



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

1	Introduction	2
2	Présentation de ClarityNotes	3
3	Compatibilité Windows	4
4	Gestion du serveur	5
4.1	Installation de l'OS sur le serveur	5
4.2	Installation d'Apache sur le serveur	5
4.3	Nom de domaine et configuration DNS	6
4.4	Installation de la BDD MySQL sur le serveur	6
4.5	Installation de phpMyAdmin sur le serveur	6
4.6	Création des tables sur la BDD	7
4.6.1	Table Utilisateurs	7
4.6.2	Table Répertoires	8
4.6.3	Table Notes	9
4.7	Connexion de la BDD à l'application	9
5	Présentation des différentes pages	10
5.1	Login Page	10
5.2	CreateAccount Page	10
5.3	Main Page	10
5.4	AddChapter/RemoveChapter Page	10
5.5	AddNote/RemoveNote Page	10
5.6	EditorNote Page	10
6	Site Internet	10
7	Organisation	10
7.1	Répartition des tâches	10
7.2	Objectif pour la soutenance finale	10
8	Conclusion	10

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 Pasdrmadjian, 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.

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'échanges 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#).

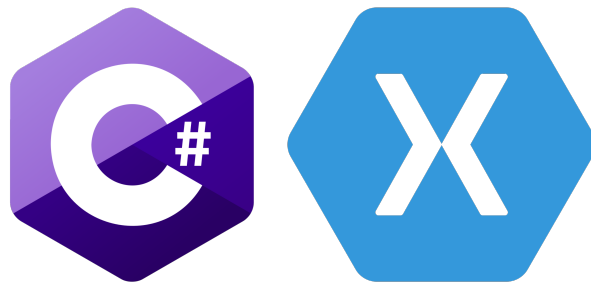


FIGURE 1 – Logo C# et Logo Xamarin

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 mise 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-héberger notre serveur, est que les fournisseurs d'accès à Internet fournisse 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 possédera une adresse IP statique.

4.1 Installation de l'OS sur le serveur

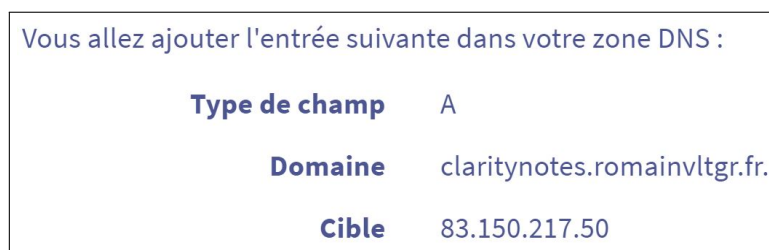
Pour le système d'exploitation du serveur, nous avons opté pour Ubuntu version serveur. Étant donné que nous n'avons pas besoin de connaissances dans la gestion et l'installation de serveur, nous avons pris la solution de facilité. En effet, Ubuntu serveur est un OS stable et fiable donc nous nous sommes tournés vers cette distribution Linux. Après quelques recherches Google, nous avons trouvé un guide d'installation, qui nous a bien aidé 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 simplicités, 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 recourt à un nom de domaine. L'un des membres du groupe possédaient 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.



The image shows a screenshot of a DNS configuration interface. At the top, it says "Vous allez ajouter l'entrée suivante dans votre zone DNS :". Below this, there is a table with three rows: "Type de champ" with value "A", "Domaine" with value "claritynotes.romainvltgr.fr.", and "Cible" with value "83.150.217.50".

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 des utilisateurs 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 organisées 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 à pousser 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, ils nous étaient indispensable de nous familiariser avec ce langage car nous allons 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 utilisé. 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 souhaitent stocker beaucoup d'informations tel 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 collectons donc finalement 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	$1 + 1 = 2 \Leftrightarrow 2 = 2$	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) sont 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 pouvant 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, notre application ClarityNotes est composée de plusieurs pages ayant chacune leur rôle et leurs fonctionnalités. Nous allons dans cette partie du rapport vous présenter les principales pages (Login Page, CreateAccount Page, Main 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

5.2 CreateAccount Page

5.3 Main Page

5.4 AddChapter/RemoveChapter Page

5.5 AddNote/RemoveNote Page

5.6 EditorNote Page

6 Site Internet

7 Organisation

7.1 Répartition des tâches

7.2 Objectif pour la soutenance finale

8 Conclusion