

Tayeb Bousfiha

Rapport de stage

*Etude et Développement d'une plate-forme
pour les étudiants*

Effectué du 1 juillet 2019 au 27 septembre 2019

Dans la Société :



À Fès

*De l'université « Institut supérieur d'ingénierie et des
affaires »*



Année universitaire : 2018/2019

Remerciements

Au terme de ce travail, nous saisissons cette occasion pour exprimer nos vifs remerciements à toute personne ayant contribué, de près ou de loin, à sa réalisation.

Nous souhaitons tout d'abord remercier Mme. MAASAOUI, la Chief Operating Officer et M. MELLOULI pour le vif accueil qu'ils nous ont accordé au sein de la société, ainsi que M. BENTALEB qui nous a encadré tout au long de notre expérience, M. ELFATNI pour ces explications et aides apportées et tout le personnel de la société « SOLVE IT TECHNOLOGIES » pour le temps qu'ils nous ont accordé afin de nous donner les explications nécessaires et répondre à nos questions.

Nous tenons à adresser aussi nos remerciements les plus profonds à tout le personnel de l'ISGA et à l'ensemble du corps professoral pour l'effort qu'il fournit pour faire de nous des futurs ingénieurs, sans oublier de remercier nos chers parents pour tous leurs sacrifices tout au long de nos études.

Liste des tableaux :

Tableau 1: Classe Etudiant	19
Tableau 2: Classe Compte	19
Tableau 3: Classe Retard	19
Tableau 4: Classe Note.....	20
Tableau 5: Classe Paiement.....	20
Tableau 6: Classe Classe.....	20
Tableau 7: Classe Absence	20
Tableau 8: Les livrables	21
Tableau 9: Les risques du stage	22
Tableau 10 : Acteurs	24

Liste des figures :

Figure 1: architecture de l'application web	11
Figure 2: Construction du REST API dans l'architecture de l'application.....	12
Figure 3: Planning initial - Page1	13
Figure 4: Planning initial - Page2	14
Figure 5 : Planning réel et écarts avec le planning initial - Page1	16
Figure 6 : Planning réel et écarts avec le planning initial - Page2	17
Figure 7: Diagramme de classe.....	25
Figure 8: Diagramme de cas d'utilisation.....	26
Figure 9: Diagramme de séquence pour un scénario d'authentification	27
Figure 10: Diagramme de séquence pour un scénario d'ajout	28
Figure 11: Diagramme de séquence pour un scénario de suppression.....	29
Figure 12: Diagramme de séquence pour un scénario de modification.....	30
Figure 13: Diagramme de séquence pour un scénario de recherche	31
Figure 14: Architecture technique	32
Figure 15: Design général.....	34
Figure 16 : Intercepteur de requête http.....	36
Figure 17 : "three dot menu"	37
Figure 18 : methode \$Lookup aggregation	38
Figure 19: Page de gestion des étudiants	43
Figure 20 : Page de gestion des suivis de chaque étudiant détaillée	44
Figure 21 : Page d'ajout d'étudiant.....	45
Figure 22 : Page de modification d'étudiant	46
Figure 23 : Page d'authentification.....	47
Figure 24 : Page d'envoi de mail de réinitialisation du mot de passe	48
Figure 25 : Page de réinitialisation de mot de passe	49
Figure 26 : Copie scannée du certificat de stage.....	52

Sommaire

Introduction	1
1. Contexte général du stage	2
1.1. Contexte du stage.....	3
1.2. Cadre du stage.....	7
2. Etude générale du projet	8
2.1. Cahier des charges.....	9
2.2. Projet confié	12
2.3. Plannings	13
2.4. Dictionnaire de données	19
2.5. Les livrables	21
2.6. Les risques	22
3. Analyse et conception	23
3.1. Identification des acteurs.....	24
3.2. Partie UML	25
3.3. Architecture technique	32
3.4. Conception	34
3.5. Enseignements/apports du stage (connaissances - compétences).....	39
4. Réalisation.....	40
4.1. Outils utilisés	41
4.2. Imprimé écran	43
Conclusion et perspectives	50
Webographie.....	51
Copie scannée du certificat de stage	52

Introduction

Dans le cadre du programme d'études et de la formation que dispose l'ISGA (Institut Supérieur d'Ingénierie et des Affaires) de Fès, et en vue de la parfaire et de l'adapter aux exigences du marché de l'emploi, les étudiants de la quatrième année ont l'obligation d'effectuer un stage au sein d'une entreprise afin d'acquérir plus d'expérience professionnelle. Dans ce sens nous avons réalisé ce stage, au sein de Solve IT Technologies, dont l'activité est le développement web, mobile et logiciel.

Dans ce rapport, nous exposons, chronologiquement, les différentes étapes de l'implémentation de notre application. Dans un premier temps, nous aborderons le contexte et l'étude générale du projet, ensuite nous présenterons le cahier des charges établi, puis nous mettrons en exergue l'analyse et la conception ainsi que la phase de réalisation de notre projet.

1. Contexte général du stage

Dans cette partie, nous présentons l'établissement qui a formulé le besoin et le cadre général du projet.

1.1. Contexte du stage

1.1.1. Solve IT Technologies

SOLVE IT Technologies est l'une des sociétés de développement web à la croissance la plus rapide à Fès. Elle assure le développement des sites Web de haute performance, des applications mobiles, des logiciels, des sites E-commerce pour différents types d'entreprise. Ses équipes de développement visent toujours à satisfaire les clients. SOLVE IT Technologies travaille avec un large éventail de clients, allant des petites entreprises aux grandes entreprises. Au cours des années, Ils ont aidé leurs clients à créer d'excellentes solutions web et mobile. Ils adaptent leurs diverses solutions à l'imagination des clients et leurs besoins pour leur présenter des solutions web, mobiles ou logiciels de gestion digne d'un succès.

1.1.2. Domaines de compétences

SOLVE IT Technologies dispose d'une équipe professionnelle qui croit en ses talents de développeur et assure les meilleures solutions à ses clients. Elle développe des sites Web réactifs, des logiciels, des applications mobiles pour tout type d'industrie, principalement pour les secteurs suivants :

_ Éducation

Avec l'évolution technologique, les méthodes traditionnelles d'enseignement et de communication sont modifiées. La plupart des institutions optent pour un environnement E-Learning. SOLVE IT Technologies développe pour ses clients les applications logicielles entièrement intégrées qui contribuent à améliorer la qualité de l'éducation et de l'administration dans les écoles, les collèges et les universités et à améliorer l'interaction entre les étudiants.

Ses équipes de développement Web comprennent les défis rencontrés par de nombreuses universités et collèges, pour résoudre ces défis et construire un meilleur environnement d'apprentissage, Ils développent des systèmes intégrés de gestion de l'information, des app E-Learning, des applications de gestion et automatisation de l'éducation. Ils maîtrisent les dernières technologies Web pour autonomiser l'éducation en ligne, les programmes de formation, améliorer l'apprentissage des étudiants et élargir l'accès à l'enseignement supérieur.

_ Santé

Les solutions de soins de santé de SOLVE IT Technologies suivent des séquences de travail progressives et itératives pour satisfaire les patients et le personnel. Les différentes étapes impliquées sont énumérées ci-dessous : Communication claire : Construire un environnement qui favorise la communication ouverte. Comprendre la perspective des autres. Établir un environnement de prise de décision. Gestion complète des tournées : Considération et compréhension Vigilance et reconnaissance.

_ Business

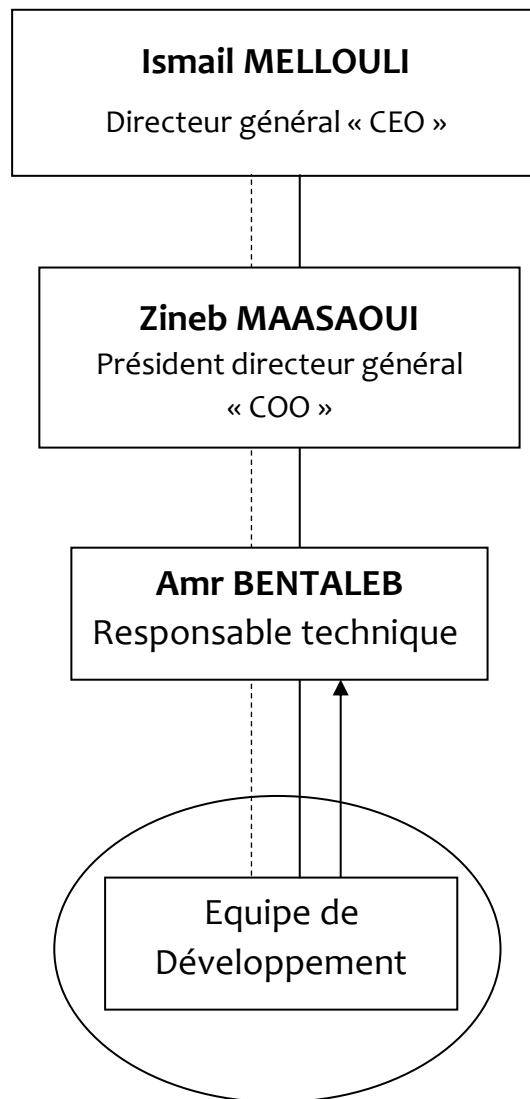
L'équipe de SOLVE IT Technologies offre des services de développement de logiciels et d'applications personnalisés rentables et de première ligne, le développement Web réactif, le développement d'applications mobiles et les services de développement de sites E-commerce pour toutes les tailles d'entreprises. Leur environnement de travail axé sur la qualité cible le succès du client grâce à une approche personnalisée de la recherche et du développement de projets. Solve IT Technologies possède une solide expérience du domaine qui fait d'eux un partenaire

idéal pour les clients. Ils mettent en œuvre leur idée et la stratégie du client dans la réalité.

Les sites E-commerces sont généralement plus faciles à mettre en place que les magasins physiques et nécessitent également moins d'investissements. Il est disponible 24 * 7 pour tout le monde. L'équipe professionnelle de Solve IT Technologies accompagne ses clients dans le développement de sites E-commerces pour leur entreprise.

1.1.3. Organigramme

Afin de parvenir aux conditions optimales du développement web, la direction à la société SOLVE IT Technologies est structurée selon des niveaux hiérarchiques et fonctionnels comme l'illustre l'organigramme suivant :



- : liaison fonctionnelle.
- : liaison hiérarchique.
- : Où nous nous situons.
- : notre client.

1.2. Cadre du stage

Notre objectif étant de se former aux technologies de la société et de se familiariser avec. Dans ce sens, nous avons suivi une autoformation et nous l'avons mise en pratique en réalisant le projet suivant :

Soit une école, qui, pour offrir une meilleure qualité de service, met en place un nouveau système nommé « STUDENT DIRECT ».

« STUDENT DIRECT » est une plateforme qui met à la disposition des étudiants et des parents, au jour le jour, toutes les informations relatives à la vie estudiantine au sein de l'école. Ces derniers auront un accès direct et en temps réel à un ensemble d'informations : Suivi des retards, Suivi des absences, Suivi des notes, Suivi des règlements, etc.

2. Etude générale du projet

Dans cette partie, nous présentons le cahier des charges et la planification du stage, ainsi que le projet confié, les livrables et les risques du stage.

2.1. Cahier des charges

2.1.1. Cahier de charge de gestion

Soit une école souhaitant proposer une meilleure qualité de service à ses étudiants. Elle souhaite instaurer un système capable de fournir à chacun de ses étudiants ses informations relatives à la vie estudiantine.

L'école doit pouvoir administrer les étudiants et leurs informations grâce à une application web.

L'application doit fournir différentes interfaces.

Une partie des interfaces web de gestion destinée à l'administrateur et la seconde de consultation qui sera consulté par les étudiants.

Les fonctions de l'administrateur consistent à ajouter, modifier, supprimer les étudiants et leurs informations : Suivi des retards, Suivi des absences, Suivi des notes, Suivi des règlements ...

L'étudiant(e) doit s'authentifier avec ses identifiants personnels fournis à l'école pour accéder à l'ensemble de ses informations.

L'étudiant(e), après authentification sera redirigé vers la page d'accueil qui lui indiquera grâce à un affichage dynamique le montant totale de l'année, la somme payée et la somme restante à payer.

La page d'accueil disposera aussi d'un menu dynamique où il(elle) devra choisir le suivi d'information à consulter.

Le suivi des retards, des absences et des règlements seront présentés sous forme de tableau en détails :

Le suivi des retards (matière du retard, séance, durée du retard en min, date du retard, raison du retard).

Le suivi des absences (matière de l'absence, séance, date de l'absence, heure de l'absence, justification de l'absence).

Le suivi des règlements (somme payée, date de paiement, type de paiement).

Le suivi des notes sera représenté par plusieurs tableaux distant les uns des autres.

Chaque tableau du suivi des notes représente un module contenant les matières du module, leur coefficient, et la note de l'étudiant de chaque matière ainsi que la note totale du module.

2.1.2. Cahier de charge technique

L'application web dispose d'un serveur qui stockera toutes les données relatives à l'étudiant et à sa vie estudiantine, ainsi que les comptes étudiants et administrateurs.

Nous utilisons le **SGBD** (Système de Gestion de Base de Données) **MongoDB**.

La connexion et la gestion de la base de données s'effectue en **JavaScript** grâce à la plate-forme de développement open source **Node.js** qui exécute le code JavaScript côté serveur et au framework web **ExpressJS** qui permet de construire des applications web basées sur Node.js.

L'application tournera en local lors de sa création sur une seule machine où il y aura le serveur web et la base de données.

L'application reposera sur une architecture utilisant tous les composants de la pile logicielle JavaScript **MEAN Stack** « **M** = **MongoDB**, **E** = **Express.js**, **A** = **Angular**, **N** = **Node.js** » :

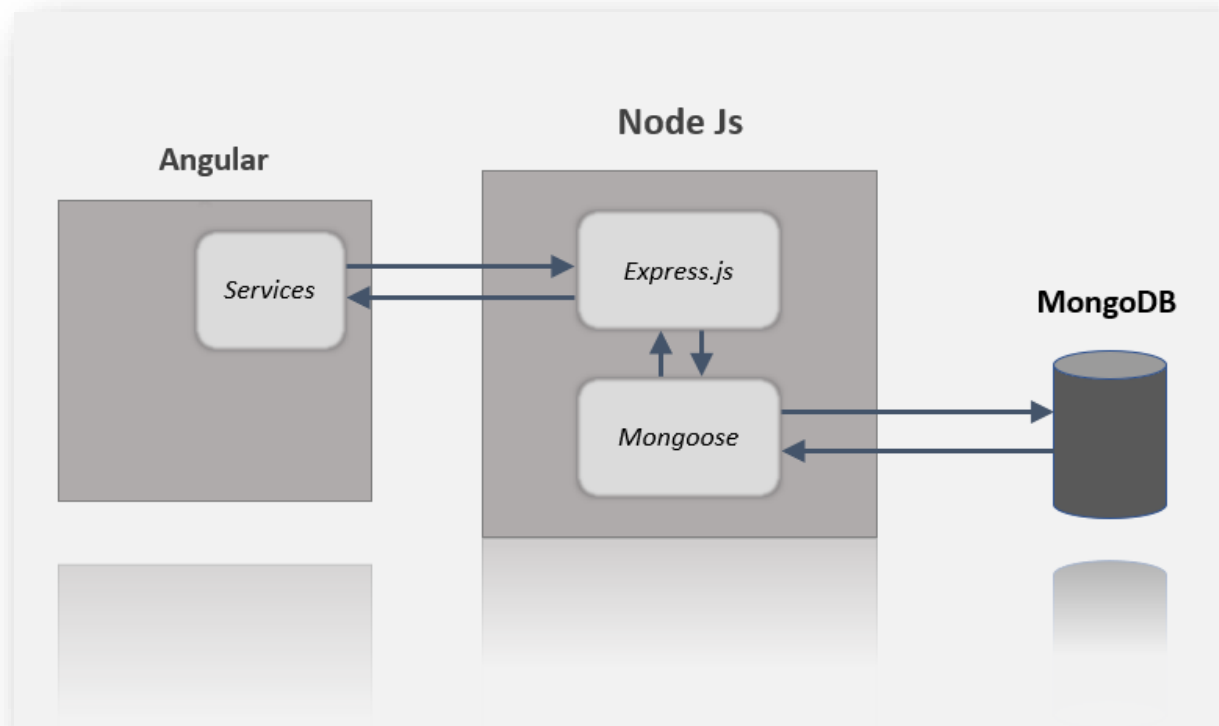


Figure 1: architecture de l'application web

Lorsqu'un client fait une demande, celle-ci est d'abord traitée par **Angular**. C'est un Framework Javascript côté client. C'est une bibliothèque écrite en JavaScript. Après cela, la requête de Angular entre dans la deuxième section qui est Node.JS.

Express.js envoie la demande à la base de données à travers **Mongoose** qui est une librairie **ODM « Object Data Modeling »** pour MongoDB et Node.js permettant de gérer les relations entre les données et de fournir la validation du schéma.

Après cela, MongoDB récupère les données et renvoie la réponse à Express.Js.

Puis Node.JS renvoie la réponse reçue dans Express.Js à Angular pour afficher le résultat.

L'application utilisera le style d'architecture logicielle REST qui alimente une application à page unique (SPA) grâce à Express.Js :

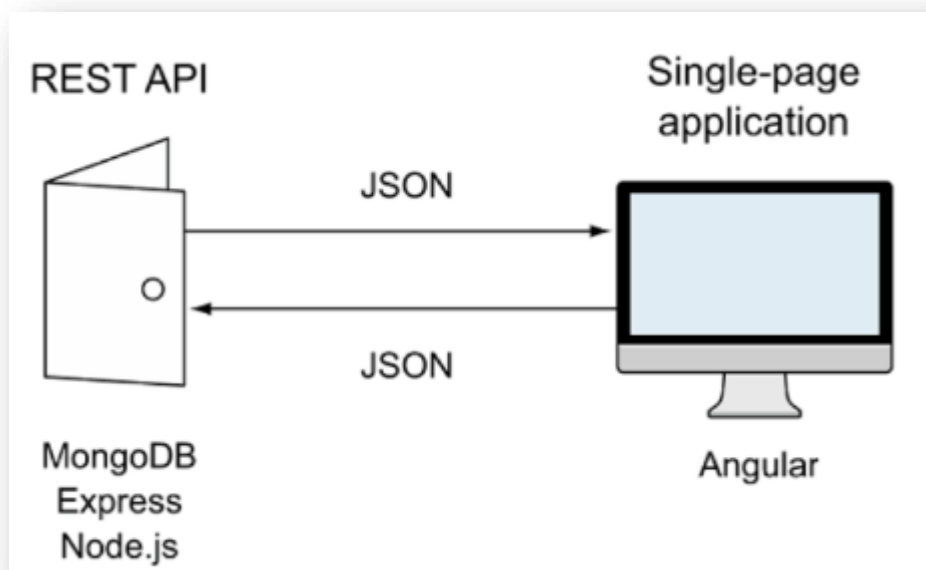


Figure 2: Construction du REST API dans l'architecture de l'application

L'API « Interface de programmation » sera construite avec MongoDB, Express et Node.js tandis que le SPA « application web monopage » avec Angular.

2.2. Projet confié

Lors, de notre stage on nous a confié le projet suivant :

_ Etude et Développement d'une plate-forme pour les étudiants.

2.3. Plannings

2.3.1. Planning initial

La figure suivante représente le planning prévisionnel du projet :

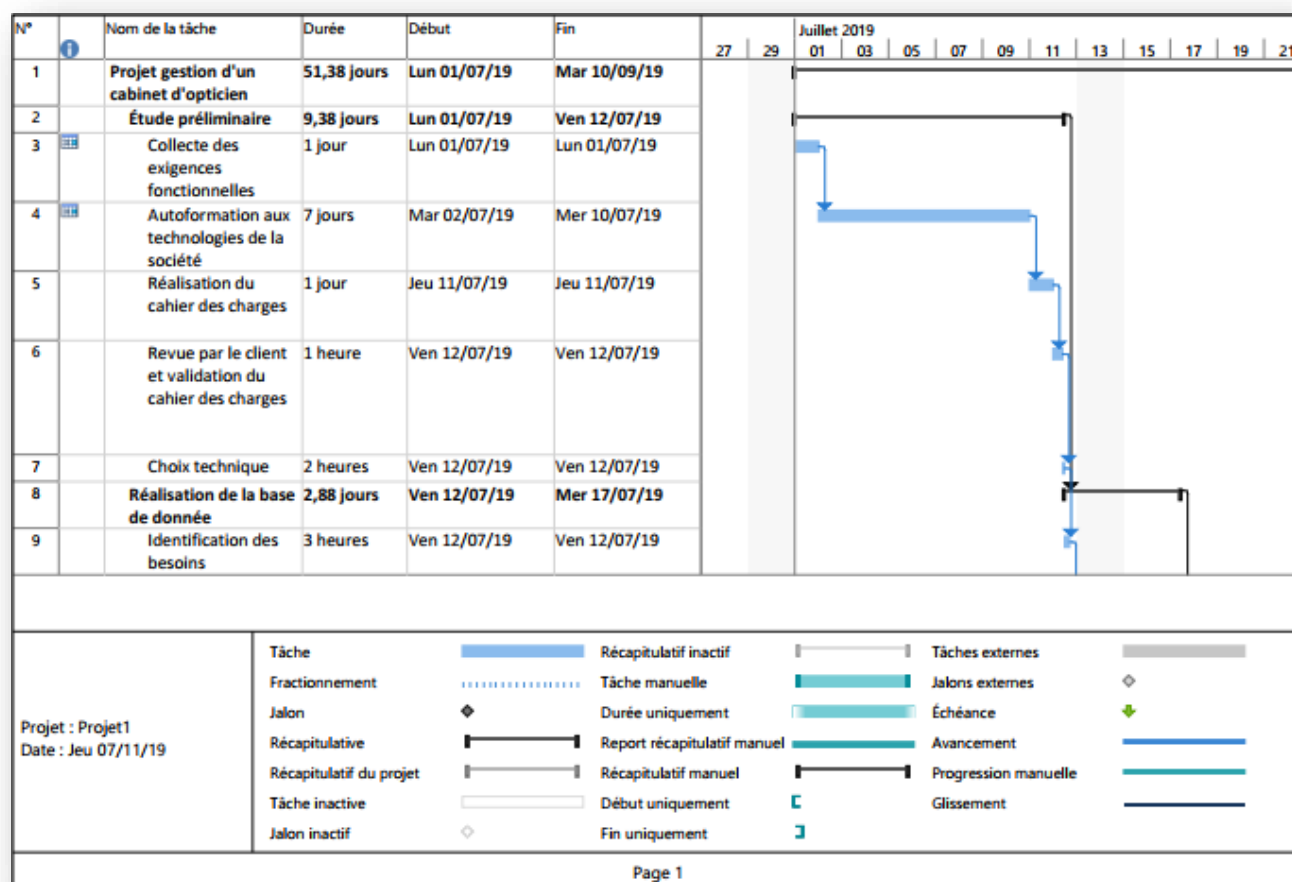


Figure 3: Planning initial - Page1

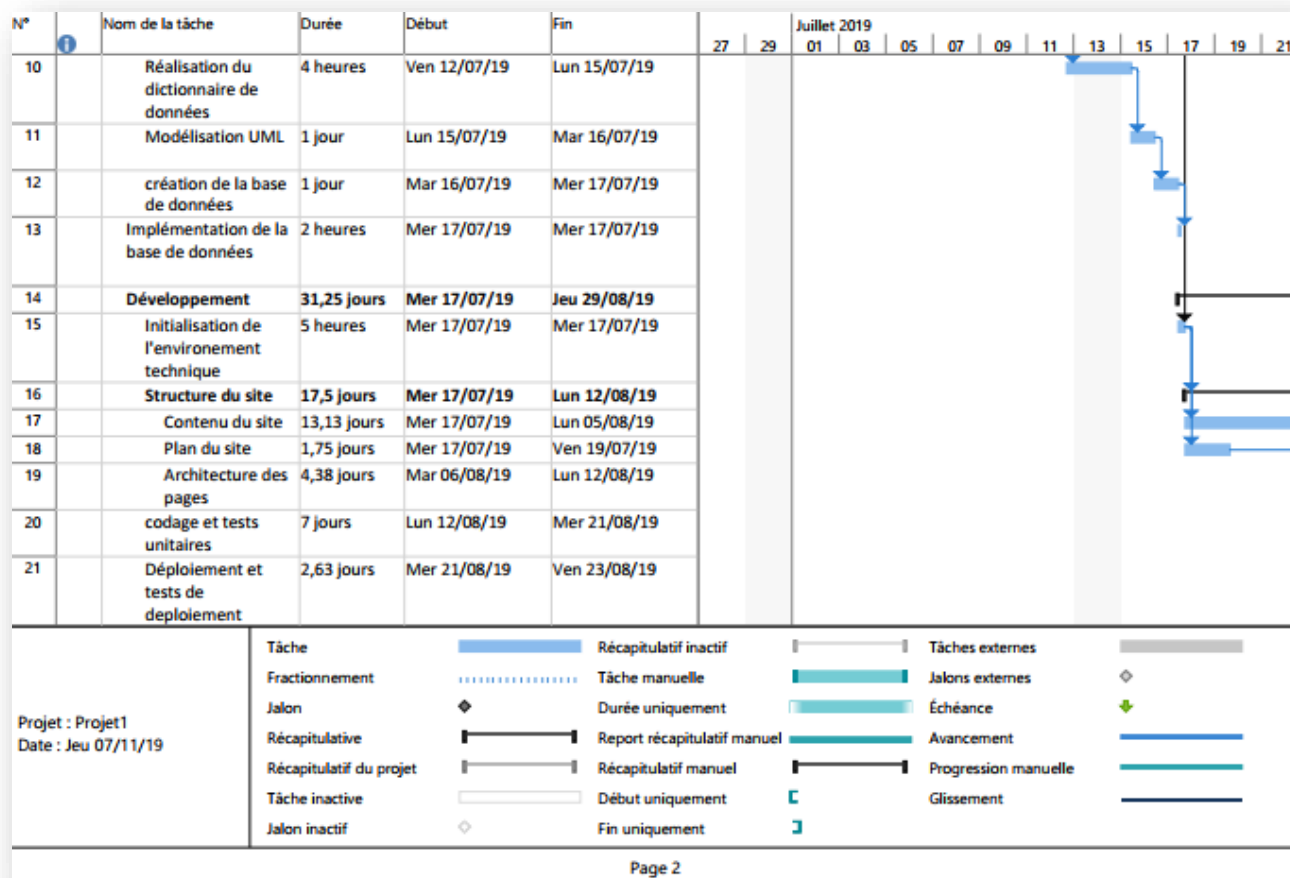


Figure 4: Planning initial - Page2

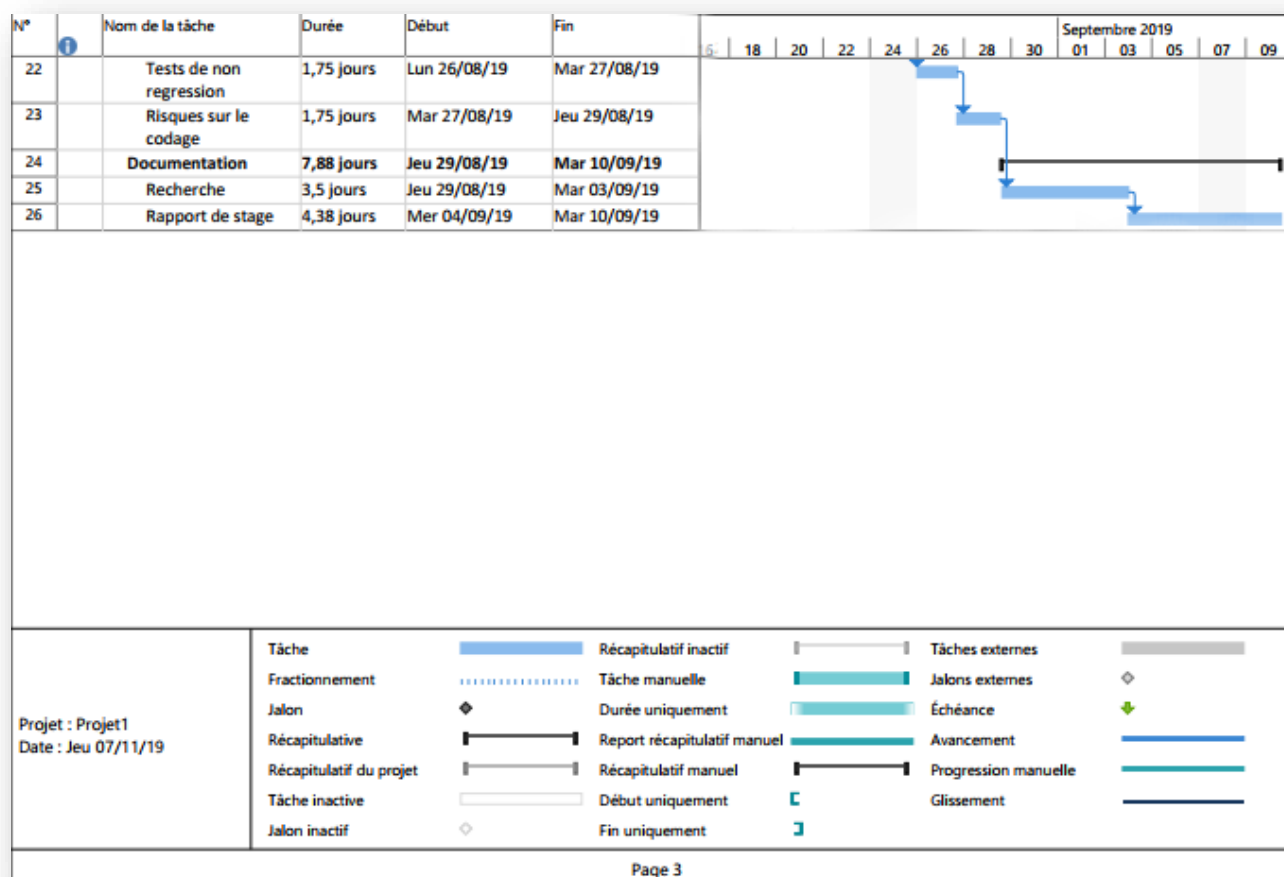


Figure 5 : Planning initial - Page3

2.3.2. Planning réel

En vue des itérations que notre projet a encourues, le planning réel constate un retard par rapport au planning prévisionnel. La figure ci-dessous illustre l'écart entre ces plannings.

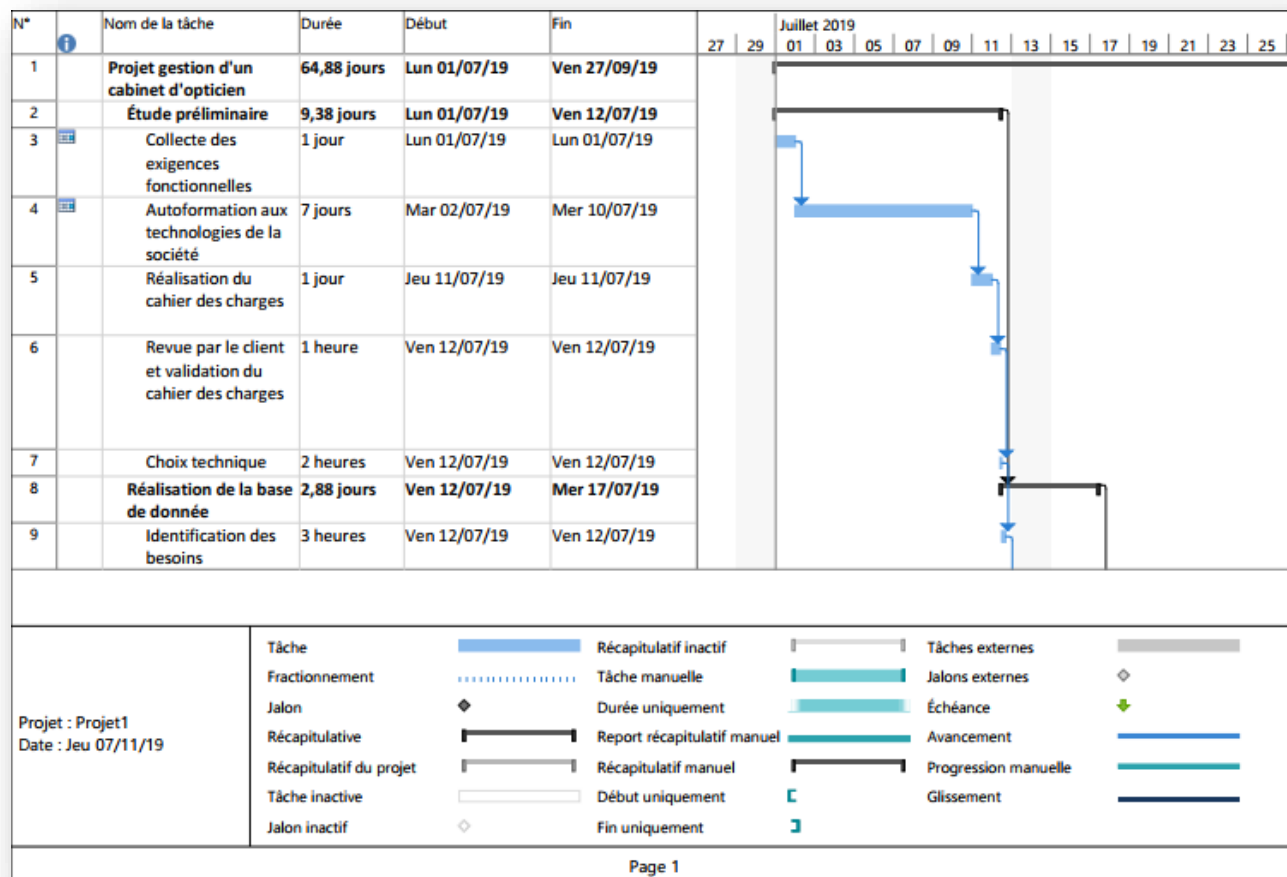


Figure 6 : Planning réel et écarts avec le planning initial - Page1

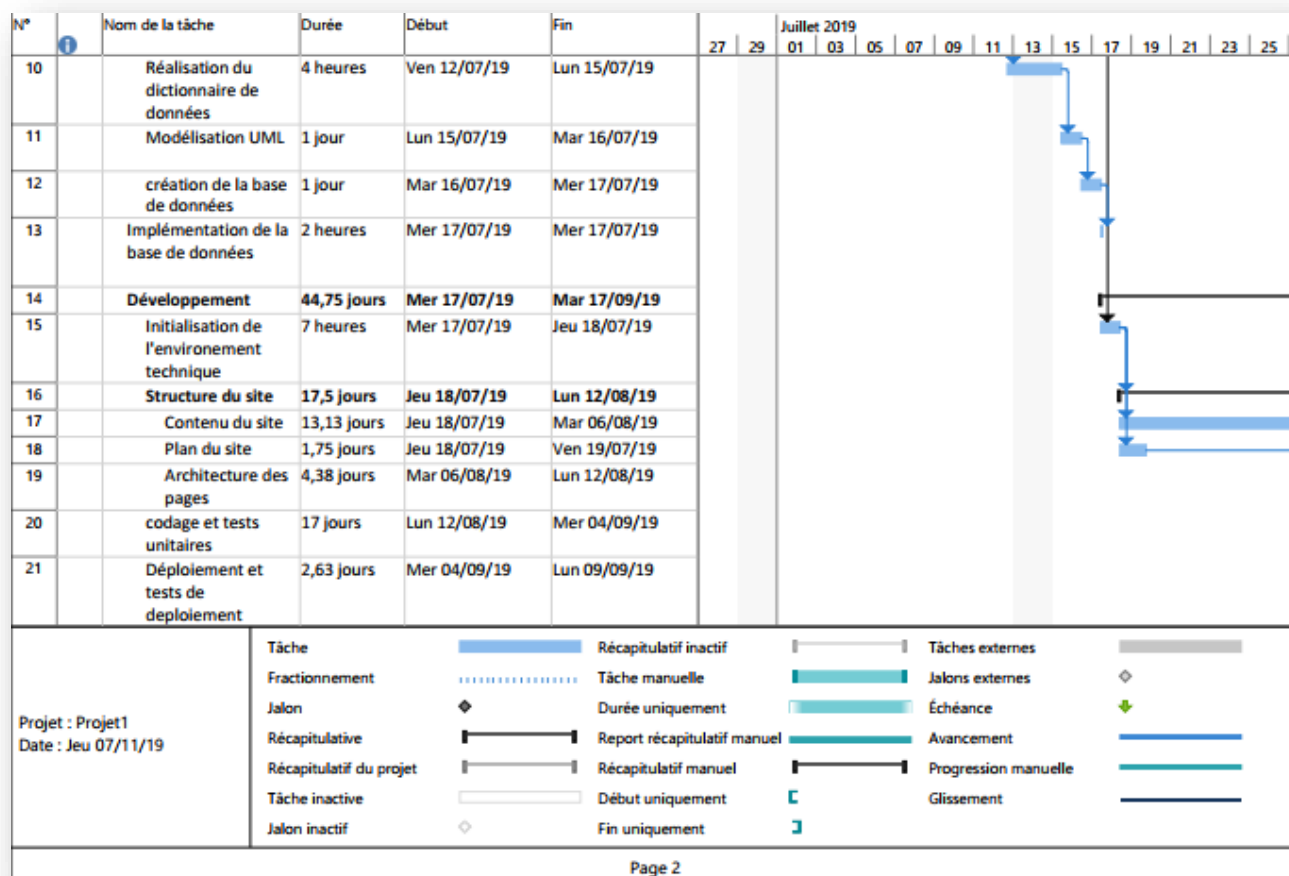


Figure 7 : Planning réel et écarts avec le planning initial - Page2

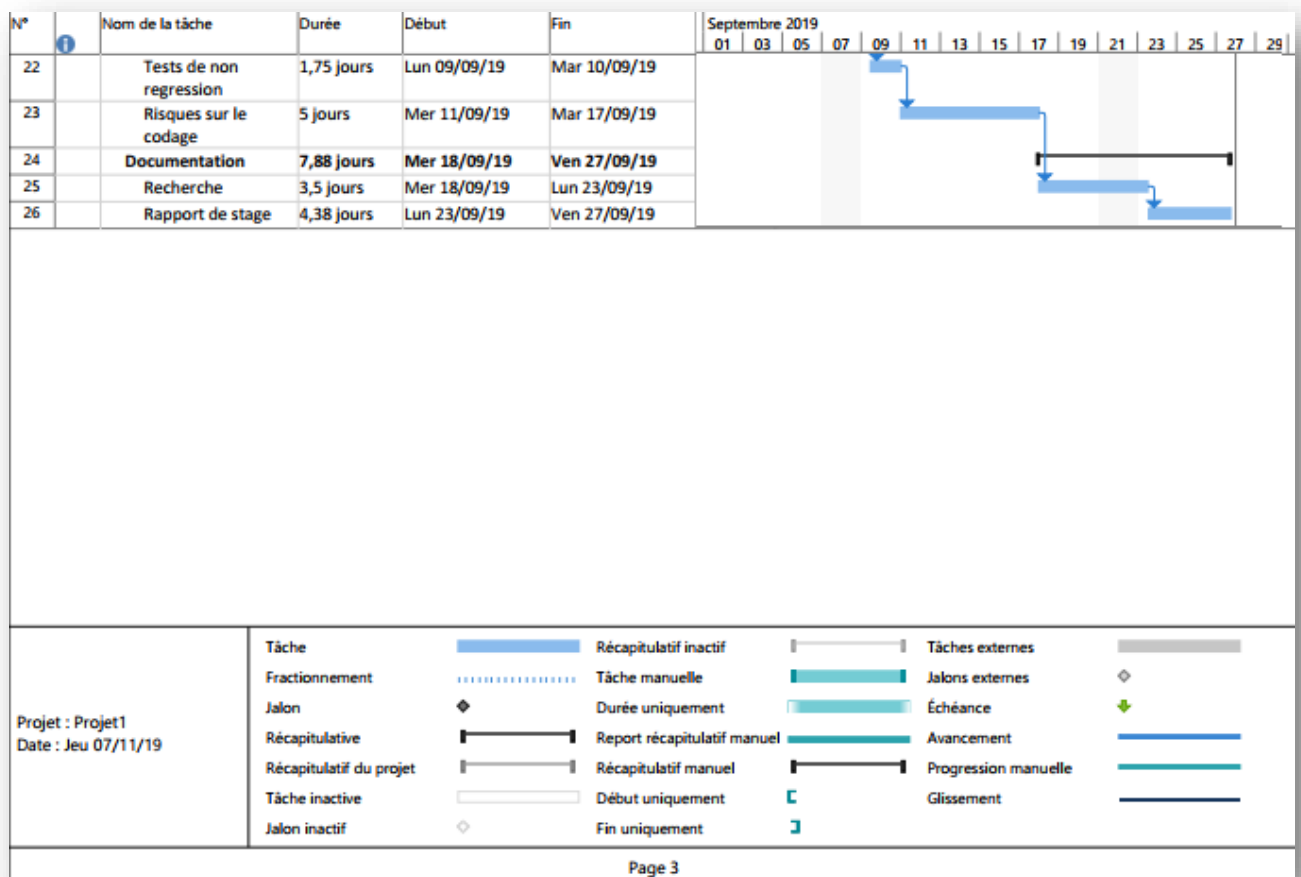


Figure 8 : Planning réel et écarts avec le planning initial - Page3

Dans le premier point, nous avons été retardés par la rédaction du document de conception. Et dans le second point, nous constatons que le risque d'itération prévu a été amorcé. En effet, la partie réalisation a été retardée faisant un effet boule de neige sur les tâches suivantes.

2.4. Dictionnaire de données

Classe : Etudiant		
Attribut		
Nom	Description	Type
_id	ID de l'étudiant	String
cin	Numéro d'identification nationale	Int
lastName	Nom de l'étudiant	String
firstName	Prénom de l'étudiant	String
sexe	Sexe de l'étudiant	String
specialty	Spécialité de l'étudiant	String
numberPhone	Numéro de téléphone de l'étudiant	String

Tableau 1: Classe Etudiant

Classe : Compte		
Attribut		
Nom	Description	Type
_id	ID du compte	String
email	Adresse mail du compte	String
login	Nom d'utilisateur du compte	String
password	Mot de passe du compte	String
role	Rôle du compte	String

Tableau 2: Classe Compte

Classe : Retard		
Attribut		
Nom	Description	Type
_id	ID du retard	String
matter	Matière du retard	String
durationInMin	Durée en minute du retard	Int
date	Date du retard	Date
session	La séance du retard	Int
reasonDelay	La raison du retard	String

Tableau 3: Classe Retard

Classe : Note		
Attribut		
Nom	Description	Type
_id	ID de la note	String
module	Module de la note	String
matter	Matière de la note	String
coefficient	Coefficient de la note	Int

Tableau 4: Classe Note

Classe : Paiement		
Attribut		
Nom	Description	Type
_id	ID du paiement	String
amountPaid	Le montant payé	Réel
paymentDate	Date du paiement	Date
modePayment	Mode de paiement	String

Tableau 5: Classe Paiement

Classe : Classe		
Attribut		
Nom	Description	Type
_id	ID de la classe	String
nameClass	Nom de la classe	String

Tableau 6: Classe Classe

Classe : Absence		
Attribut		
Nom	Description	Type
_id	ID d'absence	String
session	La séance de l'absence	Int
date	Date de l'absence	Date
matter	Matière de l'absence	String
justification	Justification de l'absence	String

Tableau 7: Classe Absence

2.5. Les livrables

Le tableau ci-dessous représente les livrables demandés, les dates de livraison prévisionnelle et réelle ainsi que le responsable de leur validation.

Phase	Livrable	Responsable de validation		Livraison prévue	Livraison réelle	Validation prévue
Etude Préalable	Cahier des charges	Mr. BENTALEB		12/07/19	12/07/19	12/07/19
Etude Préalable	Planning projet	Mr. BENTALEB		15/07/19	15/07/19	15/07/19
Conception	Applications	Mr. BENTALEB		-	-	16/07/19
Réalisation	Création d'une plate-forme pour les étudiants	Mr. BENTALEB		27/08/19	27/08/19	27/08/19
Documentation	Rapport (bilan du projet)	Mr. BENTALEB		27/08/19	27/08/19	27/08/19
Documentation	Manuel d'utilisateur	Mr. BENTALEB		-	-	-

Tableau 8: Les livrables

2.6. Les risques

Le tableau suivant présente les risques potentiels encourus durant le stage.











Risques	Type	Impact	Probabilité	Poids	Actions correctives	Actions Préventives
Technologies non maîtrisées	Organisationnel			4	Se documenter.	Se former aux outils.
Non-respect des délais	Organisationnel			8	Augmenter les heures de travail (heures supplémentaires).	Prévoir un éventuel retard en considération lors de la planification du projet.
Perte de données	Crash			16	Utiliser la dernière sauvegarde opérationnelle.	Utilisation d'outils d'intégration et de serveur de secours. Dupliquer les sauvegardes.
Panne Matérielle	Crash			4	Utiliser le matériel de secours	Prévoir un matériel de secours
Sujet incompris	Projet			4	Faire valider les livrables par l'encadrant et/ou le client.	Livrer des documents et des interfaces pour validation à chaque phase du projet.

Tableau 9: Les risques du stage

Légende :



: 1 Bas



: 2 Moyen



: 3 Elevé



: 4 Grave

_ L'impact et la probabilité vont de 1 à 4 tel que 4 est le niveau le plus dangereux.

_ Poids = Impact x Probabilité.

L'identification, la planification et la démarche ainsi réalisées ; nous pouvons établir le cahier des charges.

3. Analyse et conception

Dans cette partie, nous présentons l'analyse et conception de notre projet ainsi que l'architecture technique utilisée et pour finir les compétences acquises lors de ce stage.

3.1. Identification des acteurs

3.1.1. Définition d'un acteur

Un acteur est un utilisateur qui communique avec le système (opérateur, autre système...) et en réponse le système lui fournit le service qui correspond à son besoin.

3.1.2. Acteurs

Les acteurs qui interagissent avec l'application web sont :

Administrateur	<p>L'administrateur joue un rôle primordial et fondamental, c'est la seule personne qui dispose du droit d'exécuter les différentes tâches offertes par notre application. En effet, il peut saisir des informations en cas de nécessité et il peut aussi soit les mettre à jour soit les rechercher ou les supprimer.</p> <p>C'est une personne qui se connecte à notre application pour gérer les services suivants : Gestion des retards, Gestion des absences, Gestion des notes, Gestion des règlements.</p>
Etudiant	<p>Acteur principal, l'étudiant n'a que la possibilité de consulter ses informations relatives à la vie estudiantine au sein de l'école.</p> <p>C'est une personne qui se connecte à l'application pour consulter les informations suivantes : Suivi des retards, Suivi des absences, Suivi des notes, Suivi des règlements.</p>

Tableau 10 : Acteurs

3.2. Partie UML

3.2.1. Diagramme de classe

Il représente la structure statique d'un système. Il contient principalement les classes ainsi que leurs associations mais on peut aussi y trouver des objets. En pratique, l'intérêt majeur du diagramme de classes est de modéliser les entités du système d'information.

Nous décrivons dans cette partie la structure de notre base de données à travers le diagramme de classe suivant :

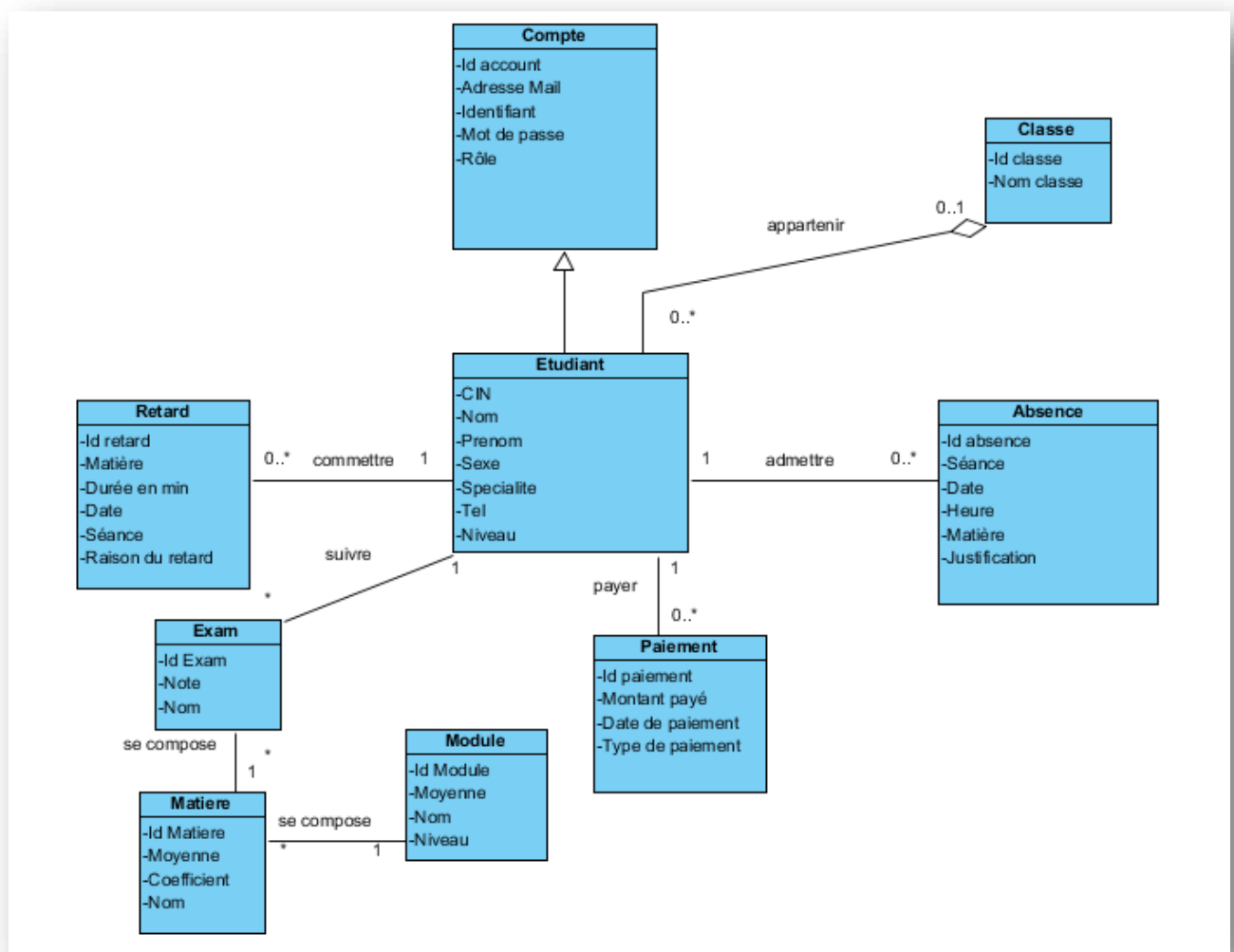


Figure 9: Diagramme de classe

3.2.2. Diagramme de cas d'utilisation

Le diagramme des cas d'utilisation représente les relations entre les acteurs et les fonctionnalités du système. Le diagramme des cas d'utilisation montre l'ensemble des processus du domaine d'étude. Chaque processus, ou plus précisément, chaque variante de processus, sera modélisé au moyen d'un diagramme de séquence et/ou d'un diagramme d'états-transitions et/ou d'un diagramme d'activités.

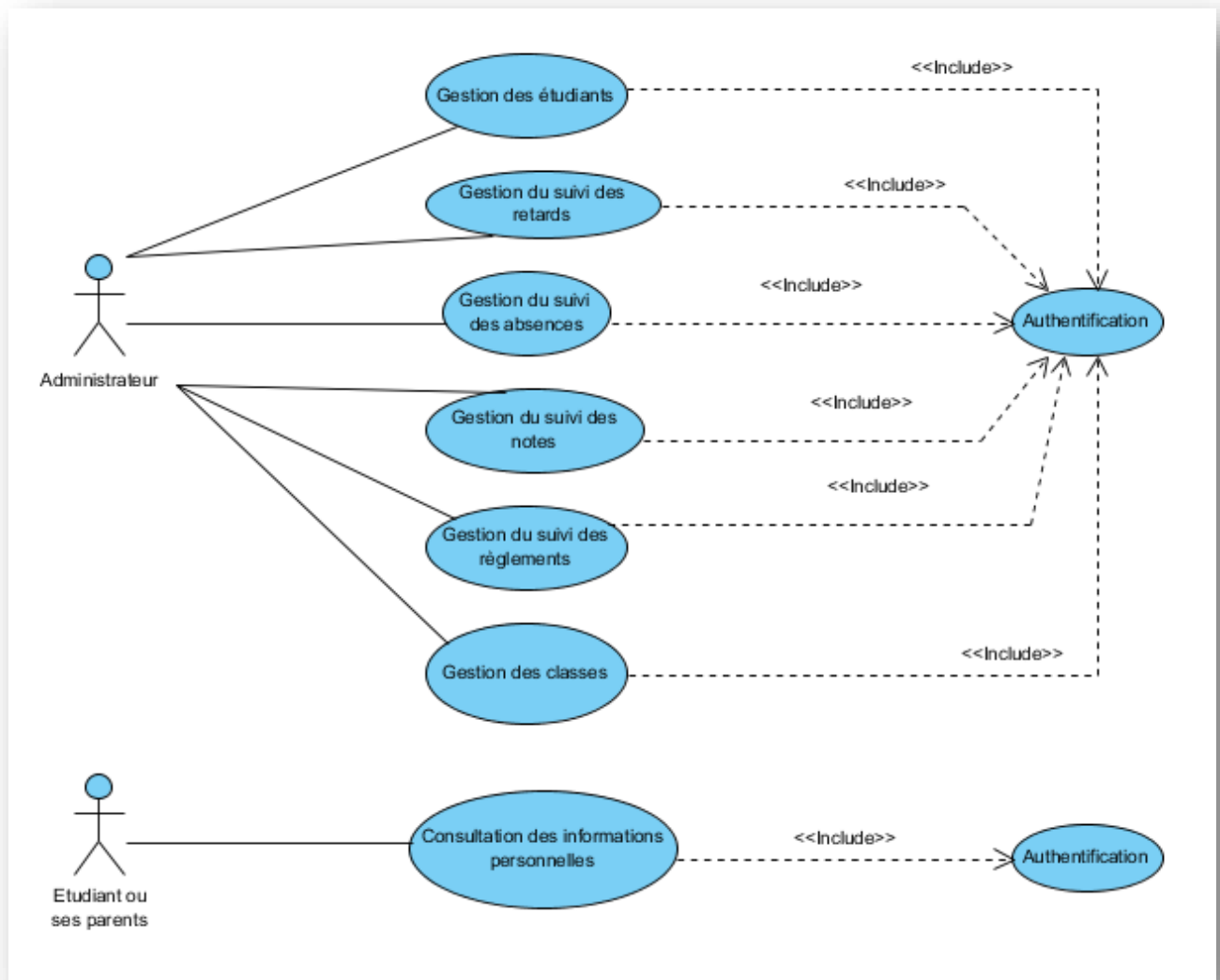


Figure 10: Diagramme de cas d'utilisation

3.2.3. Diagramme de séquence

C'est une variante du diagramme de collaboration. Il permet de mieux visualiser la séquence des messages en mettant l'accent sur les aspects temporels. Le diagramme de séquence représente la succession chronologique des opérations effectuées par un acteur pour la réalisation d'un cas d'utilisation, il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. Il est à invoquer que toutes les opérations mentionnées dans les différents diagrammes de cas d'utilisation se ressemblent, dans ce sens, on présentera les diagrammes de séquences de chaque opération pour une meilleure compréhension.

3.2.3.1. Diagramme de séquence pour un scénario d'authentification

Le diagramme, exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération d'authentification.

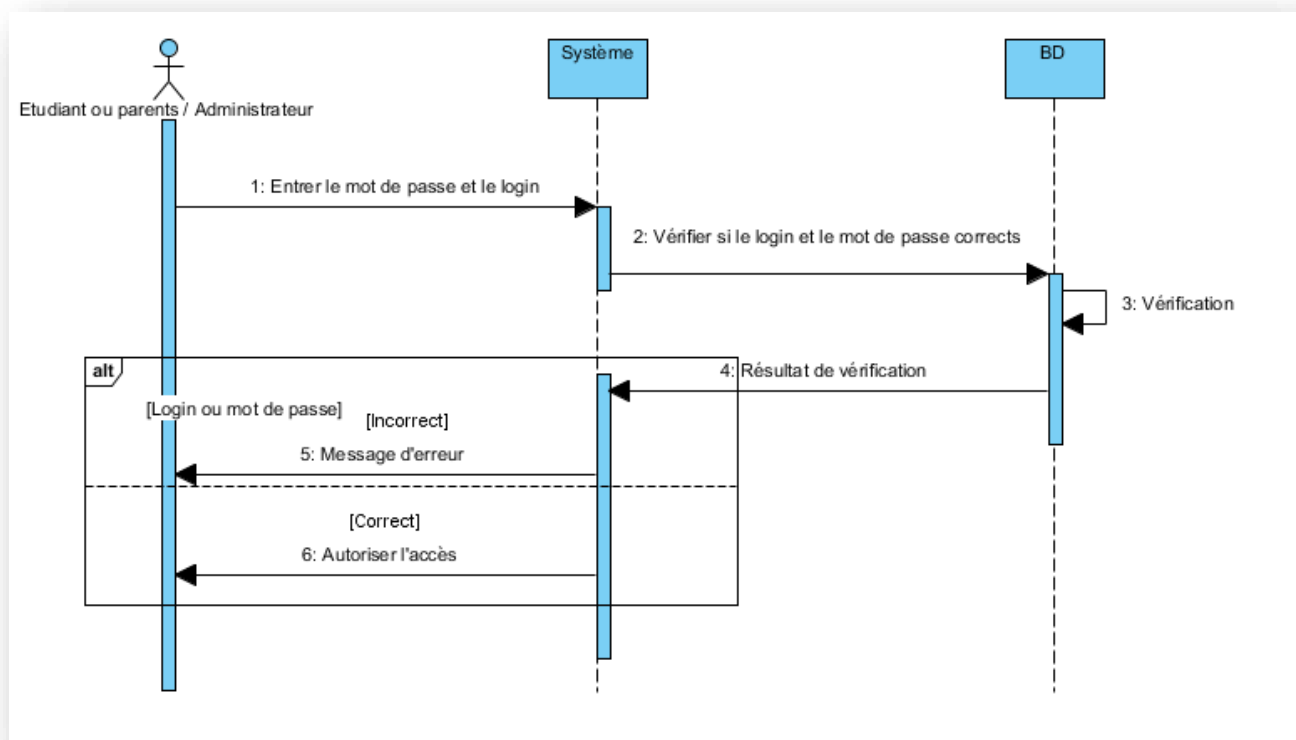


Figure 11: Diagramme de séquence pour un scénario d'authentification

3.2.3.2. Diagramme de séquence pour un scénario d'ajout

Le diagramme, exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération d'ajout.

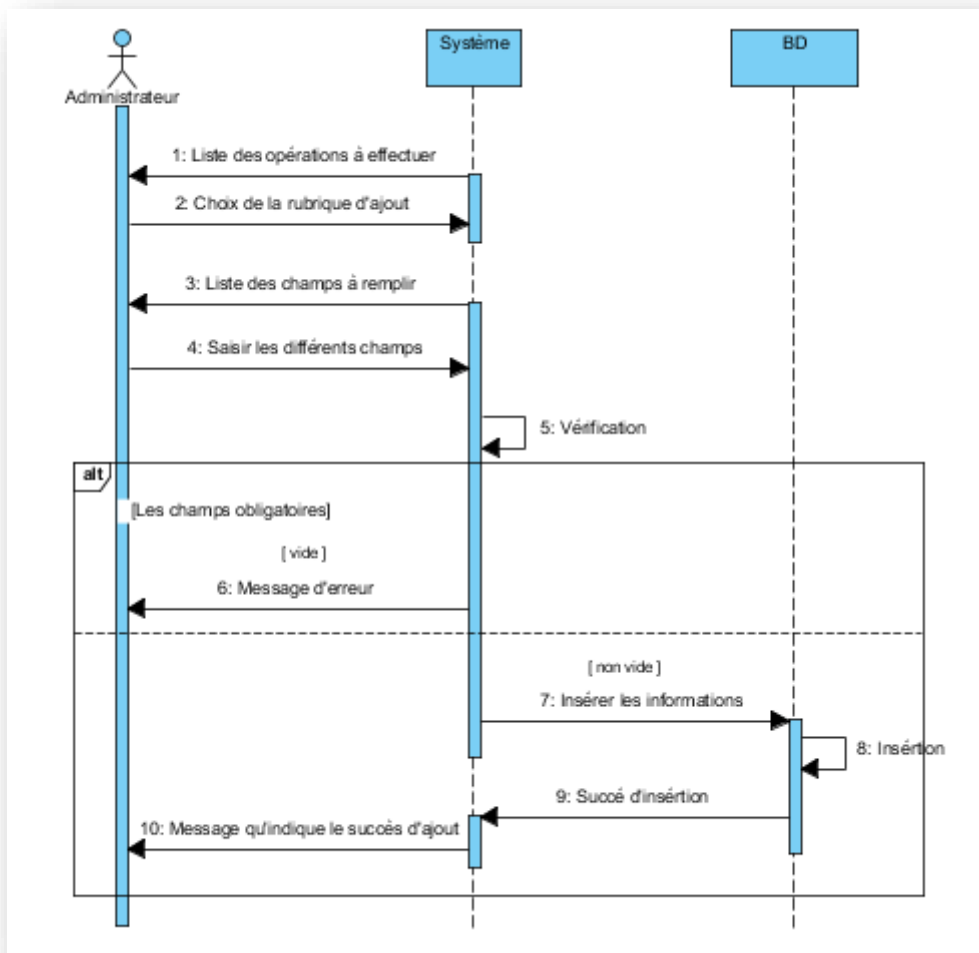


Figure 12: Diagramme de séquence pour un scénario d'ajout

3.2.3.3. Diagramme de séquence pour un scénario de suppression

Le diagramme, exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération de suppression.

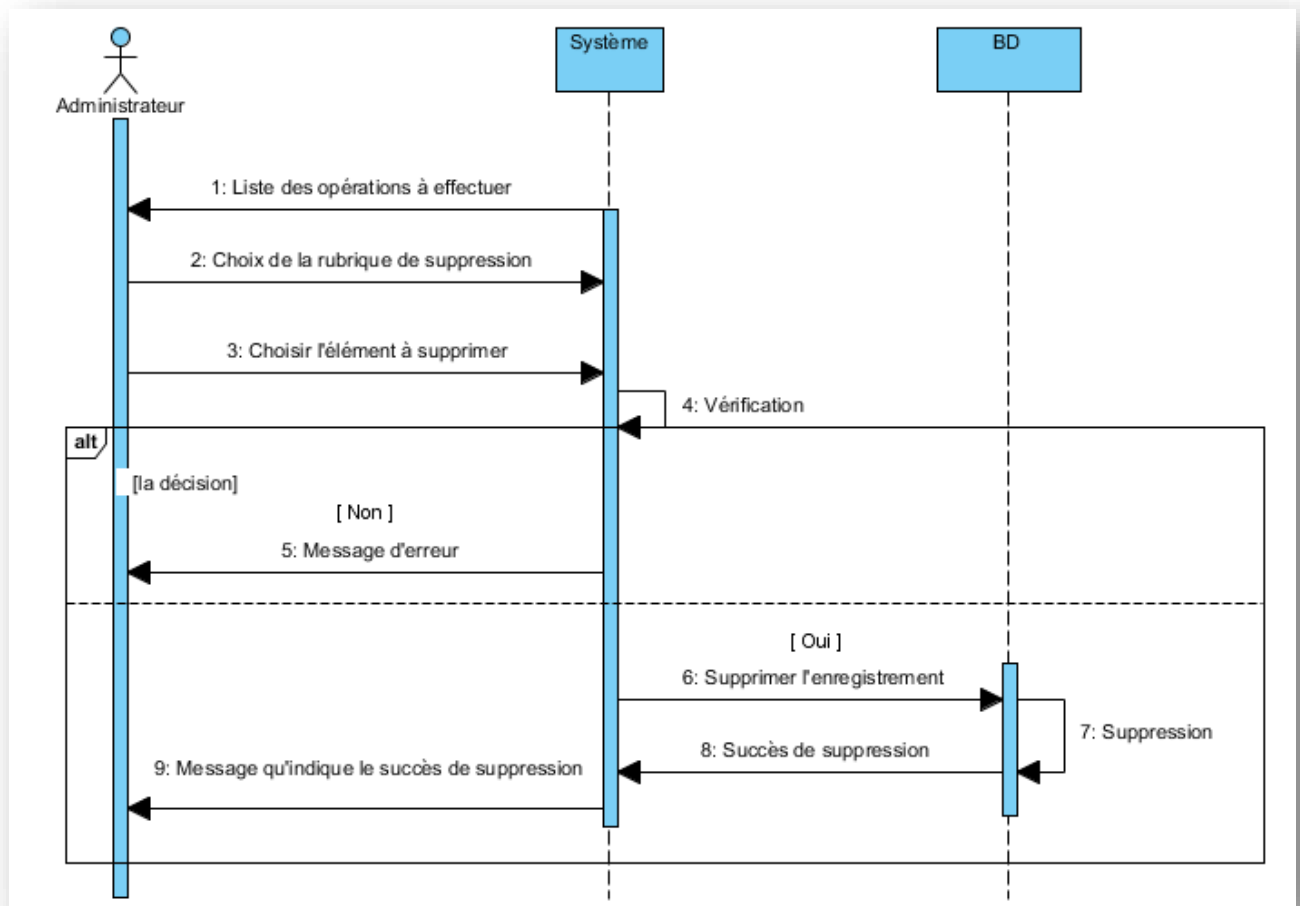


Figure 13: Diagramme de séquence pour un scénario de suppression

3.2.3.4. Diagramme de séquence pour un scénario de modification

Le diagramme, exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération de modification.

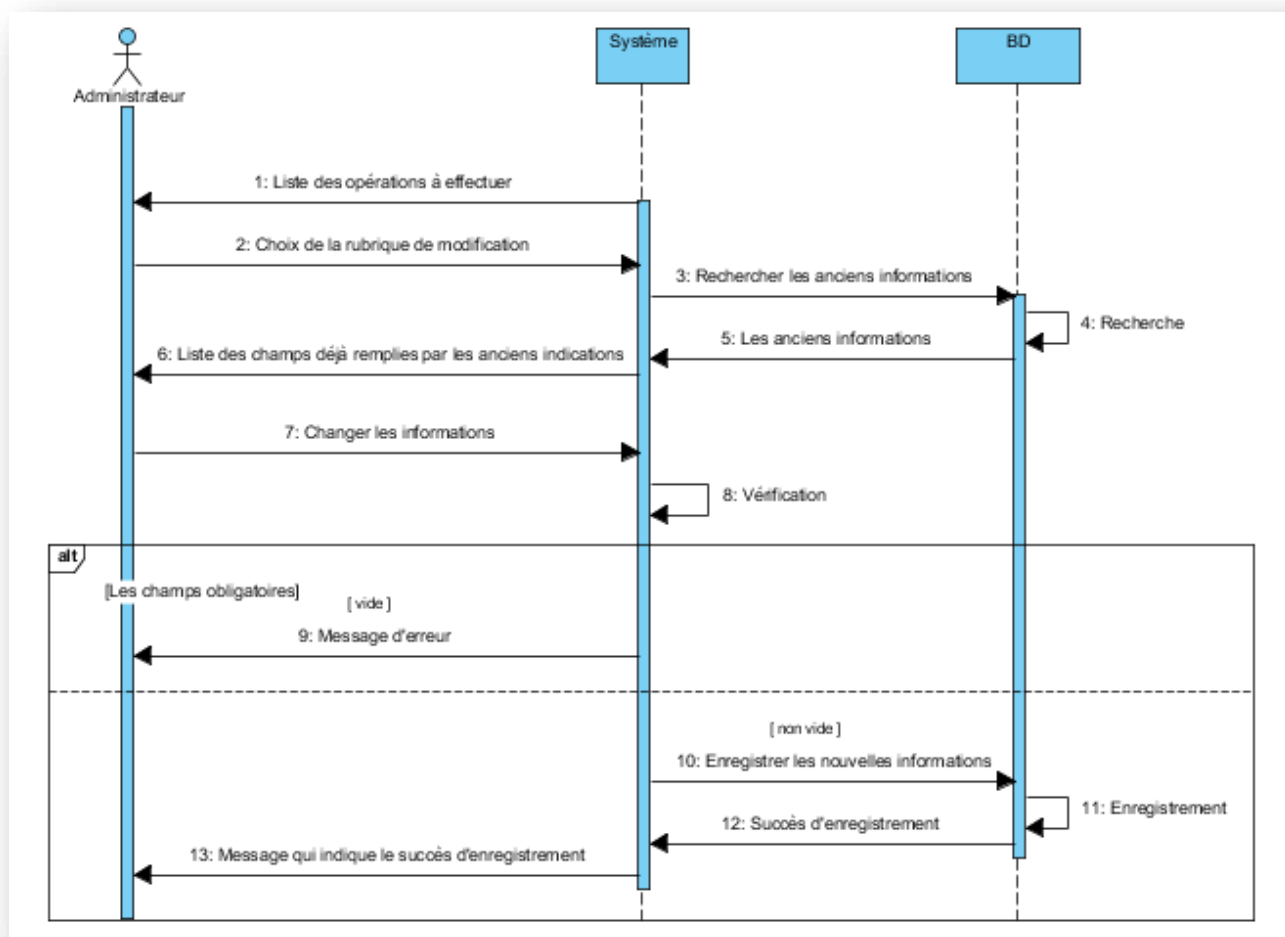


Figure 14: Diagramme de séquence pour un scénario de modification

3.2.3.5. Diagramme de séquence pour un scénario de recherche

Le diagramme, exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération de recherche.

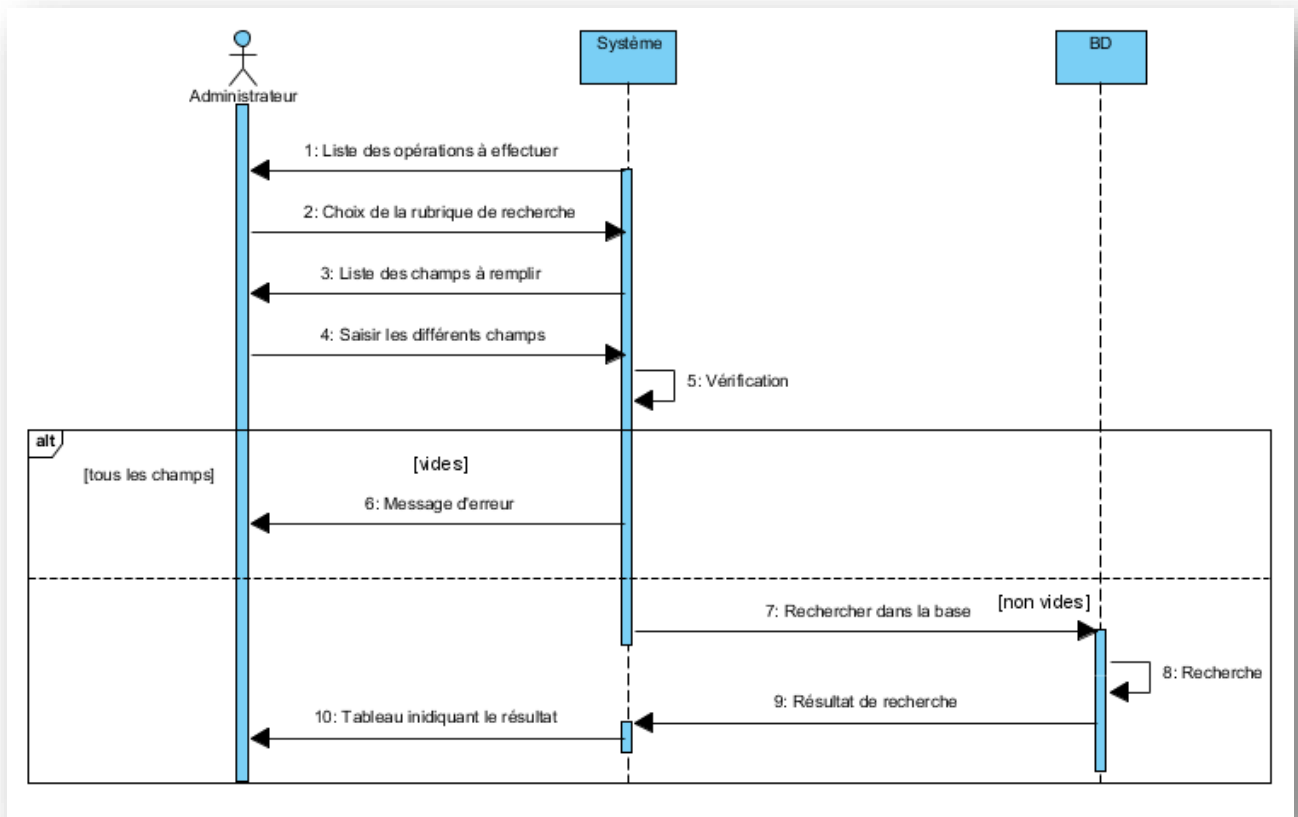


Figure 15: Diagramme de séquence pour un scénario de recherche

3.3. Architecture technique

La figure suivante présente les différents composants de notre architecture technique.

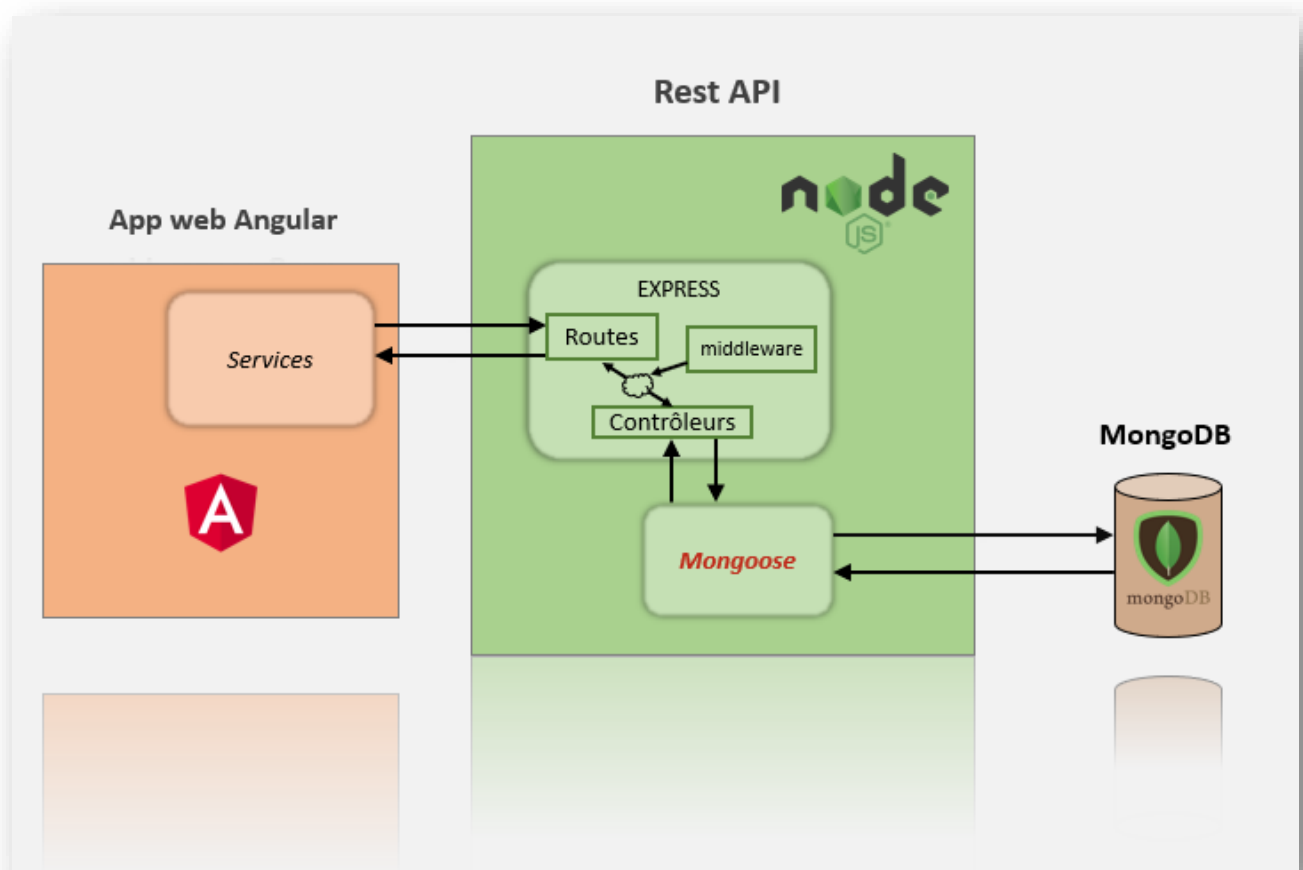


Figure 16: Architecture technique

Notre application web dispose d'une interface de programmation applicative « API » qui est la partie du programme qu'on expose officiellement au monde extérieur pour manipuler celui-ci.

Cette dernière permet d'entrer des données et de les récupérer la sortie d'un traitement. Initialement, une API regroupe un ensemble de fonctions ou méthodes, leurs signatures et ordre d'usage pour obtenir un résultat.

Nous avons choisi pour notre API le type d'architecture Transfert d'état représentationnel « REST » qui donne la possibilité de mettre en place une méthode vraiment simple afin d'accéder à des services web.

L'un des plus grands avantages de notre « REST API » est le fait que nos services web peuvent être utilisés par n'importe quel genre d'application front (web, mobile, bureau ...).

Dans notre cas nous disposons d'une application web codée avec le Framework Angular qui communique avec notre API en demandant un service web. La demande est interceptée par les routes de Express.js qui vont la renvoyer vers le bon contrôleur en étant validée par notre Middleware.

Le contrôleur gère les modèles Mongoose aux dépens de la requête qui accède directement à notre base de données MongoDB, ensuite l'opération se fait en inverse.

3.4. Conception

_ Réalisation du design :

D'abord, nous avons commencé la conception par choisir le design général du site web dont vont hériter toutes les interfaces.

Nous avons alors opté pour un design assez moderne et basic pour Bootstrap 4 et Angular, créé avec angular-cli :

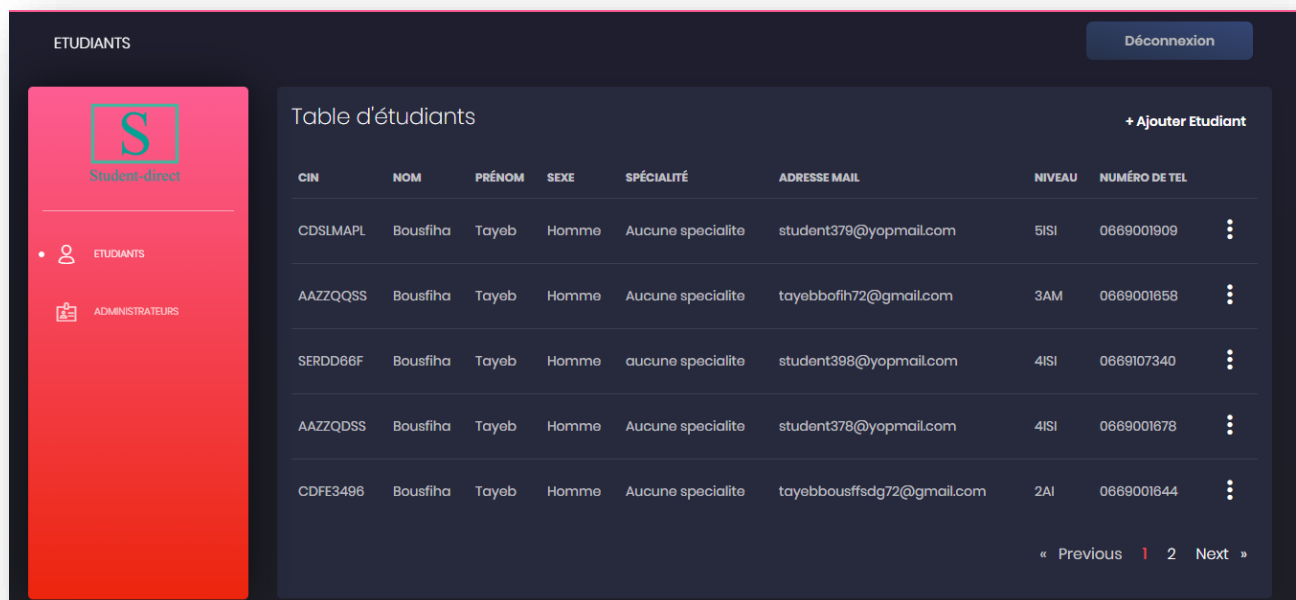


Figure 17: Design général

_ Réalisation des interfaces :

La réalisation des interfaces a été faite en plusieurs étapes, premièrement nous avons commencé par les interfaces de connexion qui vont permettre l'identification ou la réinitialisation du mot de passe au cas où il aurait été oublié, ensuite l'interface des étudiants où il sera affiché tous les étudiants ainsi qu'une autre interface pour l'affichage détaillé de chaque étudiant (Suivi des retards, Suivi des absences, Suivi des notes, Suivi des règlements) et pour finir l'interface des administrateurs où il sera affiché tous les administrateurs.

_ Développement :

Notre application composé d'un côté FRONTEND et BACKEND, l'aboutissement des interfaces représente pour nous un bon avancement dans la partie FRONTEND, après il a fallu développer l'API qui représente la partie BACKEND de notre application.

Nous avons choisi pour notre API le type d'architecture Transfert d'état représentationnel « REST » qui donne la possibilité de mettre en place une méthode vraiment simple afin d'accéder à des services web.

Pour cela nous avons choisi la base de données MongoDB, l'environnement Node.js et le framework Express.js, nous avons ensuite lancé un serveur sur node js, connecté la base de données mongoDB, créer les models, les routes et les contrôleurs pour retourner le résultat souhaité.

Pour chaque CRUD ou autre opération nous avons utilisé PostMan qui est une bonne plateforme pour le développement des API, nous avons aussi créé un middleware pour valider les données pour une plus grande sécurité.

Ensuite pour le FRONTEND nous avons choisi Angular comme framework coté client, pour commencer nous avons établi les modèles et services pour nos CRUD ensuite les composants pour remplir nos interfaces avec les réponses provenant de notre API.

Pour la sécurité de notre application nous avons choisi une des meilleures « JSON Web Token » qui est un standard qui permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. Elle s'effectue par l'algorithme HMAC ou RSA.

Lors de la connexion d'un utilisateur les identifiants sont envoyé dans une requête http du FRONTEND (Angular) vers l'API, le contrôleur qui recevra les identifiants va alors vérifier s'ils sont corrects pour ensuite attribuer un token au stockage local qui est signé avec un code secret dans l'environnement de l'API.

Pour une plus grande sécurité, à chaque requête http un service qui s'occupe de la verification du token est appelé, pour intercepter chaque requête émise de notre FRONTEND nous avons créé un intercepteur de requête http :

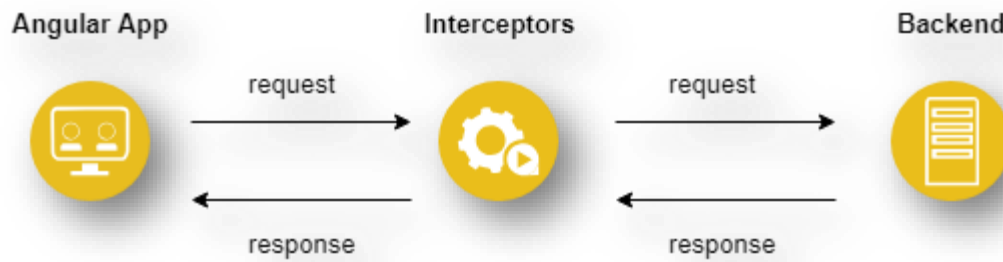


Figure 18 : Intercepteur de requête http

À chaque requête, l'intercepteur intercepte la requête et lui insère le token pour la renvoyer à l'API (BACKEND) là où le token sera vérifié au niveau du middleware comme ça aucune requête ne passera si l'utilisateur ne dispose pas du bon token.

Chaque token dispose d'une durée avant d'expirer, pour notre cas nous avons choisi une durée de 9 secondes une assez courte durée pour une sécurité maximale mais nous avons prévu le cas où le token expire il est alors rafraîchi par un service qui détecte si le token a expiré et le rafraîchi ensuite après vérification de la signature bien sûr.

Lors de l'authentification si l'utilisateur oublie son mot de passe, nous lui avons permis de le réinitialiser, l'utilisateur devra fournir son adresse email valide qui sera vérifiée dans l'API, si elle existe dans la base de données un email lui sera renvoyé avec le lien de réinitialisation sécurisé avec un token à la fin du lien qui sera inséré dans l'attribut token de la collection du USER, lorsque l'utilisateur saisira son nouveau mot de passe le token à la fin du lien sera vérifié dans l'API si il se trouve dans la base de données alors le mot de passe sera réinitialisé par le nouveau entré et sera hashé car tous les mot de passes sont hashé dans la base de données sinon il sera refusé et un message d'erreur est retourné « lien invalide ».

Notre application se compose de 6 CRUD dont la plus part se compose de la partie affichage « READ » où les données sont affichées dans un tableau avec une pagination de 5 éléments par page, l'ajout « CREATE » l'utilisateur est redirigé vers un formulaire d'ajout, modifier « UPDATE » l'utilisateur est redirigé vers le même formulaire d'ajout mais cette fois-ci avec le titre modifier est les valeurs remplies déjà par l'entité qu'on souhaite modifier, supprimer « DELETE » l'utilisateur clique sur le bouton supprimer qui va supprimer la donnée et rafraîchir l'affichage. Les options modifier et supprimer sont affichées dans un menu « three dot », menu trois points :

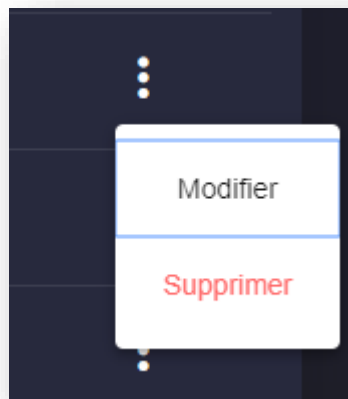


Figure 19 : "three dot menu"

Sauf pour le cas du CRUD des étudiants où on trouve le bouton afficher qui renvoie vers la page qui contient tous les détails de l'étudiant choisi (Suivi des retards, Suivi des absences, Suivi des notes, Suivi des règlements).

Lors de la création d'un étudiant l'API envoie un email à l'adresse mail saisie contenant le nom d'utilisateur et le mot de passe généré pour se connecter à l'application.

Lors de la création du CRUD de suivi des notes d'étudiants nous avons eu plusieurs relations entre la note et l'examen et la matière et le module et l'étudiant, ce qui nous a compliqué la tâche car nous utilisons une base de données NO SQL et dans une base de données NO SQL il n'y a pas de relation. Mais avec plusieurs recherches nous sommes tombés sur une méthode d'agrégation \$Lookup qui permet de relier deux collections mais d'une manière limitée, c'est seulement après une longue autoformation et recherche que nous avons pu relier en quelque sorte les 4 collections et finir le CRUD en créant notre propre requête d'agrégation :

```
const models = await moduleModel.aggregate([
  { $match: { level: student.level } },
  { "$unwind": "$matters" },
  {
    "$lookup": {
      "from": "exams",
      "as": "matters.exams",
      "let": { matterId: "$matters._id", studentId: student._id },
      "pipeline": [{
        "$match": {
          $expr: {
            $and: [
              { $eq: ["$studentId", "$$studentId"] },
              { $eq: ["$matterId", "$$matterId"] }
            ]
          }
          // { studentId: { $ifNull: { $and: [ { $eq: [ "$studentId", "$$studentId"
          //   , $eq: [ "$matterId", "$$matterId" ] } } ],
          // { $and: [ { $eq: [ "$studentId", "$$studentId" ] }, { $eq: [ "$matterId
```

Figure 20 : methode \$Lookup aggregation

Nous avons commencé dans notre requête d'agrégation par relier le niveau avec \$match ensuite d'unifier la collection de matière pour pouvoir utiliser \$lookup qui va nous permettre de la relier avec l'étudiant et l'examen, normalement \$lookup ne permet de relier que deux collections mais grâce à pipeline nous avons pu utiliser \$match pour relier et \$expr qui nous a permis d'utiliser \$and pour établir 2 conditions d'égalités.

3.5. Enseignements/apports du stage (connaissances - compétences)

Ce stage nous a apporté plusieurs compétences :

- _ Réalisation d'une API REST FULL.
- _ Coder en javascript, ECMAScript, TypeScript.
- _ Création d'un FRONTEND avec Angular.
- _ Maîtrise de la pile de logiciel MEAN STACK.
- _ Création d'une requête d'agrégation de mongoDB.
- _ Maîtrise de la communication de requêtes http entre Angular et une API REST.
- _ La sécurité avec JWT.
- _ Fixer des erreurs de codage.
- _ L'autoformation.
- _ La recherche de solution sur internet.
- _ La persévérance.

4. Réalisation

Dans ce chapitre, nous présentons les outils utilisés ainsi que quelques interfaces.

4.1. Outils utilisés

_ **MongoDB** : est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++.

_ **Express.js** : est un Framework pour construire des applications web basées sur Node.js. C'est de fait le Framework standard pour le développement de serveur en Node.js.

_ **Angular** : est un cadriciel côté client open source basé sur TypeScript dirigée par l'équipe du projet Angular à Google et par une communauté de particuliers et de sociétés.

_ **Node.js** : est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.

_ **JavaScript** : est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation de Node.js.

_ **TypeScript** : est un langage de programmation libre et open source développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript.

_ **ECMAScript** : est un ensemble de normes concernant les langages de programmation de type script et standardisées par Ecma International dans le cadre de la spécification ECMA-262.

_ **JavaScript Object Notation** : est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple.

_ **L'HyperText Markup Language** : généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS).

_ **Les feuilles de style en cascade** : généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World

Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

_ **Sass** : est un langage dynamique de génération de feuilles de style en cascade initialement développé par Hampton Catlin et Nathalie Weizenbaum. Sass est un préprocesseur CSS. C'est un langage de description compilé en CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass.

_ **Bootstrap** : est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

_ **Swagger** : est une infrastructure logicielle open-source reposant sur un vaste écosystème d'outils qui aide les développeurs à concevoir, créer, documenter et utiliser des services Web RESTful.

_ **Postman, Inc.** : est une bonne plateforme pour le développement des API, Il permet de tester les requêtes http, d'afficher le statut code et la réponse, d'envoyer une requête avec des données et beaucoup de fonctionnalités pour un bon développement d'API.

_ **Robo 3T** : est un outil open source MongoDB qui fournit des fonctionnalités innovantes pour répondre aux besoins des amateurs et de la communauté au sens large.

_ **Angular Material** : est une infrastructure de composants d'interface utilisateur et composants de conception de matériaux pour applications Web Angular mobiles et de bureau.

_ **JSON Web Token (JWT)** : est un standard qui permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. Elle s'effectue par l'algorithme HMAC ou RSA.

4.2. Imprimé écran

_ Gestion des étudiants :

Les figures ci-dessous montrent les pages dédiées à la gestion des étudiants :

CIN	NOM	PRÉNOM	SEXE	SPÉCIALITÉ	ADRESSE MAIL	NIVEAU	NUMÉRO DE TEL
CDSLMAPL	Bousfiha	Tayeb	Homme	Aucune specialite	student379@yopmail.com	5ISI	0669001909
AAZZQQSS	Bousfiha	Tayeb	Homme	Aucune specialite	tayebbofi72@gmail.com	3AM	0669001658
SERDD66F	Bousfiha	Tayeb	Homme	aucune specialite	student398@yopmail.com	4ISI	0669107340
AAZZQDSS	Bousfiha	Tayeb	Homme	Aucune specialite	student378@yopmail.com	4ISI	0669001678
CDFE3496	Bousfiha	Tayeb	Homme	Aucune specialite	tayebbousffsdg72@gmail.com	2AI	0669001644

Figure 21: Page de gestion des étudiants

DASHBOARD Notes Paiements Absences Retards + Ajouter module

Module : orienté object

MATIERE	EXAM	NOTE	COEF	MOY
c++	ghgc	10	1	7.5 +
	gvjg	5		
java	fxv	5	2	5 +
angular	hvj	9	3	9 +
Moyenne du module :		7.42		

Module : communication

MATIERE	EXAM	NOTE	COEF	MOY
SVT	EXS2	10		10 +
	EXS3	10	2	
	EXS4	10		
Math	ex1	14	5	14 +
Moyenne du module :		12.86		

Figure 22 : Page de gestion des suivis de chaque étudiant détaillée

Comme on peut constater, il existe deux pages pour la gestion des étudiants.

_ La première se compose d'un tableau où les étudiants y sont affichés par pagination de 5 étudiants maximum par page, d'un bouton Ajouter Etudiant qui nous renvoie vers un formulaire d'ajout, un « three dot » menu pour nous afficher les autres opérations possibles dont la modification qui renvoie ver le même formulaire d'ajout mais cette fois si avec le titre modifier est les valeurs remplies déjà par l'étudiant qu'on souhaite modifier, supprimer « DELETE » pour supprimer l'étudiant et rafraichir l'affichage, le bouton afficher qui permet d'afficher les informations de suivi de l'étudiant choisi en détail et qui renvoie vers la page de gestion d'étudiant choisi détaillé.

_ La deuxième se compose de 4 onglets où l'on peut naviguer comme on veut chaqu'un des onglets contient le CRUD complet du suivi approprié avec le même system de CRUD sauf pour le suivi des notes où l'affichage des notes n'est pas sous forme de tableau mais de CARD comme le montre la figure précédente, chaque CARD contient un module complet pour un effet visuel et une présentation meilleure.

Les CRUD de notre application ont tous le même procédé comme le montre l'exemple suivant du CRUD des étudiants :

_ Ajout d'un étudiant :

En cliquant sur le lien (Ajouter Etudiant) qui apparaît dans la figure 19, la page suivante s'affiche :

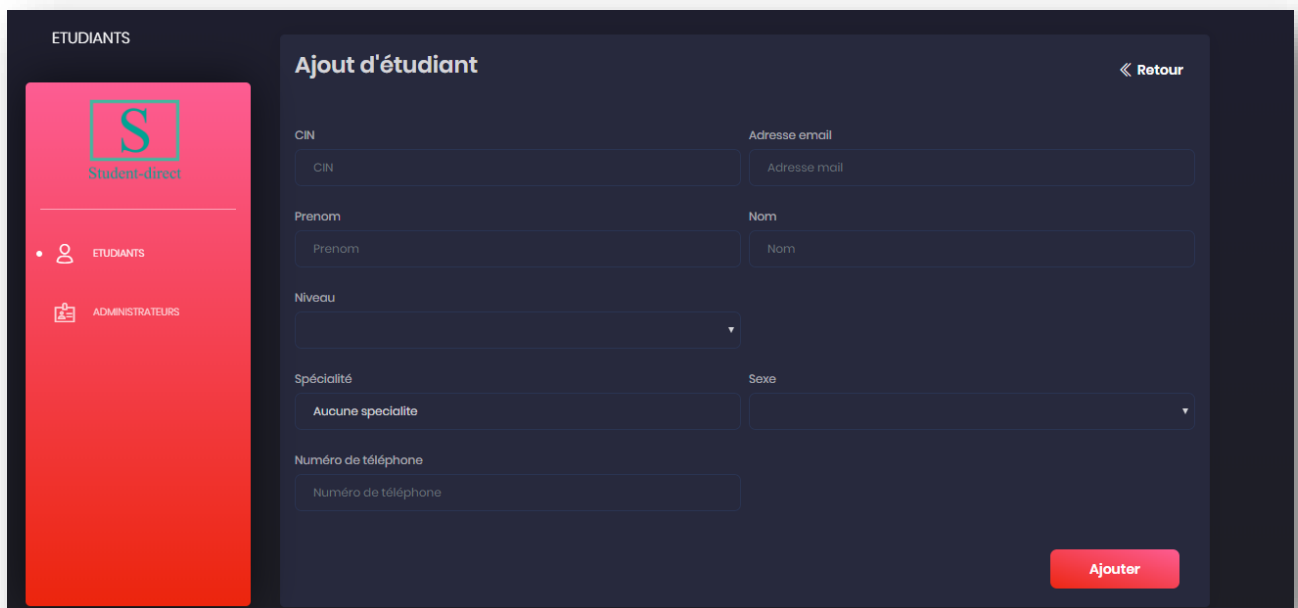


Figure 23 : Page d'ajout d'étudiant

Cette page permet à l'utilisateur de l'application de remplir les champs par les différentes informations concernant l'étudiant qu'on souhaite ajouter à la base de données, en cliquant sur le bouton « Ajouter », l'application vérifie d'abord si tous les champs sont bien remplis, s'il y'a des champs qui sont vides, l'application renvoie une erreur avertissant l'utilisateur qu'il existe un champ qu'il faut remplir, l'application vérifie aussi s'il existe déjà un patient dans la base de données qui a le même CIN saisi, si c'est le cas, une erreur est alors affichée donnant comme information que le CIN est dupliqué, même chose pour le numéro de tel et email, s'il n'existe aucune erreur, les informations saisies sont ajoutées à la base de données grâce à la fonction « save » intégrée dans « Node.js ».

_ Modification d'un client :

Quand l'utilisateur choisit de modifier un patient bien précis, la page suivante lui est affichée :

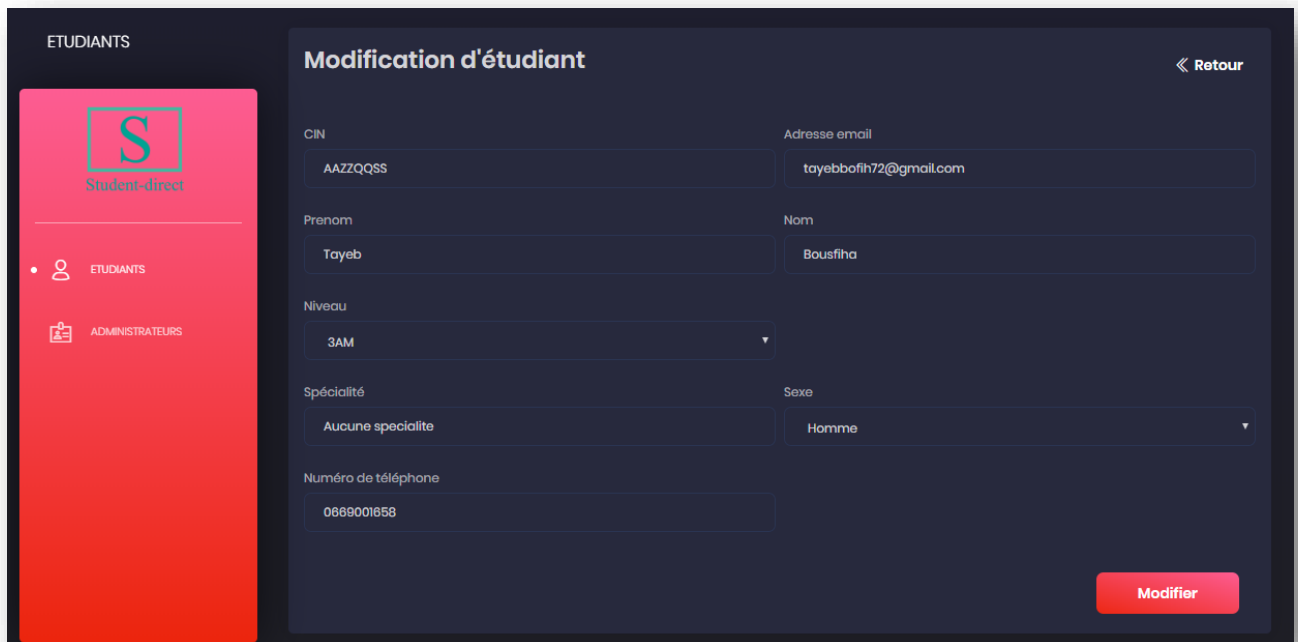


Figure 24 : Page de modification d'étudiant

Cette page récupère d'abord les informations de la ligne sélectionnée et les affiche dans le même formulaire d'ajout mais cette fois si avec le titre modifier, quand l'utilisateur termine les modifications nécessaires et clique sur le bouton « Modifier », l'application vérifie d'abord si tous les champs sont remplis et s'il n'existe pas un autre acquéreur avec le même numéro de CIN, si ces contraintes ou l'une d'entre elles n'est pas respectée.

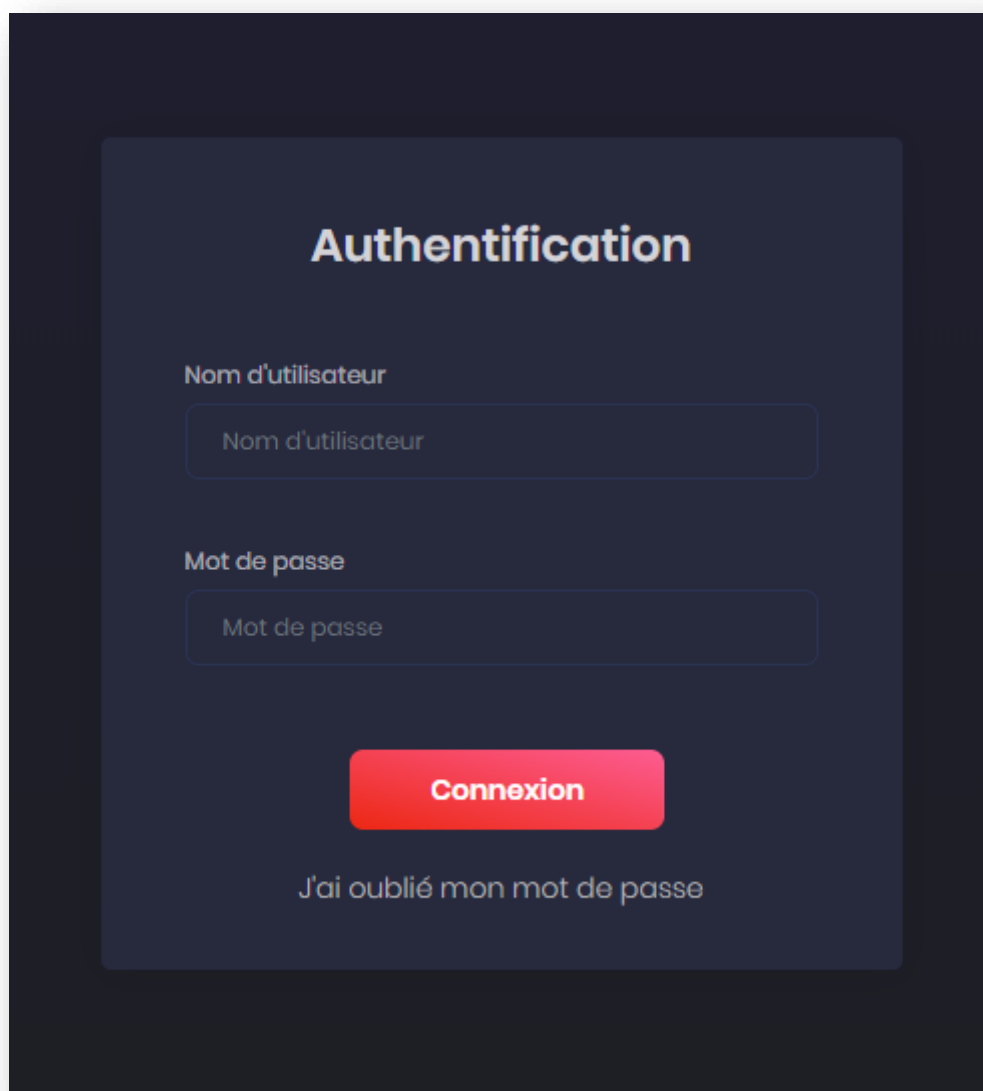
L'application avertit l'utilisateur en lui affichant les erreurs correspondant à la faute de remplissage commise, sinon, la fonction « Save » est exécutée, permettant cette fois-ci de lancer une requête SQL « UPDATE », ce qui amène à la mise à jour de la ligne souhaitée par les nouvelles informations.

Il reste à préciser que quand l'utilisateur choisit de supprimer un client, l'identifiant du client est récupéré pour qu'il soit ensuite affecté comme paramètre à la fonction

« DELETE », ce qui permet de lancer la requête « DELETE FROM », et supprimer la ligne souhaitée de la base de données.

_ Connexion :

Pour accéder à notre application l'utilisateur doit se connecter avec ses identifiants comme le montre la figure suivante :



The image shows a dark-themed authentication form. At the top, the word "Authentication" is written in a large, bold, white font. Below it, there are two input fields. The first is labeled "Nom d'utilisateur" in a smaller white font, and the second is labeled "Mot de passe". Both fields have a light blue border and contain placeholder text of the same color. Below the password field is a red button with the word "Connexion" in white. At the bottom of the form, there is a link that says "J'ai oublié mon mot de passe" in a light blue font.

Figure 25 : Page d'authentification

Si par malheur l'utilisateur oublie le mot de passe, notre application lui offre la possibilité de le réinitialiser en cliquant sur « J'ai oublié mon mot de passe » ce qui l'amènera vers la page suivante :

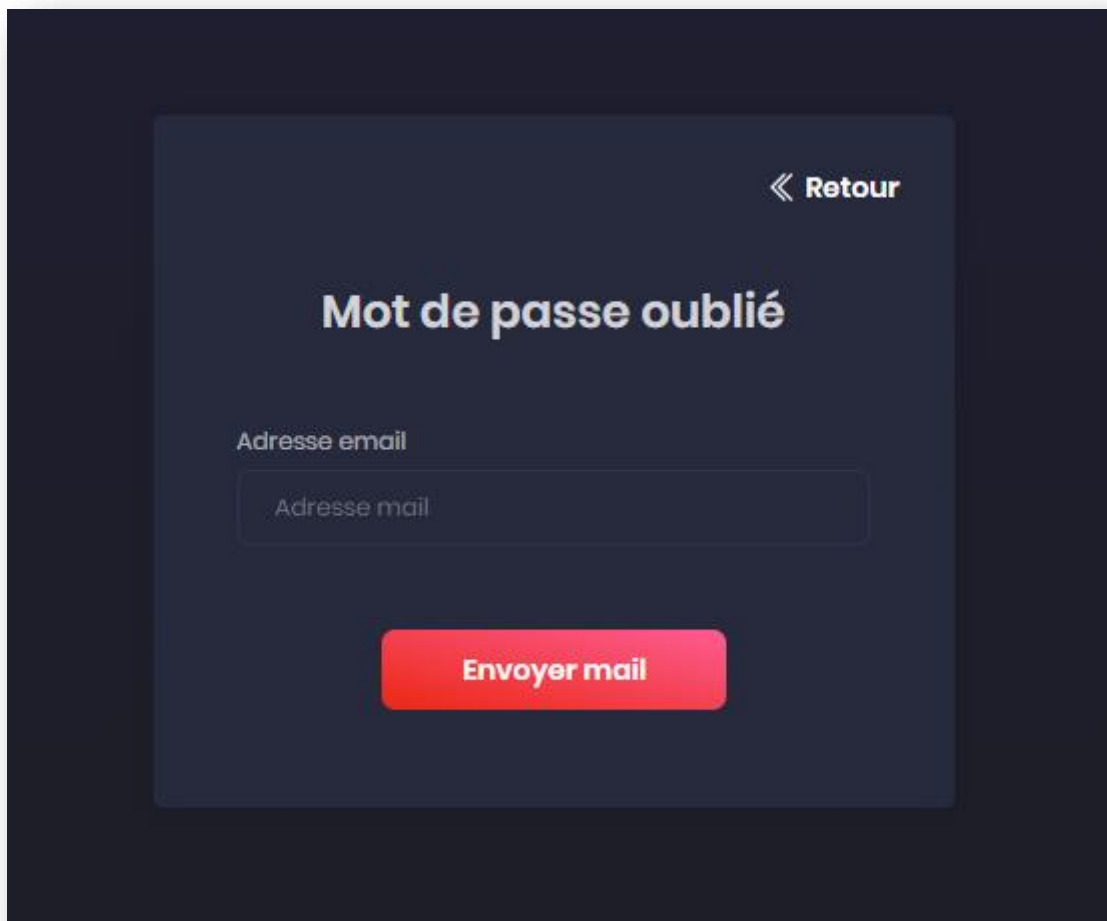
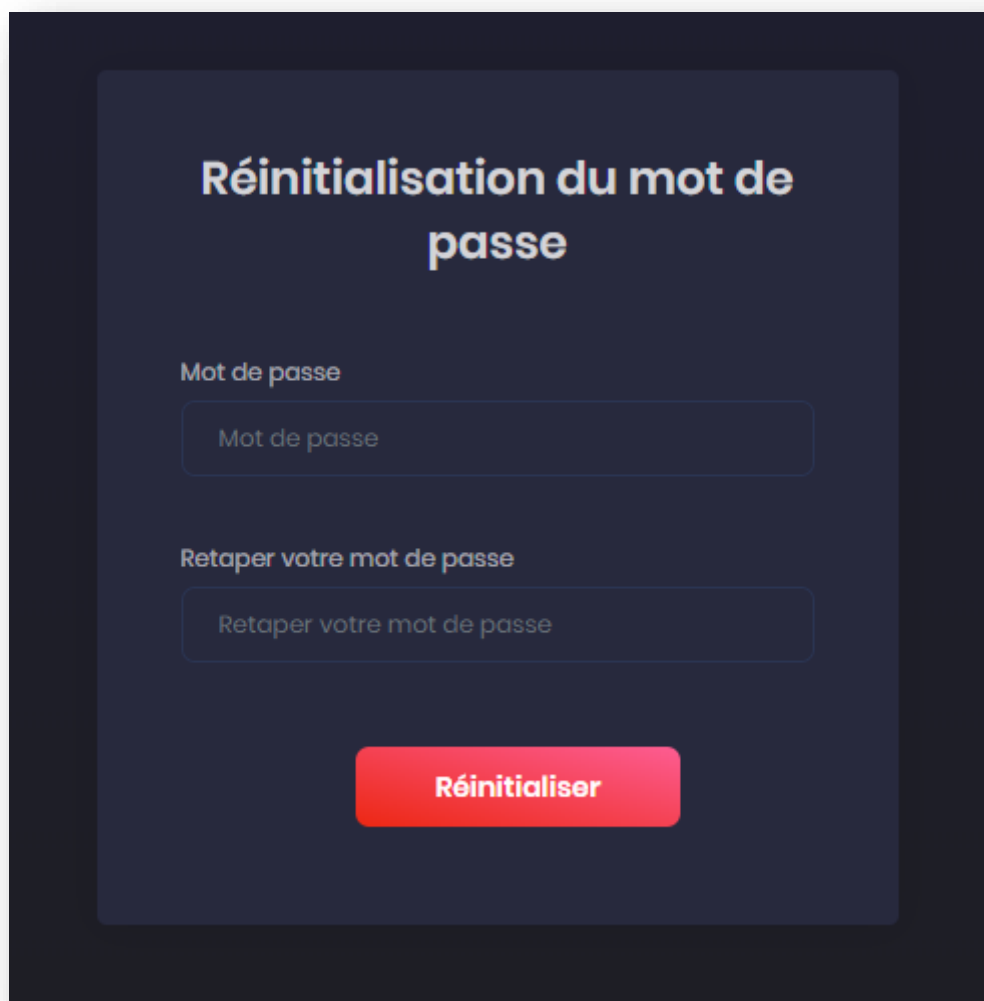


Figure 26 : Page d'envoi de mail de réinitialisation du mot de passe

Dans cette page l'utilisateur se doit d'introduire son email pour que l'API lui renvoie le lien de réinitialisation du mot de passe. Arrivé au lien de réinitialisation la page suivante s'affiche :



Réinitialisation du mot de passe

Mot de passe

Retaper votre mot de passe

Réinitialiser

Figure 27 : Page de réinitialisation de mot de passe

Il ne manquera plus que de remplir le nouveau mot de passe qui sera enregistré et hashé dans la base de données.

Conclusion et perspectives

A l'issue de ce stage, nous sommes arrivés à obtenir une vue générale sur les procédures pratiques de la création des applications web, services web, API et FRONTEND, commençant par le design, jusqu'au traitement javascript.

Lors de notre stage, nous avons été confrontés à plusieurs difficultés dans toutes les tâches que nous avons eu sans exception :

- _Difficulté dans les langages de codage Javascript, TypeScript, EcmaScript.

- _ Notre base de données n'était pas faite pour le cas des relations.

- _ Notre code n'était pas assez optimisé.

- _ Nous avons eu beaucoup d'imprévus.

- _ Nous avons été bloqués plusieurs fois malgré tous nos efforts.

Avec le temps et l'expérience, nous avons réussi à nous habituer et à nous adapter au Javascript, TypeScript, EcmaScript grâce à de nombreuses recherches, autoformations, documentations et exemples sur internet.

Pour ce qui est des perspectives nous proposons les améliorations suivantes :

- _Réalisation d'un tableau de bord.

- _Possibilité de réinitialiser le mot de passe étant connecté.

- _Améliorer le responsive pour les écrans de taille minimale.

- _Déployer l'application sur un site web.

Webographie

_ <https://www.udemy.com/the-complete-nodejs-developer-course-2/learn/lecture/>
[Auto formation (Node.Js, Express.Js, MongoDB)]

_ <http://solve-it-technologies.com/>
[Informations concernant Solve IT Technologies]

_ <https://stackoverflow.com/>
[Forum Ajax, Javascript, Node.Js, MongoDB, Express.Js, Angular]

_ <https://www.npmjs.com/>
[Documentation npm packages]

_ <https://mongoosejs.com/docs/>
[Documentation Mongoose]

_ <http://mongodb.github.io/node-mongodb-native/3.2/api/>
[Documentation Node.Js MongoDB]

_ <https://nodejs.org/dist/latest-v12.x/docs/api/>
[Documentation Node.Js]

_ <https://openclassrooms.com/fr/courses/3449001-utilisez-des-api-rest-dans-vos-projets-web>
[Documentation REST API]

Copie scannée du certificat de stage

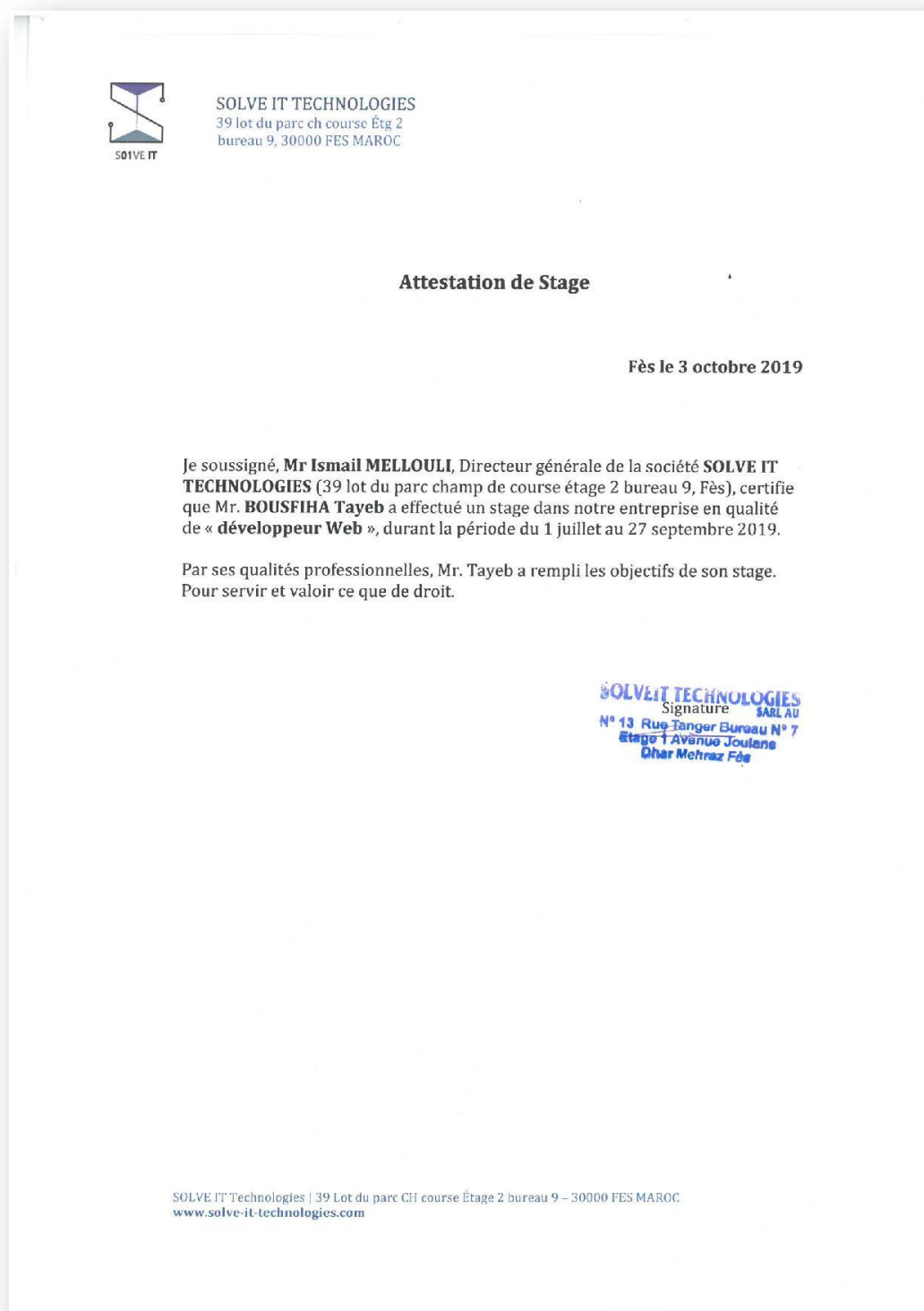


Figure 28 : Copie scannée du certificat de stage