

# Rapport de projet tutoré

## Gestion des partenariats

Tuteur : M. Bastide

Client : ISIS - Mme Fleuranceau

Cheffe de Projet : Mattone Elsa

Développeuse principale : Rampnoux Maëlle

## I. Résumé

Dans le cadre de notre première année de cycle ingénieur, nous devons réaliser un projet tutoré dont notre sujet fut la gestion des partenariats de l'école.

Nous avons, pour ce faire, un délai de deux mois avant le rendu du rapport, et une semaine supplémentaire avant la soutenance.

Dès que nous avons su quel était notre sujet, nous nous sommes penchées sur la partie gestion de projet, afin d'avoir bien cadré le sujet, ce que nous allions faire, et comment nous allions le faire. Cette partie nous a vraiment pris de nombreuses heures puisque nous voulions être sûres de l'avoir faite correctement, et de n'avoir rien oublié.

Une fois la gestion de projet faite et validée, nous avons pu nous attaquer au développement de l'application : architecture de la base de données, codage et remplissage de celle-ci, requêtes SQL, code HTML, CSS et Javascript.

## II. Sommaire

<b>Rapport de projet tutoré</b>	<b>1</b>
Résumé	2
Sommaire	3
Présentation du projet	4
Gestion de projet	5
Expression du besoin	5
Note de cadrage	5
Changement de tuteur de projet	5
Analyse de faisabilité	5
Analyse des risques	5
Maquette	6
Conception du projet	7
Cahier des charges fonctionnel	7
WBS	7
Planification et Diagramme de Gantt	7
RACI	8
Architecture de la Base de Données	9
Application Web	11
Backend	11
Classes-entités (Entity)	11
Repository	11
Controller	11
Frontend	12
Code HTML et Mustache	12
Code CSS	12
Conclusion	13
Annexes	14
Bibliographie	18

### III. Présentation du projet

Le sujet de notre projet tutoré de première année de cycle ingénieur nous demandait de travailler sur le développement d'une application Web de gestion des relations extérieures pour notre client : l'école ISIS.

Ils désirent que nous développions une base de données complète ainsi qu'une interface utilisateur qui permette de faciliter le contact avec les différents partenaires.

Il nous est donc demandé dans ce projet de mettre en place notre propre logiciel de relation clients afin de gérer la base de données des partenaires institutionnels d'ISIS, avec les coordonnées de ces partenaires, leurs adresses, les principaux contacts dans les entreprises partenaires, et l'historique de toutes les coopérations avec celles-ci (stages, alternances, etc...).

Il faudrait, de plus, garder une trace de toutes les modifications apportées à ces informations, ainsi que de l'identité des personnes effectuant ces modifications.

Il faudrait que notre application soit intuitive et qu'elle possède une fonction de tri dépendant de diverses données importantes à notre cliente, comme par exemple le nombre de stages effectués, ou le nombre d'étudiants pris en alternance.

Il nous a aussi été demandé, par le responsable du projet, de permettre aux utilisateurs de laisser des commentaires libres sur les contacts ou les partenaires, ainsi que de réaliser un système d'authentification afin de donner diverses autorisations selon le rôle de l'utilisateur (administrateur, élève, personnel).

Il s'agit donc d'un projet vraiment complet qui nous demande de mettre en œuvre tout ce que nous avons appris cette année, autant en gestion de projet qu'en Base de Données, UML, Programmation Orientée Objet et Technologies Web.

## IV. Gestion de projet

### A. Expression du besoin

Dans le cadre de ce projet, nous avons assez rapidement demandé un rendez-vous avec notre cliente et notre premier tuteur de projet, Mme Fleuranceau et M. Lamine. Nous leur avons posé quelques questions, notamment concernant les attendus de ce travail et ce qu'ils voulaient précisément dans notre application.

#### 1. Note de cadrage

Suite à ce premier rendez-vous, nous en avons eu un second avec notre cliente, à la suite duquel nous avons des idées assez claires concernant ce que nous devons faire. Nous avons alors immédiatement rédigé une courte note de cadrage. Dans celle-ci, nous avons précisé le périmètre et les attendus de ce projet, les délais et les autres contraintes. Nous y avons ensuite détaillé ce que nous pensions faire pour répondre à ce besoin, avec les moyens que nous possédions.

La rédaction de cette note de cadrage nous a permis de rendre nos idées et les limites de nos capacités dans ce projet plus claires. Nous savions déjà où se trouveraient les difficultés, et avons ainsi pu les anticiper.

#### 2. Changement de tuteur de projet

Suite au changement de notre tuteur de projet, qui est désormais M. Bastide, nous avons organisé une seconde réunion avec lui, afin de connaître ses attendus d'un point de vue technique. Nous sommes restées en contact avec lui assez souvent afin de nous assurer que ce que nous faisons restait dans le cadre de ses attentes.

### B. Analyse de faisabilité

#### 1. Analyse des risques

L'analyse des risques permet en amont de la réalisation du projet d'évaluer les éventuels risques et ainsi d'y pallier. Pour les évaluer nous avons choisi d'utiliser la matrice de risques. La matrice établit l'importance d'un risque en fonction de sa probabilité et de sa gravité.

Catastrophique	R2 R3 R4	R13			
Majeure	R10	R12	R5 R14		R6 R7
Très grave			R11	R8 R15	
Significative			R9	R1	
Mineure					
	Incroyable	Peu probable	Occasionnelle	Probable	Presque certaine

Figure 1 : Matrice des risques (cf. annexe)

Sur la matrice ci-dessus la gravité est exprimée en ordonnée, de mineure à catastrophique, la probabilité allant de incroyable à presque certaine. Enfin, le risque est représenté par les cases colorées: Acceptable à Non acceptable, respectivement de vert à rouge.

Nous avons établi 15 risques que nous avons représenté dans la matrice par des indices de R1 à R15.

Pour réaliser la matrice nous nous sommes appuyés sur le registre des risques ci-dessous (cf annexe). Dans ce registre, un risque est défini par la nature du risque. Il peut être technique, humain ou porté sur le délai.

Indice	Nature du risque	Description
R1	Technique	Mauvais choix des starters Spring
R2	Humain	Une des ressources est dans l'impossibilité de continuer le projet
R3	Technique	L'ordinateur d'une des ressources est inutilisable
R4	Technique	L'application est hackée
R5	Technique	Base de données n'est pas fonctionnel
R6	Technique	L'application ne s'ouvre pas dans le navigateur
R7	Technique	Les tests ne passent pas
R8	Technique	Les tests ne sont pas approprié
R9	Humain	Un désaccord apparaît entre les ressources
R10	Humain	Un conflit surgit entre les ressources
R11	Technique	Méconnaissance des langages et/ou logiciels employés
R12	Délai	Ne pas finir à temps le code
R13	Délai	Ne pas rendre à temps le rapport
R14	Technique	Le frontend et le backend ne peuvent pas fonctionner ensemble
R15	Délai	Rallonger la durée d'une tâche

Figure 2 : Registre des risques (cf. annexe)

## 2. Maquette

Une fois que nous avons déterminé les principaux attendus de ce projet, nous nous sommes penchées sur notre visualisation du produit fini. Cela a abouti à une maquette de l'application web (cf. annexe), à partir de laquelle nous avons codé le front-end.

## C. Conception du projet

### 1. Cahier des charges fonctionnel

Une fois l'analyse des risques faite, nous avons repris la note de cadrage et les comptes-rendus de réunions que nous avons eu avec notre cliente afin de nous atteler à la rédaction complète du cahier des charges fonctionnel.

Il s'agit de l'un des documents les plus importants de la gestion de projet qui définit clairement toutes les attentes et le travail à fournir. Il précise aussi la forme que doivent prendre les livrables et les contraintes à respecter.

### 2. WBS

Pour pouvoir correctement répondre au sujet et nous répartir le travail équitablement et selon nos compétences, nous avons dû définir des tâches et leur organisation. Pour cela, nous avons réalisé grâce au site internet Excalidraw un organigramme des tâches du projet, ou WBS (Work Breakdown Structure).

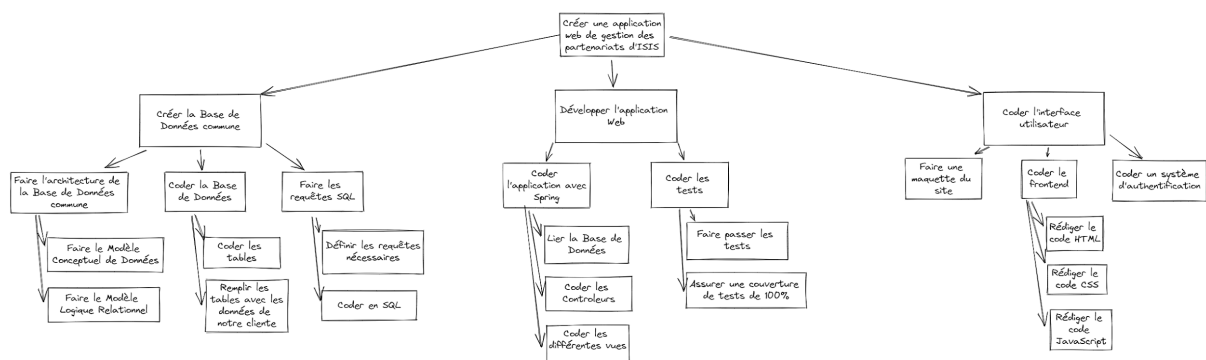


Figure 3 : WBS du projet (cf. annexe)

### 3. Planification et Diagramme de Gantt

Une fois que nous avons défini les tâches et l'ordre dans lequel elles devraient s'effectuer, il nous a fallu évaluer leur durée afin de faire la planification, et la rétroplanification, du projet.

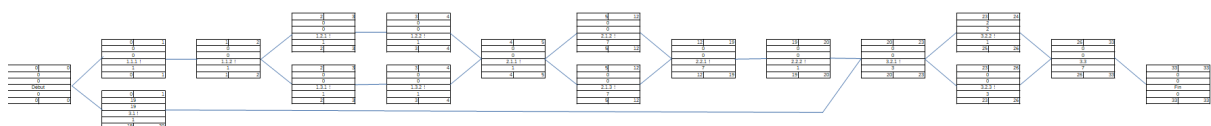


Figure 4 : Planification et Rétroplanification (cf.annexe)

Une fois que cela fut fait, nous avons toutes les données nécessaires pour réaliser le diagramme de Gantt équivalent.

Il ne s'agit cependant que d'un diagramme prévisionnel.

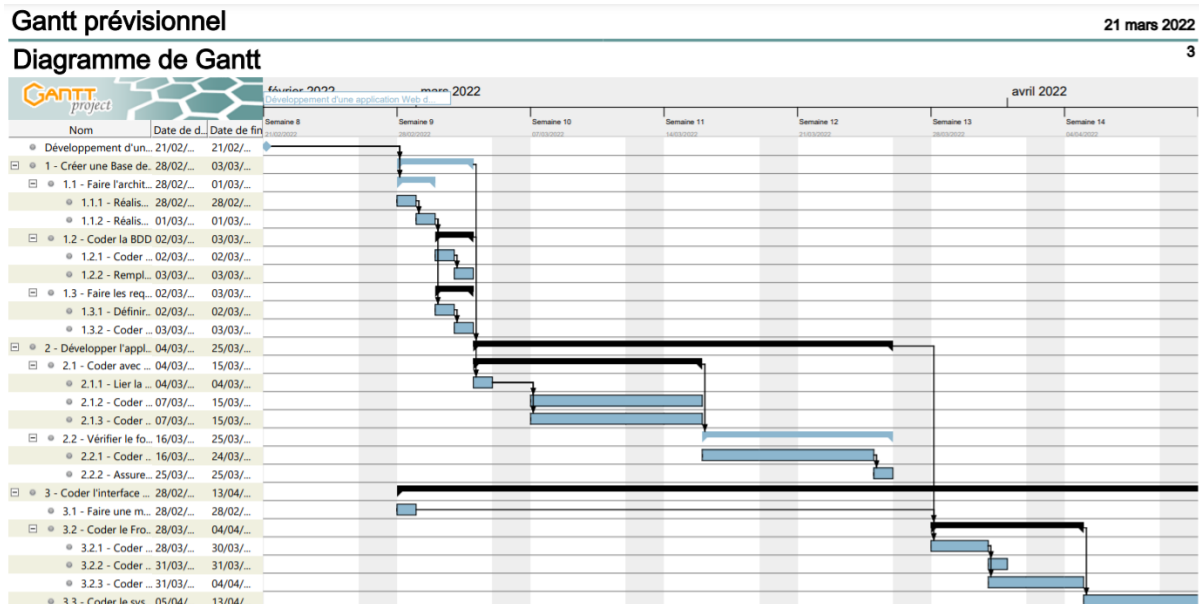


Figure 5 : Diagramme de Gantt prévisionnel (cf. annexe)

#### 4. RACI

Une fois notre WBS réalisé et les tâches clairement définies donc, nous avons pu réaliser notre RACI (Responsible, Accountable, Consulted, Informed), aussi appelé RAM (Responsibility Assignment Matrix).

Cela nous a permis de nous répartir les tâches entre nous deux, en mettant toujours l'autre en Accountable, pour que les deux parties de ce projet soient conscientes de tout ce qui a été fait, est fait et est à faire.

Nous avons aussi dû nous poser la question sur à quel moment consulter ou informer notre tuteur et notre cliente.

				Elsa	Maïlle	Tuteur	Cliente
Créer une application Web de gestion des partenariats	Créer la BDD commune	Faire l'architecture de la BDD	Faire le MCD	R	A		
			Faire le MLR	R	A		
		Coder la BDD	Coder les tables	A	R	I	
			Remplir les tables	A	R		C
	Développer l'application Web	Faire les requêtes SQL	Définir les requêtes nécessaires	R	R		C
			Coder en SQL	A	R	I	
		Coder en Spring	Lier la BDD	A	R	I	
			Coder les contrôleurs	A	R		
	Coder l'interface utilisateur	Coder les tests	Coder les différentes vues	R	A		
			Faire passer les tests	R	R		
		Faire une maquette du site	Assurer une couverture de tests de 100 %	R	R	I	
				A	R		C
		Coder le frontend	Rédiger le code HTML	R	A		
			Rédiger le code CSS	R	A		
		Coder un système d'authentification	Rédiger le code Javascript	R	A	I	
				R	R	I	C

Figure 6 : RACI (cf. annexe)



## V. Architecture de la Base de Données

Avant même de commencer à coder, il nous fallait faire toute l'architecture de la base de données.

Nous avons commencé par le Modèle Conceptuel de Données (MCD), réfléchis au brouillon et proposé à notre tuteur de projet. Celui-ci nous a aiguillées sur une manière de l'améliorer, et une fois que ce fut fait, nous l'avons mis au propre grâce à l'application Magic Draw.

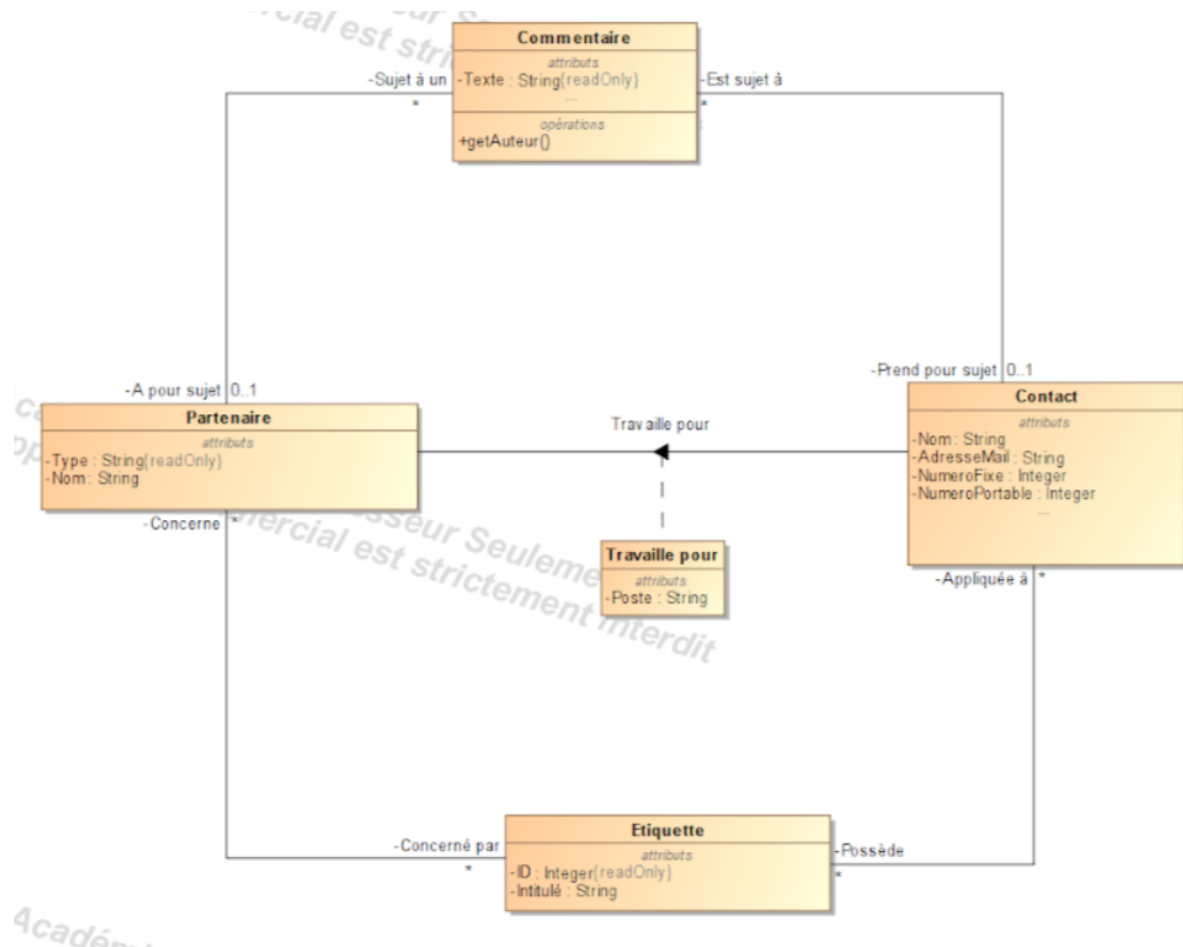


Figure 7 : MCD, première version

Nous nous sommes ensuite rendues compte que nous avions besoin d'un plus grand nombre d'attributs pour chaque table. Ce qui a donné le modèle UML suivant :

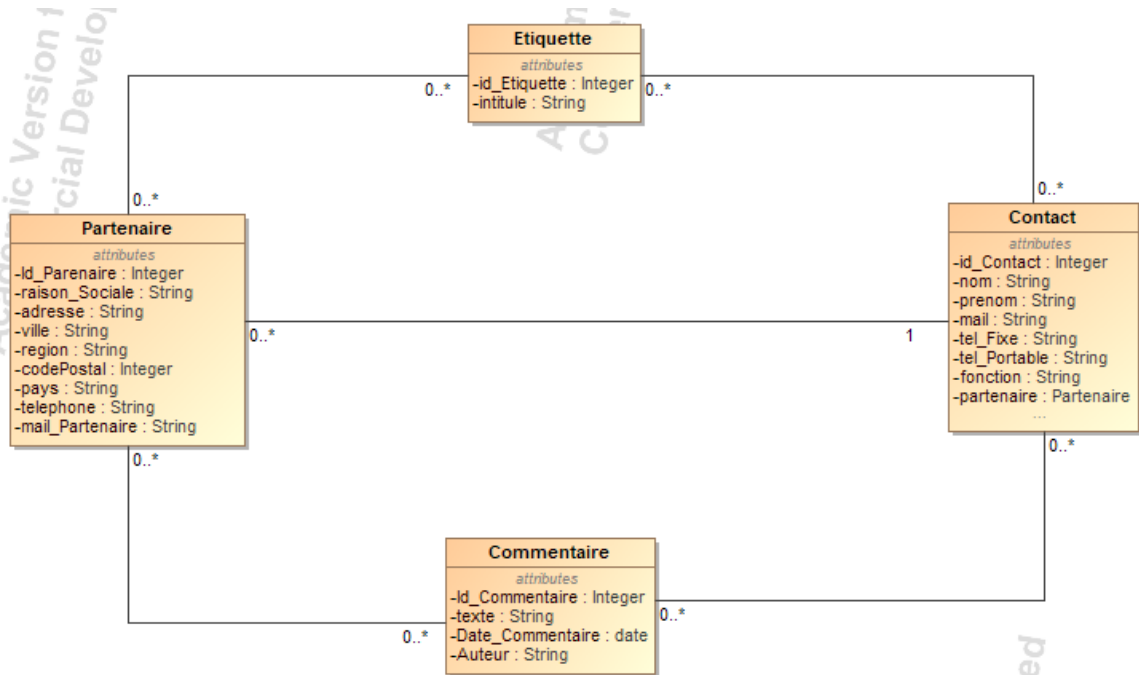


Figure 8 : Diagramme UML du MCD

Suite à cela, nous avons réalisé le Modèle Logique Relationnel (MLR).

#### MLR

Commentaire (Id\_Commentaire, texte, Date\_Commentaire, Auteur)

Partenaire (Id\_Partenaire, raison\_sociale, adresse, ville, region, codePostal, pays, telephone, mail\_Partenaire)

Contact (id\_Contact, nom, prenom, mail, tel\_Fixe, tel\_Portable, fonction, partenaire, #Id\_Partenaire)

Etiquette (id\_Etiquette, intitule)

R1(#Id\_Commentaire, #Id\_Partenaire) → Relation liant Commentaire et Partenaire

R2(#Id\_Commentaire, #id\_Contact) → Relation liant Commentaire et Contact

R3(#id\_Contact, #id\_Etiquette) → Relation liant Contact et Etiquette

R4(#id\_Etiquette, #Id\_Partenaire) → Relation liant Etiquette et Partenaire

Figure 9 : MLR

## VI. Application Web

Pour coder l'application nous nous sommes appuyés sur les exemples vus en cours de technologie web, notamment l'exemple des comptoirs(cf annexe). De ce fait, nous avons codé avec le framework Spring, en JAVA et pour le frontend avec HTML/CSS, JavaScript et Mustache.

### A. Backend

En utilisant Spring, la base de données est générée automatiquement grâce aux classes-entité. Ainsi il n'est pas nécessaire de coder en SQL.

#### 1. Classes-entités (Entity)

Nous avons donc créé cinq classes-entités : *Partenaire*, *Contact*, *Étiquette* et *Commentaire*. Chaque classe-entité a une clé primaire qui est l'id de chaque entité. Les propriétés sont identiques à celles spécifiées dans le MCD (Figure 8) Les relations qui lient les entités sont les suivantes, entre *Partenaire* et *Contact* c'est une relation 1:p bidirectionnelle, respectivement, *@OneToMany* et *@ManyToOne*. Cela illustre l'appartenance des contacts à une institution ou entreprise; entre *Partenaire* et *Contact* entretiennent une relation n:p bidirectionnelle, *@ManyToMany*, avec *Étiquette*. Plusieurs étiquettes peuvent être assignées aussi bien à un partenaire qu'à un contact. Nous aurions pu d'ailleurs utiliser seulement une relation n:p unidirectionnelle. Il en va de même entre la relation entre *Partenaire* et *Commentaire* puis *Contact* et *Commentaire*.

#### 2. Repository

Dans le dossier DAO, un repository a été créé pour chaque entité. Seulement la classe *EtiquetteRepository* est implémentée de deux fonctions de recherche. La première *trouverParIntitule* prenant en paramètre l'*intitulé* de la classe-entité *Étiquette* la fonction permettrait de rechercher une étiquette par son *intitulé* et également *trouverIntituleContenant* permettant de rechercher une étiquette si son *intitulé* contient la chaîne de caractères passé en paramètre. Toutefois dans les "repository" de *Partenaire*, *Commentaire* et *Contact* nous n'avons pas implémenté d'autres fonctions que celles générées automatiquement par Spring.

#### 3. Controller

Pour chaque table nous avons un controller associé, dans chaque fichier y sont implémentés une méthode pour lister les valeurs de la table, une méthode ajouter, supprimer, modifier et enregistrer. Pour les méthodes enregistrer (save) et supprimer (delete) sont implémentés avec une exception qui affiche un message d'erreur dans le cas d'une sauvegarde si, par exemple, l'objet existe déjà. Quant à la méthode supprimer elle affiche un message d'erreur on veut supprimer un partenaire est supprimé mais il est encore lié à des contacts. Cependant, à ce jour, ces méthodes ne fonctionnent pas mais feront l'objet de rectifications dans les jours à venir.

De plus, de nouveaux contrôles ont été créés comme *ContactPartenaireContrôler* permettant d'afficher les contacts en fonction de leur partenaire auquel ils sont liés.

## B. Frontend

### 1. Code HTML et Mustache

Pour pallier tous les risques, nous avons réparti le travail en deux : une suite de pages HTML et Mustache. Ces dernières sont utilisées pour afficher les listes des partenaires, contacts, étiquettes et commentaires et les formulaires pour l'ajout ou/et la modification.

Ainsi, même si nous ne parvenons pas à lier le front au back, nous avons les fichiers HTML à présenter.

Le principe est le même dans les deux cas, même si les fichiers Mustache s'éloignent légèrement de notre maquette initiale, contrairement aux fichiers HTML qui en sont une illustration parfaite.

### 2. Code CSS

Nous avons fait un fichier style CSS pour styliser nos pages HTML et Mustache. Nous nous sommes appliquées à prendre les couleurs du logo ISIS pour personnaliser notre application web.

Nous avons fait de même avec les fichiers HTML, mais plutôt que de faire plusieurs fichiers CSS, nous avons tout inclus dans des balises Style.

### Liste des contacts

ID	Nom	Prenom	Mail	Tel Fixe	Tel Portable	Fonction
2001	Dupont	Agnès	dupont.agnes@institut-esante.fr		06.53.48.78.96	Responsable Informatique
2002	Dupuy	Yvan	direction@ch-sabran.fr	05.36.96.36.58		Directeur
2003	Dujardin	Stéphanie	rsi@chru-tls.com	05.45.63.21.89	07.89.45.12.36	Responsable Systemes d'Informations

Figure 10 : Exemple de style CSS

## VII. Conclusion

Au cours du projet, nous avons par moments été confrontées à certains problèmes.

Premièrement, nous avons eu des problèmes liés aux délais. En effet, à cause des très courts délais, nous n'avons même pas eu le temps de nous pencher sur la programmation de l'authentification et de la sécurité de notre application. Nous avons en effet pris la décision de nous focaliser sur l'implémentation des demandes de la cliente en priorité.

Nous avons aussi été freinées par notre capacité de travail, puisque nous n'étions que deux à travailler sur ce projet. Notre charge de travail était donc assez conséquente et surtout limitée. Nous ne pouvions pas traiter assez de tâches simultanément, ce qui a prolongé notre temps de travail et nous a fait prendre du retard par rapport à nos prévisions.

La base de données originellement demandée par notre tuteur de stage n'a malheureusement pas pu se faire, et nous avons donc décidé, avec son accord, de nous concentrer sur une base de données plus personnelle et centrée sur les besoins de notre cliente uniquement.

Cependant, grâce à ce projet, nous avons aussi pu développer nos compétences, à la fois en tant que développeuses et en gestion de projet. Nous avons pu mettre en pratique tout ce que nous avons appris lors de nos deux derniers semestres d'école, pour un projet réel et pour lequel nous étions personnellement sollicitées. Cela nous a aussi préparées pour notre stage à venir.

## VIII. Annexes

Figures du : IV. B. Analyse de faisabilité :

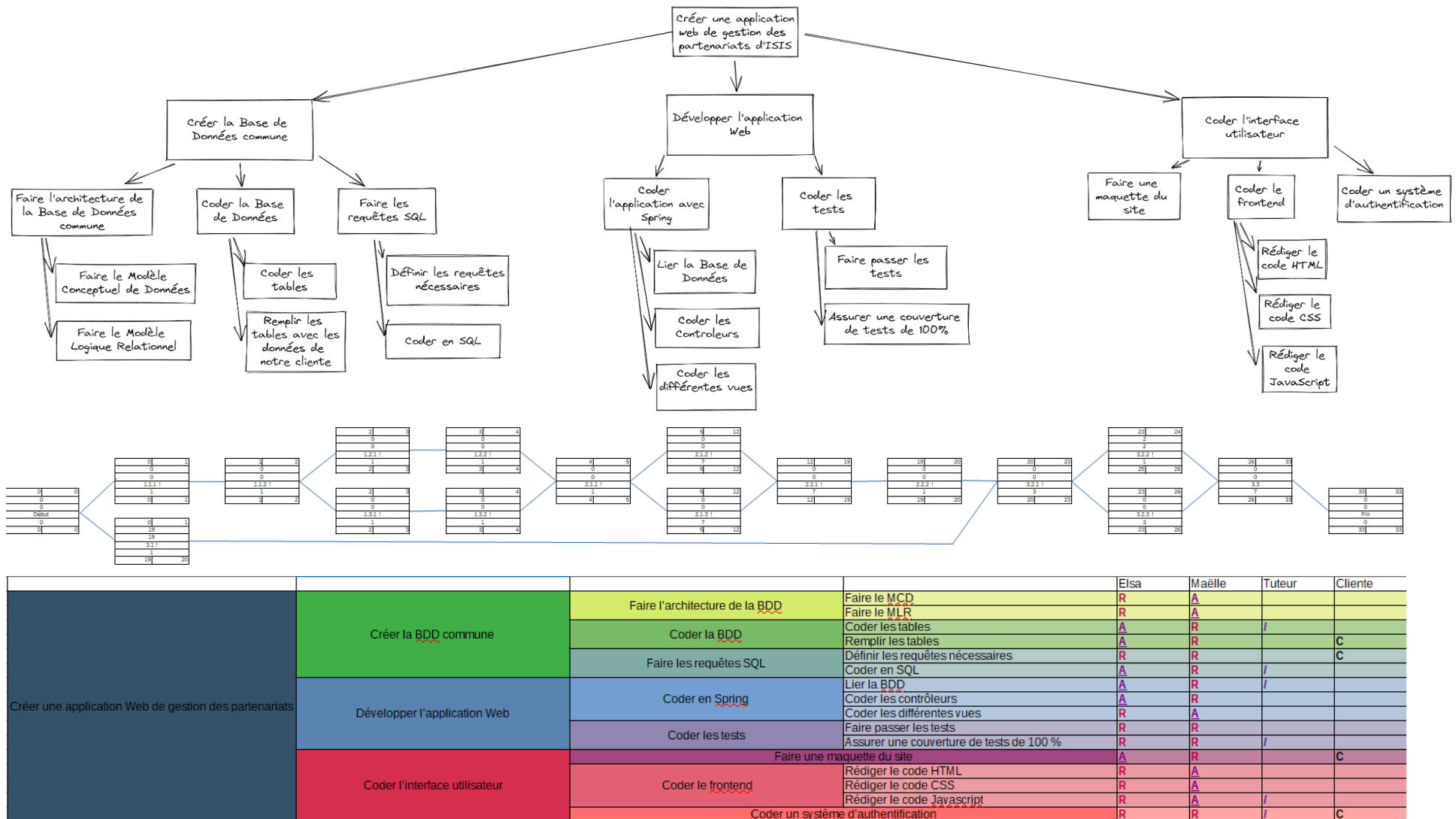
### Matrice d'analyse des risques

Catastrophique	R2 R3 R4	R13			
Majeure	R10	R12	R5 R14		R6 R7
Très grave			R11	R8 R15	
Significative			R9	R1	
Mineure					
	Incroyable	Peu probable	Occasionnelle	Probable	Presque certaine

### Registre des risques complet

Registre des risques					
Indice	Nature du risque	Description	Gravité	Probabilité	Risque
R1	Technique	Mauvais choix des starters Spring	Significative	Probable	Modéré
R2	Humain	Une des ressources est dans l'impossibilité de continuer le pro	Catastrophique	Incroyable	Modéré
R3	Technique	L'ordinateur d'une des ressources est inutilisable	Catastrophique	Incroyable	Modéré
R4	Technique	L'application est hackée	Catastrophique	Incroyable	Modéré
R5	Technique	Base de données n'est pas fonctionnel	Majeure	Occasionnelle	Elevé
R6	Technique	L'application ne s'ouvre pas dans le navigateur	Majeure	Presque certaine	Non acceptable
R7	Technique	Les tests ne passent pas	Majeure	Presque certaine	Non acceptable
R8	Technique	Les tests ne sont pas approprié	Très grave	Probable	Elevé
R9	Humain	Un désaccord apparaît entre les ressources	Significative	Occasionnelle	Bas
R10	Humain	Un conflit surgit entre les ressources	Majeure	Incroyable	Bas
R11	Technique	Méconnaissance des langages et/ou logiciels employés	Très grave	Occasionnelle	Modéré
R12	Délai	Ne pas finir à temps le code	Majeure	Probable	Elevé
R13	Délai	Ne pas rendre à temps le rapport	Catastrophique	Peu probable	Elevé
R14	Technique	Le frontend et le backend ne peuvent pas fonctionner ensemb	Majeure	Occasionnelle	Elevé
R15	Délai	Rallonger la durée d'une tâche	Très grave	Probable	Elevé

## Figures du : IV. C. Conception du projet



## Figures du : IV. B. Analyse de faisabilité :

### Gantt prévisionnel

21 mars 2022

#### Tâches

2

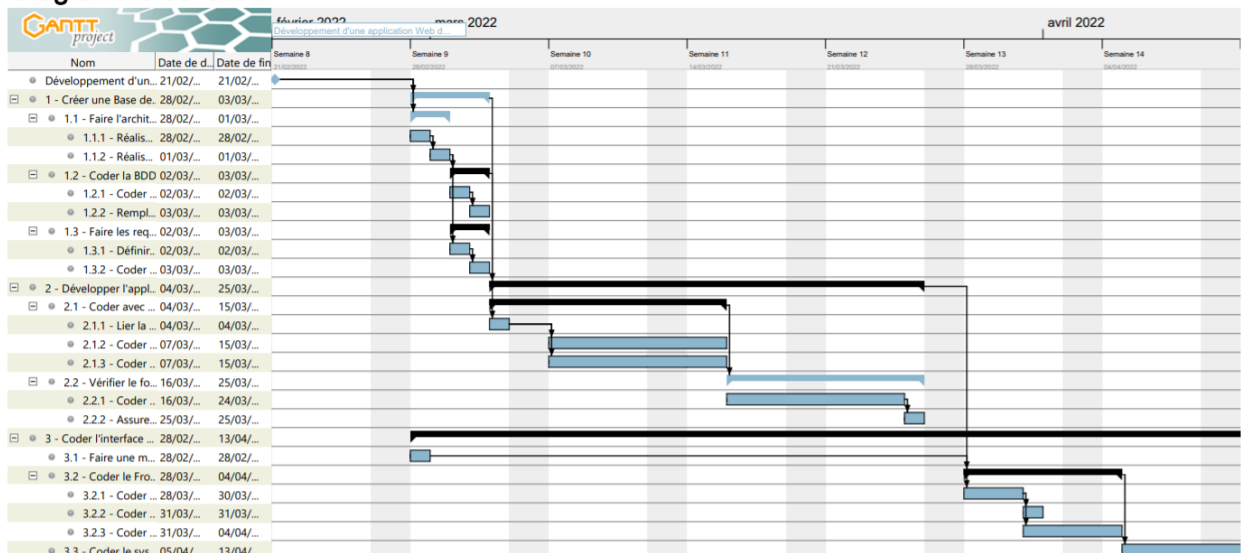
Nom	Date de début	Date de fin
Développement d'une application Web de gestion des partenariats	21/02/2022	21/02/2022
1 - Créer une Base de Données commune aux autres groupes	28/02/2022	03/03/2022
1.1 - Faire l'architecture de la BDD	28/02/2022	01/03/2022
1.1.1 - Réaliser un MCD	28/02/2022	28/02/2022
1.1.2 - Réaliser un MLR	01/03/2022	01/03/2022
1.2 - Coder la BDD	02/03/2022	03/03/2022
1.2.1 - Coder les différentes tables	02/03/2022	02/03/2022
1.2.2 - Remplir ces tables	03/03/2022	03/03/2022
1.3 - Faire les requêtes SQL	02/03/2022	03/03/2022
1.3.1 - Définir les requêtes nécessaires	02/03/2022	02/03/2022
1.3.2 - Coder en SQL	03/03/2022	03/03/2022
2 - Développer l'application Web	04/03/2022	25/03/2022
2.1 - Coder avec Spring	04/03/2022	15/03/2022
2.1.1 - Lier la BDD	04/03/2022	04/03/2022
2.1.2 - Coder les controllers	07/03/2022	15/03/2022
2.1.3 - Coder les différentes vues	07/03/2022	15/03/2022
2.2 - Vérifier le fonctionnement de l'application	16/03/2022	25/03/2022
2.2.1 - Coder les tests	16/03/2022	24/03/2022
2.2.2 - Assurer une couverture de tests de 100%	25/03/2022	25/03/2022
3 - Coder l'interface utilisateur	28/02/2022	13/04/2022
3.1 - Faire une maquette	28/02/2022	28/02/2022
3.2 - Coder le Front-end	28/03/2022	04/04/2022
3.2.1 - Coder le HTML	28/03/2022	30/03/2022
3.2.2 - Coder le CSS	31/03/2022	31/03/2022
3.2.3 - Coder le Javascript	31/03/2022	04/04/2022
3.3 - Coder le système d'authentification	05/04/2022	13/04/2022

### Gantt prévisionnel

21 mars 2022

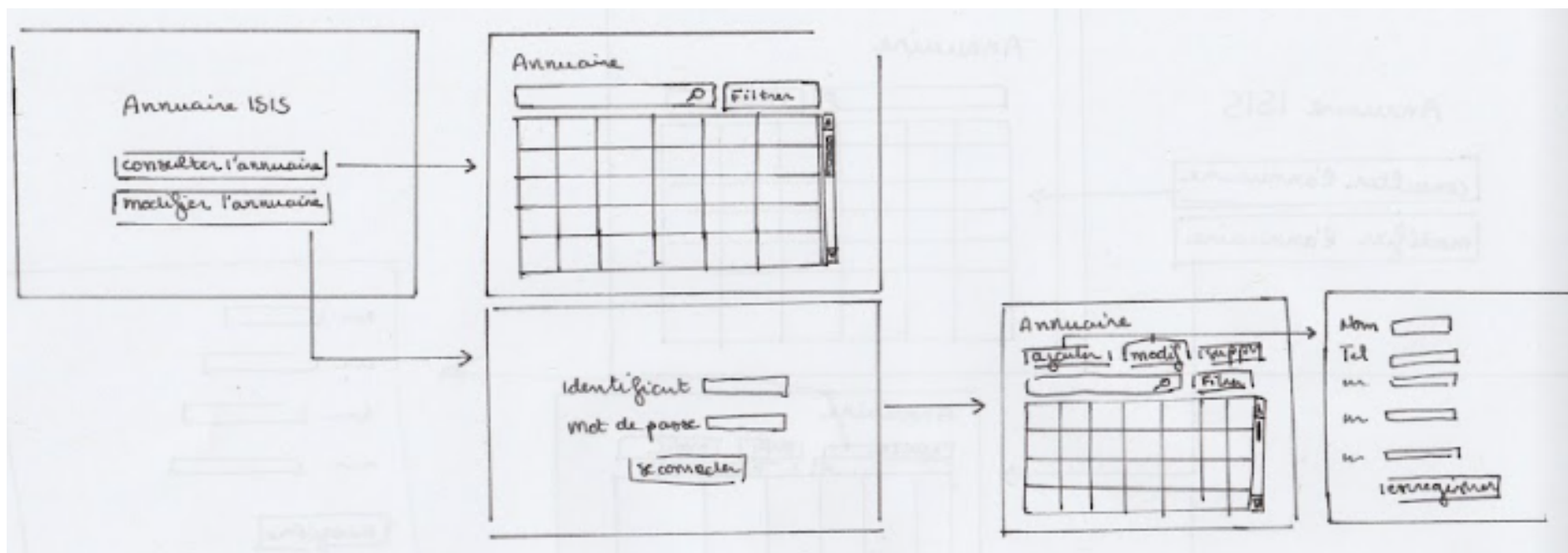
#### Diagramme de Gantt

3





Maquette de l'application Web :



Liens vers le programme comptoirs de M. Bastide dont nous nous sommes servis : <https://comptoirs-spring-mvc.herokuapp.com/>

## IX. Bibliographie

PINGAUD, Hervé. "Bases de gestion de projets" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2021

PINGAUD, Hervé. "Bases de gestion de projets : la planification" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2021

RAWAT, Eshane. "Gestion de projet informatique" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

LAMINE, Elyes. "Introduction aux bases de données, Systèmes de Gestion de Bases de Données, Théorie de Normalisation"[notes fournies dans le cours Technologie Web], ISIS, Castres, 2021

PECATTE, Jean-Marie. "Systèmes de gestion de Bases de données : le langage SQL" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2021

BASTIDE, Rémi. "Client-Serveur Web : architectures et technologies" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

BASTIDE, Rémi. "Technologies Web - accès au données avec Spring JPA" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

BASTIDE, Rémi. "AJAX sur le backend" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

PECATTE, Elisabeth. "Modules de Technologies Web" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

PECATTE, Elisabeth. "CSS" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022

PECATTE, Jean-Marie. "Module Technologie Web : le langage JavaScript" [notes fournies dans le cours Technologie Web], ISIS, Castres, 2022