

RAPPORT TECHNIQUE DE PROJET TUTEURE

Application Web pour la gestion des mobilités
internationales ISIS



Tuteur pédagogique :
M. Laurent GREGOIRE

Groupe de travail :
Rayane BENCHAREF / Lise GAUTHIER
Kilian LAMULA / Hugo MATTON

Promotion 2024

Résumé

Le présent rapport s'inscrit dans le cadre de la première année du cycle ingénieur ISIS (Informatique et Système d'Information pour la Santé) et constitue une étape primordiale dans notre formation.

Nous avons choisi le projet tuteuré proposé par notre client, Laurent GREGOIRE, concernant la création d'une application web pour la gestion des mobilités internationales à ISIS, car il s'agit d'un sujet qui nous intéressait particulièrement étant donné que dans le cadre de notre cursus, nous devons effectuer au minimum une mobilité internationale. Étant directement concernés par ce sujet, il nous semblait judicieux d'améliorer et d'approfondir les solutions déjà mises en place concernant la gestion des mobilités, du côté administratif mais aussi du côté des étudiants.

Tout au long de notre étude, nous nous sommes entretenus avec Laurent GREGOIRE afin de lui faire part de nos avancées et de lui poser les questions nécessaires à l'avancée du projet.

Afin d'appréhender ce sujet, nous avons d'abord pris connaissance des besoins de notre client et analysé la gestion actuelle des mobilités. Par la suite, nous avons réuni et formalisé toutes ces informations grâce à des éléments d'ingénierie que nous détaillerons dans ce rapport.

En parallèle, nous avons utilisé nos connaissances en gestion de projet afin de lister les tâches, planifier et analyser les potentiels risques que nous aurions pu rencontrer.

Une fois le sujet maîtrisé, nous avons commencé à conceptualiser la solution en maquant les futures interfaces de l'application web. La validation des maquettes par notre client nous a permis de donner vie au projet et de fixer les objectifs à atteindre. Cette étape a été déterminante dans le développement de l'application et la rédaction du présent rapport.

Mots-clés : mobilités internationales, application web, développement fullstack, gestion de projet, Vue.js, projet tuteuré

TABLE DES MATIERES

<i>Présentation du sujet et analyse des besoins</i>	3
Contexte du projet.....	3
Périmètre	3
Besoins.....	4
<i>Éléments d'ingénierie</i>	5
Choix du stack technologique.....	5
Diagramme UML	5
Architecture logicielle.....	6
Côté frontend	6
Côté backend.....	7
Maquettes	7
Utilisation de l'API rest	8
<i>Gestion de projet</i>	10
Analyse des risques.....	10
Organigramme des tâches	11
Répartition des tâches.....	11
Planification.....	13
<i>Prise de recul</i>	14
Difficultés rencontrées et améliorations possibles	14
Prolongements à donner au projet.....	14
<i>Conclusion</i>	15
<i>Bibliographie, Sitographie</i>	16
<i>Table des illustrations</i>	16
<i>Annexes</i>	16

PRESENTATION DU SUJET ET ANALYSE DES BESOINS

Afin de proposer une solution optimale à ce projet, il est important de bien cerner les besoins auxquels nous devons répondre. L'analyse des besoins a été réalisée en coordination avec M. Laurent GREGOIRE et Mme Malena GOMEZ, que nous présenterons ultérieurement.

CONTEXTE DU PROJET

Le projet tuteuré sur la gestion des mobilités a été commandité par l'école d'ingénieurs en informatique ISIS, basée à Castres. Notre client est M. Grégoire, professeur d'anglais, responsable langues et responsable des Relations Internationales (RI) au sein d'ISIS.

Ce projet est un projet de création de solutions. En effet, notre objectif est de répondre au besoin de notre client avec une solution spécifiquement développée pour répondre à ses problématiques.

Le projet vise à créer une application web permettant au client de gérer les mobilités des étudiants ISIS à l'étranger. En effet, au cours de leur cursus, les étudiants ISIS doivent effectuer des séjours à l'étranger (durant 2 à 6 mois). Actuellement, l'administration gère le suivi des mobilités sur tableur Excel, depuis le campus d'Albi. Les échanges entre ISIS et Champollion se font exclusivement par mail. Il paraît donc nécessaire de créer une solution permettant de réunir toutes ces informations et de les rendre disponible simultanément à Albi et à Castres.

La consultation des destinations et des places disponibles qui y sont associées sont disponibles sur l'Intranet de l'INU Champollion ainsi que sur le site d'ISIS, en accès public. Les documents administratifs destinés aux étudiants sont également disponibles sur l'Intranet. L'objectif du projet est donc de centraliser toutes ces données et de proposer une solution plus optimisée pour la gestion des mobilités à l'international.

Pour résumer globalement les besoins de notre client, nous avons créé un diagramme bête à corne.

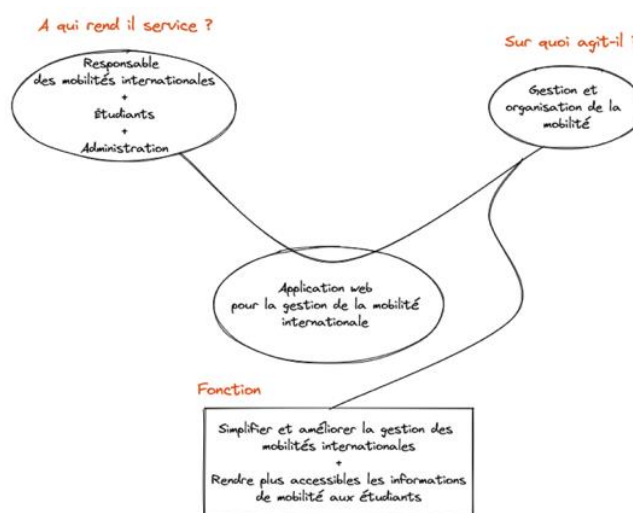


FIGURE 1 : DIAGRAMME BÊTE À CORNES

La figure ci-dessus montre bien que notre application web rend service aux responsables RI, à l'administration et aux étudiants ISIS. Elle permet de gérer et d'assurer le suivi de leur mobilité internationale. Ses fonctions principales incluent la simplification et l'amélioration de la gestion des mobilités, et la mise à disposition d'une interface plus accessible et optimisée pour les étudiants.

Certains acteurs supplémentaires vont être impliqués dans ce projet. En effet, la gestion des mobilités est gérée administrativement au sein de l'INU Champollion, par Malena Gomez (responsable administrative des relations internationales à l'INU Champollion). Nous rentrerons en contact avec elle afin de nous informer sur les réels besoins à satisfaire et les outils actuels utilisés pour la gestion des mobilités.

PERIMETRE

Dans un premier temps, le projet aura pour principaux utilisateurs notre client M. GREGOIRE ainsi que le personnel administratif, qui devra pouvoir administrer l'application web (cf. [Besoins](#)). Les étudiants auront également accès à l'application, en lecture seule. Ainsi, ils pourront consulter les différentes destinations proposées par l'école et accéder aux documents administratifs téléchargeables.

BESOINS

Notre application web a pour objectif d'assurer le suivi des séjours à l'étranger des étudiants ISIS.

L'application doit permettre aux administrateurs de gérer :

- La base de données qui contient les informations concernant :
 - Les mobilités : date de départ, durée, établissement, pays
 - Les organisations étrangères qui les accueillent : nom de l'établissement ou de l'entreprise, nombre de places disponibles par semestre et/ou an, durée restante avant la fin du contrat entre ISIS et l'établissement partenaire
- un tableau de bord permettant de :
 - Suivre rapidement les obligations de séjours internationaux des étudiants (voir quels étudiants sont partis et où).

Les administrateurs gèrent les destinations proposées : ils peuvent en ajouter, les modifier, les supprimer.

Cette application est également dédiée aux étudiants. Ces derniers pourront se renseigner sur les destinations proposées par l'école pour effectuer un semestre, une année, un stage à l'étranger ou une mission humanitaire. Seront renseignés :

- le type de mobilité (mobilité d'étude ou de stage)
- la liste et les détails des campus
- une photo du campus (facultative)
- le nombre de places disponibles par semestre ou par an (dans le cas d'une mobilité d'étude)
- le nombre de candidatures par offre de stage (dans le cas d'un stage)

Ils devront également pouvoir y retrouver l'ensemble des documents administratifs afin de préparer leur départ.

Pour formaliser les besoins de notre client, nous avons utilisé des outils de gestion de projet dont le diagramme pieuvre :

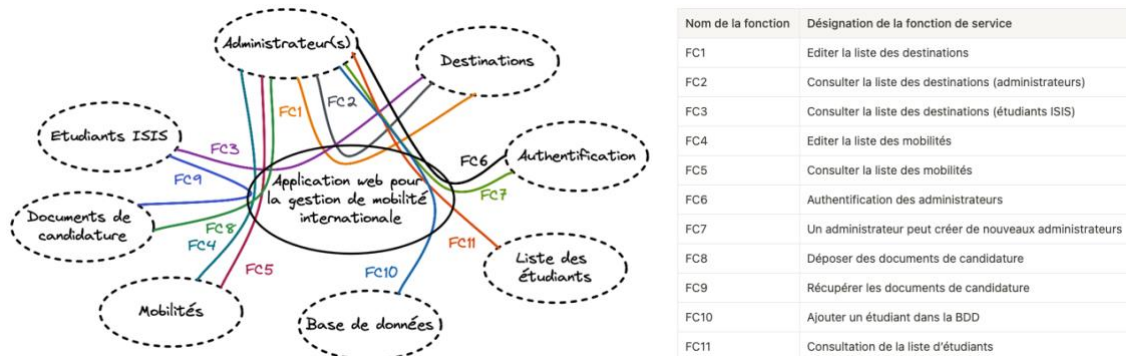


FIGURE 2 : DIAGRAMME PIEUVRE

Le diagramme ci-dessus met en évidence les services à rendre par l'application web. Ces services correspondent à des fonctions mises à disposition de l'utilisateur qui sont explicitées dans le tableau de droite. Mis en relation, les éléments du milieu extérieur (EME) qui entourent la « bulle centrale » - dans notre cas, l'application web - répondent aux besoins formulés précédemment.

L'application devra également être simple d'utilisation et ergonomique afin que le client ne perde pas de temps lorsqu'il y effectuera des tâches. De plus, elle devra être esthétique afin que les étudiants aient envie de l'utiliser (confort de lecture, informations facilement accessibles).

Le code devra être facile à comprendre et à maintenir afin que les prochains étudiants qui reprendront le projet puissent facilement l'améliorer et y ajouter des fonctionnalités.

L'ensemble des fonctionnalités sont regroupées dans le tableau en Annexe 1.

ÉLÉMENTS D'INGENIERIE

Pour mener à bien notre projet, nous avons utilisé de nombreux éléments technologiques et méthodologiques d'ingénierie que nous présentons dans cette partie

CHOIX DU STACK TECHNOLOGIQUE

Le tableau ci-dessous présente les technologies que nous avons utilisées pour développer le front-end et le back-end de notre application web.

Front-end		Back-end	
Langage front-end	HTML/CSS	Langage de programmation	Java
Framework JavaScript	Vue.js	Framework Java	Spring Boot
Framework CSS	Bootstrap	SGBD	PostgreSQL

FIGURE 3 : CHOIX DU STACK TECHNOLOGIQUE

La maîtrise de la majorité de ces langages/technologies a été acquise grâce aux enseignements ISIS mais certains ont impliqué une formation personnelle.

DIAGRAMME UML

Nous avons conceptualisé notre solution avec un diagramme UML, disponible en Annexe 2 : Diagramme UML.

Sur celui-ci, nous pouvons voir les principales entités composant notre application, parmi lesquelles figurent les étudiants et les destinations. Lorsque l'on associe ces deux entités, une mobilité est créée. Nous avons modélisé cela par une classe-association car, connaissant un étudiant, on veut connaître la liste des destinations où il a effectué une mobilité, ainsi que la date de départ, la durée, etc.

Nous avons hésité à utiliser un modèle entité-association mais, pour plusieurs raisons, nous avons abandonné cette idée. En effet, si l'on fait les configurations nécessaires, le framework Hibernate (qui gère la persistance des objets en base de données) permet de générer automatiquement la base de données qui correspond à l'UML que l'on implémente dans notre code.

Nous avons donc d'abord réalisé un diagramme UML, puis nous avons produit le code java correspondant à notre diagramme afin qu'Hibernate puisse générer automatiquement la base de données. Pour cela, nous avons rajouté des annotations dans notre code afin que le framework puisse comprendre les liens qu'il y a entre nos différentes classes et quelles sont les tables à créer dans la base de données (@Entity, @Id ...). La Figure 4 ci-dessous est un exemple de classe annotée avec Hibernate.

```
@Entity
public class Etudiant {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @NotNull
    private Integer numEtud;

    @NotNull
    private String nom;

    @NotNull
    private String prenom;

    @NotNull
    private Integer promo;

    @JsonIgnore
    @OneToMany(mappedBy="etudiant")
    @ToString.Exclude
    private List<Mobillite> mobilites = new ArrayList<>();
}
```

FIGURE 4 : ENTITE "ETUDIANT" AVEC LES ANNOTATIONS HIBERNATE

Une fois cela fait, nous avons les fondations de notre projet sur lesquelles nous avons pu nous appuyer pour développer l'application. Il ne nous a donc pas paru utile de réaliser un modèle entité-association étant donné que nous utilisons Hibernate.

ARCHITECTURE LOGICIELLE

COTE FRONTEND

Concernant l'architecture logicielle côté frontend nous avons fait le choix d'organiser nos fichiers dans des dossiers, selon la figure ci-dessous :

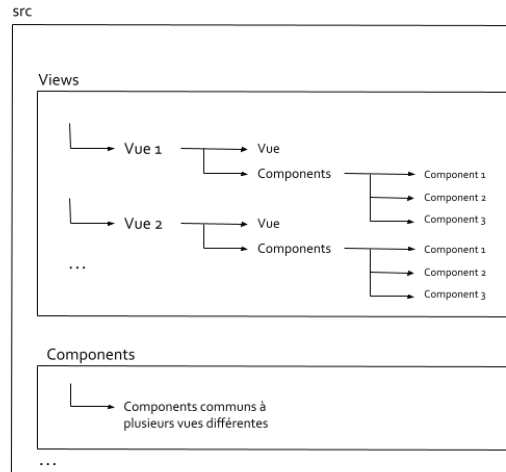


FIGURE 5 : SCHEMA ILLUSTRANT L'ARCHITECTURE LOGICIELLE GENERALE COTE FRONTEND

Comme le montre la Figure 5, dans le dossier « **src** » du frontend nous avons créé un dossier « **view** », qui contient deux dossiers : un pour la vue « **etudiant** » et un autre pour la vue « **admin** » (cf. Figure 6). Dans chacune des deux vues sont présents différents répertoires qui symbolisent chaque page de l'application. Ces répertoires contiennent le composant parent principal et les composants enfants qui les constituent, chacun dans des sous-dossiers. La Figure 6 montre l'architecture du dossier « **destinations** » situé dans la vue « **admin** ». Ce dossier comprend le composant parent **Destinations.vue** ainsi qu'un sous-dossier par composant enfant tel que **FiltreDestinations.vue** contenu dans le sous-dossier « **filtreDestinations** ».

Ensuite, nous avons créé un dossier « **components** » au même niveau que le dossier « **views** ». Celui-ci contient des composants qui sont communs à plusieurs vues tels que le composant **HeaderAdmin.vue** par exemple. À ce niveau, on trouve aussi le dossier « **img** » qui contient les images présentes sur les différentes pages. On trouve également le dossier « **router** » qui contient un fichier javascript contenant les routes d'accès aux différentes pages.

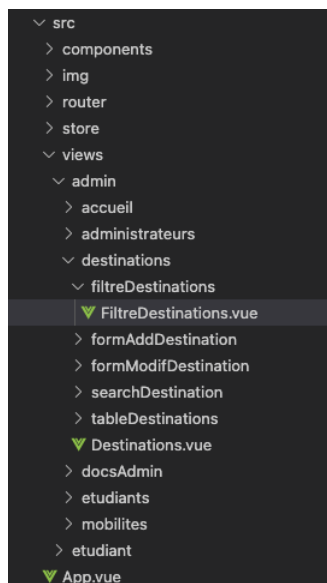


FIGURE 6 : ARCHITECTURE LOGICIELLE FRONTEND DETAILLEE DE LA VUE « DESTINATIONS »

COTE BACKEND

Pour l'architecture logicielle côté backend, les dossiers principaux se trouvent dans le dossier « **src/main/java** » (voir Figure 7). On retrouve dans ce dossier :

- Le dossier des controllers
- Le dossier des entites
- Le dossier des repositories

Au même niveau que le dossier « **java** », on retrouve les dossiers « **test** » ainsi que « **resources** » dans lequel se trouve le fichier SQL qui permet de remplir la base de données.

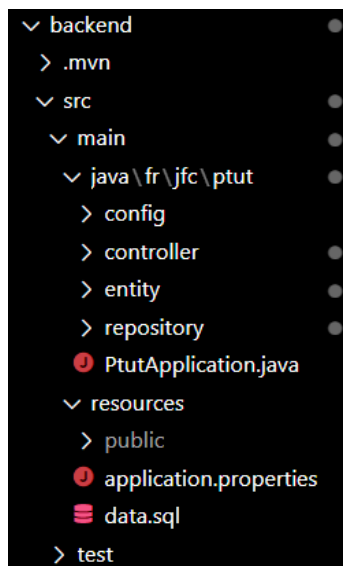


FIGURE 7 : ARCHITECTURE LOGICIELLE BACKEND

MAQUETTES

Pour réaliser les maquettes, nous avons utilisé le logiciel Figma. La figure ci-dessous est un aperçu de l'interface administrateur.

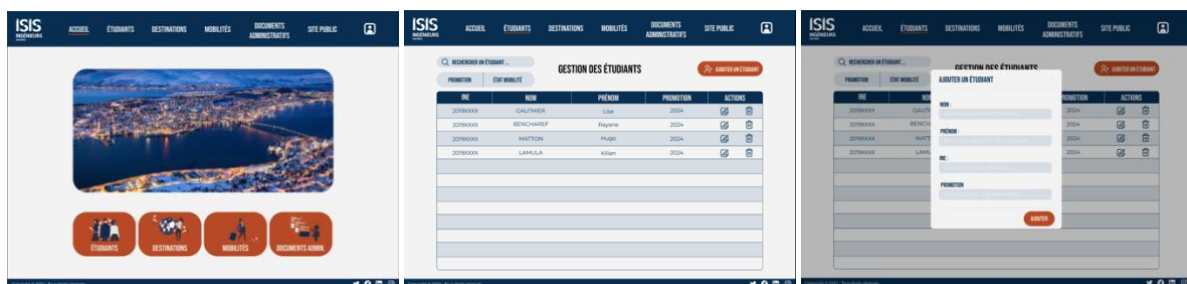


FIGURE 8 : APERÇU DES MAQUETTES DE L'INTERFACE ADMINISTRATEUR

L'intégralité des maquettes est disponible en Annexe 3 : Maquettes et via le lien cité en Bibliographie, Sitographie. Présentées au client, elles ont pour but de vérifier que les besoins du client ont été clairement identifiés et de présenter une première ébauche visuelle de l'application web.

UTILISATION DE L'API REST

Si on le configure correctement, Spring peut auto-générer une API à partir des repositories et des entités existant dans notre code. Cette API autogénérée nous permet de faire les requêtes basiques du CRUD (Create, Read, Update, Delete). Par exemple, dans notre projet nous avons une entité étudiant. L'API générée par Spring nous permet d'accéder à tous les étudiants présents dans la base de données (GET), d'accéder à un étudiant en fonction de son id (GET), d'ajouter un étudiant (POST) de supprimer un étudiant (DELETE) et de modifier un étudiant (UPDATE). Ceci n'est pas une liste exhaustive mais cela montre à quel point Spring nous facilite le travail en générant cette API. En effet, cela nous permet de gagner du temps nous n'avons pas à coder nous-mêmes des controllers « triviaux ». Cette API est visualisable grâce à l'outil Swagger selon la figure suivante.



FIGURE 9 : VISUALISATION DE L'API AUTOGENEREES POUR L'ENTITE « ETUDIANT » SUR SWAGGER

Spring Data permet également d'auto-générer des contrôleurs en ajoutant des méthodes dans les repository. Ces contrôleurs permettent d'effectuer un grand nombre de traitement, du plus simple au plus complexe, comme le montre la Figure 10.

```
public interface DestinationRepository extends JpaRepository<Destination, Integer> {  
  
    List<Destination> findByPays(String pays);  
    List<Destination> findByTypeMobilité(Type type);  
  
    /**  
     *  
     * @param nom entré par l'utilisateur  
     * @param ville entrée par l'utilisateur  
     * @return la liste des destinations dont le nom et/ou la ville correspondent  
     * à la chaîne de caractères entrée par l'utilisateur  
     */  
    List<Destination> findAllByNomEtablissementAccueilContainingIgnoreCaseOrVilleContainingIgnoreCase(String nom, String ville);  
}
```

FIGURE 10 : EXEMPLES DE CONTROLLERS POUVANT ETRE GENERES PAR L'API REST

Les deux premiers contrôleurs permettent de filtrer les destinations par pays ou par type de mobilité. Le troisième contrôleur est un peu plus complexe car il combine plusieurs *Query Methods*. En effet, il permet de lister les destinations dont le nom ou la ville, insensibles à la casse (*IgnoreCase*), contiennent la chaîne de caractères entrée par l'utilisateur (*Containing*). Le résultat sur Swagger est le suivant :



FIGURE 11 : VISUALISATION DES CONTROLLERS GENERES DANS LE REPOSITORY PAR SPRING, SUR SWAGGER

Cependant, pour certains traitements complexes, nous avons été contraints de coder nous-mêmes les controllers répondant à nos besoins. Par exemple, nous avons besoin de savoir quels étudiants avaient validé leurs mobilités et quels étudiants ne l'avaient pas validée. Pour cela nous avons codé le controller présenté en Figure 12).

```
@GetMapping("/findEtudiant/etatMobilité/{etat}")
public Set<Etudiant> getEtudiantByEtatMobilité(@PathVariable String etat){
    Set<Etudiant> res = new HashSet<>();
    List<Etudiant> etudiants = etudiantRepository.findAll();
    //Requete pour étudiants ayant validé mobilité
    if(etat.equals(anObject: "val")){
        for(Etudiant etud : etudiants){
            if(!etud.getMobilites().isEmpty()){
                res.add(etud);
            }
        }
    }
    //Requete pour étudiants n'ayant pas validé mobilité
    if(etat.equals(anObject: "nVal")){
        for(Etudiant etud : etudiants){
            if(etud.getMobilites().isEmpty()){
                res.add(etud);
            }
        }
    }
    return res;
}
```

FIGURE 12 : CONTROLEUR PERMETTANT DE FILTRER LES ETUDIANTS EN FONCTION DE L'ETAT DE LEUR MOBILITE

Nous aurions pu effectuer ce traitement en SQL dans un Repository mais il nous paraissait plus simple de le coder en Java.

Ce qu'il faut retenir c'est que Spring peut grandement simplifier la mission du développeur en auto-générant des API et des controllers, à condition que le modèle UML soit pertinent et cohérent.

GESTION DE PROJET

Pour mener à bien notre projet tuteuré, nous avons utilisé certains outils de gestion de projet que nous présentons dans cette partie. Pour réunir tous les documents liés à la gestion de projet et nous organiser quant à l'exécution des tâches, nous nous sommes servis de l'application Notion (dont le lien est accessible en Bibliographie, Sitographie).

ANALYSE DES RISQUES

Lorsque nous avons pris connaissance du projet, nous avons d'abord analysé les risques. Cette étape est essentielle car elle nous permet d'anticiper les problèmes que nous aurions pu rencontrer. Pour cela, nous nous sommes basés sur une matrice et nous avons listé les risques qui auraient pu freiner l'avancée du projet (cf. Figure 13 et Figure 14).

Gravité	5 – Très élevée					
	4 – Elevée			R4		
	3 – Moyenne		R10	R2		
	2 – Faible		R6	R9	R1, R8	
	1 – Nul			R5	R3	R7
		1 – Nul	2 – Faible	3 – Moyenne	4 – Elevée	5 – Très élevée
		Probabilité d'occurrence				

FIGURE 13 : MATRICE DES RISQUES

Risques	Référence	Score risque
Problème pour relier le back au front	R1	8
Confinement dû au COVID	R2	9
Développeur positif au COVID	R3	4
Diagramme UML erroné	R4	12
Mauvaise répartition des tâches entre les ressources humaines	R5	3
Problèmes de synchronisation GIT	R6	4
Manque d'expérience et de connaissance de certaines technologies	R7	5
Absence de réponse du client	R8	8
Sous-estimation du temps alloué à une ou plusieurs tâches	R9	6
Mauvaise compréhension des besoins du client	R10	6

FIGURE 14 : LISTE DES RISQUES

Il est important de noter que cette liste est non exhaustive bien que nous ayons essayé d'en lister un maximum.

ORGANIGRAMME DES TACHES

Après avoir analysé les risques, nous avons listé les tâches à effectuer pour atteindre les objectifs du projet. Cela nous a permis de savoir exactement sur quelles tâches nous engageons les ressources et le temps mais aussi de nous assurer de la cohérence des tâches au regard des besoins de notre client.

En détaillant les tâches, des plus générales aux plus spécifiques, nous avons obtenu l'organigramme des tâches, aussi appelé Work Breakdown Structure, de la Figure 15 (une version plus lisible est disponible en Annexe 4).



FIGURE 15 : ORGANIGRAMME DES TACHES

REPARTITION DES TACHES

Une fois les tâches clairement identifiées, il faut associer chaque tâche du projet à une ou plusieurs ressources humaines. Pour cela, nous avons utilisé la matrice RACI dont l'acronyme en anglais correspond à **Responsible, Accountable, Consulted, Informed**. En français, nous pourrions traduire respectivement ces termes par **Réalisateur, Approuvateur, Consulté, Informé**. Notre RACI est présenté sur la Figure 16.

Tâche	Hugo	Kilian	Lise	Rayane
A. Analyser le projet				
<i>A.1 Comprendre le projet</i>				
A.1.1 Demander les besoins au client	I	I	AR	I
A.1.2 Définir le périmètre	C	C	AC	R
A.1.3 Rédiger la note de cadrage	I	I	AR	R
A.1.4 Etudier la solution actuelle	R	I	A	I
A.1.5 Rédiger le CDC fonctionnel	R	C	AC	C
A.1.6 Analyse des risques	R	R	A	I
A.1.7 Faire la bête à cornes	R	R	A	I
A.1.8 Faire le diagramme pieuvre	C	R	A	I
<i>A.2 Mettre en place une organisation</i>				
A.2.1 Faire le WBS	R	R	A	I
A.2.2 Faire le RACI	R	R	A	I
A.2.3 Faire le diagramme de GANTT	I	I	AR	R
A.2.4 Choisir le stack technologique	R	R	AR	R

B. Concevoir				
B.1 Maquetter	C	C	AR	C
B.2 Back				
B.2.1 Faire modèle entité-association	R	C	CA	C
B.2.2 Faire le diagramme UML	R	R	CA	R
C. Réaliser				
C.1 Front				
C.1.1 Définir architecture de projet vue	R	C	AR	C
C.1.2 Créer interface statique	R	R	AR	R
C.1.3 Rendre interface dynamique	R	R	AR	R
C.2 Back				
C.2.1 Faire le mapping UML JPA	R	R	CA	C
C.2.2 Créer repositories	C	R	A	R
C.2.3 Créer controllers	C	C	RA	R
C.2.4 Relier API à BDD postgres	R	C	CA	C
C.2.5 Mettre en place authentification	R	R	RA	R
D. Tester				
D.1 Tester repositories	R	R	CA	C
D.2 Tester controller	C	C	RA	R
D.3 Vérifier couverture de tests	I	R	RA	I
D.4 Tester sur plusieurs navigateurs	I	R	A	I
D.5 Tester présence et fonctionnement des fonctionnalités	C	C	CA	R
E. Documenter				
E.1 Rédiger documentation	R	R	RA	R
E.2 Rédiger manuel d'utilisation	R	R	AR	R
F. Présenter et livrer				
F.1 Préparer le support de présentation	R	R	RA	R
F.2 Rédiger rapport technique	R	R	R	R

FIGURE 16 : MATRICE RACI

PLANIFICATION

La planification est une étape clé de la gestion de projet. Il s'agit de planifier en fonction des contraintes et des dépendances entre les tâches. Elle permet d'assurer le suivi du projet et l'adaptation du planning en fonction des retards et des avances.

Ainsi, pour mener à bien notre projet et respecter les délais, il a été nécessaire d'adopter une méthode de planification efficace. Nous avons le choix entre planification et rétro-planification mais nous avons pris la décision de choisir la première option. En effet, nous avons déjà pu expérimenter la planification lors d'un projet JAVA en deuxième année et pour la majorité d'entre nous, cela a bien fonctionné. De plus, selon nous, la rétro-planification implique une bonne connaissance en matière de code et de gestion de projet. Il est aussi nécessaire d'avoir une bonne estimation du temps passé sur chaque tâche et par extension, d'être conscient de nos capacités.

Pour réaliser ce projet, nous devons disposer de 60 heures en autonomie, réparties du 21 février au 15 avril 2022. Nous prévoyions de consacrer $\frac{2}{3}$ de ce temps (soit environ 40 heures) à la réalisation et les tests du projet et les 20 heures restantes à la compréhension du besoin, la mise en place d'une organisation (diagramme de Gantt et choix du stack technologique), la conception des maquettes, de l'UML et du diagramme entité-association, la réalisation de la documentation, la rédaction et le dépôt des livrables.

Plusieurs jalons ont également été planifiés, planifiant ainsi les points clés du projet. Parmi eux nous pouvons citer la validation du cahier des charges et des maquettes par le client, le dépôt des différents livrables, etc.

Pour cela, nous avons utilisé plusieurs outils de planification dont le réseau Critical Path Method (CPM) et des diagrammes de GANTT.

Nous avons d'abord tracé le réseau CPM (disponible en Annexe 5) en nous appuyant sur le RACI et en estimant la durée de chaque tâche. Cet outil nous a notamment permis d'identifier le chemin critique, c'est-à-dire la séquence de tâches de disposant d'aucune marge.

Une fois le chemin critique identifié, nous avons généré 3 diagrammes de GANTT à l'aide du logiciel GANTT Project. Deux d'entre eux sont des diagrammes prévisionnels (l'un en heures de travail : voir Annexe 6, l'autre en jours : voir Annexe 7) tandis que le troisième est le diagramme de GANTT réel (voir Annexe 8), c'est-à-dire celui qui représente la durée réelle des tâches et de leur enchaînement, dans la réalité. Nous avons généré tous ces diagrammes car il nous semblait pertinent de pouvoir comparer le GANTT réel au GANTT prévisionnel. De plus, le GANTT prévisionnel en heures nous permettait de nous situer par rapport aux heures de projet tuteuré restantes dans notre emploi du temps. Le GANTT prévisionnel en jours nous donnait une indication quant à l'avancée du projet vis-à-vis de la date butoir.

Lors de la planification, il y avait donc 2 enjeux : respecter les délais du projet en estimant du mieux possible la durée de chaque tâche, mais aussi allouer correctement les ressources humaines aux tâches afin de ne pas surcharger les membres du groupe. La répartition des tâches est présentée sur les diagrammes de ressources des Annexes Annexe 6, Annexe 7 et Annexe 8.

Dès le début du projet nous étions conscients que le diagramme de GANTT prévisionnel n'allait pas être respecté. Cependant, au cours des semaines nous avons réalisé qu'au lieu des 60 heures normalement prévues pour le projet tuteuré, seulement une quarantaine nous avaient été attribuées. Ainsi, nous avons dû revoir nos priorités en termes de fonctionnalités et nous adapter pour pouvoir rendre le projet à temps. Il s'agit d'un risque que nous aurions dû prévoir : cela nous aurait permis de passer moins de temps sur la gestion de projet et de nous concentrer davantage sur la partie développement web.

PRISE DE RECUL

Ce projet a permis à chacun des membres du groupe de s'enrichir à la fois techniquement et personnellement. En effet, nous avons consolidé nos connaissances en termes de gestion de projet et de développement web. Certains membres du groupe ont même développé de nouvelles compétences en apprenant, de manière autonome, de nouveaux langages/technologies ou en essayant de développer des fonctionnalités comme l'upload d'images et de documents.

Sur le plan personnel, ce projet nous a permis d'acquérir de l'expérience en termes de travail en équipe. Pour certains d'entre nous, il s'agissait du premier projet informatique et nous avons trouvé intéressant le fait de se confronter à la réalité d'un projet.

DIFFICULTES RENCONTREES ET AMELIORATIONS POSSIBLES

Durant le projet, nous nous sommes confrontés à plusieurs difficultés. Un de nos principaux problèmes a été le retard sur le planning initialement prévu. En effet, compte tenu du fait qu'il nous a manqué 20 heures dans notre emploi du temps et que nous avons compté ces 20 heures dans notre planification, nous n'avons pas pu développer l'intégralité des fonctionnalités annoncées au client. De plus, nous avons considérablement sous-estimé le temps passé à déboguer les erreurs de code.

Durant le projet nous avons rencontré des problèmes de tous type mais à ce jour, tous ont été résolus. Cela a été possible grâce à des forums sur Internet tels que StackOverflow, mais aussi grâce à M. BASTIDE et à nos camarades de promotion qui ont pu nous aider lorsque nous rencontrions des erreurs qu'eux-mêmes avaient réussi à résoudre.

Une des consignes de notre projet tuteuré était de mettre en œuvre un « back-end » commun avec les autres projets tuteurés liés à la gestion administrative et pédagogique d'ISIS. Malheureusement, par manque de temps, nous n'avons pas pu construire ce back-end commun avec nos camarades. Cependant, il pourrait être intéressant que les promotions futures reprennent nos projets tuteurés et essayent de les adapter de façon à ce qu'ils soient reliés à un back-end commun.

À l'heure actuelle, notre solution est fonctionnelle mais des améliorations sont encore possibles. Elles concernent notamment des fonctionnalités auxquelles nous avons pensé mais que nous n'avons pas eu le temps de développer. Les améliorations possibles, selon nous, sont les suivantes :

- L'administrateur pourrait modifier une destination en indiquant les étudiants qui y ont effectué une mobilité (s'ils ont donné leur accord au préalable) et en ajoutant le lien vers le site de l'établissement, vers le département informatique de l'établissement et vers le syllabus
- Lorsqu'ils consultent les détails d'une destination, les étudiants auraient accès aux noms des étudiants ayant effectué une mobilité à cette destination ainsi qu'aux liens cités précédemment
- Les filtres pourraient être combinés
- Lorsqu'un contrat avec un établissement est expiré, la destination resterait référencée mais ne serait plus accessible
- Possibilité d'uploader un fichier du côté étudiant
- Possibilité de filtrer les destinations ERASMUS ou NON ERASMUS côté étudiants
- Ajouter des fonctionnalités administratives telles que le suivi d'obtention de VISA, validation du TOEIC, validation des crédits à l'étranger, documents administratifs à jour ...
- Ajouter une fonctionnalité pour la prise en compte d'une mobilité internationale validée hors ISIS

PROLONGEMENTS A DONNER AU PROJET

L'avantage de notre projet est qu'il est utile et qu'il peut faciliter le travail des responsables des relations internationales. Il pourrait d'ailleurs être envisageable, dans le futur, de rendre notre application disponible à l'ensemble des campus de l'INUC. De plus, actuellement notre solution est déjà fonctionnelle mais comme nous l'avons dit précédemment, de nombreuses améliorations peuvent y être apportées. Pour cela, nous pourrions imaginer que notre projet soit repris par d'autres promotions d'ISIS dans les années à venir.

CONCLUSION

Un projet tuteuré est un projet conséquent qui implique un réel travail en équipe et une mise à contribution des compétences et connaissances de chacun des membres du groupe. En s'inscrivant dans le cursus de première année du cycle ingénieur ISIS, ce premier projet tuteuré nous permet d'acquérir une expérience complète en termes de gestion et de conduite de projet.

Ainsi, à travers des phases de réflexion, veille, conception, modélisation et création, nous avons pu apprendre à travailler en équipe, tant au point de vue de l'ambiance de travail que dans la répartition des tâches. Ayant déjà travaillé ensemble, l'esprit d'équipe et la cohésion se sont rapidement installés, ce qui nous a permis de mener à bien notre projet.

Nous avons également pris conscience de l'importance de la communication dans un tel projet. En effet, pour cerner correctement les besoins de notre client, nous avons dû poser les bonnes questions, ce qui a contribué au développement des qualités de communication de chacun. Il est aussi important de souligner l'importance de la communication entre les membres du groupe car sans celle-ci, le projet aurait pu ne pas aboutir. En ce sens, le chef de projet a réellement pu expérimenter les défis auxquels il devra faire face dans sa vie professionnelle. En effet, celui-ci a dû veiller à la fois au bon déroulement du projet en se référant à la planification initiale mais aussi à la production des documents nécessaires à la rédaction de ce rapport. Il a également échangé avec le client et le tuteur école afin de les tenir au courant de l'avancée du projet. Par ailleurs, l'ensemble de notre groupe de travail tient à remercier notre tuteur école, M. Bastide, pour sa disponibilité et sa réactivité lorsque nous l'avons sollicité par mail pour qu'il nous vienne en aide.

De plus, le respect des délais a été, sans aucun doute, le plus grand défi de ce projet puisqu'il a fallu respecter au mieux le planning pour fournir un projet abouti en temps voulu.

Enfin, même si l'intérêt premier d'un projet tuteuré est de mettre à contribution nos connaissances et d'en acquérir de nouvelles, nous nous devons de livrer une application web fonctionnelle répondant au cahier des charges. Pour faciliter la prise en main de notre application web par les utilisateurs, nous avons pour projet de rédiger un manuel d'utilisation illustré, détaillant les manipulations à effectuer pour utiliser les fonctionnalités proposées par cette dernière.

Aujourd'hui, nous sommes fiers du résultat obtenu et nous espérons sincèrement que ce projet pourra être repris et amélioré par les promotions futures. En effet, une fois déployée, nous sommes convaincus que cette application sera utile et facilitera le travail des administrateurs ainsi que l'accès à l'information pour les étudiants.

BIBLIOGRAPHIE, SITOGRAPHIE

- Cours de gestion de projet de M. PINGAUD
- Cours de gestion de projet informatique de M. RAWAT
- Lien du dépôt GITHUB contenant l'ensemble des documents présentés dans ce rapport, le cahier des charges ainsi que le code : <https://github.com/lisegt/GestionMobilites.git>
- Lien Notion : <https://www.notion.so/Projet-tutor-04c53b61cdeb4a5b88727f5602d7a1a5>
- Lien Figma vers les maquettes : <https://www.figma.com/file/V612DwojKa7DlfDPwl6vaf/Maquette-Projet-Tutor%C3%A9?node-id=2%3A2>

TABLE DES ILLUSTRATIONS

Figure 1 : Diagramme bête à cornes	3
Figure 2 : Diagramme pieuvre	4
Figure 3 : Choix du stack technologique	5
Figure 4 : entité "etudiant" avec les annotations hibernate	5
Figure 5 : Schéma illustrant l'architecture logicielle générale côté frontend	6
Figure 6 : Architecture logicielle frontend détaillée de la vue « destinations »	6
Figure 7 : Architecture logicielle backend	7
Figure 8 : Aperçu des Maquettes de l'interface administrateur	7
Figure 9 : Visualisation de l'API autogénérée pour l'entité « Etudiant » sur Swagger	8
Figure 10 : Exemples de controllers pouvant être générés par l'API Rest	8
Figure 11 : Visualisation des controllers générés dans le repository par Spring, sur Swagger	8
Figure 12 : Contrôleur permettant de filtrer les étudiants en fonction de l'état de leur mobilité	9
Figure 13 : Matrice des risques	10
Figure 14 : Liste des risques	10
Figure 15 : Organigramme des tâches	11
Figure 16 : Matrice RACI	12

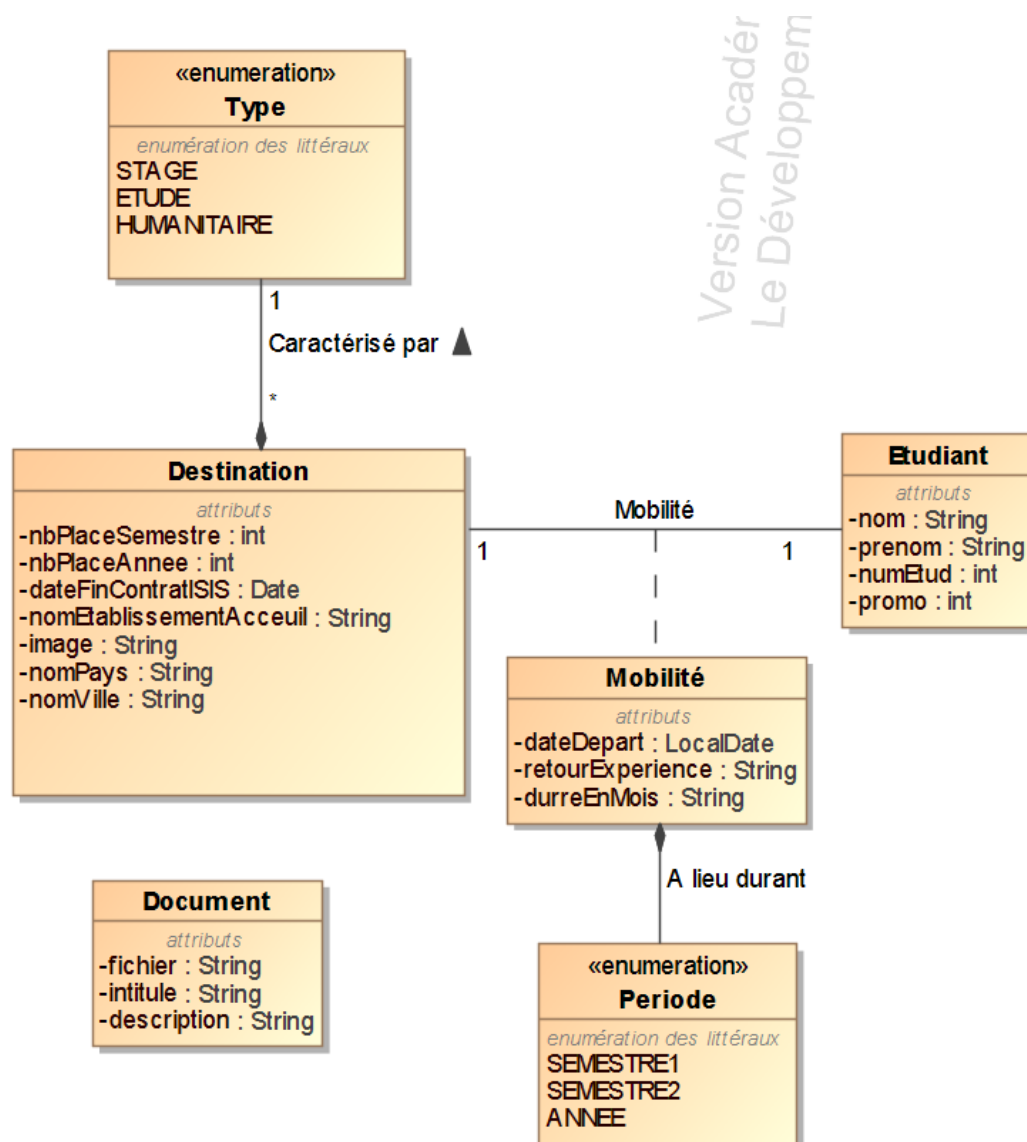
ANNEXES

Annexe 1 : Liste des fonctionnalités	1
Annexe 2 : Diagramme UML	2
Annexe 3 : Maquettes	3
Annexe 4 : Organigramme des tâches	6
Annexe 5 : Réseau CPM	7
Annexe 6 : Diagramme de GANTT prévisionnel en heures	8
Annexe 7 : Diagramme de GANTT prévisionnel en jours	9
Annexe 8 : Diagramme de GANTT réel	10

ANNEXE 1 : LISTE DES FONCTIONNALITES

Interface administrateur			
Gestion des étudiants	Afficher tous les étudiants	INE, nom, prénom, promotion	<u>Filtres :</u> <ul style="list-style-type: none"> Recherche nom/prénom Promotion Etat de la mobilité
	Ajouter un étudiant	INE, nom, prénom et promotion de l'étudiant	
	Modifier le profil d'un étudiant	INE, nom, prénom et promotion de l'étudiant	
	Supprimer un étudiant		
Gestion des destinations	Afficher toutes les destinations	Nom de l'établissement, ville, pays, type de mobilité, nombre de places par semestre/an, statut du contrat	<u>Filtres :</u> <ul style="list-style-type: none"> Recherche nom de l'établissement ou ville Pays Type de mobilité Statut du contrat
	Ajouter une destination	Nom de l'établissement, ville, pays, type de mobilité, nombre de places par semestre/an, date de fin de contrat, illustration (facultative)	
	Modifier une destination	Nom de l'établissement, ville, pays, type de mobilité, nombre de places par semestre/an, date de fin de contrat, illustration (facultative)	
	Supprimer une destination		
Gestion des mobilités	Afficher toutes les mobilités	Nom, prénom, promotion de l'étudiant, type de mobilité, destination, date de départ, période, durée, état de la mobilité	<u>Filtres :</u> <ul style="list-style-type: none"> Recherche étudiant Promotion Destination Etat de la mobilité
	Ajouter une mobilité	Promotion, étudiant, type de mobilité, destination, période, durée en mois, date de départ	
	Ajouter un retour d'expérience		
	Supprimer une mobilité		
Gestion des documents administratifs	Afficher tous les documents administratifs	Nom du document, description	
	Ajouter un document	Nom du document, description, fichier PDF	
	Modifier un document	Nom du document, description, fichier PDF	
	Supprimer un document		
Gestion des administrateurs	Afficher les administrateurs	Identifiant	
	Ajouter un administrateur	Identifiant, mot de passe	
	Supprimer un administrateur		
Authentification	Se connecter en tant qu'administrateur	identifiant, mot de passe	

Interface étudiant			
Accès aux destinations	Afficher les destinations	Nom de l'entité d'accueil, ville, pays	<u>Filtres :</u> <ul style="list-style-type: none"> Pays Type de mobilité Toutes
	Afficher les détails d'une destination	Type de mobilité, nombre de places disponibles, retour(s) d'expérience	
Accès aux documents administratifs	Afficher les documents administratifs	Nom du document	
	Afficher les détails d'un document	Description, lien de téléchargement	



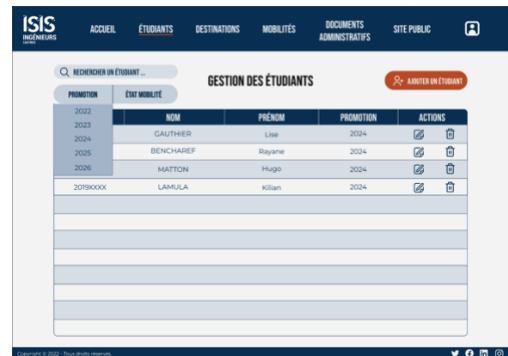
Interface administrateur

Accueil

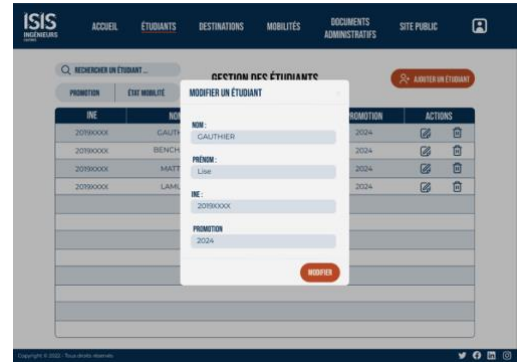
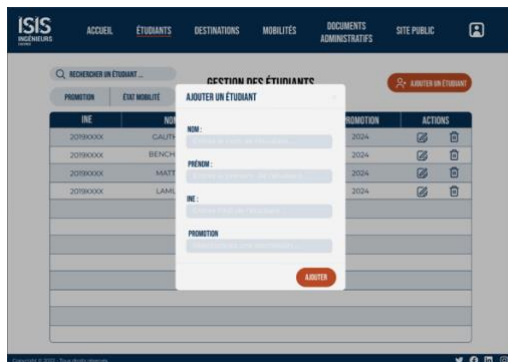


Gestion des étudiants : Formulaire d'ajout d'un étudiant

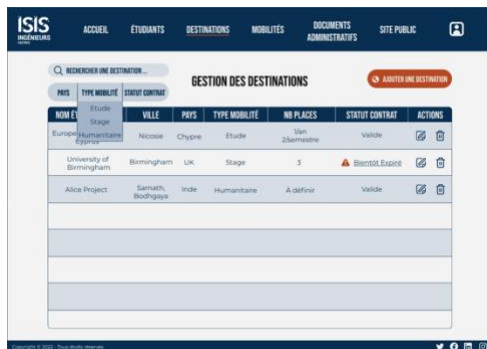
Gestion des étudiants



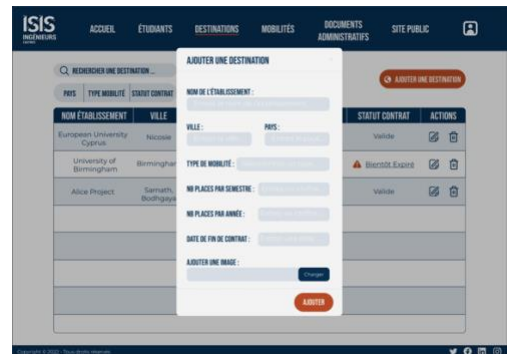
Gestion des étudiants : Formulaire de modification d'un étudiant



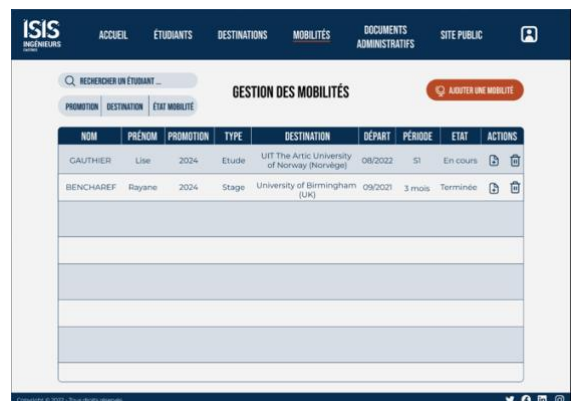
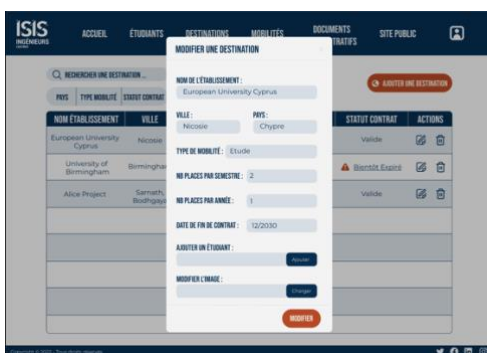
Gestion des destinations



Gestion des destinations :
Formulaire de modification d'une destination



Gestion des mobilités



Gestion des mobilités : Formulaire d'ajout d'une mobilité

The screenshot shows the 'Ajouter une mobilité' form overlaid on the 'Gestion des mobilités' page. The form includes fields for 'PROMOTION', 'ÉTUDIANT', 'DESTINATION', 'DÉPART (MOIS/ANNEE)', and 'PÉRIODE'. An 'AJOUTER' button is at the bottom right of the form. The background table lists students like GAUTHIER and BENCHAREF with their mobility details.

Gestion des mobilités : Formulaire d'ajout d'un retour d'expérience

The screenshot shows the 'Ajouter un retour d'expérience' form overlaid on the 'Gestion des mobilités' page. The form has a single text area for the experience and an 'AJOUTER' button. The background table is the same as the previous screenshot.

Gestion des documents administratifs

The screenshot shows the 'Gestion des documents administratifs' page. It features a table with columns 'NOM DU DOCUMENT', 'DESCRIPTION', and 'ACTIONS'. The first row shows 'Dossier de candidature 2022/2023' with a detailed description and a link to the application page. An 'AJOUTER UN DOCUMENT' button is in the top right.

Gestion des documents administratifs :
Formulaire d'ajout d'un document administratif

The screenshot shows the 'Ajouter un document' form overlaid on the 'Gestion des documents administratifs' page. The form has fields for 'NOM DU DOCUMENT' and 'DESCRIPTION', and an 'AJOUTER' button. The background table is the same as the previous screenshot.

Gestion des documents administratifs :
Formulaire de modification d'un document administratif

The screenshot shows the 'Modifier le document' form overlaid on the 'Gestion des documents administratifs' page. The form has fields for 'NOM DU DOCUMENT', 'DESCRIPTION', and an 'AJOUTER LE DOCUMENT' button. The background table is the same as the previous screenshot.

Gestion des administrateurs

The screenshot shows the 'Gestion des administrateurs' page. It features a table with columns 'IDENTIFIANT' and 'ACTIONS'. The first row shows 'adresse.admin@univ-jfc.fr'. An 'AJOUTER UN ADMIN' button is in the top right.

Gestion des administrateurs :
Formulaire d'ajout d'un administrateur

The screenshot shows the 'Ajouter un administrateur' form overlaid on the 'Gestion des administrateurs' page. The form has fields for 'IDENTIFIANT' and 'MOT DE PASSE', and an 'AJOUTER' button. The background table is the same as the previous screenshot.

Interface étudiant

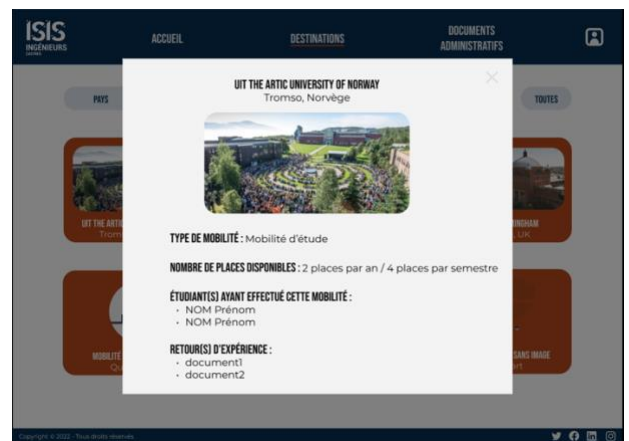
Accueil



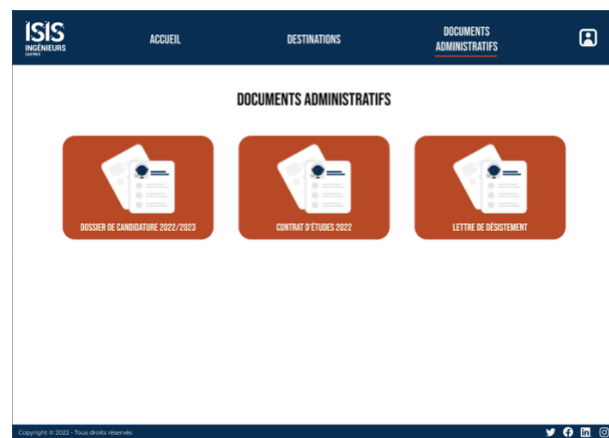
Destinations



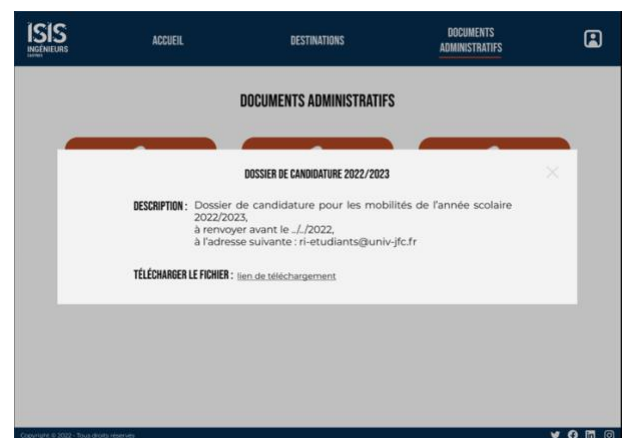
Destinations : Détails d'une destination



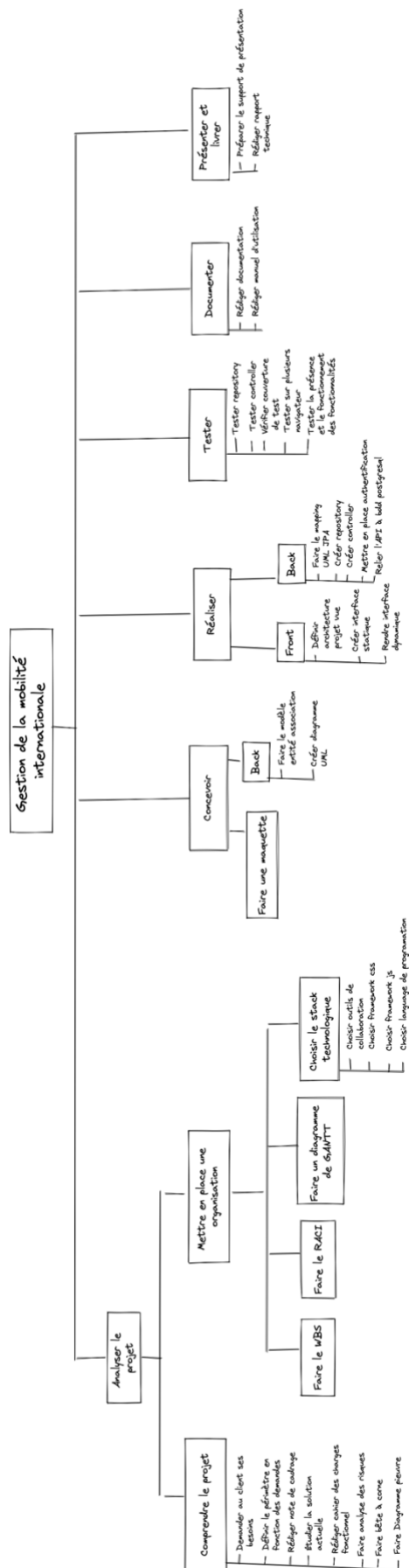
Documents administratifs



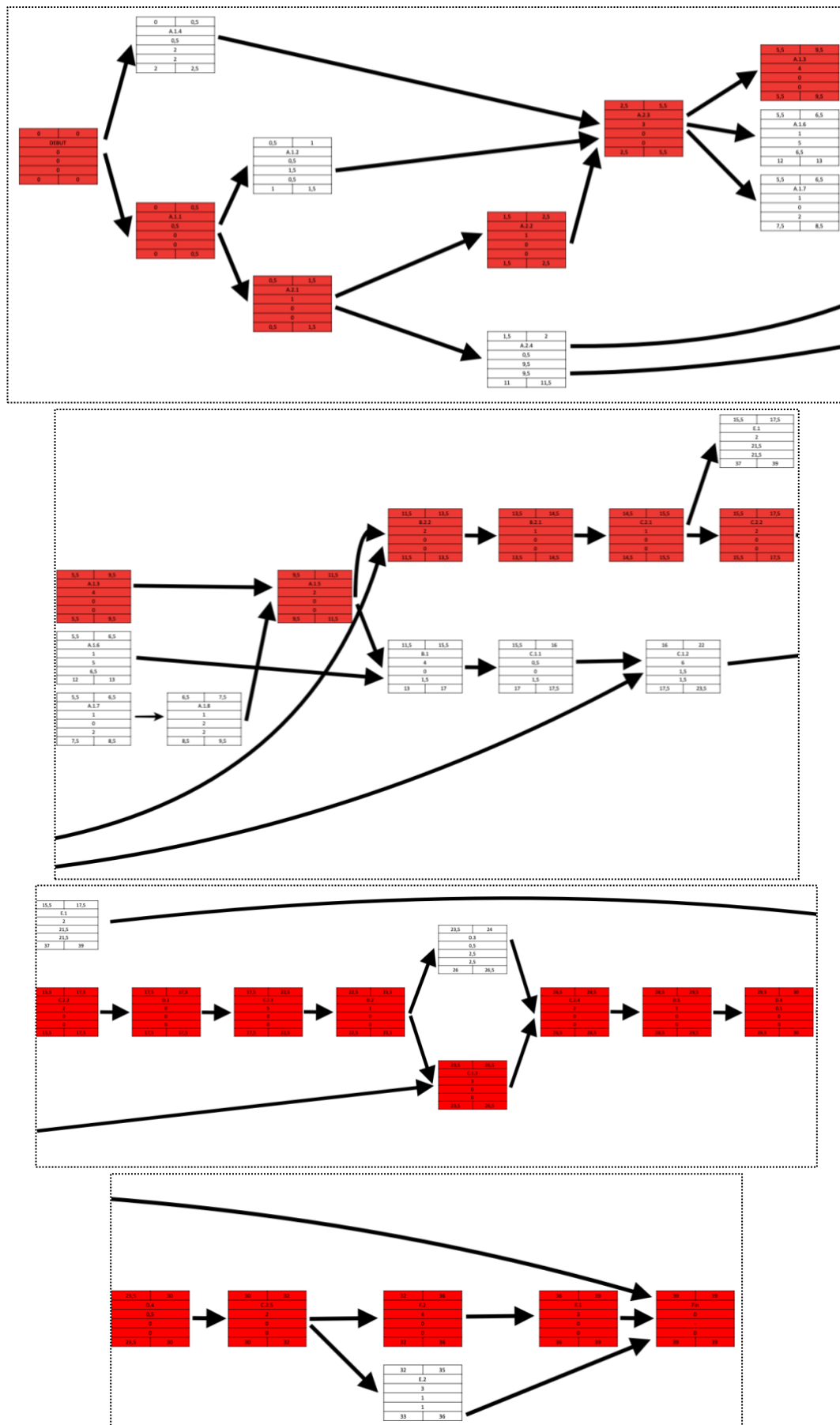
Documents administratifs : Détails d'un document administratif



ANNEXE 4 : ORGANIGRAMME DES TACHES



ANNEXE 5 : RESEAU CPM



GANTT Project

Gantt Project

01'22 02'22

03'22

		Semaine 6 01/02/2022	Semaine 7 08/02/2022	Semaine 8 15/02/2022	Semaine 9 22/02/2022	Semaine 10 01/03/2022	Semaine 11 08/03/2022	Semaine 12 15/03/2022
✓	Gestion des mobilités	01/02/2022	13/03/2022					
○	Début	01/02/2022	01/02/2022					
✓	A. Analyser le projet	01/02/2022	12/02/2022					
✓	FIN	01/02/2022	12/02/2022					
○	A.1.1 Demander les besoin...	01/02/2022	01/02/2022					
○	A.1.2 Définir le périmètre	02/02/2022	02/02/2022					
○	A.1.3 Rédiger la note de c...	07/02/2022	10/02/2022					
○	A.1.4 Etudier la solution a...	01/02/2022	01/02/2022					
○	A.1.5 Rédiger le CDC fonc...	11/02/2022	12/02/2022					
○	A.1.6 Analyse des risques	07/02/2022	07/02/2022					
○	A.1.7 Faire la bête à cornes	07/02/2022	07/02/2022					
○	A.1.8 Faire le diagramme ...	08/02/2022	08/02/2022					
✓	A.2 Mettre en place une orga...	02/02/2022	06/02/2022					
○	A.2.1 Faire le WBS	02/02/2022	02/02/2022					
○	A.2.2 Faire le RACI	03/02/2022	03/02/2022					
○	A.2.3 Faire le diagramme ...	04/02/2022	06/02/2022					
○	A.2.4 Choisir le stack tech...	03/02/2022	03/02/2022					
✓	B. Concevoir	13/02/2022	16/02/2022					
○	B.1 Maquetter	13/02/2022	16/02/2022					
✓	B.2 Back	13/02/2022	15/02/2022					
○	B.2.1 Faire modèle entité-...	15/02/2022	15/02/2022					
○	B.2.2 Faire le diagramme ...	13/02/2022	14/02/2022					
✓	C. Réaliser	16/02/2022	06/03/2022					
✓	C.1 Front	17/02/2022	28/02/2022					
○	C.1.1 Définir architecture ...	17/02/2022	17/02/2022					
○	C.1.2 Créer interface stat...	18/02/2022	23/02/2022					
○	C.1.3 Rendre interface dy...	26/02/2022	28/02/2022					
✓	C.2 Back	16/02/2022	06/03/2022					
○	C.2.1 Faire le mapping U...	16/02/2022	16/02/2022					
○	C.2.2 Créer repositories	17/02/2022	18/02/2022					
○	C.2.3 Créer controllers	20/02/2022	24/02/2022					
○	C.2.4 Relier API à BDD po...	01/03/2022	02/03/2022					
○	C.2.5 Mettre en place aut...	05/03/2022	06/03/2022					
✓	D. Tester	19/02/2022	04/03/2022					
○	D.1 Tester repositories	19/02/2022	19/02/2022					
○	D.2 Tester controller	25/02/2022	25/02/2022					
○	D.3 Vérifier couverture de tests	26/02/2022	26/02/2022					
○	D.4 Tester sur plusieurs navig...	04/03/2022	04/03/2022					
○	D.5 Tester présence et foncti...	03/03/2022	03/03/2022					
✓	E. Documenter	17/02/2022	09/03/2022					
○	E.1 Rédiger documentation	17/02/2022	18/02/2022					
○	E.2 Rédiger manuel d'utilisat...	07/03/2022	09/03/2022					
✓	F. Présenter et livrer	07/03/2022	13/03/2022					
○	F.1 Préparer le support de pr...							

GANTT project		01/22 02/22												03/22												Rendre le matériel électrique											
		Semaine 6				Semaine 7				Semaine 8				Semaine 9				Semaine 10				Semaine 11				Semaine 12											
		06/03/2020				13/03/2020				20/03/2020				27/03/2020				03/04/2020				10/04/2020															
>	Hugo MATTON	Développeur	SOK 75%								33%				SOK 75% SOK 75% 25%				25%				25%				SOK 25%										
>	Kilian LAMULA	Développeur	SOK 75%								33%				SOK 75% 75% 25%				SOK 25%								25%										
>	Rayane BENCHAREF	Développeur	25% SOK				SOK				33%				75% 25% 75%				SOK SOK 25%				25%				SOK 25%										
>	Lise GAUTHIER	Chef de projet	SOK SOK				SOK				75% SOK 25% 75%				SOK SOK 25%				25%				25%				SOK 25%										

ANNEXE 7 : DIAGRAMME DE GANTT PREVISIONNEL EN JOURS

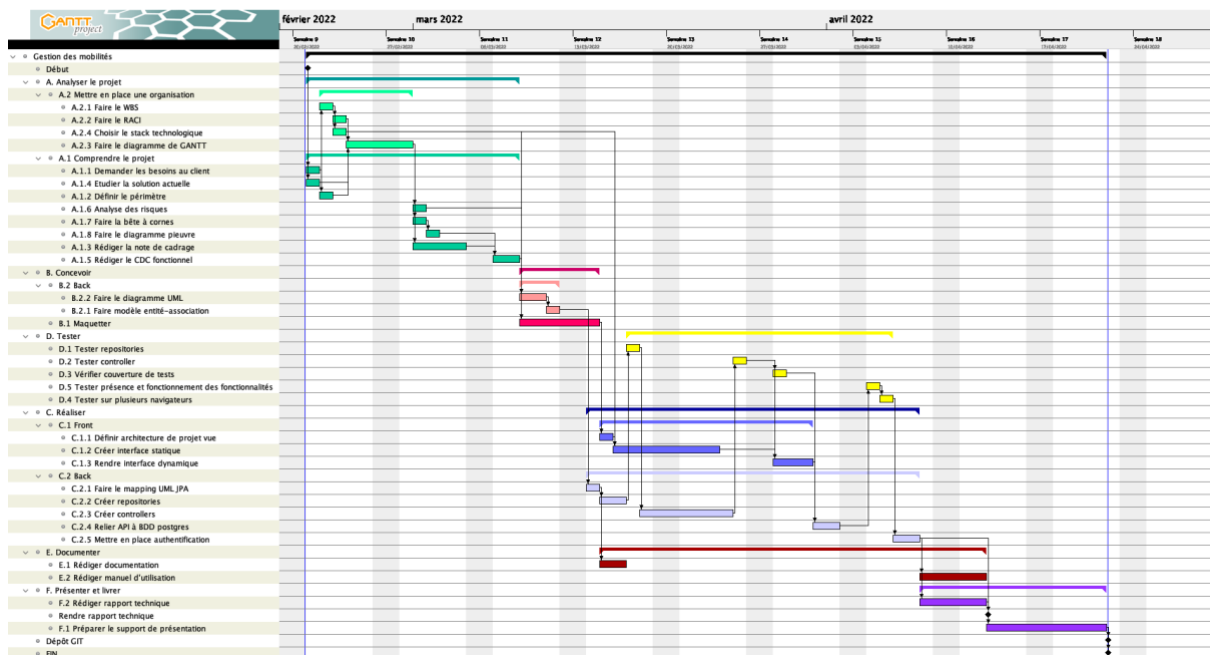


Diagramme de tâches

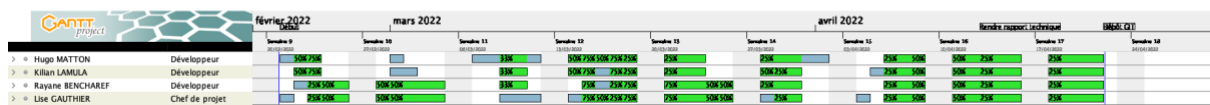


Diagramme de ressources

ANNEXE 8 : DIAGRAMME DE GANTT REEL

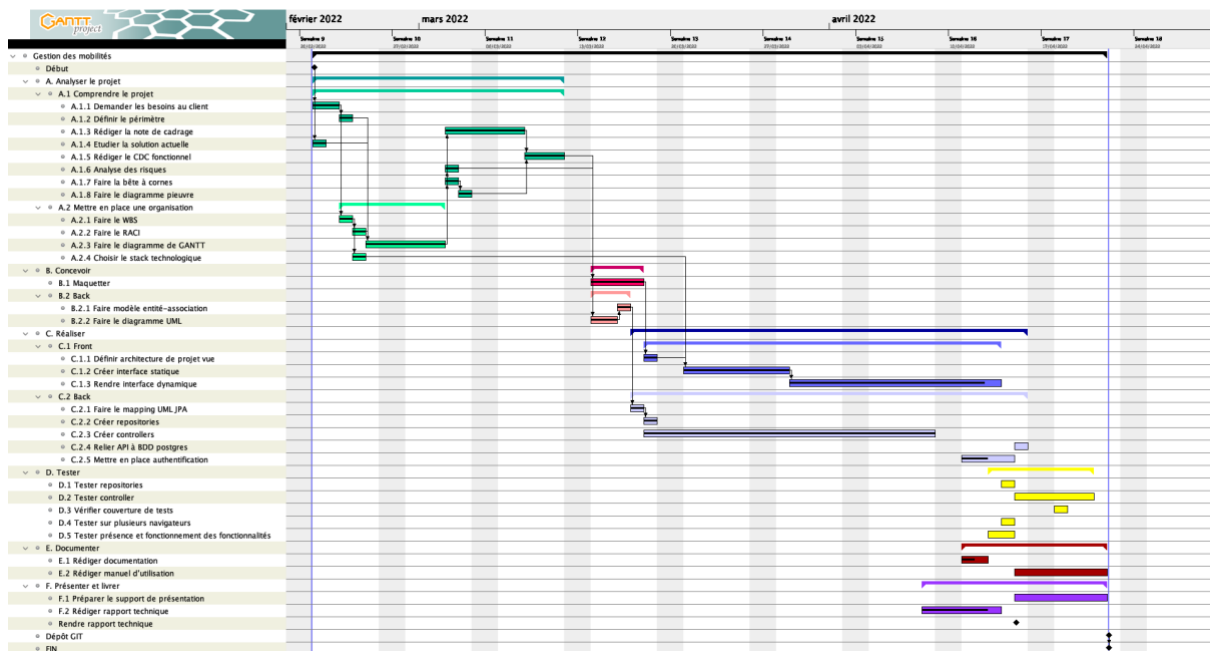


Diagramme de tâches

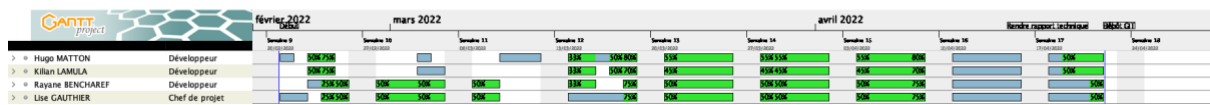


Diagramme de ressources