



# Clarity Notes

## Projet S2

### Rapport Soutenance 2

Lucas Besson, Khôren Pasdrmadjian  
Vianney Marticou, Romain Vallette-Grisel

Professeur référent : Nouha Chaoued

Avril 2022

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Présentation de ClarityNotes</b>	<b>4</b>
<b>3</b>	<b>Compatibilité Windows</b>	<b>5</b>
<b>4</b>	<b>Gestion du serveur</b>	<b>6</b>
4.1	Installation de l'OS sur le serveur . . . . .	6
4.2	Installation d'Apache sur le serveur . . . . .	6
4.3	Nom de domaine et configuration DNS . . . . .	7
4.4	Installation de la BDD MySQL sur le serveur . . . . .	7
4.5	Installation de phpMyAdmin sur le serveur . . . . .	7
4.6	Création des tables sur la BDD . . . . .	8
4.6.1	Table Utilisateurs . . . . .	8
4.6.2	Table Répertoires . . . . .	9
4.6.3	Table Notes . . . . .	10
4.7	Connexion de la BDD à l'application . . . . .	10
<b>5</b>	<b>Présentation des différentes pages</b>	<b>11</b>
5.1	Login Page . . . . .	11
5.2	CreateAccount Page . . . . .	12
5.3	Root Page . . . . .	12
5.4	RemoveChapter/RemoveNote Page . . . . .	13
5.5	AddChapter/AddNote Page . . . . .	13
5.6	EditorNote Page . . . . .	14
5.7	Settings Page . . . . .	15
5.8	Liaisons entre les différentes pages . . . . .	15
<b>6</b>	<b>Site Internet</b>	<b>16</b>
6.1	Installation du template sur le serveur Web . . . . .	16
6.2	Conception graphique . . . . .	16
6.3	Edition du template . . . . .	17
<b>7</b>	<b>Organisation et avancement</b>	<b>18</b>
7.1	Répartition des tâches . . . . .	18
7.2	Avancement du projet . . . . .	18
7.3	Objectif pour la soutenance finale . . . . .	19
<b>8</b>	<b>Conclusion</b>	<b>20</b>

# 1 Introduction

Au cours de notre deuxième semestre au sein de l'EPITA, les étudiants sont amenés à réaliser un projet en groupe de trois ou quatre. Nous quatre, Lucas Besson, Khôren Pasdradjian, Vianney Marticou et Romain Vallette-Grisel, nous nous sommes réunis en décembre 2021 pour former un groupe.

Nous avons fait le choix de réaliser une application mais nous reviendrons sur la présentation de celle-ci dans la partie suivante. Ce rapport de seconde soutenance a pour but de présenter notre projet, de montrer notre avancement et d'évoquer les difficultés et les facilités rencontrées au cours de la réalisation de ce projet scolaire.

Nous profitons de cet introduction pour notifier que Lucas Besson nous a quitté le vendredi 18 février 2022. En effet, Lucas Besson a pris la décision de ne plus suivre sa scolarité à EPITA et de se tourner vers de nouvelles aventures. Nous tenons à le remercier pour la contribution qu'il a eu pour le projet (aide pour trouver l'idée de ce projet, aide à la conception du cahier des charges, et aide à l'installation du serveur). Son départ a nécessité un réajustement de la répartition des tâches et de nos objectifs finaux, nous reviendrons sur cela en temps et en heure dans ce rapport.

Au cours de ce rapport de deuxième soutenance, nous réaliserons dans un premier temps une présentation du projet ClarityNotes puis nous évoquerons comment nous avons rendu l'application compatible avec Windows. Par la suite, nous expliquerons comment nous avons administré notre serveur (installation serveur Web, base de données, configuration DNS...). Après avoir traité cet aspect, nous vous présenterons les différentes pages de notre application et évoquerons les difficultés rencontrées pour les mettre en place. À la suite de cela, nous vous montrerons notre site Internet et expliquerons sa conception. Nous finirons ce rapport en évoquant la phase organisationnelle de notre groupe (répartition des tâches et objectif pour la soutenance finale). Enfin, nous conclurons notre rapport de soutenance.

À la demande du jury, nous joignons les deux répertoires Git de notre projet :

- Site Internet : `git@github.com:Laregledujeu/ClarityNotes-Web.git`
- Application : `git@github.com:Laregledujeu/clarity_notes.git`

## 2 Présentation de ClarityNotes

Au cours de notre premier semestre au sein de l'EPITA, nous avons remarqué que le partage de prise de notes en sein des élèves de la classe n'était pas toujours optimal et centralisé. Notre projet de second semestre est donc de réaliser une application qui permettrait de partager et de créer des notes à un ensemble de personnes défini. En effet, nous avons pu voir à quel point la diversité des méthodes d'échange de données peut constituer une barrière pour les étudiants dont les systèmes d'information ne sont pas leur principal objet d'étude. Cela nous a donc convaincu du fait qu'une application qui permettrait de gérer et centraliser ces échanges de manière simple et concise pouvait s'avérer d'une grande utilité. Nous assumons pleinement et avec force, avoir pris le choix de réaliser une application plutôt qu'un jeu-vidéo comme la majorité de nos camarades de classe.

ClarityNotes est une application développée en C# avec l'aide de Xamarin. Cependant, qu'est-ce que le C# et Xamarin ? Le C# est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET. Quant à lui, Xamarin est un outil de développement cross-platform qui permet de générer des versions natives iOS, Android et Windows d'une même application en utilisant un seul langage de programmation (en l'occurrence ici le C#).

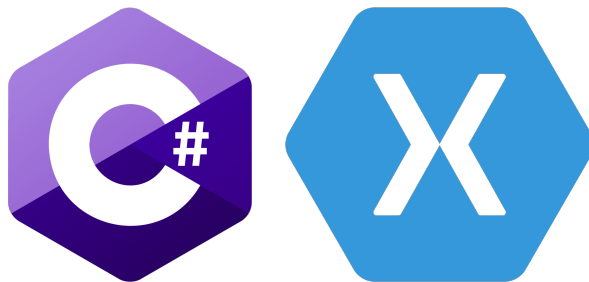


FIGURE 1 – Logo C# et Logo Xamarin

ClarityNotes propose les fonctionnalités suivantes à ses utilisateurs :

- L'utilisateur peut se créer un compte, s'y connecter et le supprimer.
- L'utilisateur peut organiser ses notes par l'intermédiaire de répertoires qu'il a la possibilité de créer et de supprimer.
- L'utilisateur peut créer, éditer, visionner et supprimer des notes.
- Les notes peuvent être mises en forme (italique, soulignage, couleurs, barré).
- Toutes les données relatives aux utilisateurs, aux répertoires et aux notes sont stockées sur une base de données MySQL sécurisée.
- La possibilité d'accéder à un site Internet présentant notre application, ses fonctionnalités et affichant des avis et retours d'expérience d'utilisateurs.

### 3 Compatibilité Windows

Au début du projet, l'accord nous avait été donnée de réaliser notre application sous format mobile. Cependant, nous ne savions pas que nous étions contraint également de réaliser une version qui soit compatible avec le système d'exploitation Windows. Nous avons appris cette contrainte lors de la première soutenance qui s'est déroulée en avril dernier.

Afin de pouvoir rendre l'application compatible avec Windows, nous avons dû réaliser une refonte totale de notre application. En effet, notre façon de développer et de réaliser notre application rendait l'application fonctionnelle avec les appareils Android mais pas sur Windows.

Pour contrer ce soucis de compatibilité, nous avons dû utiliser les variables d'environnement de Xamarin, que nous ne connaissions pas jusqu'à présent. En effet, il existe une variable nommée `app.current.mainpage.height`. Cette dernière permet de récupérer la hauteur de l'écran qui utilise l'application. Grâce à cela, nous pouvons désormais faire varier la taille du texte et par la même occasion de tous les composants graphiques (tel que les boutons par exemple) en fonction de cette variable. Néanmoins, nous avons dû re-implémenter toutes les interfaces de l'application.

Cette refonte globale, nous a permis de réorganiser notre code de façon plus propre et plus conventionnelle. Avant la refonte globale, nos variables avaient des noms peut clairs, et le code était peu aéré. La relecture du code n'était donc pas évidente, surtout lorsque le code était écrit par un autre membre du projet, et que nous n'avions pas connaissance de l'intérêt du code écrit.

Étant donné que nous sommes dans l'obligation de rendre notre application fonctionnelle sur Windows et qu'une comptabilité avec les versions mobiles n'est pas obligatoire et secondaire, nous avons pris la décision de nous concentrer et de mettre notre force de travail au profit d'un fonctionnement sous Windows plutôt que sous Android et iOS. Cependant, nous essayerons tout de même de fournir une version mobile fonctionnelle afin de permettre à nos utilisateurs d'avoir la possibilité d'utiliser ClarityNotes sur ordinateur ainsi que sur mobile.

## 4 Gestion du serveur

Pour ce projet, nous sommes obligés d'utiliser un serveur. En effet, nous avons besoin d'héberger un site Internet présentant notre application et nous avons besoin de stocker les informations des utilisateurs et les notes de nos utilisateurs dans un espace centralisé. Notre but initial était d'auto-héberger le serveur chez un des membres de notre groupe sur un Raspberry Pi. Cependant, un problème de taille s'est présenté à nous, celui de l'adresse IP. En effet, tous les appareils reliés à Internet possède une adresse IP publique. Cette adresse IP est comparable à une adresse postale. Si je vous envoie une lettre, j'indique votre adresse sur l'enveloppe. Cela permet au facteur de savoir où déposer la lettre. En informatique, pour faire communiquer deux périphériques appartenant au même réseau, nous utilisons leur adresse IP. Or notre soucis, dans le cas où nous auto-hébergeons notre serveur, est que les fournisseurs d'accès à Internet fournissent des adresses IP dynamiques. C'est à dire que l'adresse IP change fréquemment. Il était donc impossible d'auto-héberger notre serveur car nous avons besoin d'un service proposant une adresse IP statique (en opposition à une adresse IP dynamique). Nous avons donc finalement opté pour louer à un hébergeur (tel qu'OVH par exemple) un VPS (Virtual Private Server) qui possèdera une adresse IP statique.

### 4.1 Installation de l'OS sur le serveur

Pour le système d'exploitation du serveur, nous avons opté pour Ubuntu version serveur. Étant donné que nous n'avons pas besoin de connaissances dans la gestion et l'installation de serveur, nous avons pris la solution de facilité. En effet, Ubuntu serveur est un OS stable et fiable donc nous nous sommes tournés vers cette distribution Linux. Après quelques recherches Google, nous avons trouvé un guide d'installation, qui nous a bien aidés pour installer cet OS sur notre VPS.

### 4.2 Installation d'Apache sur le serveur

Comme dit précédemment, nous avons besoin d'un site internet pour présenter notre application à de potentiels nouveaux clients. Pour cela, nous avons besoin d'être en capacité d'héberger notre site Internet sur notre VPS. Pour se faire, nous avons dû installer Apache HTTP Server : un logiciel libre permettant d'héberger un site Internet sur un serveur. Pour se faire, nous nous sommes cette fois-ci référés à une vidéo YouTube expliquant comment installer ce logiciel libre sur un VPS. Nous tenons à ajouter qu'avant d'installer Apache, nous nous étions tournés vers NGINX. Cependant, ayant eu des soucis de compatibilité avec ce dernier, nous avons désinstallé NGINX pour finalement utiliser Apache.

### 4.3 Nom de domaine et configuration DNS

Grâce au VPS que nous louons chez un hébergeur, nous possédons une adresse IP statique. Cependant, pour des raisons de simplicité, nous n'allons pas demander aux personnes souhaitant accéder à notre site Internet de taper dans leur moteur de recherche une adresse IP (composée uniquement de chiffres et de points). Pour remédier à cela, nous avons fait eu à un nom de domaine. L'un des membres du groupe possédait déjà un nom de domaine, nous avons donc tout simplement ajouter un sous-domaine redirigeant vers l'IP de notre VPS. L'utilisateur n'a pas à écrire `http://83.150.217.50` mais `http://claritynotes.romainvltgr.fr` dans son moteur de recherche. Bien qu'écrire directement l'adresse IP envoie aussi sur notre site Internet, utiliser un nom de domaine à l'avantage d'être plus facile à écrire et plus facile à retenir.

Vous allez ajouter l'entrée suivante dans votre zone DNS :	
Type de champ	A
Domaine	claritynotes.romainvltgr.fr.
Cible	83.150.217.50

FIGURE 2 – Configuration de la redirection DNS vers l'IP du VPS

### 4.4 Installation de la BDD MySQL sur le serveur

Dans le cadre de notre projet, nous avons besoin de stocker les informations et les notes de nos utilisateurs dans un espace centralisé. Pour se faire, nous avons installé MySQL sur le VPS. MySQL est un système libre de gestion de bases de données relationnelles SQL développé et supporté par Oracle. Une base de données peut-être comparée à un ensemble de fichiers Excel dans lequel nous avons des tables (comparable à des tableaux Excel) dans lesquelles nous pouvons stocker de façon organisée tout type de données (du texte, des nombres, des dates...).

### 4.5 Installation de phpMyAdmin sur le serveur

Maintenant que nous avons installé la base de données MySQL sur le serveur, nous allons installer phpMyAdmin. Cependant, vous vous posez sûrement la question suivante : "Qu'est-ce que phpMyAdmin ?". Pour résumer, phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL. En d'autres termes, cela nous permet, par l'intermédiaire d'une interface Web, d'administrer notre base de données. L'installation s'est réalisée sans difficultés majeures. Pour se faire, nous avons suivi un tutoriel.

## 4.6 Création des tables sur la BDD

Maintenant que nous avons installé la base de données MySQL sur notre serveur et que nous avons phpMyAdmin afin de l'administrer, nous devons créer des tables afin de stocker les données nécessaires au bon fonctionnement de notre application ClarityNotes. Pour se faire, nous avons créé trois tables : Users, Directories et Notes. Une table est un ensemble de données organisé sous forme d'un tableau où les colonnes correspondent à des catégories d'information (une colonne peut stocker des numéros de téléphone, une autre des noms...) et les lignes à des enregistrements, également appelés entrées.

Notre soif d'apprendre nous a poussé à ne pas nous tourner vers la simplicité en créant les tables grâce à l'interface prévue à cet effet sur phpMyAdmin mais plutôt en réalisant des requêtes SQL (Structured Query Language). De plus, il nous était indispensable de nous familiariser avec ce langage car nous allions en avoir besoin prochainement pour faire interagir la base de données MySQL avec notre application. Pour nous familiariser avec le langage SQL, nous avons utilisé les sites <https://sql.sh> et <https://www.w3schools.com> qui regorgent d'exercices et d'explications concernant les différentes requêtes SQL existantes. Nous avons aussi regardé des vidéos YouTube et lu des tutoriels sur Internet.

```
CREATE TABLE users(  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    username VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    password VARCHAR(255) NOT NULL);
```

FIGURE 3 – Exemple de requêtes SQL afin de générer une table

### 4.6.1 Table Utilisateurs

Notre application nécessite la création d'un compte utilisateur afin d'être utilisée. Pour se faire, nous devons donc stocker dans notre base de données MySQL les informations concernant les utilisateurs. Au sein de notre groupe, les avis ont été divers concernant quels types d'informations nous devons collecter sur nos utilisateurs. Certains souhaitent se résoudre au strict minimum défendant ainsi la protection de la vie privée et d'autres souhaitaient stocker beaucoup d'informations telles que le genre, l'âge, le prénom de nos utilisateurs, dans le but de pouvoir mieux cerner quel type d'individus cible notre application, et ainsi mieux cibler par la suite nos campagnes publicitaires. Nous nous sommes résolus à mettre en avant la protection de la vie privée de nos utilisateurs, et pour mieux gagner leur confiance, nous ne collectons donc que les données fondamentales en se contentant du minimum.



Dans notre table Utilisateurs, nous stockons donc un identifiant pour chaque utilisateur. Chaque utilisateur a un identifiant qui est différent des autres utilisateurs. Nous stockons aussi l'adresse email de l'utilisateur ainsi que son nom d'utilisateur et son mot de passe. Toutes ces données sont récupérées auprès de l'utilisateur lorsqu'il fait le choix de se créer un compte sur notre application, afin d'utiliser nos services.

id	username	email	password
1	romainvltgr	romain.vallette-grisel@epita.fr	ZqJpDU2y348p
2	vianneymartcou	vianney.martcou@epita.fr	b28i93WHRhjX
3	khorenpasdrmadjian	khoren.pasdrmadjian@epita.fr	n5Q29jrB6PNe

FIGURE 4 – Représentation de la table SQL Users

#### 4.6.2 Table Répertoires

Notre application doit proposer à nos utilisateurs un moyen simple de classer leurs notes afin de rester organisés. En effet, beaucoup de nos utilisateurs aiment être organisés et cette fonctionnalité est donc importante. Nous avons donc décidé de mettre en place un système de rangement par l'intermédiaire de répertoires. Un répertoire est comme un dossier pouvant être vide, ou contenir des notes. Les utilisateurs auront la possibilité de créer des répertoires afin d'organiser leurs notes, ou au contraire, de supprimer des répertoires déjà existants. Nous reviendrons sur ces fonctionnalités plus tard dans notre rapport de soutenance.

Afin d'avoir connaissance des répertoires existants, nous avons donc dû créer une table Répertoires. Dans notre table Répertoires, nous stockons donc un identifiant pour chaque répertoire. Chaque répertoire a un identifiant qui est différent des autres répertoires. Nous stockons aussi le titre du répertoire ainsi que sa date de création et l'identifiant de l'utilisateur ayant créé le répertoire.

id	title	creation_date	creation_id_author
1	EPITA	01-04-2022 00:00:00	1
2	Personnel	02-04-2022 12:30:00	1
3	Projet	03-04-2022 15:27:00	1

FIGURE 5 – Représentation de la table SQL Directories

### 4.6.3 Table Notes

L'objectif principal de notre application est de pouvoir créer, éditer, organiser et supprimer des notes. De plus, les notes seront partagées entre les différents utilisateurs de notre application. Nous avons donc dû stocker les notes dans notre base de données MySQL. Pour se faire, nous avons créé à nouveau une table, qui cette fois-ci contiendra toutes les notes de nos utilisateurs.

Dans notre table Notes, nous stockons donc un identifiant pour chaque note. Chaque note a un identifiant qui est différent des autres notes. Nous stockons aussi l'identifiant du répertoire contenant la note, le contenu de la note, la date de création de la note, la date à laquelle la note a été mise à jour pour la dernière fois, l'identifiant du créateur de la note et enfin l'identifiant de l'utilisateur ayant pour la dernière fois édité la note.

id	title	id_directory	content	creation_date	update_date	creation_id_author	update_id_author
1	Cours Maths	1	1 + 1 = 2 <=> 2 = 2	07-04-2022 17:57:35	08-04-2022 06:35:48	1	1
2	Conseil	2	Pensez à boire.	07-04-2022 18:14:15	08-04-2022 05:28:23	1	1
3	Soutenance	3	ClarityNotes	07-04-2022 19:41:07	08-04-2022 12:29:32	1	1

FIGURE 6 – Représentation de la table SQL Notes

## 4.7 Connexion de la BDD à l'application

Maintenant que nous avons une base de données MySQL, phpMyAdmin afin de l'administrer facilement et que les tables (utilisateurs, répertoires et notes) soient créées, il est temps de connecter la base de données MySQL à l'application.

Par défaut, le langage C# ne propose pas de fonctionnalités permettant de communiquer avec une base de données MySQL. Pour remédier à ce soucis majeur, nous avons cherché sur Internet une alternative et nous avons découvert les outils clients NuGet. NuGet est le gestionnaire de packages pour .NET. Les outils clients NuGet permettent de produire et de consommer des packages. Nous avons trouvé un package intitulé `MySql.Data` qui permet de communiquer en C# sous la forme de requête SQL avec notre base de données MySQL.

Il existe deux types de requêtes SQL. Celles qui consistent à interroger notre base de données pour obtenir des informations (sur un utilisateur, un répertoire ou une note dans notre cas). Ces requêtes commencent par le mot clef **SELECT**. Enfin, il y a les requêtes SQL qui permettent de mettre à jour la base de données (ajouter, supprimer, mettre à jour des champs de valeur). Quant à elles, ces requêtes commencent par l'un des mots clefs suivants : **INSERT**, **DELETE** ou **UPDATE**. Les requêtes SQL peuvent aussi contenir des conditions grâce au mot clef **WHERE** suivi de la condition afin de ne pas cibler systématiquement tous les éléments de la table.

En nous documentant sur le langage SQL, nous avons découvert l'existence des injections SQL. Les injections SQL sont des méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données. Elles permettent d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et ainsi en compromettre la sécurité. La sécurité des données de nos utilisateurs étant l'une de nos priorités, nous avons tenté de trouver une solution pour empêcher d'être vulnérable à ce type d'attaques. Nous avons découvert, en lisant la documentation du package NuGet `MySQL.Data` que des fonctionnalités déjà présentes permettent d'éviter ces attaques. Nous les avons donc utilisées.

## 5 Présentation des différentes pages

Comme la majorité des applications existantes, notre application ClarityNotes est composée de plusieurs pages ayant chacune un rôle défini et leurs propres fonctionnalités. Nous allons dans cette partie du rapport de soutenance vous présenter les principales pages (Login Page, CreateAccount Page, Root Page, ...) et interfaces de notre application et évoquer les fonctionnalités majeures de chacune d'entre elles. Nous évoquerons aussi les difficultés et les facilités rencontrées pour l'implémentation de chacune d'elles.

### 5.1 Login Page

La Login Page correspond à la toute première page que l'utilisateur rencontre lorsqu'il démarre l'application. Sur cette dernière, il a la possibilité de se connecter en utilisant son nom d'utilisateur et son mot de passe. Une fois ces champs remplis, la base de données MySQL est interrogée afin de savoir si un compte Utilisateur correspondant à ces données existe. Si c'est le cas, l'utilisateur se retrouve connecté à son compte et est redirigé vers la Root Page. Cependant si la base de données renvoie une réponse négative, un message d'erreur informe l'utilisateur qu'aucun compte ne correspond aux données entrées par ce dernier et il est invité une nouvelle fois à essayer de se connecter. Dans le cas où l'utilisateur est nouveau, et que par conséquent, il n'a pas de compte, il a la possibilité de cliquer sur un bouton intitulé **S'inscrire** qui le re-dirigera vers une autre page de notre application que nous décrirons dans la sous-section suivante.

Pour implémenter cette page de connexion, comme l'ensemble de notre application, il a fallu créer une base de données MySQL sur notre VPS. Une fois la base de données opérationnelle, il a fallu réussir à l'interroger. Pour se faire, nous nous sommes référés aux connaissances acquises pendant la période d'apprentissage du langage SQL que nous nous étions réservés. Dans le cas présent, nous avons utilisé une requête **SELECT** nous permettant de récupérer tous les comptes utilisateur, puis nous avons par la suite filtré les résultats, pour tenter de trouver un compte correspondant aux valeurs rentrées par l'utilisateur.

## 5.2 CreateAccount Page

La CreateAccount Page correspond à la page permettant à l'utilisateur de se créer un compte sur notre service afin de pouvoir, par la suite, utiliser notre application et profiter pleinement des diverses fonctionnalités offertes par notre application. Cette page est accessible depuis la Login Page en cliquant sur le bouton **S'inscrire**.

La page est constituée d'un formulaire composé de quatre champs à renseigner : un nom d'utilisateur, une adresse mail, le mot de passe et un champ afin de confirmer le mot de passe choisi. Si, au minimum, l'un des quatre champs n'est pas renseigné, le bouton **S'inscrire** n'est pas cliquable. De plus, pour qu'il devienne cliquable, le contenu des deux champs concernant le mot de passe doit être identique. Si c'est le cas, le contenu de ces deux champs devient bleu, autrement il est rouge.

La création d'un nouveau compte utilisateur nécessite cependant quelques vérifications. D'abord, deux utilisateurs ne peuvent pas avoir le même nom d'utilisateur ou la même adresse mail. De ce fait, à chaque nouvelle inscription, nous devons vérifier qu'aucun compte ne possède déjà le nom d'utilisateur ou l'adresse mail renseignés par l'utilisateur souhaitant s'inscrire. Dans le cas où un utilisateur possède déjà le nom d'utilisateur souhaité ou l'adresse mail renseignée, l'utilisateur souhaitant s'inscrire est averti par l'intermédiaire d'un pop-up avec un message d'erreur.

Nous tenons à vous dire que nous avons pris le choix de demander le moins d'informations à l'utilisateur souhaitant s'inscrire afin de préserver et de respecter au mieux sa vie privée. La vie privée de nos utilisateurs est l'une de nos priorités. À l'heure actuelle, les mots de passe de nos utilisateurs sont affichés en clair dans notre base de données MySQL. Nous souhaitons que, pour la soutenance finale, ces derniers soient hashés afin de garantir la protection des données personnelles.

## 5.3 Root Page

La Root Page est la page principale de notre application. L'utilisateur est redirigé sur celle-ci une fois qu'il s'est connecté à son compte en passant par le formulaire présent sur la Login Page.

Sur la partie gauche de cette page, nous pouvons trouver un menu constitué de boutons. D'abord, nous trouvons les boutons correspondant aux différents répertoires contenant des notes. Tous ces répertoires ont été récupéré en effectuant une requête **SELECT** vers la table **Directories** de la base de données MySQL. Ensuite, nous trouvons un bouton **+** afin de créer un nouveau répertoire. Ce bouton redirige l'utilisateur vers la AddChapter Page. De plus, nous trouvons aussi un bouton **-** permettant de supprimer un répertoire existant. Ce bouton, une fois cliqué, redirige l'utilisateur vers la RemoveChapter Page. Enfin, le dernier bouton de ce menu est le bouton **SET**, qui une fois cliqué, redirige l'utilisateur vers la page Settings Page.

Sur la partie droite de cette page, nous pouvons d'abord retrouver toutes les notes contenues dans le répertoire sélectionné. Dans le cas où aucun répertoire n'a été sélectionné par l'utilisateur, les notes contenues dans le premier répertoire s'afficheront. Si l'utilisateur ne possède aucun répertoire, un message l'invitant à en créer un s'affichera. Dans le cas où l'utilisateur consulte le contenu d'un répertoire mais que ce dernier n'a pas de note, un message invitant l'utilisateur à créer une note s'affichera. Dans le cas où le répertoire contient des notes, des boutons avec le titre de chaque note s'afficheront. Il suffit dans ce cas à l'utilisateur de cliquer sur le bouton correspondant à la note pour ainsi être redirigé vers l'Editor Page. Sous la liste des notes, deux boutons sont présents. Le premier, présent à gauche, permet de créer une nouvelle note. En cliquant dessus l'utilisateur sera redirigé vers l'Add-Note Page. Le second, présent à droite, permet de supprimer une note existante, en cliquant dessus l'utilisateur sera redirigé vers la RemoveNote Page.

## 5.4 RemoveChapter/RemoveNote Page

Les pages RemoveChapter/RemoveNote sont accessibles à l'utilisateur depuis deux boutons présents sur la Root Page. Ces deux pages permettent de supprimer respectivement un chapitre ou une note. Ces deux pages ont la même structure et le même fonctionnement, c'est pour cette raison que nous les avons réunies dans la même sous-section de notre rapport de soutenance.

Pour ces deux pages, nous récupérons, par l'intermédiaire d'une requête **SELECT** vers la base de données, la liste de tous les chapitres (pour le cas de la RemoveChapter Page) et la liste de toutes les notes (dans le cas de la RemoveNote Page). Nous avons implémenté un menu déroulant permettant à l'utilisateur de sélectionner le chapitre ou la note qu'il souhaite supprimer. Un message d'alerte informe l'utilisateur que la suppression est définitive et irrémédiable et que la suppression d'un répertoire supprime par conséquent toutes les notes le composant. Afin de supprimer les éléments, nous effectuons des requêtes **DELETE** vers la base de données. Nous ne gardons donc plus les éléments, par principe de respect pour nos utilisateurs et en corrélation avec les valeurs de notre projet.

## 5.5 AddChapter/AddNote Page

Les pages AddChapter/AddNote sont accessibles à l'utilisateur depuis deux boutons présents sur la Root Page. Ces deux pages permettent d'ajouter respectivement un chapitre ou une note. Ces deux pages ont la même structure et le même fonctionnement, c'est pour cette raison que nous les avons réunies dans la même sous-section de notre rapport de soutenance.

Avant d'avoir mis en place la base de données MySQL, nous stockions les notes de l'utilisateur en local sur son ordinateur ou son appareil mobile. Les répertoires étaient représentés par des dossiers et les notes par des fichiers au format textuel. Ainsi, nous utilisions les chemins relatifs afin d'accéder aux différents répertoires et aux différentes notes. Cependant, nous devions rendre possible la présence d'espaces dans le nom des répertoires et des notes. Ainsi, au moment de créer les répertoires et fichiers en local, nous remplacions les espaces par des sous-tirets (communément appelés *underscores*). Puis, au moment d'afficher le nom des répertoires et des notes sur notre application, nous remplacions les tirets-bas par des espaces. Cependant, ce procédé, réglant un souci, en créa un nouveau : les noms ne pouvaient plus contenir de tirets-bas, ou alors ils seraient automatiquement considérés comme des espaces. Heureusement, dès que nous sommes passés à l'utilisation de la base de données, ce souci fut réglé. En effet, le titre du répertoire ou de la note est désormais stocké sous forme de chaîne de caractères dans la table correspondante de la base de données (**Directories** dans le cas des répertoires et **Notes** dans le cas des notes).

Concernant l'aspect visuel de ces deux pages, elles possèdent chacune un champ d'entrée dans lequel l'utilisateur est invité à renseigner le nom du répertoire ou de la note qu'il souhaite créer. Il suffit ensuite de cliquer sur un bouton afin de créer le répertoire ou la note. Dans le cas où un répertoire ou une note existe déjà sous ce nom, l'utilisateur est informé par l'intermédiaire d'un pop-up lui indiquant que le répertoire ou la note n'a pas pu être créé.

## 5.6 EditorNote Page

L'EditorNote Page est l'une des pages fondamentales de notre application. L'utilisateur est redirigé sur celle-ci une fois qu'il clique sur le nom d'une note présente sur la Root Page. Elle permet de visionner, mais surtout d'éditer, la note.

Cette page est composée d'un éditeur permettant d'écrire, de souligner, colorier, barrer, mettre en gras et/ou en italique du texte. Une fois l'édition réalisée, il suffit de cliquer sur le bouton présent en haut à droite de l'éditeur. Ainsi, la note sera enregistrée sur la base de données, par l'intermédiaire d'une requête **UPDATE**.

Pour l'implémentation de l'éditeur, nous nous sommes tournés vers un package NuGet nommé **TEditor**. Nous avons donc lu la documentation du package avant de comprendre comment l'utiliser et pouvoir utiliser au mieux les fonctionnalités qu'il pouvait nous offrir. Il est très important à nos yeux que l'éditeur de notes soit assez complet pour permettre à nos utilisateurs d'avoir les principales fonctionnalités attendues dans un éditeur.

## 5.7 Settings Page

La Settings Page est accessible à l'utilisateur depuis la Root Page. L'utilisateur est redirigé sur celle-ci une fois qu'il clique sur le bouton **SET** présent dans le menu gauche. Cette page permet à l'utilisateur de régler les paramètres relatifs à son compte (gestion du mode sombre, modification du mot de passe, déconnexion, etc).

Les fonctionnalités de cette page ne sont pas encore totalement implémentées. Nous envisageons de stocker, dans notre base de données MySQL, les paramètres de chaque utilisateur dans une nouvelle table nommée **Settings**. Cette page étant encore en développement, nous envisageons de la finaliser et de la rendre totalement fonctionnelle pour la soutenance finale.

## 5.8 Liaisons entre les différentes pages

Comme nous l'avons vu jusqu'à présent, notre application est composée de nombreuses pages. Afin de mieux comprendre le lien entre les différentes pages, nous avons réalisé un graphe. Chaque noeud représente une page de notre application et chaque arrête représente la possibilité de passer d'une page à une autre. Le Login est représenté en rouge afin de le dissocier des autres, car c'est sur cette page que l'utilisateur est situé lorsqu'il démarre notre application.

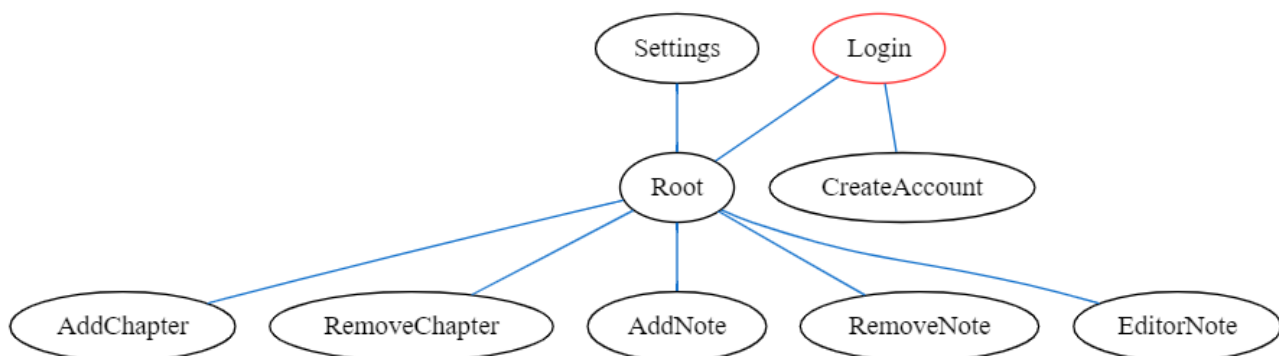


FIGURE 7 – Graphe représentant les liens entre les différentes pages de l'application

## 6 Site Internet

Dans le cadre de notre projet, il est indispensable d'avoir un site Internet faisant office de vitrine pour nos services pour faire découvrir notre application et nos services à d'éventuels futurs nouveaux clients.

Un de nos principaux soucis pour le site Internet a été le départ du projet de Lucas. En effet, Lucas était la seule personne qui avait déjà pratiqué des langages Web, tel que du HTML, CSS et du JavaScript. La tâche du site Internet lui avait donc été réservée. Nous disons bien "était" car nous nous sommes initiés aux langages du Web, de manière plus ou moins forcée. Mais c'est aussi ça un projet, une suite de soucis et de problèmes ainsi qu'une suite de solutions et de résolutions.

### 6.1 Installation du template sur le serveur Web

Pour la conception du site Internet, n'ayant que très peu de connaissances dans ce domaine, nous avons opté pour l'utilisation d'un template afin de nous guider dans la conception visuelle et esthétique du site Internet. Nous avons trouvé de nombreux templates, certains utilisant Bootstrap (une collection d'outils utiles à la création du design de sites et d'applications Web). Nous avons, dans le cadre de la première soutenance, décidé d'utiliser celui-ci : Weather Apps – Landing Page Free Bootstrap Template. Cependant, lors de la dernière soutenance, le jury nous a informé, comme dit précédemment, que nous devions nous concentrer sur la version Windows de notre application et donc mettre celle-ci en avant. Cependant, le template choisi permettait de mettre en avant uniquement la version mobile. Nous avons donc rechercher un nouveau template Bootstrap et avons opté pour celui-ci : Small Apps - Bootstrap App Landing Template. Nous avons installé le template sur un serveur Web prêté gracieusement par Vianney durant toute la durée du projet.

### 6.2 Conception graphique

La conception d'un site Internet nécessite de nombreux éléments graphiques tels que la conception d'images représentant notre application. En effet, il est évident que nous ne pouvions pas garder les images par défaut du template, étant donné qu'elles faisaient référence à une autre application que la nôtre. Nous avons eu par exemple à insérer une photo de profil que chaque membre du projet sur la page **Membres**. Ces dernières ont dû être recadrée au format 600 pixels par 600 pixels, afin de respecter le format souhaité par le template. La luminosité de nos photos ont aussi été retravaillée.

Pour concevoir nos éléments graphiques, nous avons utilisé Photoshop, un logiciel issu de la prestigieuse suite Adobe. Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur.



## 6.3 Edition du template

Avant d'éditer le template, nous avons dû comprendre comment était structuré un code HTML, CSS et JavaScript. Pour le HTML, nous avons dû nous familiariser avec la notion de balisage, dont les balises orphelines comme par exemple la balise `<a>`). Concernant le CSS, nous avons dû apprendre les notions de classe et d'identifiant, ainsi que celles de padding, border et margin.

Une fois que nous nous sommes un peu familiarisés avec ces langages, nous avons pu commencer à éditer le code HTML, CSS et JavaScript du template. Nous avons d'abord retiré un grand nombre de fonctionnalités qui ne nous intéressaient. En effet, le template Bootstrap était très complet et contenait une vingtaine de pages différentes. Nous avons décidé d'en garder cinq d'entre-elles : la page **Accueil**, la page **Membres**, la page **FAQ** et la page **Contact**. Nous avons supprimé toutes les autres pages. De ce fait, nous avons dû refaire la barre de navigation présente en haut de chaque page du site. Afin d'y mettre uniquement les pages que nous avons décidé de garder. Quant à lui, le footer a aussi été revu et simplifié. Celui du template était pas assez épuré et renvoyait vers de nombreuses pages dont nous avons pris la décision de nous séparer.

Lors de l'édition du template, nous avons dû éditer tous le texte présent afin qu'il soit en corrélation avec notre projet. Pour se faire, nous avons édité les fichiers HTML un par un. Nous avons ajouté la possibilité aux visiteurs de notre site Internet de télécharger notre cahier des charges ainsi que nos différents rapports de soutenance. Nous avons aussi rendu accessible nos répertoires GitHub au public, rendant ainsi notre projet open-source.

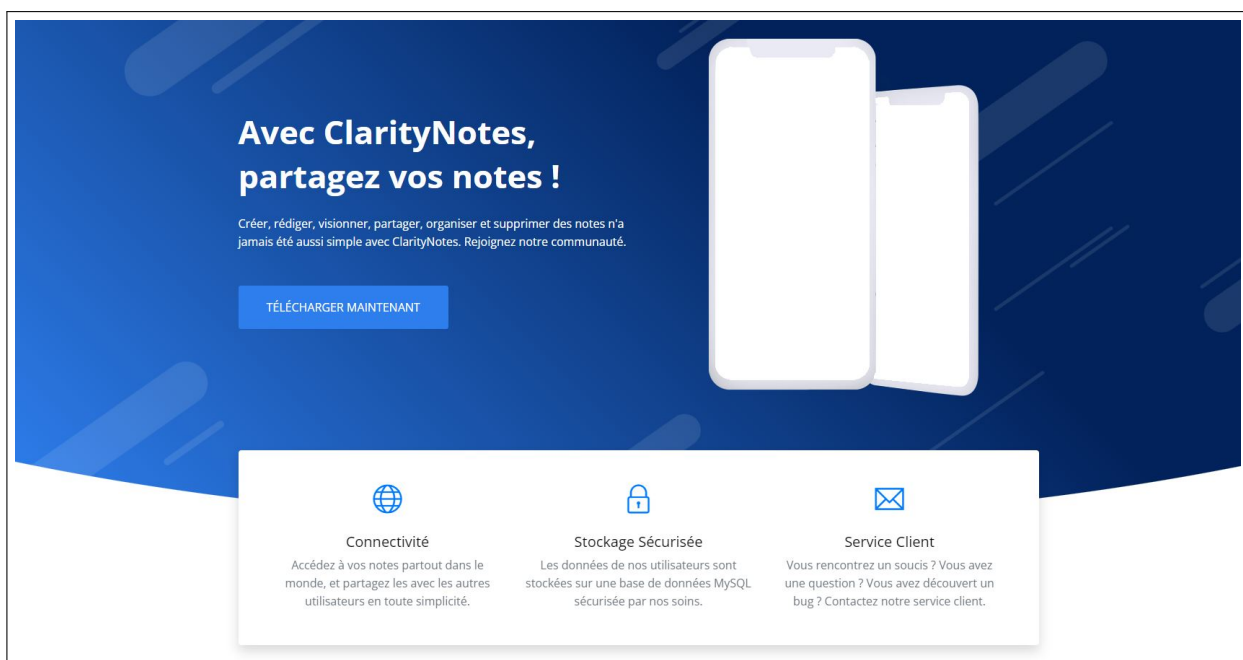


FIGURE 8 – Page d'accueil du site Internet ClarityNotes

## 7 Organisation et avancement

### 7.1 Répartition des tâches

Le projet étant copieux et dense, nous avons dû le diviser en tâches et attribuer chaque tâche à un ou plusieurs membres du groupe. Le départ de Lucas a nécessité de réorganiser notre groupe, ainsi, la répartition des tâches n'est plus la même qu'indiquée dans le cahier des charges. Le tableau ci-dessous représentent donc les principales tâches du projet ainsi que les personnes missionnées pour chacune d'elles.

<i>Tâches</i>	Khôren	Vianney	Romain
Gestion et installation du serveur VPS		X	
Gestion et connexion à la base de données			X
Implémentation des pages de l'application	X	X	X
Réalisation du site Internet			X
Implémentation de l'éditeur de notes		X	X
Implémentation de la fonctionnalité QR Code	X		
Implémentation de la fonctionnalité de traduction	X		
Mise en place des offres Gratuite / Premium	X	X	

FIGURE 9 – Tableau représentant la répartition des différentes tâches

### 7.2 Avancement du projet

Dans l'optique de mieux visualiser l'avancement de notre projet et de chacune des tâches le composant, nous avons réalisé un tableau représentant l'avancement en pourcentage de chacune des principales tâches constituant le projet. Nous pouvons ainsi constater que certaines tâches sont plus avancées que d'autres.

<i>Tâches</i>	Avancée (en %)
Gestion et installation du serveur VPS	100 %
Gestion et connexion à la base de données	75 %
Implémentation des pages de l'application	80 %
Réalisation du site Internet	85 %
Implémentation de l'éditeur de notes	100 %
Implémentation de la fonctionnalité QR Code	0 %
Implémentation de la fonctionnalité de traduction	0 %
Mise en place des offres Gratuite / Premium	0 %

FIGURE 10 – Tableau représentant l'avancement de chacune des tâches

### 7.3 Objectif pour la soutenance finale

Dans cette partie de notre rapport de soutenance, nous allons évoquer nos objectifs pour la soutenance finale. D'abord, nous tenons à dire que nous avons décidé de garder les fonctionnalités les moins importantes à implémenter pour la dernière soutenance, comme la fonctionnalité de traduction intégrée, la gestion des QR codes, ainsi que la mise en place de deux offres pour les utilisateurs, une payante et une gratuite. Vous trouverez ci-dessous le détail concernant les deux offres que nous souhaitons mettre en place.

Offre premium	Offre gratuite
Mode sombre et mode clair disponibles	Mode clair uniquement
Pas de publicité	Une publicité à la connexion
Création de notes illimité	Création de notes limité
Création de répertoires illimité	Création de répertoires limité

FIGURE 11 – Tableau présentant l'offre gratuite et l'offre payante

Nous devons aussi implémenter la page Settings accessible depuis le bouton **SET** présent sur la Root Page, ainsi que toutes ses fonctionnalités. Nous aimerions pour la soutenance finale avoir une interface utilisateur agréable, intuitive et fonctionnelle sur tout type de support (Windows ainsi que Android et iOS). De plus, nous aimerions rendre l'offre Premium activable depuis le site Internet simplement en faisant renseigner le nom d'utilisateur ou l'adresse mail de l'utilisateur dans un champ prévu à cet effet. Une requête SQL ajoutera par la suite l'offre Premium à l'utilisateur.

## 8 Conclusion

Pour conclure, on peut dire que même si nous avons eu de nombreux défis, notamment à cause de notre manque de compétences, nous avons continué à apprendre de nous-même afin de dépasser les problématiques auxquelles nous étions exposés ce qui est particulièrement valorisant.

En effet, cela n'a pas été facile de réagir au départ de Lucas, car cela a induit de répartir à nouveau les tâches au sein du groupe. De plus, comme nous avons pu l'exprimer dans ce rapport, nous avons donc dû faire face à de nombreux problèmes pour lesquels nous n'avions pas les compétences requises. Cela nous a forcé à nous surpasser pour régler ses problèmes et à trouver des solutions.

Sur le plan technique, nous avons commencé à voir émerger certaines fonctionnalités liées au serveur. Ce qui nous permis de découvrir de nombreux aspects, la configuration d'un VPS ainsi que d'une base de données. Cela rend le projet particulièrement stimulant et instructif. D'autant plus que le travail qui nous attend concernera les fonctionnalités les plus intéressantes à implémenter ce qui renforce l'aspect stimulant du projet.