



Rapport de stage

Développeur fullstack

MAGUEUR Marc

Mai 2023 à Juillet 2023

Tuteur de stage : GUIGNAND Franck

Professeur superviseur : RAMIARA Jean-François

Responsable de section : PUEL Christophe

Établissement : Institut Limayrac - BTS SIO (première année)

Entreprise d'accueil : Alsatis (Alsatis Entreprise)

Rapport de stage

1. Fiche 1 : Présentation de la structure d'accueil.....	3
1.1. Présentation générale.....	3
1.2. Cœur de métier.....	4
1.3. Organisation de la structure.....	4
1.4. Chiffres clés.....	5
2. Fiche 2 : Présentation du contexte du stage.....	6
2.1. Maître de stage.....	6
2.2. Positionnement dans l'organisation.....	6
2.3. Missions réalisées.....	7
3. Fiche 3 : Environnement technique.....	8
3.1. Ressources matérielles.....	8
3.1.1. Architecture.....	8
3.1.2. Schéma réseau.....	8
3.2. Ressources logicielles.....	8
4. Fiche 4 : Activité 1.....	9
4.1. Préface.....	9
4.2. Compétence(s) mise(s) en œuvre.....	9
4.3. Cahier des charges.....	9
4.3.1. Présentation et objectifs.....	9
4.3.2. Public visé.....	9
4.3.3. Aspects fonctionnels et techniques	10
4.4. Démarche / Mode opératoire.....	10
4.5. Preuves de la réalisation.....	10
4.5.1. Modèle de données.....	11
4.5.2. IHM - Brouillon des interfaces.....	16
4.5.3. Code.....	21
4.5.3.0. Préface.....	21
4.5.3.1. web.php.....	21
4.5.3.2. Middleware.....	24
4.5.3.3. Models.....	24
4.5.3.4. View.....	26
4.5.3.5. Controller.....	26
5. Fiche 5 : Retour d'expérience.....	28
5.1. Remerciements.....	28
5.3. Pistes de progrès.....	28

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	2 / 28

1. Fiche 1 : Présentation de la structure d'accueil

1.1. Présentation générale

Dans le cadre de de formation BTS SIO option SLAM à l'Institut Limayrac, j'ai effectué un stage dans la société Alsatis ayant son siège au 11 Rue Michel Labrousse à Toulouse. Mon stage s'est effectué sous la tutelle de mon maître de stage et également responsable technique, M. GUIGNAND Franck.

Alsatis, a été créée le 28 décembre 2004 par quatre associés, dont Monsieur Vincent SABATHIER (fondateur historique). Aujourd'hui elle est dirigée par Monsieur Antoine ROUSSEL (aussi membre fondateur).

Lors de la création d'Alsatis, les associés ont constaté que combler le manque de connectivité en zone blanche pour le grand public était un réel enjeu. Alsatis s'est donné pour mission l'accès à internet grâce au développement des différentes solutions pour les personnes habitant dans les zones blanches.

2004 : Création de la société Alsatis et du centre de profit Alsatis Réseaux qui a pour but de concevoir, de déployer et exploiter des infrastructures télécom radios pour lutter contre la fracture numérique aux côtés des Collectivités.

2005 : Déclaration auprès de l'ARCEP sous la licence L33-1.

2008 : Mise en place de la Direction Alsatis Télécom afin de proposer des offres commerciales au Grand Public éligible à ces réseaux radios.

2008 : Alsatis remporte un premier marché départemental en Isère.

2009 : Naissance de Bloosurf, petite sœur d'Alsatis qui sévit aussi dans les zones rurales du Maryland en portant la même offre de valeurs que sa grande sœur.

2009 : Alsatis remporte un deuxième marché départemental en Charente.

2010 : Création de la Direction Alsatis Entreprises qui adresse les entreprises PME/PMI/ETI et même grands comptes avec une offre de services IP sur mesure.

2011 : La Holding Groupe Alsatis est créée pour supporter et animer les différentes filiales déjà créées et qui s'y rattachent naturellement

2012 : Création de la société Confluences Fibre qui assure l'exploitation d'un réseau fibre optique à vocation professionnelle pour le compte de la communauté de communes de Sarreguemines Confluences dans le département de la Moselle (57).

2013 : Alsatis vend son réseau radio Haut Débit au CD38.

2015 : Vente clientèle Satellite Grand Public à Nordnet, filiale de Orange.

2018 : Alsatis remporte l'appel d'offre de la Vendée afin de déployer, exploiter et maintenir les sites Radio qui permettront de couvrir les zones dites « blanches » de ce département.

2019 : Alsatis gagne 3 marchés de référence dans l'aménagement numérique en France :

- Reconduction marché CD38 en LTE
- Nouveau marché Zones blanches en Charente
- Succès sur marché CD11 en LTE

Etudiant	MAGUEUR Marc	Tuteur	GUIGNAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	3 / 28

1.2. Cœur de métier

Créé en 2011, Groupe Alsatis a vu le jour afin d'être l'animatrice des différentes structures qui la compose.

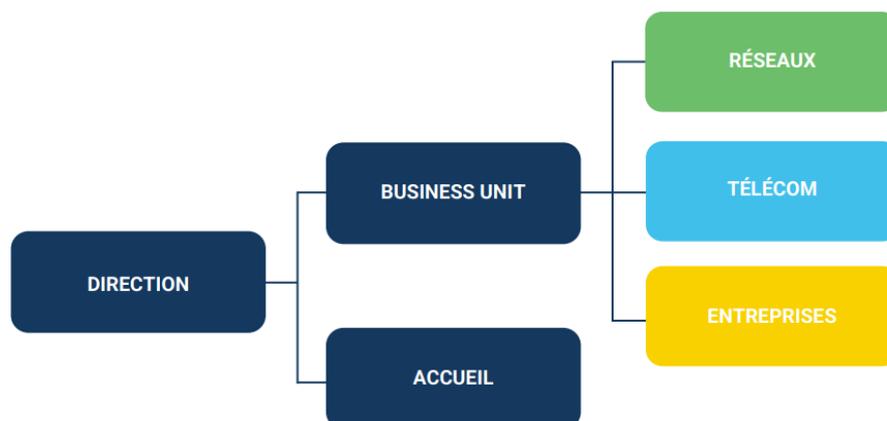
Aujourd'hui, le Groupe est composé de quatre filiales aux activités complémentaires dans les domaines du numérique et des réseaux de télécommunications fixes et mobile :

- **ALSATIS** : est la filiale principale qui est composée de 3 business Unit : Alsatis Réseaux/Exploitation, Alsatis Entreprises et Alsatis Télécom
- **THORIS** : Créé en 2014 est une Société Civile Immobilière qui permet à la structure de supporter et gérer son parc immobilier.
- **CONFLUENCES FIBRE** : a été créée dans le but d'exploiter un réseau de fibre optique
- **BLOOSURF** : situé à Pocomoke City dans le Maryland (Etats-Unis), cette filiale a la même activité que la filiale ALSATIS et a pour but de couvrir en internet très haut débit l'état du Maryland.

Le reste du rapport portera uniquement sur alsatis étant donné que c'est dans cette entreprise du groupe que j'ai effectué mon stage.

1.3. Organisation de la structure

Aujourd'hui, Alsatis c'est une équipe d'experts qualifiés, regroupés au sein de trois centres de profits. L'organigramme ci-dessous vous explique leur fonctionnement :



Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	4 / 28

Chaque branche possède ses propres missions. Alsatis Réseaux est l'activité en charge de l'ingénierie, l'étude de faisabilité, le déploiement et l'exploitation des infrastructures réseaux, notamment :

- Le déploiement de réseaux télécoms multi technologies,
- La gestion transparente et active des Délégations de Service Public,
- L'exploitation technique de réseaux radios,
- La supervision et la maintenance 24h/24 et 7j/7,
- Avec 14 centres d'intervention locaux pour une couverture nationale sous 4h.

Alsatis Télécom est le FAI qui adresse le grand public :

- Avec des offres Haut et Très Haut Débit Illimité Internet et Téléphonie ;
- Une option TV selon les cas
- Avec une hotline locale 6j/7.

Alsatis Entreprises est un FAI et opérateur télécom pour les entreprises et les collectivités locales. Elle permet à ses clients d'avoir :

- Une offre sur mesure adaptée au besoin de la structure grâce à nos commerciaux BtoB présents à Toulouse, Lyon, Nantes et Paris,
- Une offre Haut et Très Haut Débit Illimité sur tous types de supports : fibre optique, satellite, radio, xDSL, etc.
- De la téléphonie sur IP en configuration Centrex (IPBX hébergé par Alsatis) ou trunk SIP,
- Une interconnexion physique ou virtuelle de sites distants,
- Un hébergement de serveurs dédiés ou virtuels dans nos salles blanches,
- Une vidéoconférence Haute définition,
- Une configuration spécifique de réseaux LAN (QoS, redirection de ports, sécurisation de liens, balancement de charge, etc.)

1.4. Chiffres clés

Alsatis en quelques chiffres c'est :

- 1600 Sites exploités
- Un centre de supervision 24/7 (NOC)
- 15M € de CA en 2021
- 1500 Clients
- 4 Agences
- 3 510 950€ de capital
- 80 Collaborateurs

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	5 / 28

2. Fiche 2 : Présentation du contexte du stage

2.1. Maître de stage

Mon stage s'est déroulé au sein du support et service technique de d'Alsatis Entreprise. Tout au long de mon stage, j'ai été accompagné par mon maître de stage, Franck GUIGNAND.

Au sein d'Alsatis, M.GUIGNAND occupe le poste Responsable Technique (cadre), ses missions consistent principalement en :

- Gestion de service
- Achat et négociation

2.2. Positionnement dans l'organisation

Tout au long de mon stage, j'ai été rattaché à la gestion technique et support technique. Cette équipe se compose de 6 personnes.



TROMBINOSCOPE



Maxime BETEMPS et Charles BANQUET sont chefs de projets (clients). Ils sont chargés de répondre aux demandes techniques de leurs clients. Si jamais un nouveau client arrive, il est directement rattaché à un projet, lui-même rattaché à un chef de projet.

Leur travail varie entre la configuration, l'installation et l'ingénierie de réseaux. Ils sont aussi, de manière non officielle, agents de support N3.

Tristan TISON est principalement agent de support technique (N1+N2) mais aussi chef de projet. Ses missions principales sont d'assister les clients qui appellent le support mais il s'occupe aussi de clients comme MMs BETEMPS et BANQUET.

Etudiant	MAGUEUR Marc	Tuteur	GUIGNAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	6 / 28

Quentin DE GRENIER est agent de support (N1+N2). Il s'occupe principalement d'aider les clients quand ils appellent le support mais peut aussi s'occuper de tâches de gestion de réseaux quand il n'a pas de client à s'occuper et de tâches de l'un des chefs de projet s'ils sont débordés.

Maxime DOBIGNY est alternant et même s'il est noté comme agent de support technique, il s'occupe de décharger de certaines tâches les chefs de projets mais aussi développe des outils internes pour aider les chefs de projets dans leurs missions quotidiennes.

2.3. Missions réalisées

Alsatis Entreprise est une entreprise qui s'est spécialisée dans la revente de services qu'elle achète la plupart du temps à d'autres opérateurs spécialisés. A aucun moment, Alsatis Entreprise n'est propriétaire des liens qu'ils revendent. Ils sont juste exploitants. A ce titre, il est important pour eux de rentrer dans ses frais et donc de tenir une trace du prix d'achat du matériel / du lien avant d'y appliquer une marge et de le revendre.

Un fichier excel regroupant tout ce qu'Alsatis Entreprise pouvait et avait acheté a été mis en place avant mon arrivée. Son fonctionnement était simple, sur une page on rentrait les informations générales du client (nombre de sites, clients existant, sites déjà raccordés, etc...) et les autres pages étaient seulement des longs tableaux compliqués remplis de formules de calcul dans tous les sens.

Ce que cherchais mon responsable de stage était une manière de plus facilement éditer, partager et d'utiliser le fichier excel. La solution que j'ai proposé était de développer un logiciel informatique sous la forme d'un site web qui serait accessible uniquement sur le réseau local (comme un intranet) avec un système d'authentification pour éviter que toute personne sur le même réseau puisse accéder au logiciel.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	7 / 28

3. Fiche 3 : Environnement technique

3.1. Ressources matérielles

3.1.1. Architecture matérielle

Pour des raisons de sécurité, Alsatis n'a pas pu me fournir d'informations sur leur architecture matérielle utilisée au sein de leurs locaux et/ou de leurs différents serveurs en datacenter.

Pour mener à bien ma mission principale, un ordinateur portable, un clavier sans fil et une souris sans fil m'ont été prêtés. Avec l'ordinateur et l'utilisation de XAMPP j'ai pu simuler l'environnement de production (un serveur Apache et MySQL)

Lors du développement de mon logiciel, le déploiement sur un serveur dédié en interne était discuté mais je n'ai jamais eu l'opportunité d'effectuer le déploiement ou d'en savoir plus.

3.1.2. Schéma réseau

Pour des raisons de sécurité, Alsatis n'a pas pu me fournir le schéma réseau utilisé au sein de leurs locaux et/ou de leurs différents serveurs en datacenter.

Ma mission principale ne possède pas de schéma réseau car j'étais dans un environnement de développement sur mon ordinateur. Je n'interagissent donc pas sur le réseau interne de l'entreprise. En plus, l'application ne sera pas publique mais juste déployée sur un serveur en interne, il n'y a donc pas besoin de schéma réseau.

3.2. Ressources logicielles

Alsatis m'a fourni certaines ressources logicielles pour l'accomplissement de ma mission principale. J'ai quand même dû fournir certaines ressources logicielles personnelles.

Les ressources qu'Alsatis m'a fourni sont :

- Un compte teams
- Une adresse email @alsatis.com
- Un compte avec des permissions limités me permettant de m'authentifier auprès de divers logiciels internes

Durant mon stage, j'ai utilisé comme ressources logicielles personnelles :

- XAMPP : Petit logiciel utilisé pour créer un environnement de développement avec un serveur php et une base de données MySQL.
- PhpMyAdmin : Site web me permettant d'administrer la base de données du projet
- PhpStorm : Éditeur de code intelligent qui m'a permis de développer l'entièreté du site
- Tailwind CSS : Framework front-end que j'ai pour habitude d'utiliser dans des gros projets qui me permet d'alléger la charge de travail sur tout ce qui est frontend (interfaces graphiques)

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	8 / 28

4. Fiche 4 - Activité 1 : Développement un logiciel sous la forme d'un site web

4.1. Préface

Je sais qu'il n'est pas commun de faire une préface à une fiche d'activité mais je pense nécessaire de signaler que la version finale du logiciel est disponible à tous. Si vous voulez explorer l'application dans son entièreté (et je pense notamment pour voir toutes les IHM car elles ne seront pas toutes listés) vous pouvez vous rendre à l'URL suivante :

<https://r.mrkm.dev/alsatis/saas>

ou directement

<https://preview.saas-alsatis.marc-magueur.dev>

4.2. Compétence(s) mise(s) en œuvre

Gérer le patrimoine informatique	Non ▾
Répondre aux incidents et aux demandes d'assistance et d'évolution Une fois le cahier des charges établi, le client a décidé, à plusieurs reprises, de faire des demandes d'évolution, qui au final, se sont retrouvées dans le cahier des charges. (4.2 et 4.3)	Oui ▾
Développer la présence en ligne de l'organisation	Non ▾
Travailler en mode projet Durant toute la durée de développement du logiciel, j'ai suivi une méthodologie de travail très stricte et je rapportait tout ce que je faisais sur github (preuve: 4.3)	Oui ▾
Mettre à disposition des utilisateurs un service informatique A la fin de mon stage, le logiciel en question était prêt à être déployé pour être mis à disposition des différents acteurs en ayant besoin. (preuve: 4.2 et 4.4)	Oui ▾
Organiser son développement professionnel Formation sur Laravel et sur tailwindcss pour mener le projet	Oui ▾

4.3. Cahier des charges

4.3.1. Présentation et objectifs

Le rôle de l'application web est de permettre aux différents commerciaux / employés de l'avant vente de facilement donner à un client une idée des prestations qu'il peut avoir. Cet outil aura aussi pour but de faciliter la mise à jour des prix de revente pour éviter de donner des informations erronés à un client ou faire un devis/facture sur des informations erronés

La date butoir était initialement prévue pour le 30 juin 2023 mais dû à certaines demandes floues et à certaines certaines demandes qui d'évolution arrivés assez tardivement, la date butoir a été décalé au 13 juillet 2023 (+ 9 jours ouvrés)

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	9 / 28

4.3.2. Public visé

La cible principale est le service commercial et le service d'avant vente. Le public en question est toujours sur ordinateur et est toujours connecté au réseau local de l'entreprise, que ce soit car ils sont physiquement dans l'entreprise, soit car ils sont connectés au réseau de l'entreprise via un VPN.

4.3.3. Aspects fonctionnels et techniques

Le site sera divisé en 4 modules fonctionnels. Ces derniers seront développés indépendamment les uns des autres.

- Client : Le module client permettra de créer une fiche client et d'y ajouter des prestations afin de faire un pseudo devis
- Prestations : Le module prestation permettra de consulter ou d'éditer (en fonction des permissions attribués à l'utilisateur en question) les prestations disponibles
- Changelog : Sur demande du client, un module permettant de faire des changelogs doit être présent. Il servira aux utilisateur de consulter les modifications récentes faites au prestations
- Utilisateurs : Ce module servira à la gestion des comptes utilisateurs

Le site demandera une authentification sur toutes les pages à l'exception de la page de connexion et de déconnexion.

La création du site se fera avec les technologies suivantes :

- HTML - Pour faire la structure des pages
- Blade - Pour la structure des pages
- CSS - Pour styliser les pages
- Tailwind CSS - Pour faciliter l'utilisation du css
- JS - Pour animer et rendre les pages interactives
- PHP - Pour tout ce qui est gestion serveur
- Laravel - Pour faciliter la maintenance et le développement
- MySQL - Pour la base de donnée

Le site n'aura pas besoin d'être responsive et les mises à jour (qu'elles soient de sécurités ou non) devront être effectuées par Alsatis une fois la date butoir atteinte. Le suivi d'implémentation des fonctionnalités ainsi que le planning du projet se feront via github

4.4. Démarche / Mode opératoire

Durant tout le développement du logiciel, la méthodologie AGILE KABAN a été mise en place par le biais de "projets GitHub" qui permettent d'utiliser une "issue" et lui assigner un état (À faire, En cours et Terminé)

Cela permettait aussi (comme la méthode AGILE l'impose) de plus facilement répondre aux demandes du client qu'elles soient faites le premier jour où la dernière semaine. Ca permettait aussi de plus facilement visualiser les tâches restantes et les tâches déjà accomplies.

Title	Assignees	Status	Labels
1 Permettre l'affichage des promotions d'une manière particulières (front & back) #8		Todo	enhancement
2 exprot csv :D #27		Todo	enhancement
3 Ajout de la possibilité d'ajouter des options sur les prestations (obligatoires ou non) #1	MaitreRouge	In Progress	enhancement
4 Ajout des changelogs ? #45	MaitreRouge	In Progress	enhancement
5 Ajout de la possibilité de totalement customiser les prix lors de la création du devis #2	MaitreRouge	Done	enhancement
6 Ajouter la possibilité de bloquer certaines options/prestations si l'engagement est trop petit... #3	MaitreRouge	Done	enhancement
7 possible de mettre 0 en qte d'un produit #5	MaitreRouge	Done	bug
8 Ajout d'un bouton pour supprimer un devis #7	MaitreRouge	Done	enhancement

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	10 / 28

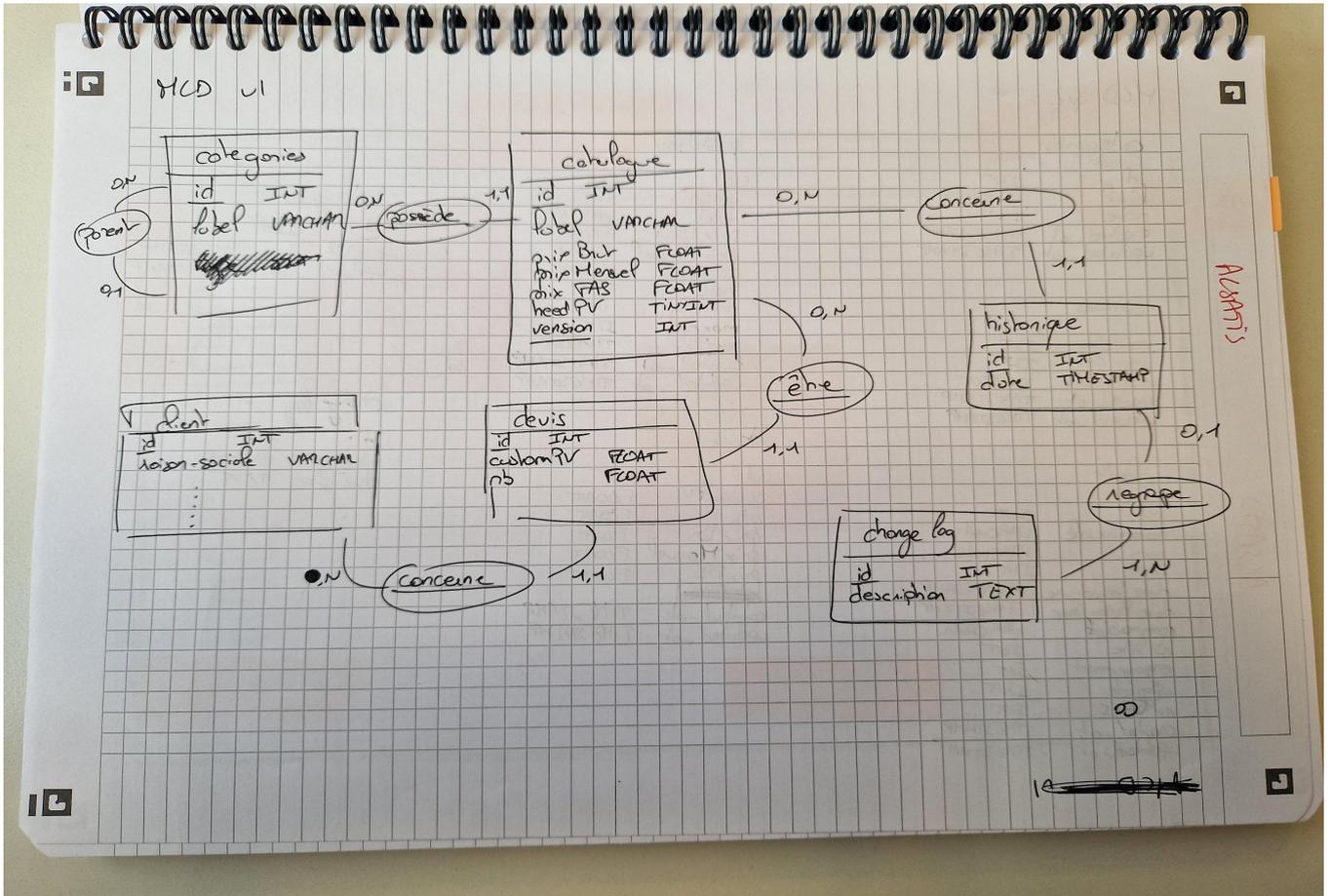
4.5. Preuves de la réalisation

Exemple de preuves (liste non exhaustive) :

- SLAM : Modèle de données, extrait de code, IHM,...
- SISR : Schéma réseau, fichier de configuration, plan d'adressage,...

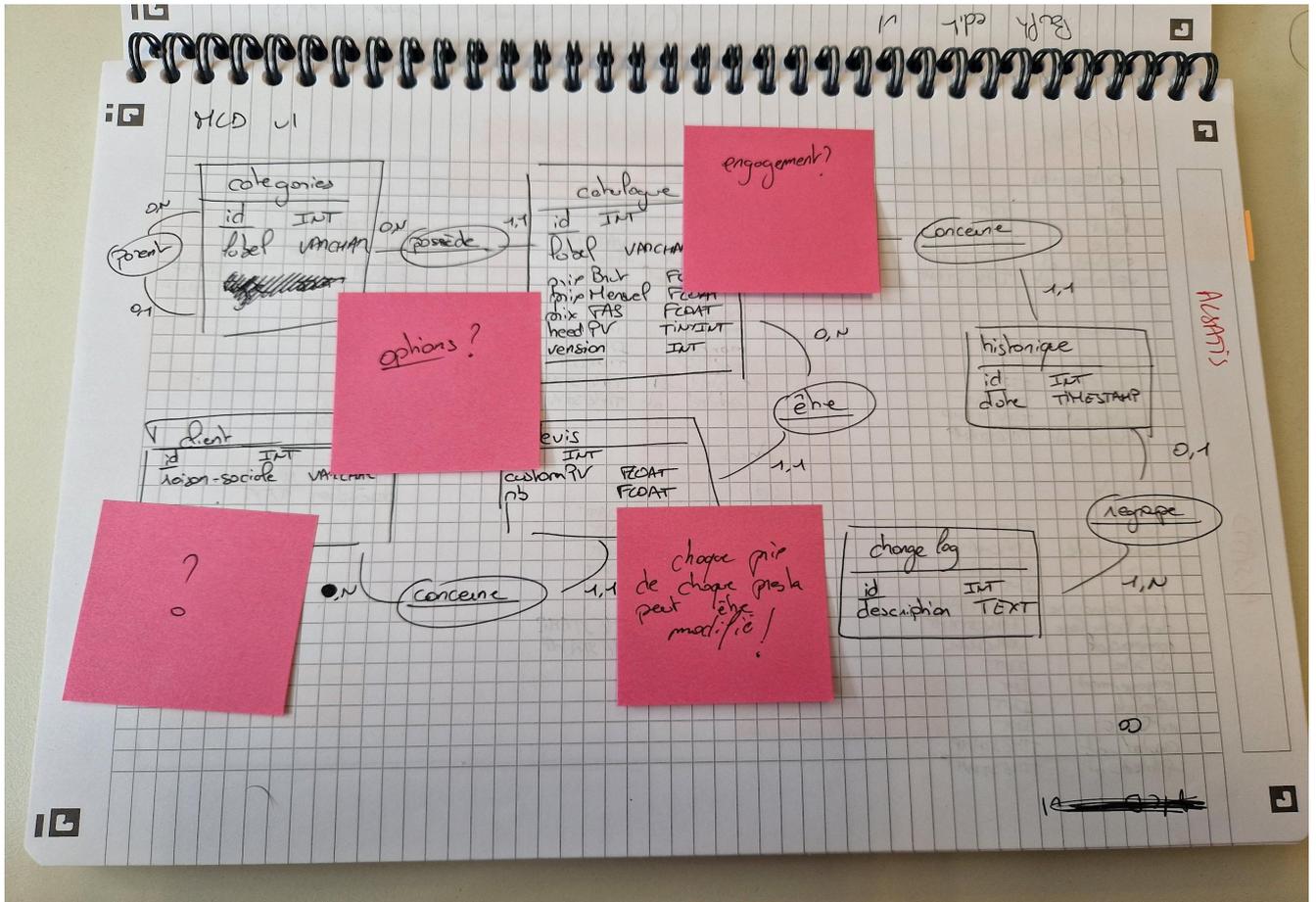
4.5.1. Modèle de données

Après avoir fait le cahier des charges, je me suis lancé dans la conception du modèle de données. Malheureusement, la méthodologie AGILE KABAN, m'obligeant à s'adapter au client en permanence m'a fait faire des changements drastiques dans le MLD :



Première itération du MLD (sans notes)

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	11 / 28



Première itération du MCD mais avec des notes

A chaque MLD que je faisais, j'essayais d'englober toutes les fonctionnalités que le client demandait mais chaque demande d'évolution demandait que je change le MLD

Ce MLD comprends déjà beaucoup de fonctionnalités qui sont restés dans l'application

- Catalogue
Plus tard renommé en prestations, catalogue est la table qui s'occupe de stocker et gérer toutes les prestations de l'application. Chaque prestation doit appartenir à une catégorie (1,1), peut être rattachée à un historique (0,N) et peut être ajoutée à un devis d'un client (0,N)
- Categories
Dans l'application, chaque prestation appartient à une catégorie. Chaque catégorie doit elle-même appartenir à une catégorie parent et chaque parent doit appartenir à la catégorie principale. Ça paraît compliqué mais c'est plutôt simple quand on regarde l'interface :

Il n'existe que 3 catégories racines et on ne peut ni les modifier ni les supprimer.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	12 / 28

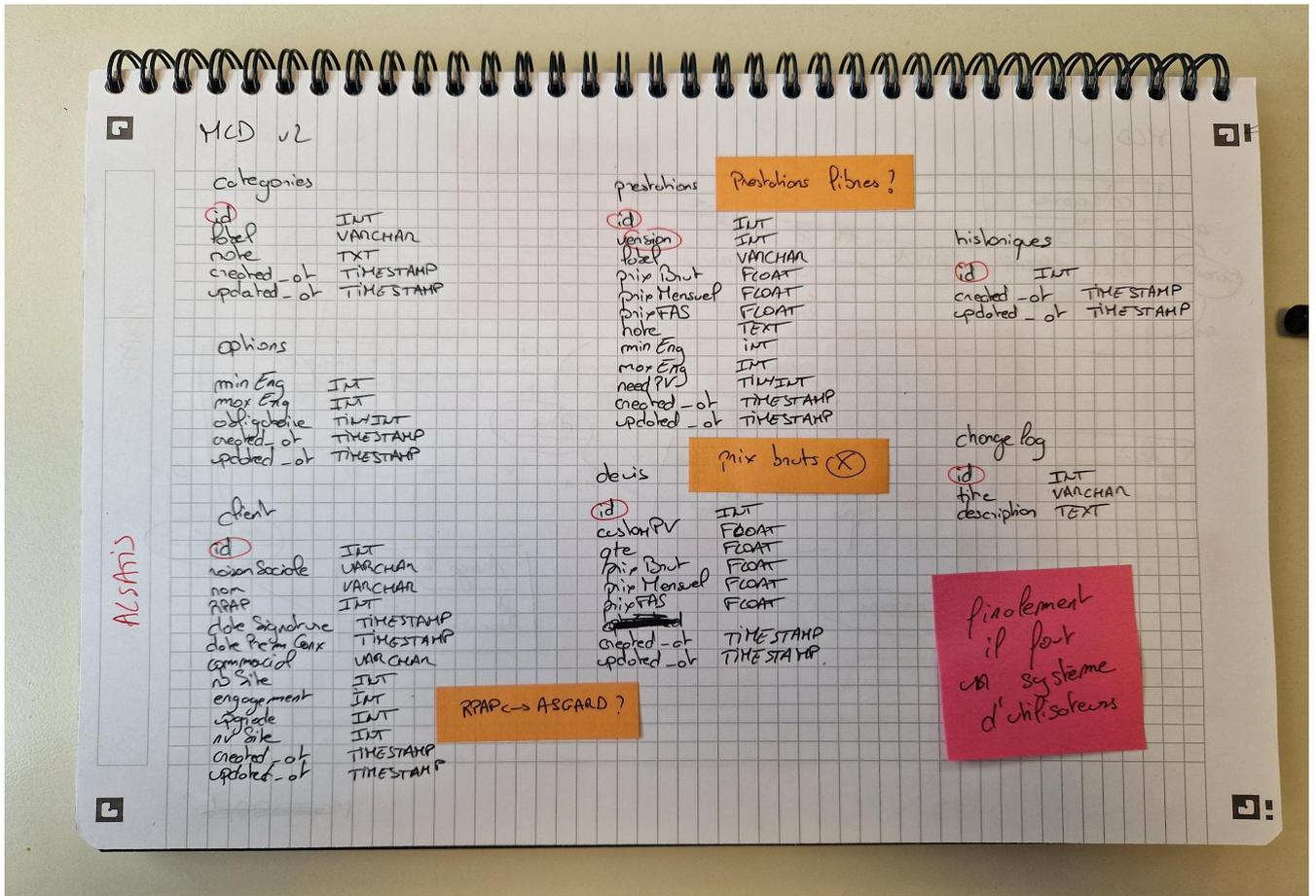


- Historique :
Le système d'historique crée un historique dès qu'une prestation est modifiée. Chaque historique est lié à une seule et unique prestation et chaque historique est dans un second temps regroupé dans un changelog (0,1)
- Changelog
Une fois qu'on a fait toutes les modifications qu'on voulait sur les prestations, pour tout sauvegarder, il faut créer un changelog dans lequel, va automatiquement s'ajouter tout l'historique non présent dans un changelog et on va manuellement donner un titre et une description au changelog pour que tout le monde puisse comprendre ce qui a été changé
- Devis
Chaque devis est rattaché à un client (1,1) et c'est au commercial / avant-vente d'ajouter des prestations au devis. Chaque prestation va créer une nouvelle entrée dans la table devis. Il faut voir la table devis comme la trace de toutes les prestations ajoutées au client. Dans cette version du MLD, on ne peut que renseigner une quantité et un prix de vente.
- Client
Chaque client est relié à un seul et unique devis (1,1). Je sais que ca peut paraître contre intuitif et se dire que ca serait plus facile de faire un client vers plusieurs devis (0,N ou 1,N). Mais dans notre cas, chaque devis est réalisé par rapport à la situation d'un client à la création d'un devis. Par exemple, on peut avoir l'entreprise "mrkm.dev" qui fait une demande de devis pour 3 sites et l'année d'après pour 5. Les prix vont différer et on cherche à avoir "une fiche" client par devis.

Chaque post-it est directement une demande d'évolution du client par rapport à ce qu'il avait originellement demandé. De haut en bas, on peut lire :

- engagement :
Le client voulait absolument que les clients (du logiciel) soit engagés et que ca figure sur la fiche client. Il voulait aussi que chaque prestation (table catalogue) ait un engagement
- options ? :
Le client voulait un système d'option pour que chaque prestation puisse ou non, être vendue avec des options supplémentaires.
- Chaque prix de chaque prestation peut être modifié ! :
Le client voulait que chaque commercial / avant-vente, puisse modifier le prix d'une prestation quand elle était ajoutée à un client. Par exemple vendre une prestation dont le prix conseillé est 30 et la vendre 35

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	13 / 28



Deuxième itération du MCD avec des notes

Pour ce qui est des modifications :

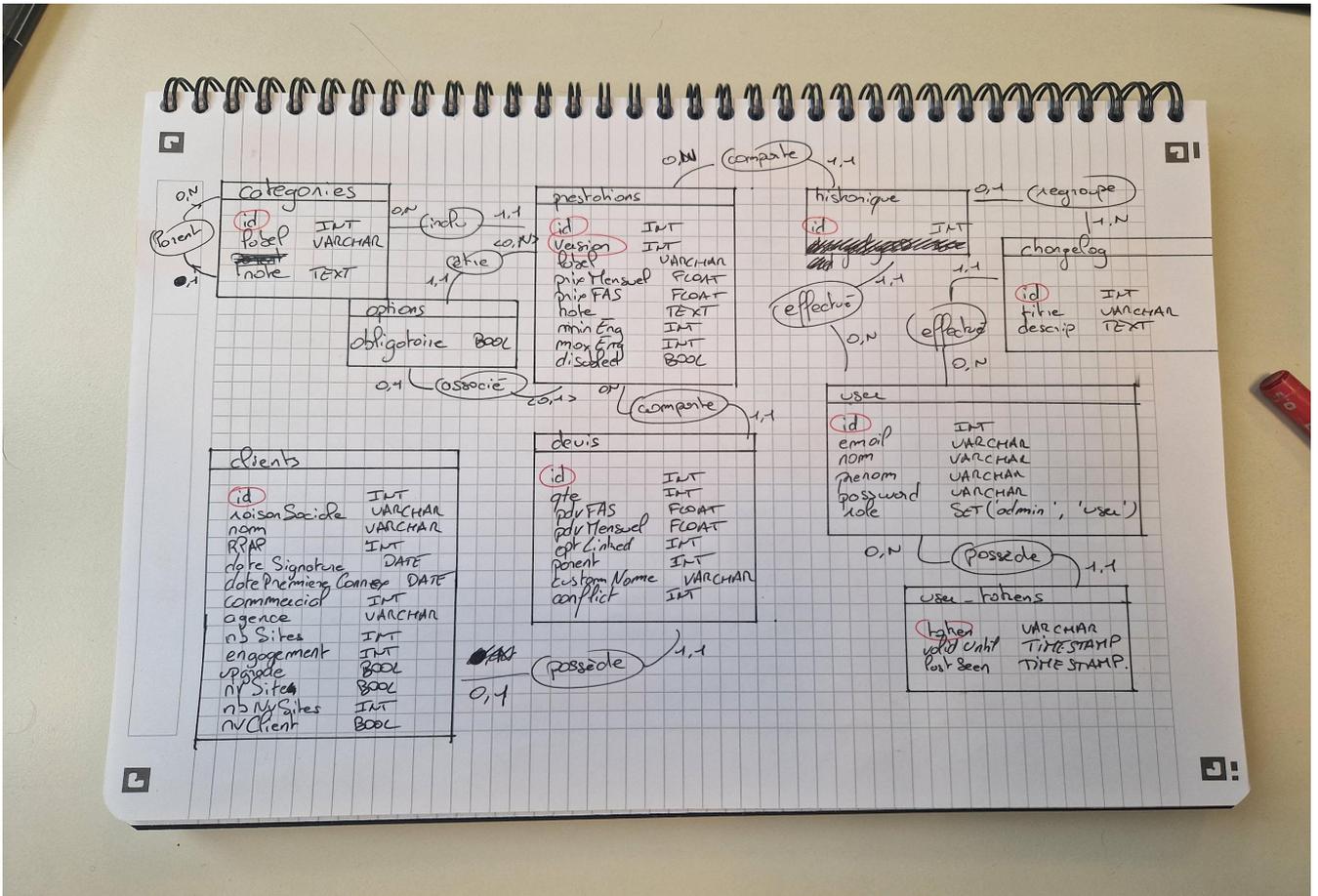
- Categories est comporte maintenant un champ note pour permettre de faire passer un message ou d'attirer l'attention de certains sur certaines particularités d'une certaine catégorie.
- Catalogue à été renommé en prestations, comporte aussi un champ note, version est désormais une clé primaire (oubli dans le premier MLD), ajout des engagements
- Devis permet maintenant de modifier entièrement le prix d'une prestation grace a ses champs prixBut, prixMensuel et prixFAS. Accessoirement, nb a été renommé en qte
- Client possède maintenant ses champs et un engagement
- Création de la table Option. Dans cette application, chaque option sera en réalité une prestation avec un lien d'héritage. Chaque entrée dans option aura une clé primaire composée (id de la prestation que l'option hérite et id de la prestation sur laquelle l'option s'applique).

Pour les post-its (de haut en bas) :

- prestations libres ?
Ce petit post-it évoque une nouvelle fonctionnalité que le client avait initialement demandé mais que je n'avais pas compris. Le but est que chaque commercial / avant-vente puisse directement créer une prestation qui n'existe pas (par exemple des frais de dossier). Ces prestations libres pourraient dans l'idéal avoir un nom, des prix et une quantité qu'on puisse changer.
- prix brut X
Le client ne voit pas l'intérêt des prix bruts et juge inutile que cette information soit stockée.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	14 / 28

- finalement il faut un système d'utilisateurs
Au final, le client veut un système d'utilisateurs avec un système de permission pour pas que certains utilisateurs puissent faire certaines actions.
- RPAP <-> Asgard
Fonctionnalité que j'ai proposé qui avait pour but de lier l'identifiant d'une fiche client au client dans leur logiciel de gestion client pour que certains champs (nom, nombre de sites, structure et plus...) se complètent automatiquement. Cette fonctionnalité n'a pas été retenue



Troisième et finale itération du MCD (sans notes)

Ce MCD n'est pas le MCD idéal et j'en ai conscience. Je pense notamment à la table option qui aurait pu être une relation prestation (0,N) <-> prestation (0,N) avec le champ obligatoire en option mais du côté MLD ça ne changerait rien.

Pour les modifications effectuées comparé à la version précédente on retrouve :

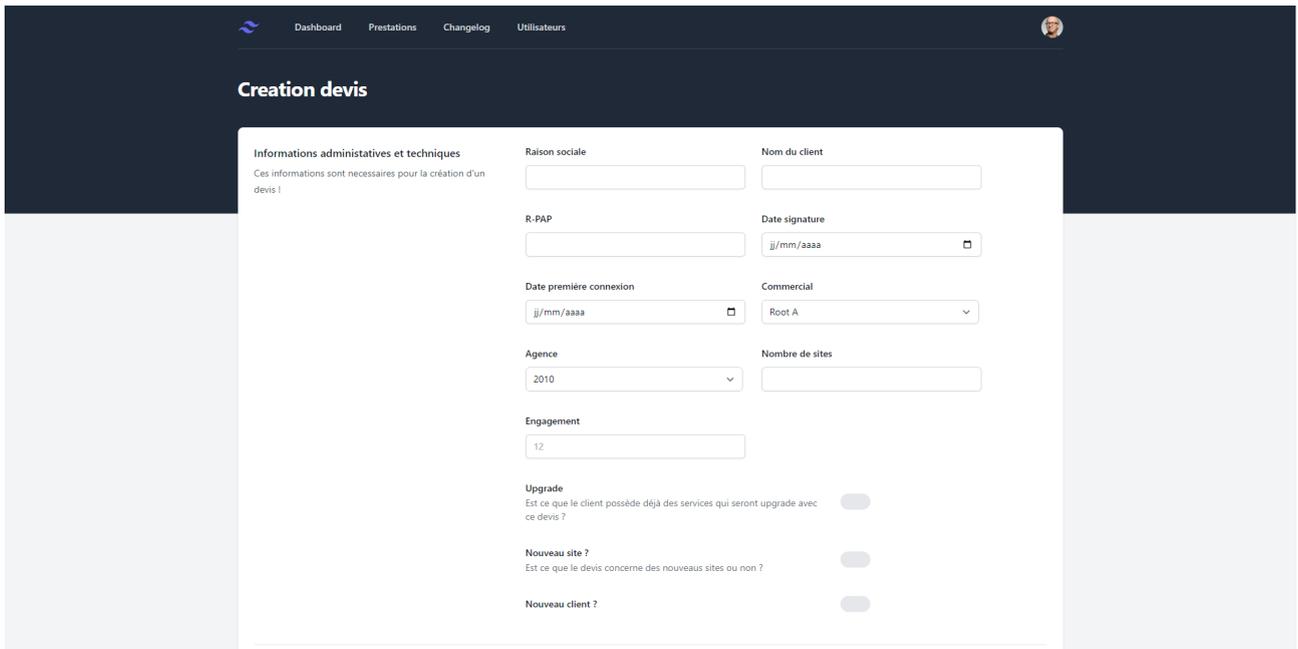
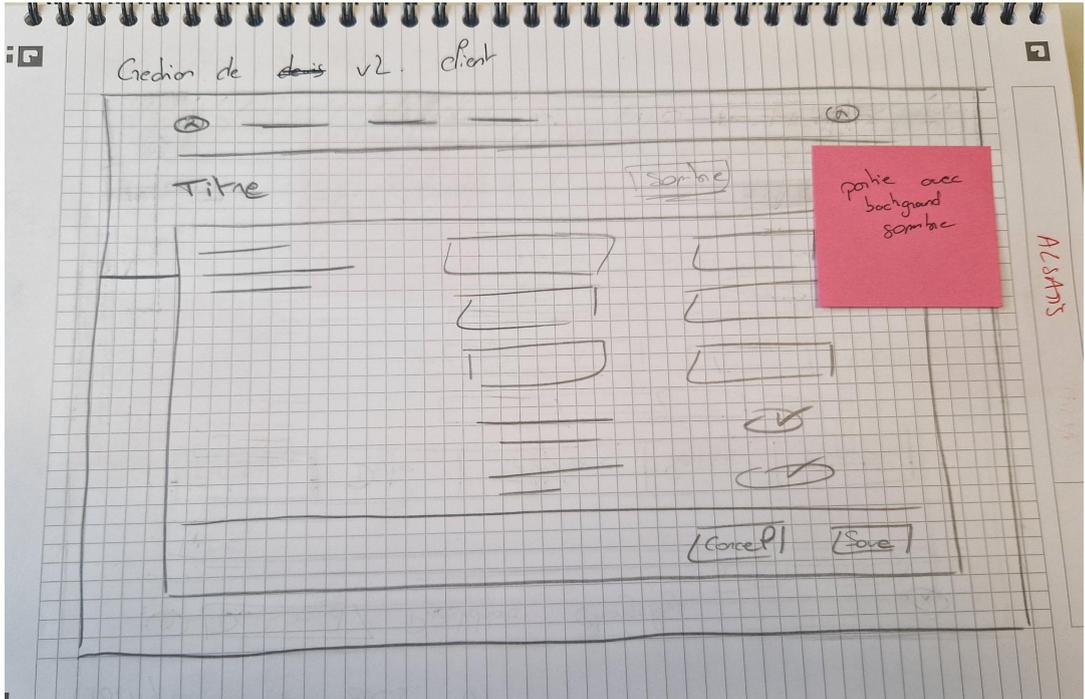
- Création de la table user qui aura pour but de stocker les utilisateurs et leurs données personnelles.
- Création de la table user_token qui à pour but de stocker les tokens d'authentification des différents utilisateurs. Ces derniers ont une date d'expiration et un timestamp de dernière utilisation
- Retrait du prixBrut dans les tables devis et prestation.
- Ajout des relations que j'avais oublié dans la deuxième itération
- Ajout de la colonne "disabled" dans la table prestation qui sert a noter les prestations qui ne sont plus en vente

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	15 / 28

4.5.2. IHM - Brouillon des interfaces

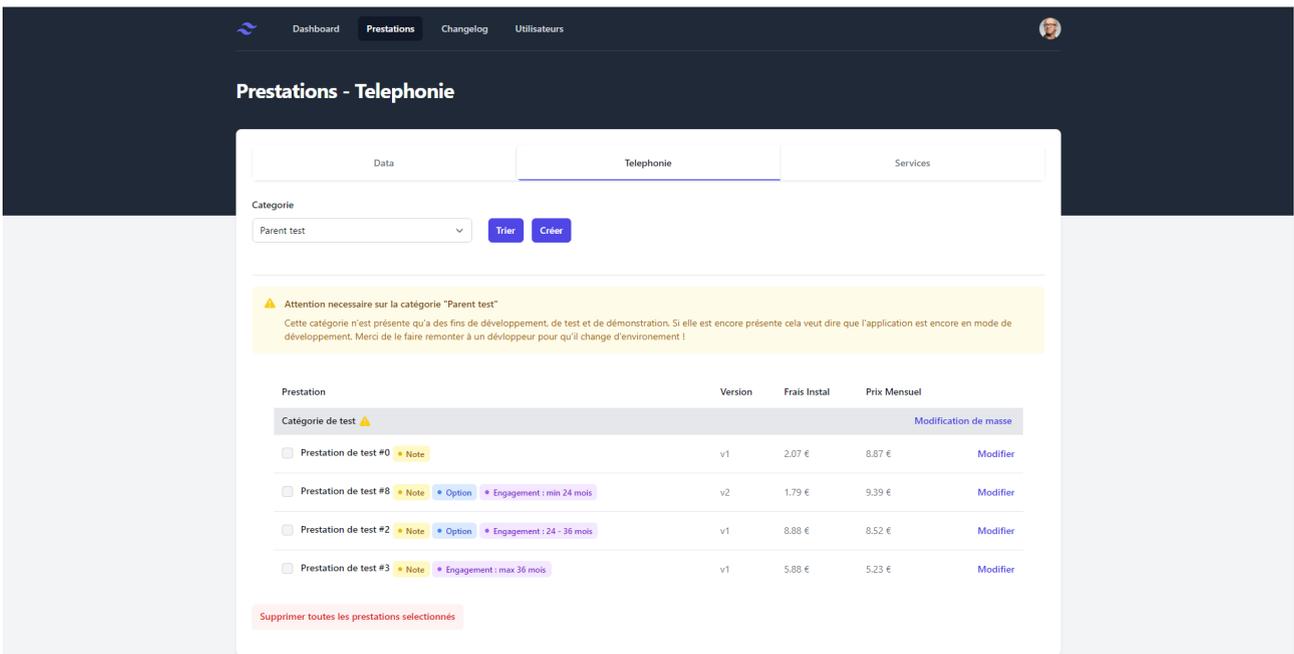
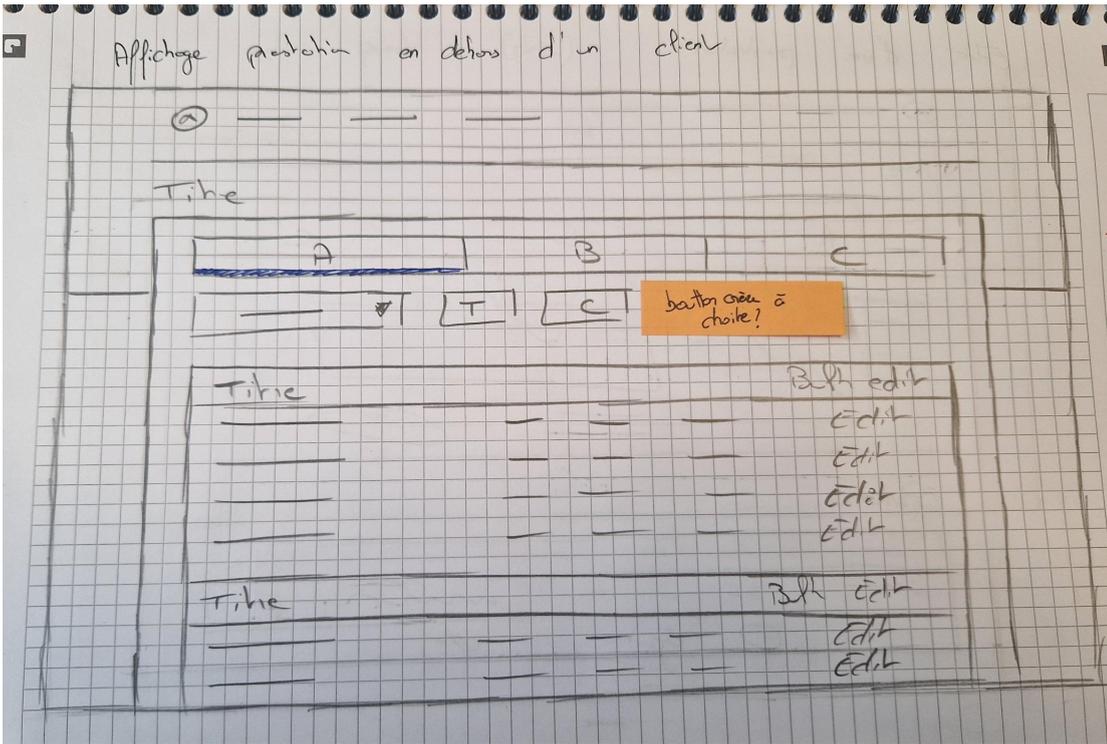
Pour réaliser le logiciel, il a fallu imaginer et construire des interfaces bien spécifiques. Dans cette partie je vais inclure les brouillons et les interfaces terminées.

Pour éviter de saturer le rapport d'images, j'ai décidé d'inclure dans ce rapport uniquement les interfaces sur lesquelles j'ai le plus travaillé. Mais il faut savoir que pour cette application, j'ai développé un total de 17 interfaces. Pour gagner du temps et pouvoir inclure le plus de fonctionnalités possibles, les interfaces d'édition n'existent pas à proprement parler, on utilise juste les interfaces de création dans laquelle on mets les informations de l'objet à modifier et on envoie le formulaire sur une route différente.



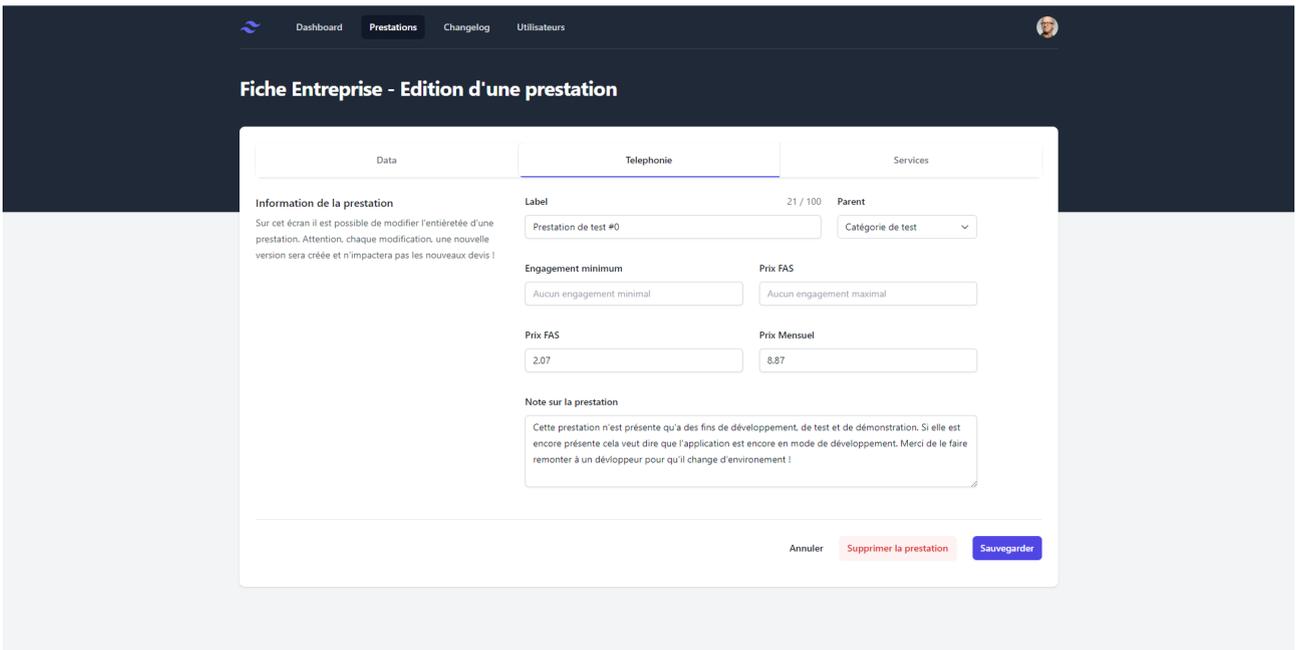
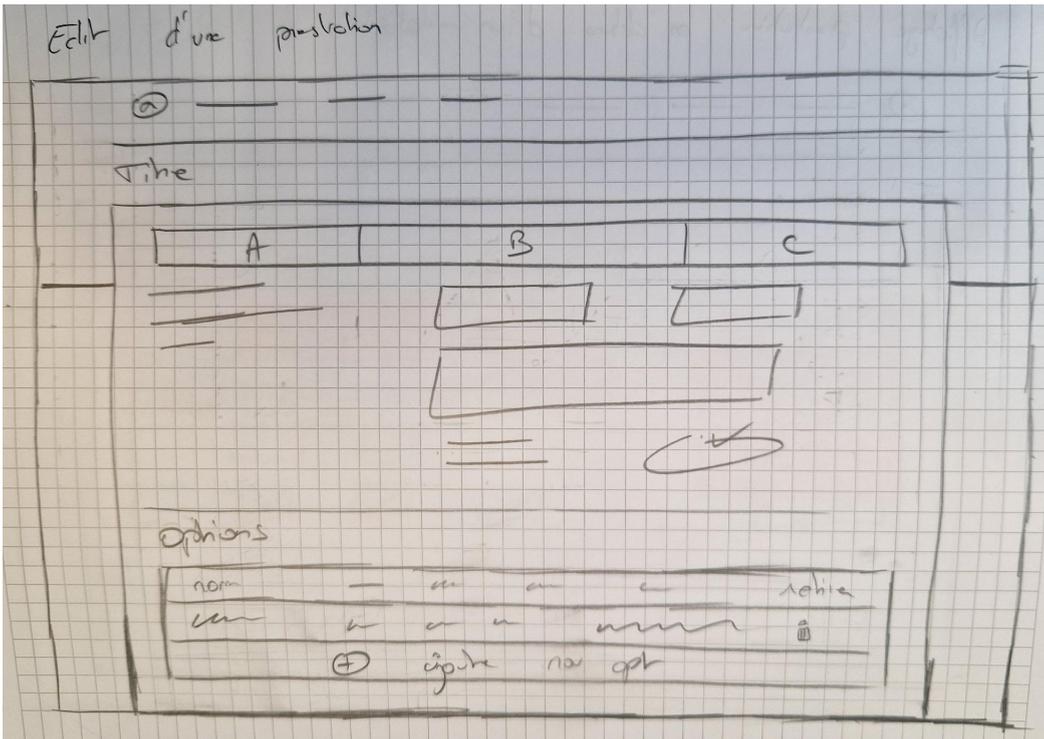
Brouillon et interface de la page de création des fiches client

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	16 / 28



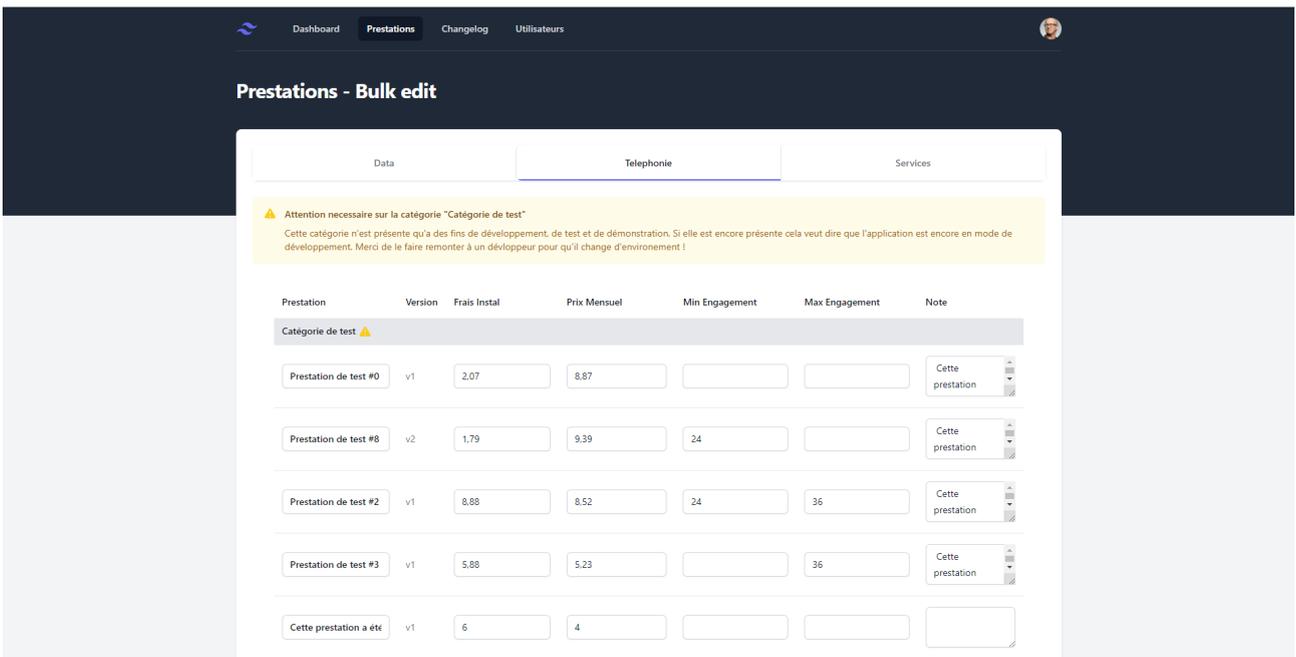
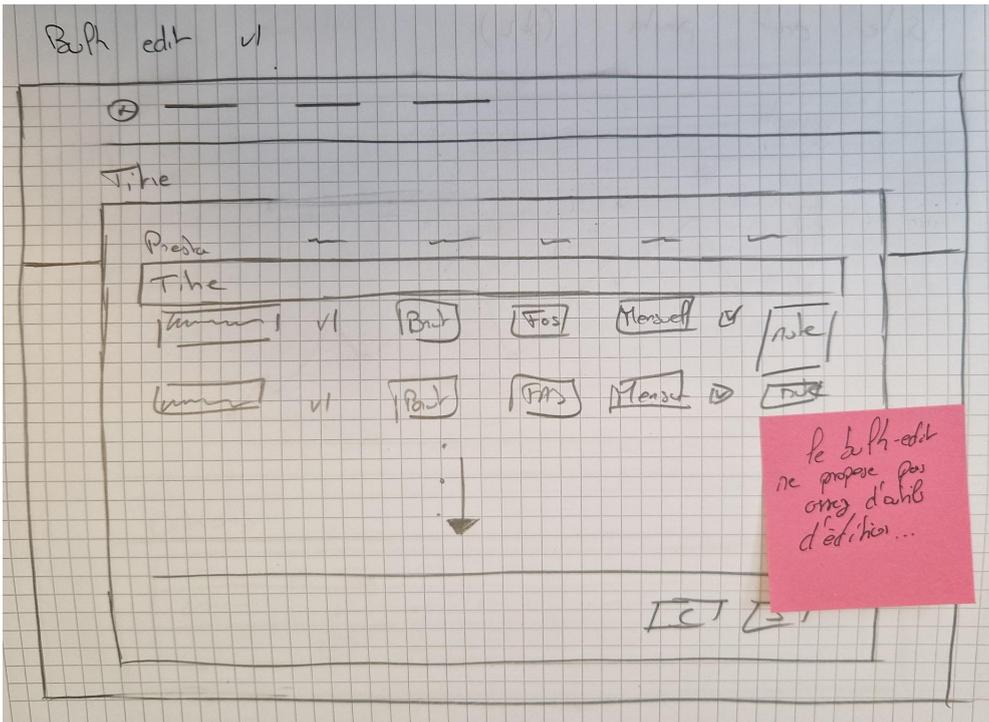
Brouillon et interface de gestion des prestations en dehors d'une fiche client

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	17 / 28



Brouillon et interface de l'édition d'une prestation

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	18 / 28



Brouillon et interface de l'édit de masse

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	19 / 28

4.5.3. Code

4.5.3.0. Préface

Pour ce projet, j'ai utilisé PHP avec le framework Laravel 10 en backend et pour le frontend j'ai utilisé une combinaison de blade (syntaxe particulière prise en charge par Laravel qui permet de facilement mélanger html et php) et tailwind css (un framework css qui génère le css automatiquement en fonction des classes qui sont utilisés).

Avant de détailler le code dans son entièreté, je pense qu'il est important d'expliquer les points les plus importants de Laravel.

Tout d'abord, Laravel est un framework php orienté objet. C'est-à-dire que tout (à l'exception des fonctions natives php) est un objet. Dans notre application beaucoup d'objets sont utilisés car c'est extrêmement pratique et ça rend le processus très rapide, en tant que développeur et en tant qu'utilisateur final. Laravel est aussi, comme la plupart des framework php, (Symfony, Codeigniter, etc...) basé sur un fonctionnement "MVC". C'est une manière de dire qu'on sépare le cerveau (Controller) du coeur (Model) et de l'interface (View).

La requête passe donc par ce processus de MVC mais regardons plus en détail comment une requête est gérée dans Laravel.

Supposons que vous utilisiez Laravel pour construire votre application et que vous construisez une page "contact us". La fonction "contact us" comprend généralement deux requêtes, une requête GET pour afficher le formulaire et une requête POST pour traiter les données soumises.

Chaque requête passe par le routeur, dans Laravel le routeur est un fichier appelé web.php, dans ce fichier Laravel détermine que Controller va traiter la requête en se basant sur le chemin par exemple /contact-us et la méthode de requête par exemple GET.

Une fois dans le Controller, c'est là qu'on va traiter notre requête. Le code peut être n'importe quoi, vous pouvez mettre à jour la base de données, envoyer un email, envoyer un message SMS, ajouter deux nombres, peu importe ce que fait votre application, mais dans notre cas, le contrôleur renverra le formulaire de contact ou validera les données et enverra un email. Le routeur a donc dirigé la requête vers un contrôleur, et le contrôleur a fait la partie logique de l'application, nous devons maintenant renvoyer un retour à l'utilisateur.

La valeur renvoyée peut être de nombreuses choses, nous pouvons renvoyer une "view" (fichier HTML/Blade) si la requête était destinée à afficher une page, par exemple une requête GET sur le chemin /contact-us, ou alors, nous pouvons renvoyer une redirection HTTP avec un message de succès ou d'erreur.

Si on décompose notre "MVC", notre application comporte

- 10 Modèles
- 17 Views/Interfaces
- 5 Controllers

Il est à noter que dans cette partie, étant donné la quantité de code présente dans le projet, des extraits de code seront fournis dans ce rapport. Les sous parties suivantes ne sont présentes uniquement pour aider le lecteur à comprendre le code qui a été écrit. Le code du logiciel est disponible à l'adresse suivante : <https://r.mrkm.dev/alsatis/code>

4.5.3.1. web.php

Le fichier web.php est un peu comme la colonne vertébrale de notre application car il fait office de routeur. Pour chaque requête il va dire quel controller appeler. Voici un extrait du web.php de notre application :

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	21 / 28

```

01. Route::get('/', function () {
02.     return redirect("/login");
03. });
04.
05. Route::get("/logout", [UserController::class, "processLogout"])
06. Route::get("/login", [UserController::class, "showLogin"])->middleware("guest");
07. Route::post("/login", [UserController::class, "processLogin"])->middleware("guest");
08.
09. //EntrepriseController (devis)
10. Route::middleware(["slash", "validClientId", "auth"])->group(function () {
11.
12.     Route::get("/dashboard", [EntrepriseController::class, showDashboard"])
13.         ->withoutMiddleware(["validClientId"]);
14.     Route::get('/new', [EntrepriseController::class, "showCreate"])
15.         ->withoutMiddleware(["validClientId"]);
16.     Route::post('/new', [EntrepriseController::class, "processCreate"])
17.         ->withoutMiddleware(["validClientId"]);
18.     Route::get("/delete/{id}", [EntrepriseController::class, "deleteAll"]);
19.     Route::get("/delete/{id}/devis", [EntrepriseController::class, "deleteAllDevis"]);
20.
21.     Route::prefix('/edit/{id}/')->group(function () {
22.         //exemple.com/edit/1
23.         Route::get('', function ($id) {
24.             return redirect("/edit/$id/fiche");
25.         });
26.
27.         //Plus de routes ici
28.
29.     });
30. });

```

Dans ces 30 lignes il se passe beaucoup de choses donc si on découpait ce code en blocs et qu'on expliquait en détail ce que chaque bloc fait ?

Déjà de la ligne 1 on déclare une route avec la méthode get, la syntaxe est plutôt simple. "get" est une fonction de l'objet Route. Dans la fonction on va spécifier 2 paramètres, un chemin et une action. Le chemin doit forcément être un string et ça va être ce qui va venir après notre nom de domaine ou votre IP. Dans notre ça c'est "/". En d'autres termes, on va exécuter l'action de cette route si on est sur <http://exemple.com/> L'action, dans ce cas, va être une fonction anonyme. C'est-à-dire une fonction qu'on va déclarer sur le tas. Dans Laravel on va très peu utiliser ce genre de syntaxe car tout orienté objet. C'est juste que dans ce cas, c'est plus rapide et plus pratique. Dans cette fonction, on va directement renvoyer une redirection. Et voilà ! Maintenant on sait comment déclarer une route avec la méthode get et la rediriger !

Maintenant, si on s'intéresse à la ligne 5, on voit que c'est plutôt similaire que ce qu'on vient de voir. On déclare une route en get, avec un chemin (<http://exemple.com/logout>) mais ici l'action est différente. Pour cette route (et les prochaines), on va donner comme action un Controller avec une fonction. Concrètement, ça veut dire que quand on visitera cette url, Laravel va exécuter la fonction "processLogout" qui est présente dans le UserController.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	22 / 28

Bon, maintenant qu'on a vu comment déclarer une route qui effectue une action dans un controller, on va enfin voir le plus intéressant, exécuter une route avec un middleware. Laravel est un framework très intéressant et permet très facilement de déclarer des middlewares. Ces middlewares permettent d'effectuer un travail de vérification de la requête avant d'effectuer toute autre action. Dans notre cas, ligne 6, on déclare notre route mais à la fin, on va faire appel à un middleware. Ce middleware en particulier (guest) est un middleware que j'ai développé qui va vérifier que l'utilisateur qui effectue cette requête n'est pas connecté. Si jamais il est connecté, le middleware peut bloquer la requête et afficher une erreur ou simplement rediriger le navigateur. On parlera un peu plus des middlewares un peu plus tard.

Juste en dessous de ça, on tombe sur la ligne 7 relativement similaire à celle qu'on vient de voir à une exception près, c'est qu'au lieu d'avoir marqué "Route::get" on tombe sur "Route::post". Cela veut dire qu'il faut que la requête soit en POST pour déclencher cette route

On parlait juste avant des outils incroyables que Laravel met à disposition, on peut parler des groupes, comme on peut le voir ligne 10. Cette ligne est plutôt particulière car même si elle commence par "Route", on ne va pas venir déclarer une route mais un groupe (comme on peut le voir plus loin sur cette même ligne) de routes. Ce qui se passe c'est qu'on va déclarer une liste de middlewares qu'on va vouloir exécuter pour chaque route qui se trouve dans ce groupe. Dans notre cas, on veut que chaque route de ce groupe effectue les middlewares "slash", retire tout slash à la fin de la requête, "validClientId", qui vérifie que l'identifiant client (si présent dans la requête) est valide et "auth" qui s'assure que chaque requête provient d'un utilisateur authentifié.

Les lignes 12 et 13 sont exactement ce qu'elles décrivent, elles déclarent une route (l12) et directement la ligne 13 interdit l'utilisation du middleware "validClientId" sur cette route (car il n'y a pas d'identifiant à vérifier).

La prochaine ligne qui nous intéresse est la ligne 21. La syntaxe de cette ligne est extrêmement similaire à la ligne 10. On vient ici aussi créer un groupe de routes sauf qu'ici on ne requiert pas l'utilisation de middlewares. A la place, on va créer ce que l'on appelle un préfixe. Un préfixe permet de plus facilement créer des routes qui ont le même début. Par exemple, sur cette ligne on dit que toutes les lignes du groupe qui va suivre commencent par "/edit/{id}". Vous avez vu comment "id" est entouré d'accolades ? Dans Laravel, si jamais un mot est entre des accolades, ce mot devient une variable et Laravel va considérer qu'il peut mettre tout ce qu'il veut. Par exemple, les requêtes "/edit/test" et "/edit/8" Sont deux requêtes qui sont valides et qui rentrent dans ce groupe. D'où la présence de "validClientId" qui va vérifier que notre variable id est bien un identifiant client.

Il est aussi possible d'ajouter un préfixe et faire appel à des middlewares dans la même déclaration de groupe. A ce moment, la déclaration ressemblerait à ca :

```
01. Route::middleware(["auth"])->prefix("/users")->group(function () {
02.
03.     //Routes ici
04.
05. });
```

Il existe aussi d'autres fichiers "routeurs" comme api.php, channels.php et console.php. Respectivement ils sont utilisés si jamais une api est développée, un serveur de "message broker" (websockets) ou une application à exécuter dans une console est développée. Pendant ma mission, je n'ai pas eu besoin d'utiliser un autre routeur autre que web.php.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	23 / 28

4.5.3.2. Middlewares

Comme abordé juste avant, Laravel met en place des outils très pratiques à la disposition des développeurs pour leur faciliter la vie. Dans cette partie, je vais parler très rapidement du principe des middlewares.

Les middleware permettent de filtrer les requêtes HTTP entrant dans notre application.

Par défaut, Laravel inclut des middlewares qui permettent de vérifier si l'utilisateur est identifié ou non. Par exemple, le middleware auth redirigera tous les utilisateurs non authentifiés vers la page de connexion.

```

01. <?php
02. namespace App\Http\Middleware;
03.
04. class FiltreIpMiddleware {
05.
06.     /**
07.      * Run the request filter.
08.      *
09.      * @param \Illuminate\Http\Request $request
10.      * @param \Closure $next
11.      * @return mixed
12.      */
13.     public function handle($request, Closure $next)
14.     {
15.         if ($request->getClientIp() != '0.0.0.0') {
16.             return redirect('login');
17.         }
18.         return $next($request);
19.     }
20. }
21. }

```

Dans ce middleware que j'ai spécialement conçu pour cet exemple, on y trouve qu'une seule fonction, la fonction handle. Quand on va appeler un middleware dans Laravel, Laravel va tout seul appeler la fonction handle avec la requête en premier paramètre et la variable qui permet de "valider" la requête.

La réponse des middleware est extrêmement encadrée car c'est eux qui font suivre ou pas la requête. Un peu comme un téléphone arabe.

Cette fonction est assez explicite en elle-même. On va regarder que l'ip du client n'est pas "0.0.0.0". Dans le cas où l'ip ne correspond pas à "0.0.0.0", on renvoie une redirection vers la page de connexion (login). Si jamais l'ip est "0.0.0.0", on laisse la requête aller vers le prochain middleware. Si jamais le middleware qu'on vient d'exécuter est le dernier, la requête passera directement vers un controller.

4.5.3.3. Models

Maintenant qu'on a vu comment les routes sont gérées et comment certaines requêtes sont validées, on va enfin pouvoir s'intéresser à la première lettre de l'acronyme MVC.

L'ORM Eloquent est un composant essentiel du framework Laravel. L'acronyme "ORM" signifie "Object-Relational Mapping", qui est une technique utilisée pour faciliter l'interaction avec une base de données relationnelle en représentant les tables de la base de données sous forme d'objets dans le code source de l'application.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	24 / 28

Dans le contexte de Laravel, Eloquent permet aux développeurs de travailler avec des bases de données en utilisant des objets et des méthodes plutôt qu'en écrivant directement des requêtes SQL. Cela rend la gestion de la base de données beaucoup plus simple et intuitive.

Par exemple, dans notre projet, on va avoir le modèle "User" qui va s'occuper de gérer un utilisateur présent (dans la plupart) dans la base de données.

Un modèle est déclaré comme suit :

```
<?php

namespace App\Models;

use Carbon\Carbon;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Illuminate\Support\Facades\Cookie;

class User extends Authenticatable
{
    public function passwordVerify(string $password)
    {
        return password_verify($password, $this->password);
    }

    public static function fromToken(string $token): ?User
    {
        $token = UserToken::find($token);
        if (empty($token)) {
            return null;
        }
        return User::find($token->uid);
    }

    public function deleteAllTokens(): void
    {
        UserToken::where("uid", $this->id)->delete();
    }

    public function isAdmin(): bool
    {
        return ($this->role === "admin");
    }
}
```

Cet exemple reprend de manière réduite le vrai modèle user utilisé dans l'application. Comme on peut le voir, le modèle est en réalité une classe avec des fonctions qui lui sont propres. Dans le code, on peut l'appeler de manière suivante :

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	25 / 28

```

$user = ((User::where("email", "test@test.test")->get())[0])??null;
// ou
if (!$user->passwordVerify($request["password"])) {
    // Action à effectuer si le mot de passe est invalide
}

```

Avec les modèles, on peut écrire des requêtes sql très facilement. Dans la première ligne on cherche les utilisateurs qui possèdent le mail "test@test.test", on les récupère (->get()) et on prend le premier.

Les lignes 3 à 5 servent à vérifier le mot de passe avec la fonction définie dans le modèle.

4.5.3.4. View

Dans Laravel, une vue est un fichier qui contient du code HTML (et éventuellement d'autres balises de templating) mélangé avec des variables ou des instructions de contrôle. Les vues sont utilisées pour afficher le contenu que vous souhaitez présenter aux utilisateurs dans leur navigateur. Laravel utilise un moteur de templating appelé "Blade". Il permet d'écrire des vues de manière plus expressive et lisible grâce à des fonctionnalités telles que les directives, les boucles et les conditionnels. Les fichiers Blade ont généralement une extension ".blade.php".

Blade permet d'inclure une vue à l'intérieur d'une autre. Cela signifie que nous pouvons définir des parties réutilisables de votre interface utilisateur, appelées "partials", et les inclure dans d'autres vues. Cela favorise la réutilisation du code et simplifie la gestion des mises à jour d'interface utilisateur. Dans notre cas, j'ai fait le header et le background dans une vue que j'appelle tout le temps. Cela me permet de me faire gagner du temps lors du développement et de ne pas avoir "re-déclarer" le css et le js.

Ces vues peuvent aussi recevoir des données à partir du contrôleur qui gère la requête. Cela permet de transmettre des informations dynamiques à la vue pour qu'elle les affiche. Par exemple, lorsque j'affiche toutes les prestations, c'est le controller qui donne à la vue un array que la vue affiche grâce à un foreach.

En parlant des directives, Blade propose différentes directives pour faciliter la gestion des structures de contrôle dans les vues. Voici quelques directives couramment utilisées :

@if, @else, @elseif, @endif: Pour les blocs conditionnels.

@foreach, @endforeach: Pour les boucles.

@for, @endfor: Pour les boucles "for".

@include: Pour inclure d'autres vues/partials.

{{ }}: Pour afficher des variables ou des expressions dans le HTML. (remplace le <?= ?> et est bien plus sécurisé que ce dernier)

4.5.3.5. Controller

Un contrôleur est une classe PHP qui gère la logique de l'application en réponse à une requête HTTP. Chaque contrôleur est associé à une ou plusieurs actions (méthodes) qui définissent le comportement lorsque l'application reçoit une requête spécifique. Les contrôleurs sont activés lorsque votre application reçoit une requête HTTP pour une route spécifique. Le système de routage de Laravel associe les URL aux méthodes de contrôleur correspondantes pour répondre aux différentes actions.

Les contrôleurs peuvent accéder aux données envoyées par le client via les paramètres de la requête (query paramètres, données de formulaire, etc.) et les utiliser pour effectuer des opérations. Par exemple, un contrôleur peut recevoir les informations d'un formulaire d'inscription, valider les données et les enregistrer dans la base de données. Les contrôleurs interagissent souvent avec les modèles pour accéder à la base de données et récupérer ou modifier les informations stockées. Cela permet de séparer la logique de base de données du contrôleur et de garder le code plus organisé.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	26 / 28

Une fois que le contrôleur a traité la requête et obtenu les données nécessaires, il peut renvoyer une réponse adaptée. Cela peut être sous la forme d'une vue, où le contrôleur passe les données nécessaires à la vue pour qu'elle les affiche, ou sous la forme d'une réponse JSON, utile pour les applications front-end qui consomment des API.

Etudiant	MAGUEUR Marc	Tuteur	GUINAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	27 / 28

5. Fiche 5 : Retour d'expérience

5.1. Remerciements

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui m'ont accompagné tout au long de mon stage au sein du support et service technique d'Alsatis Entreprise.

Je souhaite adresser mes remerciements les plus chaleureux à mon maître de stage, Franck GUIGNAND, Responsable Technique au sein d'Alsatis. Sa présence et ses conseils avisés ont grandement contribué à enrichir mon expérience professionnelle et à mener à bien mes missions.

Je voudrais également exprimer ma reconnaissance envers toute l'équipe du support technique, avec qui j'ai été en contact durant mon stage. Maxime BETEMPS, Charles BANQUET, Tristan TISON, Quentin DE GRENIER et Maxime DOBIGNY ont été des collègues exemplaires, toujours prêts à partager leurs connaissances et à m'apprendre toujours plus sur du réseau ou du support.

Enfin, j'ai aussi envie de remercier d'une manière particulière Alain ZARRAGOZA qui était mon maître de stage lors de mon stage d'observation de seconde et qui m'a particulièrement aidé et conseillé sur mon parcours personnel et professionnel. C'est lui qui m'a donné envie de faire mon stage à Alsatis et c'est lui qui m'a mis en relation avec Franck GUIGNAND.

Ce stage a été une expérience enrichissante et formatrice, et je suis profondément reconnaissant envers toutes les personnes qui ont contribué à sa réussite.

5.2. Pistes de progrès

Ce stage m'a permis d'ouvrir les yeux sur certaines de mes pratiques qui n'étaient pas forcément très utilisées en entreprise. Les pistes de progrès que j'ai réussi à dégager de ce stage sont :

- L'importance d'un cahier des charges défini en dur avant la création du projet
- La documentation du code (je documentais pas forcément énormément mon code avant et ce n'était pas forcément très lisible)
- Renforcer la validation des entrées utilisateur pour éviter les erreurs potentielles et garantir la sécurité de l'application
- Améliorer la gestion des erreurs et des exceptions pour fournir des messages d'erreur clairs et informatifs aux utilisateurs.
- Travailler sur l'expérience utilisateur en améliorant l'ergonomie et le design de l'interface (mes interfaces ne sont pas très travaillées et certaines peuvent être hostiles aux nouveaux utilisateurs)

Etudiant	MAGUEUR Marc	Tuteur	GUIGNAND Franck	Entreprise	Alsatis Entreprises
Année Scolaire	2022/2023	Section	1SIO SLAM	Numéro de page	28 / 28