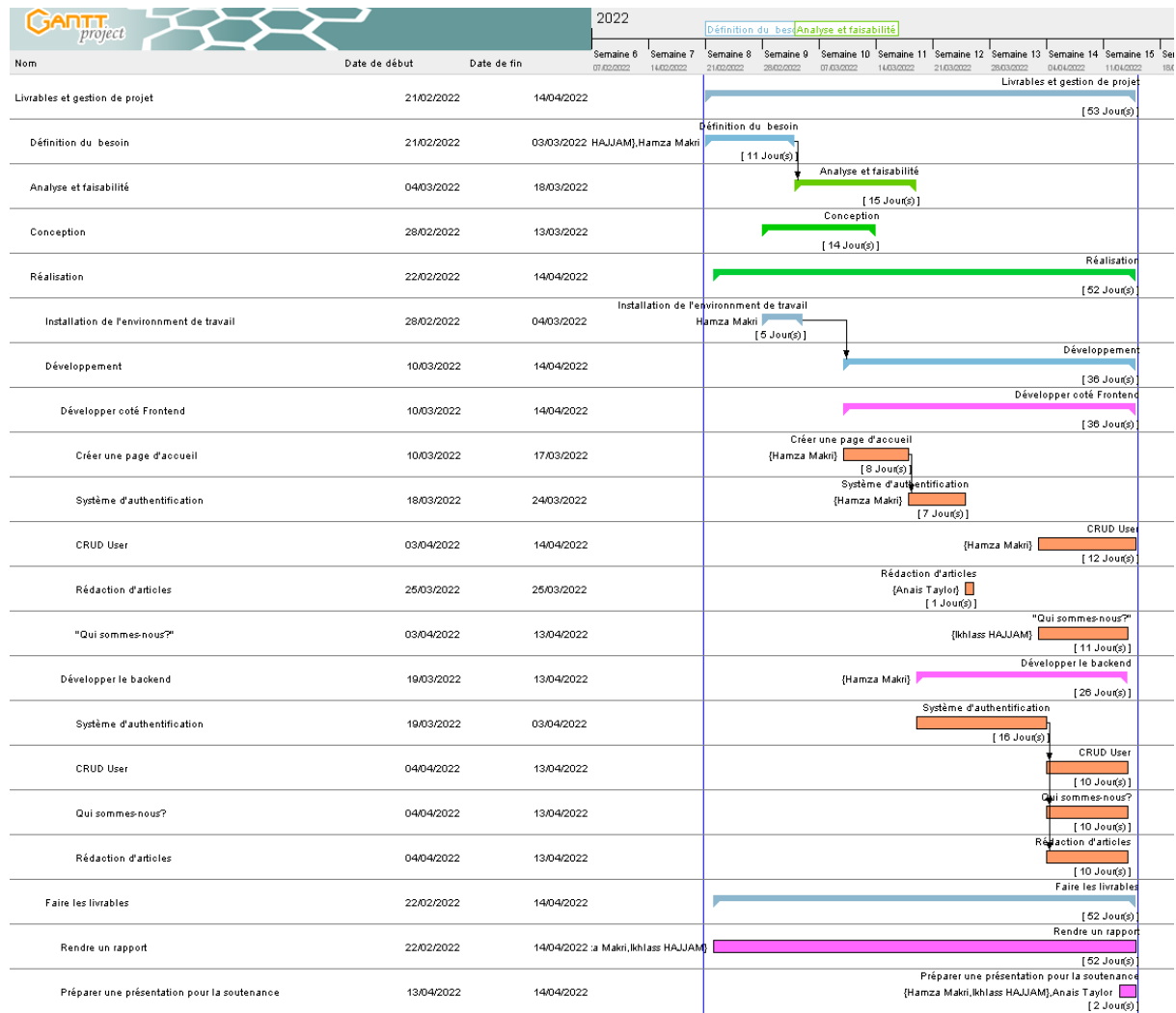
Annexe 3 : Organigrammes des tâches (WBS)



Annexe 4 : Diagramme de Gantt prévisionnel



Annexe 5 : Diagramme de Gantt réel:



```
classDiagram
    class User {
        attributes
        -user_id : Integer
        -first_name : String
        -last_name : String
        -email : String
        -password : String
        -avatar_path : String
    }
    class Role {
        attributes
        -role_id : Integer
        -name : String
    }
    class Permission {
        attributes
        -name : String
        -permission_id : Integer
    }
    class Article {
        attributes
        -article_id : Integer
        -title : String
        -Content : String
        -img : String
    }
    class Pole {
        attributes
        -pole_id : Integer
        -name : String
        -description : String
    }
    class user_role {
        attributes
        -user_id : Integer
        -role_id : Integer
    }
    class role_permission {
        attributes
        -permission_id : Integer
        -role_id : Integer
    }
    User "1" -- "*" Role : -utilisateur / -poste
    Role "*" -- "*" Permission : -role / -permission
    User "1" -- "*" Article : -ajouter / -article
    Role "*" -- "*" Pole : -role_pole / -pole
    user_role "1" -- "1" User
    role_permission "1" -- "1" Role
```

```

classDiagram
    class tfredérick : User {
        first_name = "Tristan"
        last_name = "Frédéric"
        poste = president
        user_id = 0
    }
    class president : Role {
        name = "Président"
        permission = supprimerUtilisateur, ajouterUtilisateur
        pole = bureau
        role_id = 0
        utilisateur = tfredérick
    }
    class bureau : Pole {
        name = "pole Bureau"
        pole_id = 1
        role = president
    }
    class supprimerUtilisateur : Permission {
        name = "supprimer"
        permission_id = 0
        role = president
    }
    class ajouterUtilisateur : Permission {
        name = "ajouter"
        permission_id = 1
        role = president
    }
    tfredérick --> president : : user_role
    president --> bureau : : role_pole
    president --> supprimerUtilisateur : : role_permission
    president --> ajouterUtilisateur : : role_permission

```

27

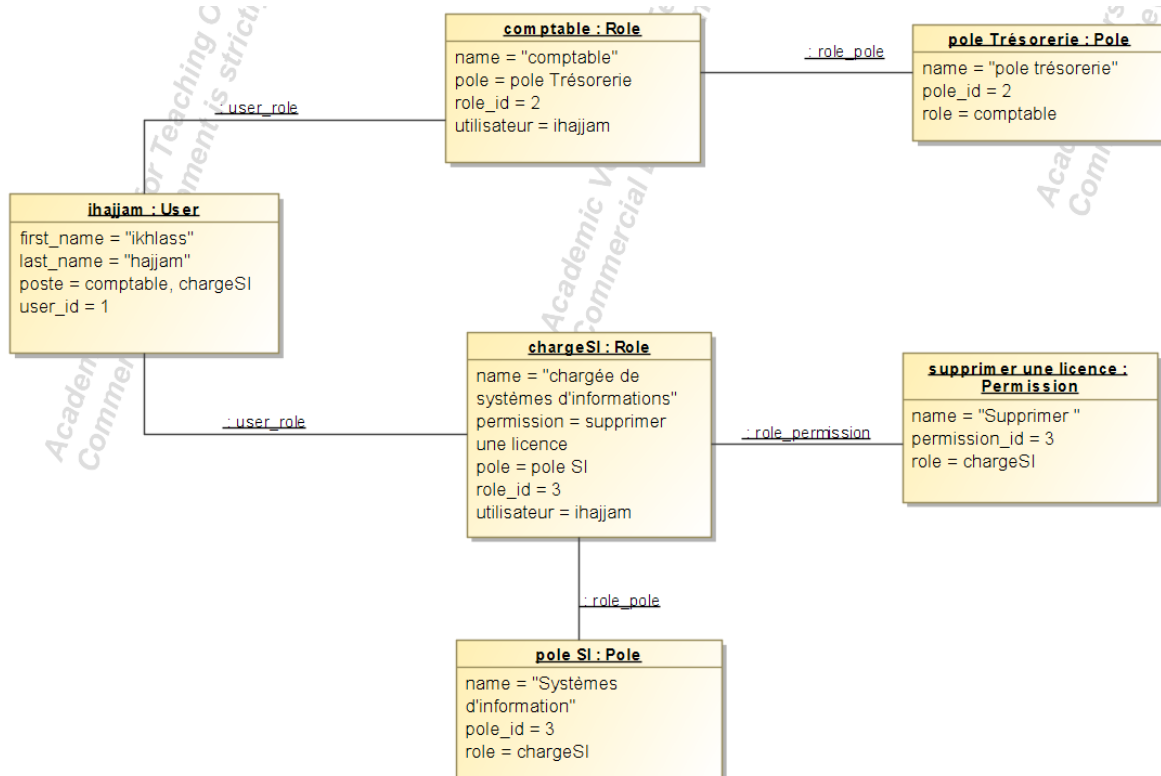


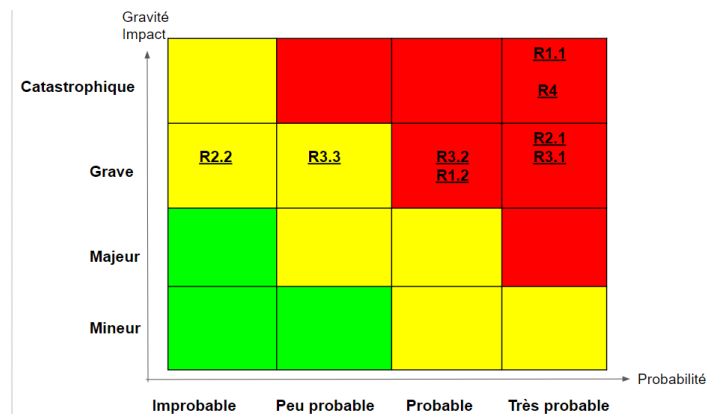
Diagramme d'objets 2:

Annexe 7 : Tableau de criticité

Évènement redouté		Impacts	Niveau de criticité	Stratégie de mitigation
<u>R1: Risques techniques:</u>	R1.1: Mauvaise maîtrise des outils et langages imposés	Familiarisation et apprentissage des outils trop longue, perte de temps	ÉLEVÉ	Auto formation + aide de notre tuteur ainsi que des autres enseignants
	R1.2: Outils informatiques non adaptés au projet	Objectifs inatteignables	ÉLEVÉ	Adaptation des fonctionnalités
<u>R2: Risque lié à la gestion de projet:</u>	R2.1: Manque d'expertise de la maîtrise d'oeuvre (Manque d'expérience en conduite de projet)	Mauvaise organisation et répartition des tâches, perte de temps	ÉLEVÉ	Mise au point régulière sur l'avancée du projet
	R2.2: Demandes de changements (notamment d'objectifs) en cours de projet	Perte de temps, vision confuse de la ligne directrice du projet	MODÉRÉ	Réunion de mise au point pour s'assurer que les nouveaux objectifs sont compris + mitigation des objectifs en cas de perte de temps
<u>R3: Risque lié aux ressources humaines:</u>	R3.1: Manque de compétences techniques et de connaissances pour la partie réalisation	Perte de temps en formation des membres	ÉLEVÉ	Auto-formation + aide de nos enseignants + mitigation des objectifs + répartition des tâches en fonction des compétences de chacun
	R3.2: Mésententes sur les objectifs, tensions, manque d'investissement	Manque de cohésion au sein du groupe	BAS	(Communication)

	R3.3: Absence d'un membre ou plusieurs membres	Retard, désorganisation dans le travail et augmentation de la charge de travail pour les autres membres du groupe	MODÉRÉ	Répartir la charge de travail entre les autres membres de l'équipe pour pouvoir avancer
<u>R4: Risque sur les délais</u>	R4: Délai trop court	Objectifs non atteints, projet incomplet ou "bâclé"	ÉLEVÉ	Mitigation des objectifs Surestimer le temps nécessaire pour l'accomplissement des tâches lors de la planification ⇒ Diagramme de Gantt

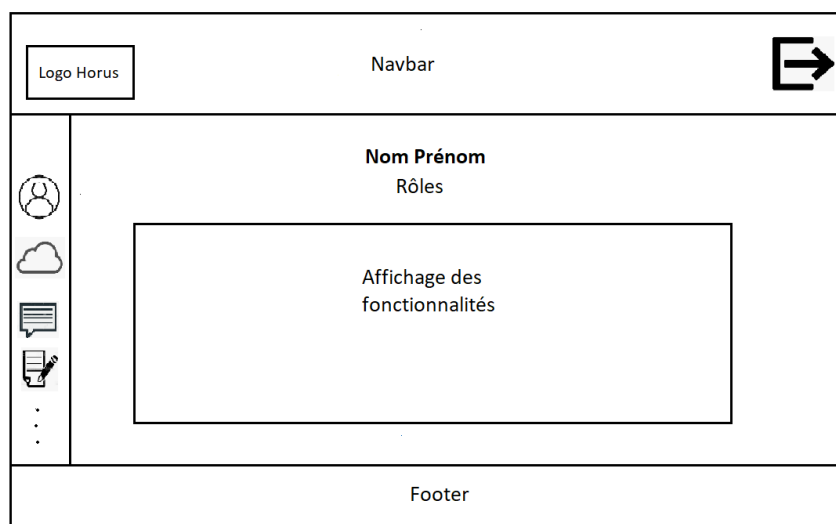
Annexe 8 : Matrice de criticité



Annexe 9 : Maquette - Page d'Accueil



Annexe 10 : Maquette - Page de connexion



Annexe 11 : Maquette - “Qui sommes-nous?”



Annexe 12 : Matrice RACI réelle :

Tâches Réalisées			Hamza Makri (chef de projet)	Anais Taylor	Ikhlass Hajjam
Les éléments essentiels pour la Gestion de projet:	Note de cadrage		R	R	C
	Organigramme des tâches		R/A/C	A	R
	La matrice RACI		A	C	R
	Diagramme de Gantt		A	R	R
	Analyse des risques		R	R	R
	Matrice de criticité		C	I	R
Eléments techniques:	Maquettes		A	C	R
	Diagramme de classe UML		R/C	I	R
	Diagramme d'objets		A	I	R
La rédaction du rapport:	I- Cadrage:	Définition du contexte	R	R	C
		Expression des besoins	A	C	R
		Analyse de l'existant interne	R/A	R	I
		Analyse de l'existant externe	A	R	R
	II-Analyse et faisabilité:	Définition des objectifs	R	R	R
		Contraintes	R	R	
		Risques	R	C	
		Tableau de criticité des risques et mitigation	R	I	R
		Matrice de criticité des risque	A/C	I	R
	III-Conception:	Organigramme des tâches	R/A/C	I	R
		Diagramme de Gantt	A/C	I	R

		La matrice RACI	A	C	R
		Maquettage	A/C	C	R
	IV-Réalisation	Choix des moyens techniques - Cohérence et pertinence	R	I	I
		Architecture logiciel	R	I	I
		Système d'authentification et sécurité	R	I	I
		Difficultés rencontrées	R	I	R
		Retard sur le planning initial	R	I	R
		Difficultés techniques	R	I	I
		Solutions apportés	R	I	R
	V. Etat d'avancement du projet	Objectifs atteints	R	R	R
		Améliorations	R	R	I
		Suite du projet	R	R	I
	VI. Conclusion		R/A/C	R	I
	Résumé		R/A	I	R
	Sommaire /Remerciements		A	C	R
Développement	Installation de l'environnement du travail		R	I	I
	Coté backend	Création de la page d'accueil	R	I	I
		Système d'authentification	R	I	I
		CRUD user	R	I	I
		Page "Articles"			

		La page "Qui sommes-nous?"	R	I	I
	Coté frontend	Création de la page d'accueil	R	I	I
		Système d'authentification	R	I	I
		CRUD user	R	I	I
		Rédaction d'article			
		La page "Qui sommes-nous?"	A/C	I	R

R	Réalisateur
A	Approbateur
C	Consultant
I	Informé

Annexe 13 : Site Web de la junior entreprise de L'ESCadrille :



Annexe 14 : Site Web de la junior entreprise de HEC :



Annexe 15 : Système de connexion de la Junior Entreprise de l'Insa Toulouse:



Connectez-vous

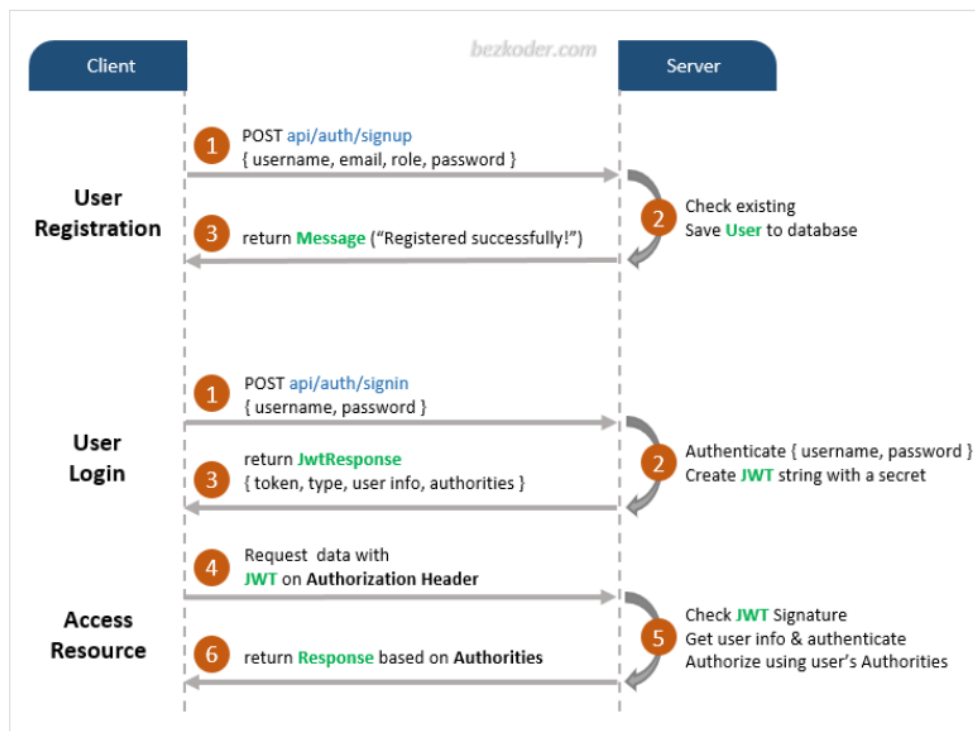
☐ Se souvenir de moi

Connexion

[Vous n'avez pas encore de compte ? Inscription](#)

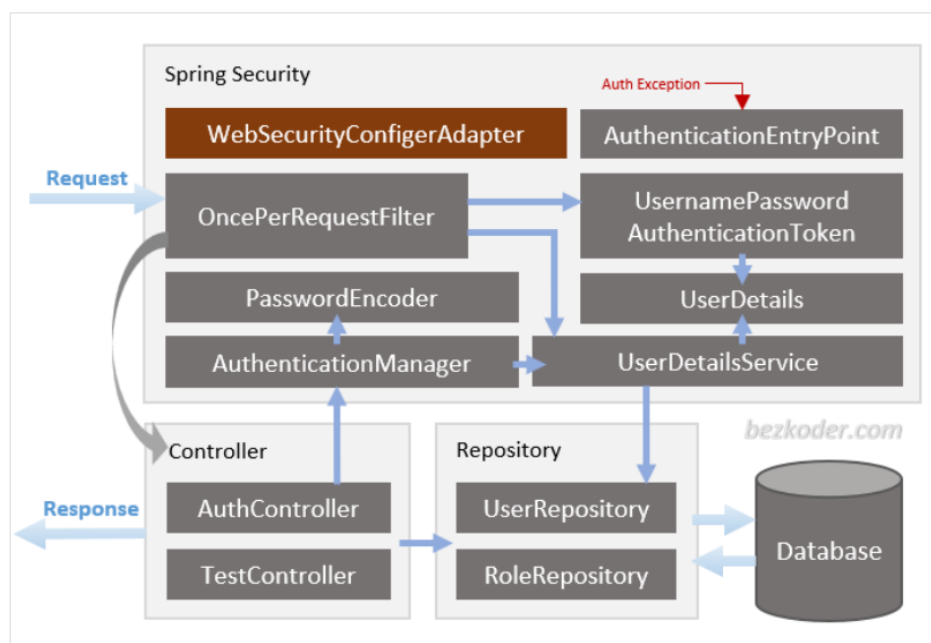
[Mot de passe oublié](#)

Annexe 16 : Schéma des requêtes échangées entre Client et Serveur pour le système d'authentification (avant personnalisation) :



Source : <https://www.bezkoder.com/spring-boot-jwt-authentication/>

Annexe 17 : Schéma de l'architecture logiciel du système d'authentification :



Source : <https://www.bezkoder.com/spring-boot-jwt-authentication/>