



Concepteur Développeur d'Applications

Dossier Projet



Concevoir et développer la persistance des données
en intégrant les recommandations de sécurité

Concevoir et développer des composants d'interface
utilisateur en intégrant les recommandations de sécurité

Concevoir et développer une application en couche répartie
en intégrant les recommandations de sécurité.

Réalisé par :
Dimitri GUEROUT
CDA06

PAE
du 27-03-2023
au 09-06-2023

Résumé

CDA06 Concepteur développeur d'applications

ENTREPRISE: DSI

Auteur: Dimitri GUEROUT

L'entreprise DSI est une entreprise adaptée, qui possède un service développement de solutions logicielles.

Cette dernière souhaite inclure dans les solutions proposées à ses clients une interface graphique qui permet de gérer les droits d'accès aux différentes méthode d'un projet de manière simple et ergonomique.

Au niveau administrateur, l'application Web aura pour rôle de saisir, modifier et limiter les accès aux différentes méthodes de l'Application.

Ce projet est réalisé avec l'**IDE Visual studio code**, en langage **PHP**.

Dans un premier temps je crée un diagramme des cas d'utilisation en **UML**. Puis je réalise un diagramme de séquence sur la création d'utilisateurs et un scénario textuel de la création d'un groupe.

Dans un deuxième temps je réalise la **rétro conception** de la base de données fournie, en utilisant le **SGBDR SQL Server**. Je réalise un **MCD** et un **MLD** de la solution qui va devoir intégrer le projet.

Je crée une Application Web avec architecture développée en couche de type **MVC**.

1. Le Modèle(Model) pour interagir avec la base de données.
2. Le contrôleur pour la logique métier.
3. La vue en langage **TWIG** de **Symfony** pour l'interface utilisateur.

Pour renforcer la sécurise j'intègre la bibliothèque **CodeIgniter\SHIELD** du framework **CodeIgniter4.0**.

J'utilise **GitLab** pour le suivi du projet et Le versioning de l'application.

Pour finir je prépare un jeu d'essai représentatif et je réalise les différents tests pour m'assurer du bon fonctionnement et la robustesse de l'application Web tout en respectant les recommandations de sécurité.

Mots-clés : **IDE** (environnement développement intégré) **Visual Studio Code**, **PHP**(hypertext precrocessor), **UML**(langage de modélisation unifier), **SGBDR**(système gestion base de donnée relationnel), **SQL**(Strutured query Language) , **MCD** (Modèle Conceptuel de données), **MLD**(Modèle Logique de données), **MVC** (Modèle-Vue-Controller), **TWIG**, **Symfony**, **CodeIgniter SHIELD**, **Framework CodeIgniter 4.0**, **GitLab**.

241 mots

Abstract

CDA06 Concepteur développeur d'applications

ENTREPRISE: DSI

Auteur: Dimitri GUEROUT

DSI is an adapted company with a software solutions development department.

The company wanted to include a graphical interface in the solutions it offered its customers, enabling them to manage access rights to the various methods of a project in a simple, user-friendly way.

At administrator level, the Web application will have the role of entering, modifying and limiting access to the various methods of the Application.

This project is being carried out using the **Visual Studio Code IDE**, in **PHP**.

First of all, I'm creating a use case diagram in **UML**. Then I created a sequence diagram for user creation and a text scenario for group creation.

I then reverse-engineer the database provided, using the **SQL Server RDBMS**, and create a **MCD** and **MLD** of the solution that will have to be integrated into the project.

I created a Web Application with a layered **MVC** architecture.

The Model to interact with the database.

The controller for the business logic.

The view in **Symfony's TWIG** language for the user interface.

To increase security, I've integrated the **CodeIgniter\SHIELD** library from the **CodeIgniter4.0 framework**.

I'm using **GitLab** to monitor the project and version the application.

Finally, I prepare a representative test set and carry out the various tests to ensure that the Web application works properly and is robust, while complying with the security recommendations.

Mots-clés : **IDE** (environnement developement intergrer) **Visual studio code**, **PHP**(hypertext precroessor), **UML**(language de modelisation unifier), **SGBDR**(systeme gestion base de donnée relasional), **SQL**(Strutured query Language) , **MCD** (Modèle Conceptuel de données), **MLD**(Modèle Logique de données), **MVC** (Modèle-Vue-Controller), **TWIG**, **Symfony**, **CodeIgniter SHIELD**, **Framework CodeIgniter 4.0**, **GitLab**.

Remerciement

Je tiens à vous remercier pour l'écoute de présentation de ce dossier.

Je voudrais dans un premier temps remercier toute l'équipe de DSI pour son accueil et son professionnalisme.

Je remercie tout particulièrement Mr Sky Gunning, pour m'avoir accepté et accompagné au cours du stage.

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce projet :

Mr Robin Bayonnove pour l'aide et le professionnalisme dont il a su faire preuve à mon égard.

Mr Ramdan AIT SLIMANI pour l'aide apportée à la rédaction de ce dossier.

Ma famille, pour leur soutien constant et leurs encouragements.

Table des matières

1 Introduction.....	11
1.1 Présentation personnelle	11
1.2 Formation CDA	12
2 Présentation de la PAE	13
2.1 Présentation de l'entreprise	13
2.2 Objectifs	13
2.3 Liste des compétences du référentiel couvertes.....	14
3 Contexte et besoins du projet.....	15
3.1 Contexte du projet.....	15
3.2 Besoins de l'entreprise.....	15
4 Analyse du besoin.....	17
4.1 Cas d'utilisation	17
4.2 MCD MLD	18
4.3 Base de données	20
5 Développement.....	24
5.1 CodeIgniter 4	24
5.2 Architecture	25
5.3 Installation de la bibliothèque CodeIgniter\Shield	26
5.4 Fonctionnalités Attendues	29
5.5 Le Model	31
5.6 Le contrôleur	34
5.7 La vue Twig	36
5.8 Le CSS.....	38
5.9 Les permissions groupe.....	40
6 Sécurité	45
6.1 Le CSRF	45
6.2 La session.....	45
6.3 Session avec BDD	48
6.4 Auth-Rates filtre	49
7 Gestion du projet	50
7.1 GitLab	50
7.2 Méthode scrum	52
8 Bilan.....	53
10.1 Apports et difficultés rencontrées	53
10.2 Perspectives	53

Table des matières

9 Partie fil rouge	54
9.1 Liste des compétences du référentiel couvertes.....	54
9.2 Les outils utilisés	54
9.3 Contexte du projet.....	54
9.4 Contexte et besoins de l'entreprise.....	55
10 Analyse du besoin.....	55
10.1 Cas d'utilisation	55
11 MLD MCD.....	56
11.1 Le MLD.....	56
11.2 Le MCD	57
12 Mise en place de la base de données.....	58
12.1 Création des tables	58
12.2 Élaboration du jeu d'essai	58
13 Réalisation	59
13.1 Requêtes les plus significatives et résultats obtenus	59
14 Gestion de la sécurité	61
14.1 Sauvegarde de la base de test	61
15 Bilan.....	62
15.1 Apports et difficultés rencontrées	62

1 Introduction

1.1 Présentation personnelle

Je m'appelle Dimitri, j'ai 32 ans, je viens de Limoges (87).

Mon parcours professionnel : j'ai obtenu un BEP carrosserie et un CAP peinture automobile (en candidat libre).

À la suite j'ai exercé différents postes de travail. Suite à un accident du travail, j'ai fait une reconversion professionnelle au Centre ESRP EPNAK de Limoges (Feret du longbois) où j'ai passé un Bac pro AMA (Artisanat et Métier d'Art) option communication visuelle plurimédia.

Après l'obtention du bac j'ai intégré le Centre de Millau (2isa) pour passer le titre Concepteur Développeur d'Application.

Pour moi venir dans ce centre est une suite indispensable pour arriver à mon objectif de créer ma propre entreprise.

1.2 La formation CDA

J'ai rejoint la formation Concepteur Développeur d'Applications (CDA) au Centre de Rééducation Professionnelle de l'Institut Informatique Sud-Aveyron à Millau (2ISA) le 17 mai 2021.

La formation a commencé par :

- 2 semaines de préparatoire où l'on nous a présenté la formation et une remise à niveaux.
- Puis 11 semaines de socle de base où l'on a vu la programmation procédurale.
- La programmation orientée objet en C#.
- Les bases de données relationnelles sur le système de gestion de base de données .

Pour avoir le Titre professionnel Concepteur Développeur d'applications, la formation se déroule en 91 semaines avec 24 en entreprise.

Elle est découpée en trois activités :

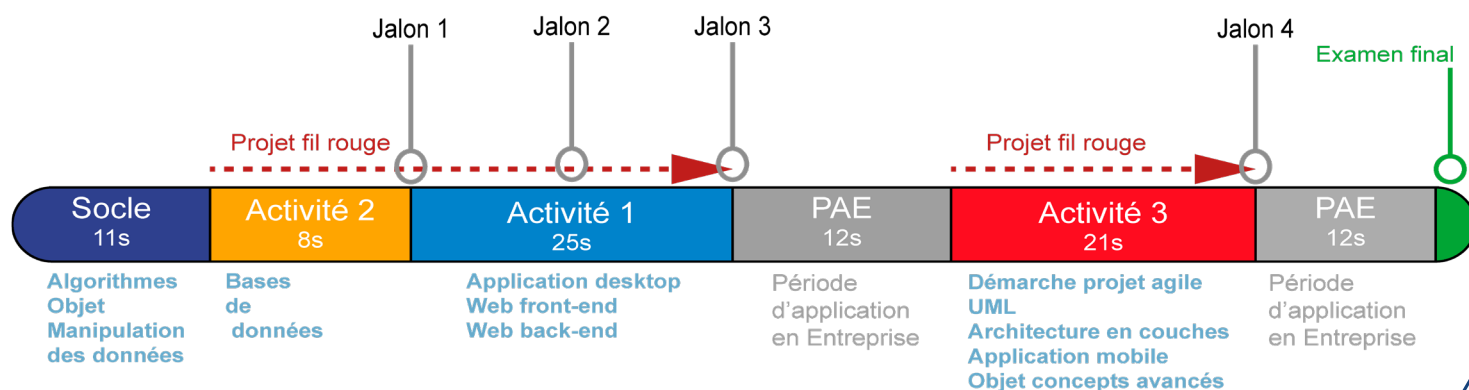
Activité 2 : Concevoir et développer la persistance des données

Activité 1 : Concevoir et développer des composants d'interface utilisateur

Activité 3 : Concevoir et développer une application multicouche répartie

Pour chacune des activités j'utilise et acquière de nouvelles compétences, en parallèle nous avons un « fil rouge » qui est une mise en situation qui nous amène à construire une application en plusieurs étapes, nous devons fournir un livrable écrit du fil rouge correspondant aux EPCF.

Un jury de professionnel est convoqué après la dernier PAE pour nous évaluer sur notre savoir et nos compétences professionnelles. Ils pourront délivrer le certificat de compétences professionnelles (CCP).



2 Présentation de la PAE

2.1 Présentation de l'entreprise

DSI a été fondé en 1994 par Jean-Louis RIBES, c'est une SARL avec l'agrément d'État « Entreprise Adaptée » et « Entreprise Solidaire d'Utilité Sociale » ; c'est à ce titre que cette société a un taux d'emploi de travailleurs handicapés de 84%.

Elle est implantée au niveau national (Midi-Pyrénées, Ile de France, Aquitaine, PACA, Pays de la Loire).



L'entreprise compte plus de 800 salariés et connaît une croissance de 15% en moyenne par an depuis sa création, avec un chiffre d'affaires de +30 M€ pour 2018.

Ses activités sont regroupées en 4 pôles : Projets numériques, Interactivités Digitales, BTP, Services industriels.

2.2 Objectifs

L'objectif de concevoir et de développer une application organisée en couches réparties est de rendre l'application plus modulaire, flexible, maintenable et performante en utilisant une architecture bien structurée.

Pour ce faire, je développe une Application en architecture (MVC) qui va me permettre de communiquer avec une base de données, et qui est une application organisée en couches. Ces couches sont utiles pour la séparation des responsabilités.

2.3 Liste des compétences du référentiel couvertes

Cette PAE est faite pour montrer l'acquisition et les compétences professionnelles qui font partie du référentiel. Il s'agit de :

« Concevoir et développer la persistance des données en intégrant les recommandations de sécurité »

- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données

« Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité ».

- Maquetter une application
- Développer des composants d'accès aux données
- Développer la partie back-end d'une interface utilisateur web
- Développer la partie front-end d'une interface utilisateur web

« Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité ».

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement
- Concevoir une application.
- Développer des composants métiers.
- Construire une application organisée en couches
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application

3 Contexte et besoins du projet

3.1 Contexte du projet

Le projet nous a été donné en début de PAE, il se divise en plusieurs étapes pour élaborer diverses fonctionnalités de l'application .

Un ticket nous a été attribué pour connaître la fonctionnalité que l'on doit implémenter dans le projet. Le chargé de projet est là pour toutes questions et besoins de précision.

3.2 Contexte et besoin de l'entreprise

Le site actuel ne dispose pas de droit utilisateur une fois connecté on peut accéder à toutes les pages du site ainsi qu'à toutes les fonctionnalités.

Il nous est demandé de développer une solution pour la mise en place des droits utilisateurs. Un ticket qui nous a été attribué pour mettre en place ces droits.

Les consignes détaillées sont sur le ticket ce qui fait office de cahier des charges. On a choisi de mettre en place une solution par groupe d'utilisateur et donner que des accès à ces groupes.

- L'administrateur : il doit disposer de toutes les fonctionnalités.
- Les utilisateurs : ils disposeront que des droits qui sont attribués au groupe.
- Il est possible de créer de nouveau groupe et de leur attribuer des accès.
- Un utilisateur peut appartenir à différents groupes.

3.2.1 Outils utilisés



- Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.



- Laragon est un environnement de développement web qui regroupe les serveurs suivants : Apache HTTP Server : Serveur HTTP, tournant par défaut sur le port 80. MySQL : Serveur de base de données relationnelles SQL, tournant par défaut sur le port 3306.



- Software Ideas Modeler est un logiciel de diagramme et de modélisation pour l'analyse, la conception et la documentation des systèmes et processus logiciels.



- Fork est une application flexible, ultra-rapide et multi-plateforme qui facilite grandement la gestion des processus GitHub et leur exécution.

2.3.2 Technologies utilisées



- PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP.



Symfony

- Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP.



- CodeIgniter est un framework libre écrit en PHP. Il suit le motif de conception MVC et s'inspire du fonctionnement de Ruby on Rails.



- Twig est un moteur de templates pour le langage de programmation PHP, utilisé par défaut par le framework Symfony.



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

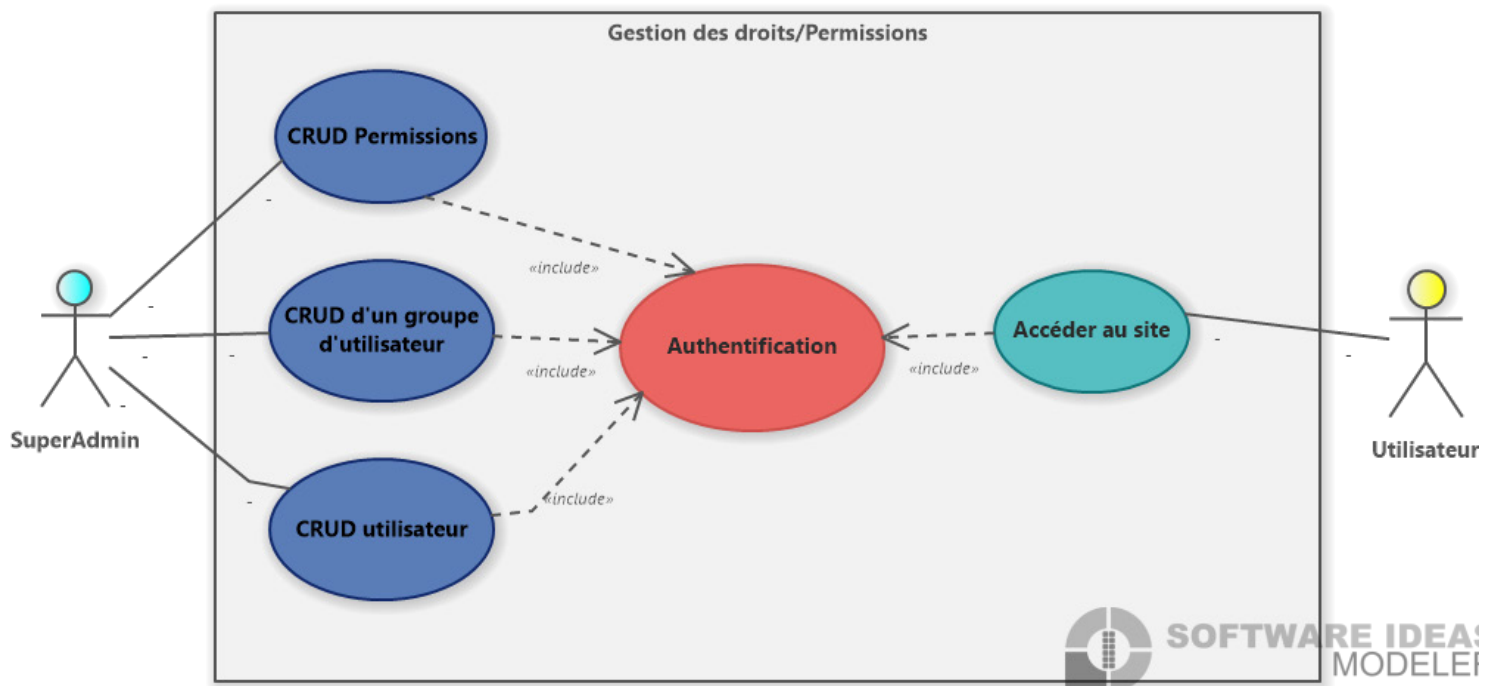
4 Analyse du besoin

Pour élaborer les droits d'utilisation on a utilisé CodeIgniter 4 Shield qui est une bibliothèque d'autorisation pour le framework PHP CodeIgniter 4.

Elle permet de gérer les permissions d'accès aux différentes parties d'une application web développée avec CodeIgniter. Il a besoin de nouvelles tables pour pouvoir l'utiliser. Pour comprendre comment il fonctionne, j'ai fait de la rétro-ingénierie pour analyser le système et en extraire les informations de sa structure et son comportement.

4.1 Cas d'utilisation

En ce qui concerne le fonctionnement de l'application, je décris les fonctionnalités dans un diagramme regroupant les cas d'utilisation.



Sur le diagramme, deux acteurs sont identifiés :

1. L'administrateur

Il disposera d'un accès pour gérer les autorisations, les groupes et les utilisateurs.

2. Les utilisateurs

Ils pourront accéder au site et à certaines fonctionnalités en fonction de leur groupe.

4.2 MLD MCD

J'extrais les informations qui me seront utiles pour la réalisation de l'application et pour savoir quelles tables je dois aller chercher sur la base de données. Je réalise le MLD pour connaître chaque clé primaire et avoir leur type.

4.2.1 MLD

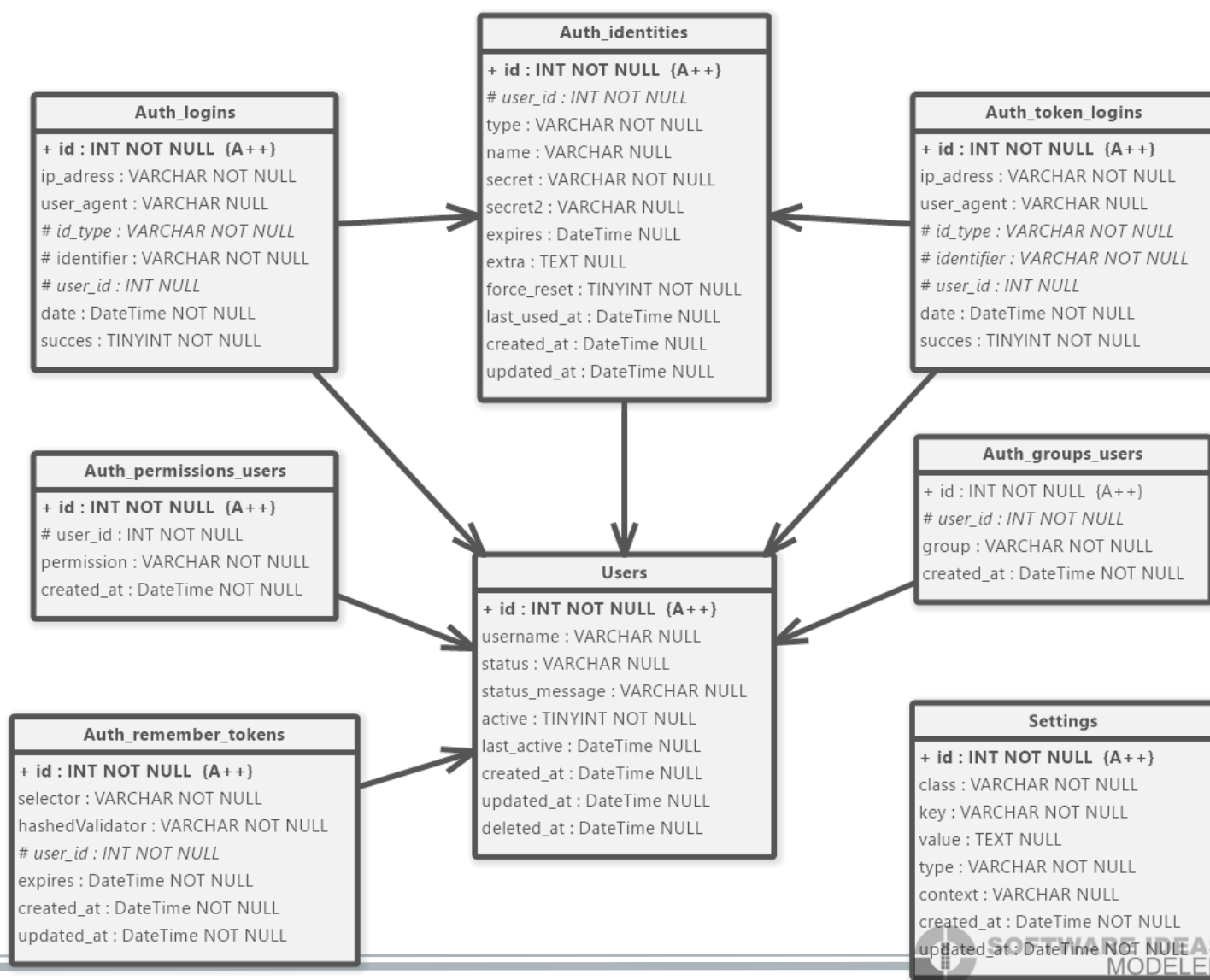
je déduit le schéma de la base de données.

Dans les tables on peut trouver une ou plusieurs clés étrangères qui sont l'identifiant d'autres tables. On y retrouve :

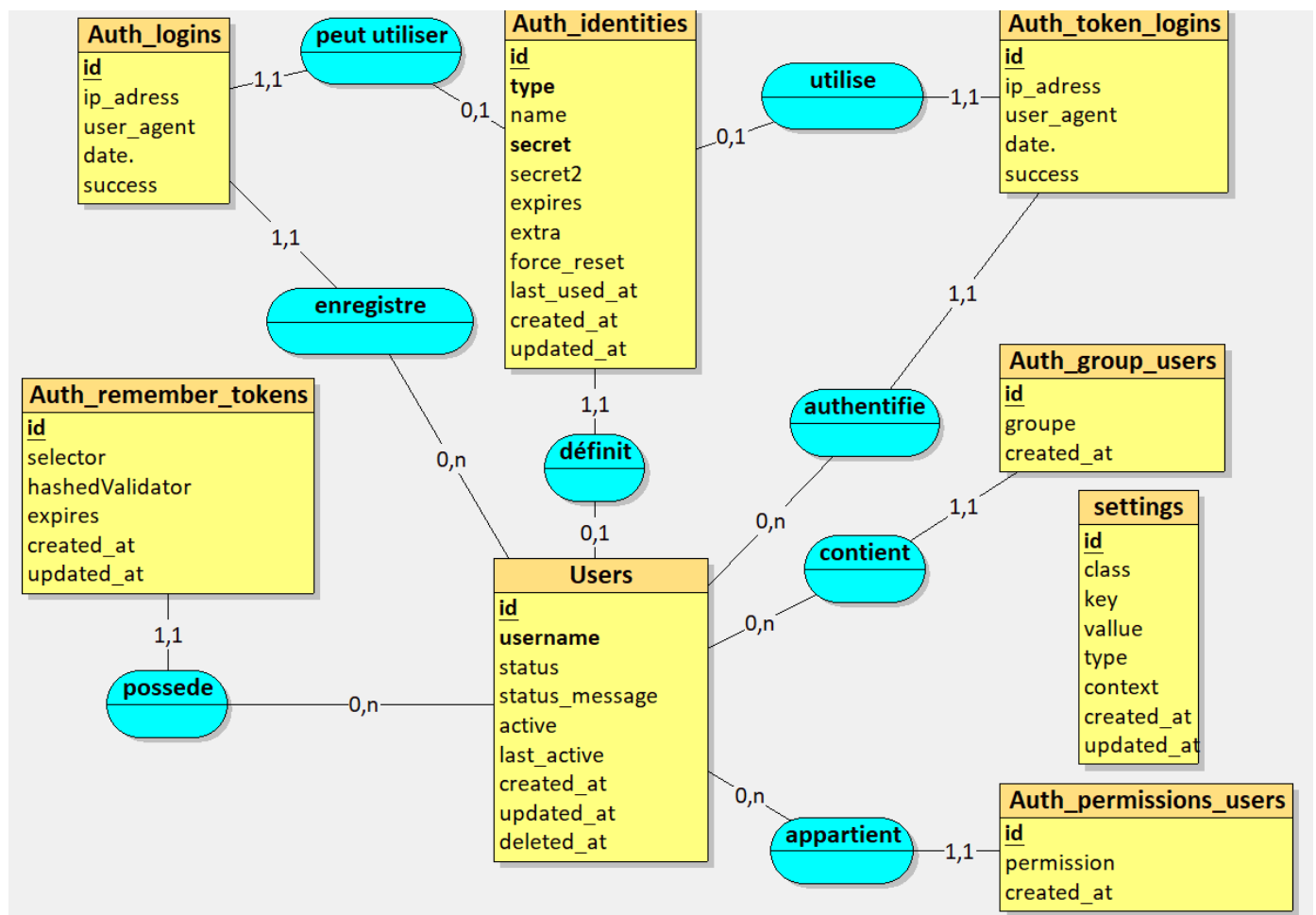
- les tables
- les attributs
- les clés primaires
- les clés étrangères

Les clés primaires sont représentées en premier dans chaque table.

Les clés étrangères sont représentées avec un #.



4.2.2 MCD



On y trouve :

- des entités qui ont un identifiant unique naturel
- des attributs qui dépendent directement de l'identifiant
- les cardinalités de chaque côté expriment le nombre minimum et le nombre maximum d'occurrences pouvant exister pour une entité.

4.3 Base de données

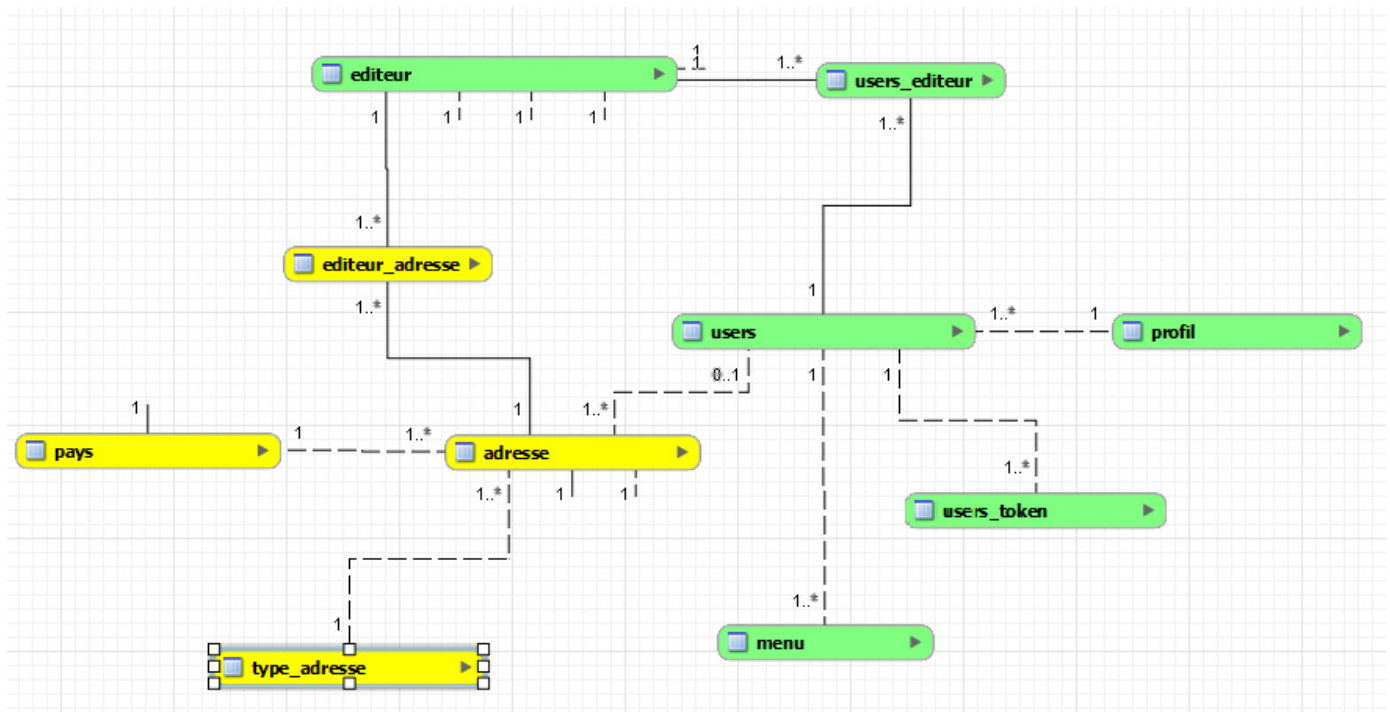
Pour la base de donnée qui été donnée au départ, elle n'intègre pas les tables de codeigniter\shield.

J'ai effectué une rétro-ingénierie ce qui consiste à analyser et comprendre la structure interne, son fonctionnement et son contenu, sans avoir accès aux informations initiales ou à la documentation officielle.

1. J'ai examiné la base de données existante et collecté autant d'informations que possible sur sa structure et son comportement.
2. J'analyse le contenu pour comprendre les données stockées.

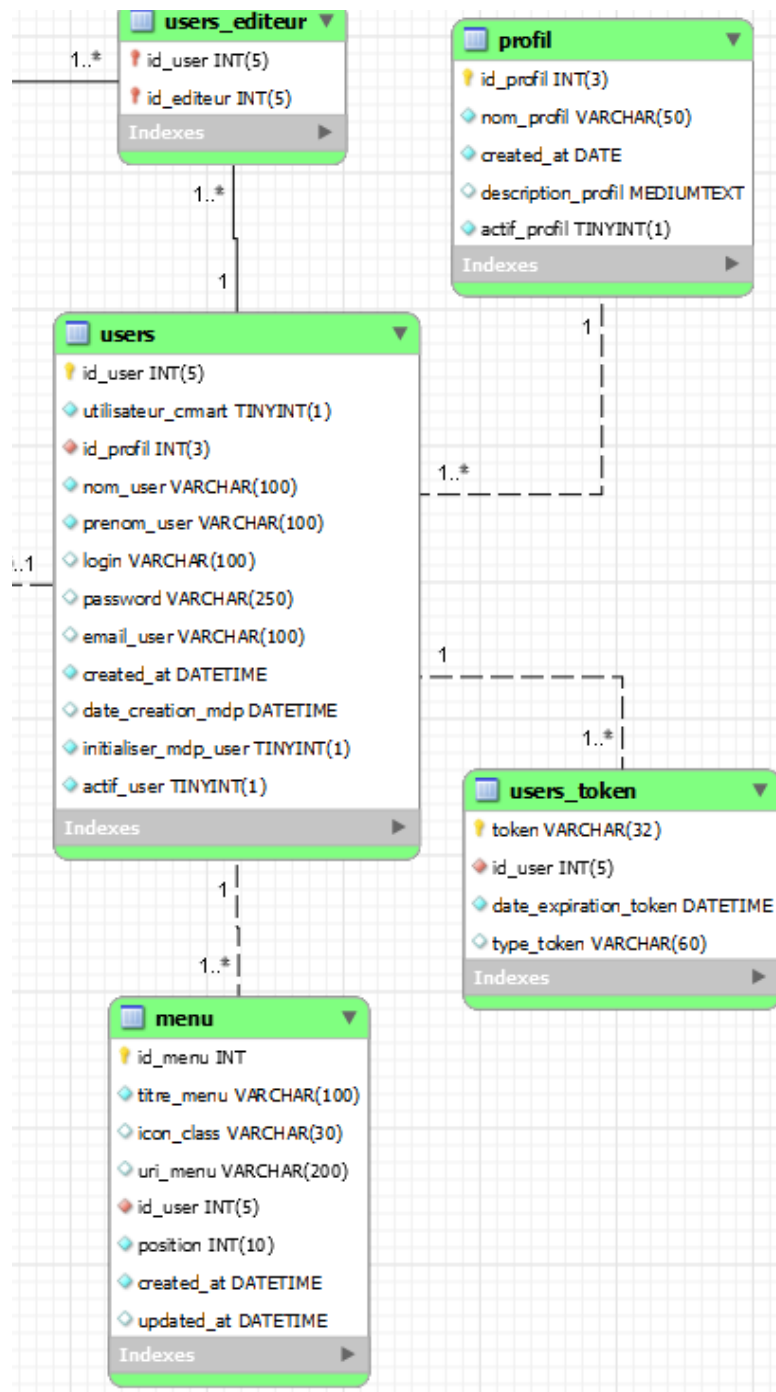
J'ai du récupérer le mcd de la base qui nous a été fournie de base pour analyser le fonctionnement des tables et leurs relations.

Exemple d'un extrait du mcd sans les données



Pour le projet je vais devoir recréer de nouvelles entités avec de nouveaux attributs qui correspondent à la méthode de fonctionnement de Shield.

Par rapport au mcd de fonctionnement de Shield, je constate qu'il va y avoir des entités qui vont disparaître et d'autres qui vont devoir changer d'attributs.



Exemple d'un extrait du mcd avec les données avant changement

J'ai récupéré le scrit sql de la base pour pouvoir avoir toutes les tables et les insert, puis je vide le schéma et j'exécute Shield en invite de commande pour installer les nouvelles tables sur le schéma.

```
CREATE TABLE IF NOT EXISTS `users` (
  `id_user` INT(11) NOT NULL AUTO_INCREMENT,
  `utilisateur_crmart` INT(11) NOT NULL,
  `id_profil` INT(11) NOT NULL,
  `nom_user` VARCHAR(255) NOT NULL,
  `prenom_user` VARCHAR(255) NOT NULL,
  `login` VARCHAR(255),
  `password` VARCHAR(255),
  `email_user` VARCHAR(255),
  `created_at` DATETIME NOT NULL,
  `deleted_at` DATETIME DEFAULT NULL,
  `updated_at` DATETIME DEFAULT NULL,
  `actif_user` TINYINT(1) NOT NULL,
  PRIMARY KEY (`id_user`),
  CONSTRAINT `fk_user_profil` FOREIGN KEY (`id_profil`) REFERENCES `profil` (`id_profil`));

INSERT INTO `users` (`id_user`, `utilisateur_crmart`, `id_profil`, `nom_user`, `prenom_user`,
`login`, `password`, `email_user`, `created_at`, `deleted_at`, `updated_at`, `actif_user`)
VALUES
(NULL, 1, 1, 'admin', 'admin', 'admin',
'$2y$10$9anfiCNCDU2uzeyhOvYHLeBkgzyvBzVHHpsKX92YhQjpEpyYIDV1y', 'admin@
crmart.fr.local', '2023-04-11 08:39:38', NULL, NULL, 1);
```















exemple script create table "user" et un insert

🔑 id_user	INT(5)
💎 utilisateur_crmart	TINYINT(1)
💎 id_profil	INT(3)
💎 nom_user	VARCHAR(100)
💎 prenom_user	VARCHAR(100)
💎 login	VARCHAR(100)
💎 password	VARCHAR(250)
💎 email_user	VARCHAR(100)
💎 created_at	DATETIME
💎 date_creation_mdp	DATETIME
💎 initialiser_mdp_user	TINYINT(1)
💎 actif_user	TINYINT(1)

Les colonness de la table "user" dans la base avant changement



















Une fois les nouvelles tables enregistrées pour remettre la base de données dans un état utilisable, je rajoute les anciennes tables, et j'enlève toutes les tables qui ne sont plus utilisées.

schéma de la base avant changement

 pays
 periodicite
 profil
 reglement
 societe
 stock
 tva
 type_adresse
 type_commande
 type_contact
 type_reglement
 users
 users_editeur
 users_token

On constate que certaines tables ont été enlevées car elles n'avaient plus d'utilité à rester dans le projet

schéma de la base modifiée

 auth_logins
 auth_permissions_users
 auth_remember_tokens
 auth_token_logins
 civilite
 client
 client_adresse
 contact
 editeur
 editeur_adresse
 menu
 migrations
 pays
 settings
 type_adresse
 type_contact
 users
 users_editeur

La table users qui a changé pour pouvoir utiliser par Shield correctement dans le projet.

Nom	Type de données
id	INT
username	VARCHAR
status	VARCHAR
status_message	VARCHAR
active	TINYINT
last_active	DATETIME
created_at	DATETIME
updated_at	DATETIME
deleted_at	DATETIME

5 Développement

5.1 CodeIgniter 4

CodeIgniter 4 est un framework de développement web open-source utilisé pour créer des applications web performantes et évolutives en utilisant le langage de programmation PHP. Il s'agit de la dernière version majeure du framework CodeIgniter, qui apporte de nombreuses améliorations et fonctionnalités par rapport à la version précédente, CodeIgniter 3.

- CodeIgniter 4 suit l'architecture MVC, qui divise l'application en trois composants principaux. Le modèle est responsable de la gestion des données, la vue représente l'interface utilisateur et le contrôleur gère la logique métier et interagit avec les modèles et les vues.
- CodeIgniter 4 fournit une bibliothèque de validation puissante qui facilite la validation des données entrées par les utilisateurs. On peut définir des règles de validation prédéfinies ou créer des règles personnalisées pour vérifier les données saisies par l'utilisateur.
- Le framework offre une couche d'abstraction pour interagir avec les bases de données. On peut facilement exécuter des requêtes SQL, gérer les migrations de base de données et utiliser des requêtes de génération automatique.

5.1.1 CodeIgniter 4\Shield

Shield fournit un moyen simple et efficace de définir des règles d'autorisation pour contrôler qui peut accéder à quelles ressources ou effectuer quelles actions dans votre application. Il offre des fonctionnalités telles que la définition de rôles, l'assignation de rôles aux utilisateurs, la gestion des autorisations basée sur les rôles et les permissions, etc.

En utilisant CodeIgniter 4 Shield, on peut restreindre l'accès à certaines pages ou fonctionnalités de l'application en fonction des rôles et des permissions définis. Par exemple, on peut limiter l'accès à une page d'administration aux utilisateurs ayant le rôle d'administrateur, ou autoriser seulement certains utilisateurs à effectuer des opérations spécifiques.

Pour utiliser CodeIgniter 4 Shield, on doit l'installer en tant que dépendance dans le projet CodeIgniter 4, puis configurer les règles d'autorisation selon les besoins. La documentation officielle de CodeIgniter 4 Shield peut nous fournir des instructions détaillées sur son installation, sa configuration et son utilisation.

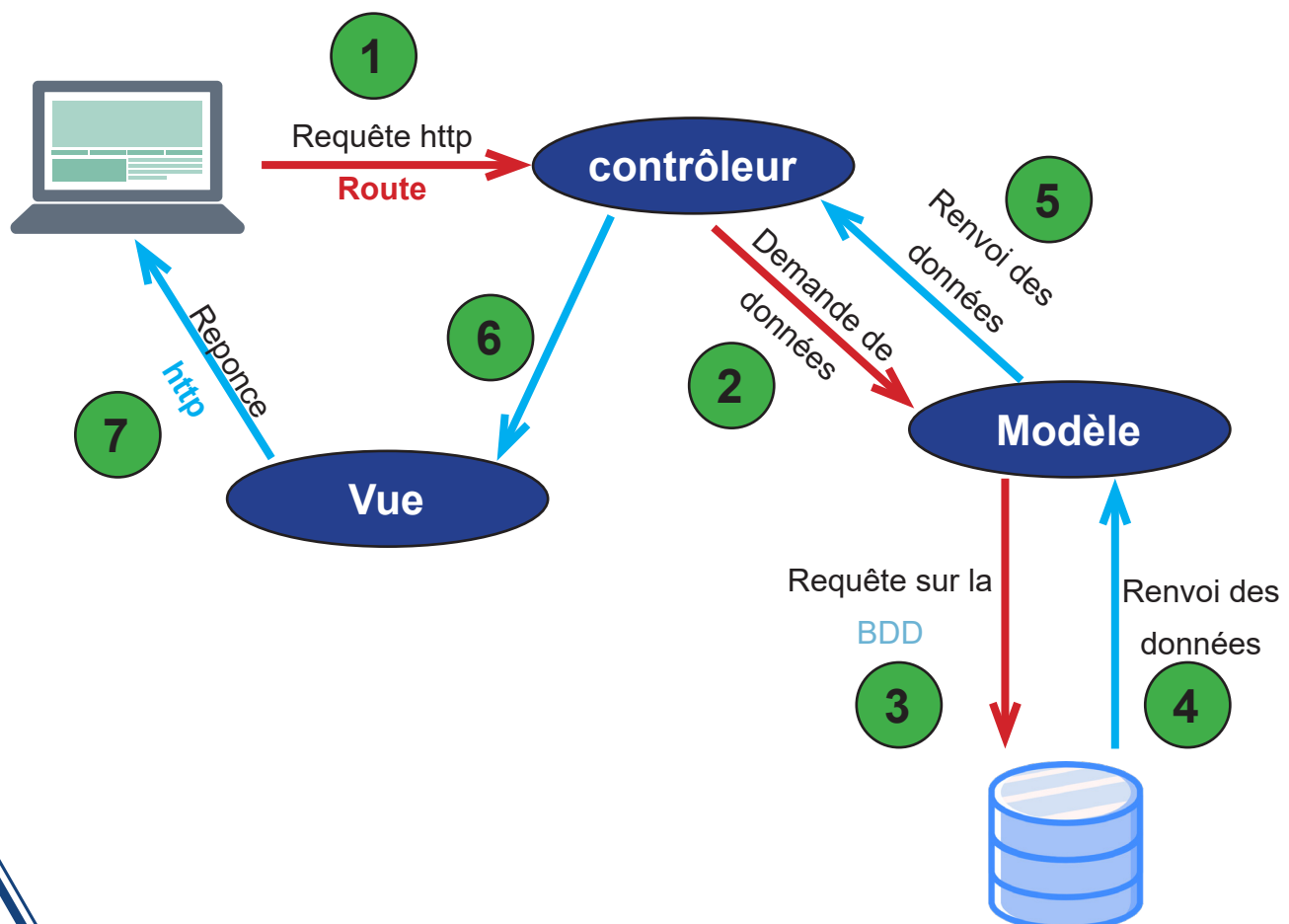
5.2 Architecture

Pour répondre à la demande, l'architecture choisie est une architecture en couche MVC .

Le but du MVC est de fournir une structure claire et organisée à une application, en séparant les responsabilités et en permettant une évolutivité, une maintenance et une collaboration plus faciles.

Cette architecture a été choisie pour :

- La séparation des préoccupations : Le MVC divise l'application en trois composants distincts, le modèle, la vue et le contrôleur, chacun ayant des responsabilités spécifiques.
- Réutiliser du code : En séparant les différentes couches de l'application, il devient possible de réutiliser les composants indépendamment les uns des autres..
- Facilité de maintenance : Grâce à la séparation des responsabilités, il est plus facile de localiser et de résoudre les problèmes spécifiques.



5.2.1 MVC

- **Model** : Il représente la couche de données et la logique métier d'une application. Sa responsabilité principale est de gérer les données, qu'il s'agisse de les récupérer, de les manipuler, de les valider ou de les enregistrer. Le modèle encapsule la logique métier spécifique à l'application, en appliquant les règles et les opérations nécessaires sur les données.

- **View** est chargée de présenter les données au travers de l'interface utilisateur. Elle se concentre sur la manière dont les informations sont affichées à l'utilisateur sans se soucier de leur récupération ou de leur traitement. La vue joue un rôle clé dans la création d'une interface attrayante et interactive, en affichant les données de manière appropriée et en permettant à l'utilisateur d'interagir avec l'application. Sa séparation des autres composants permet une meilleure organisation du code et facilite la maintenance de l'interface utilisateur.

- **Contrôleur** est chargé de gérer les interactions entre l'utilisateur, la vue et le modèle. Son rôle principal est de recevoir les actions de l'utilisateur, d'interagir avec le modèle pour récupérer ou mettre à jour les données, et de mettre à jour la vue en conséquence. Le contrôleur joue un rôle clé dans la coordination des composants de l'architecture MVC, permettant ainsi une gestion fluide des fonctionnalités de l'application et des flux de navigation. Il facilite également la séparation des préoccupations en isolant la logique de traitement des interactions de l'utilisateur du reste de l'application.

5.3 Installation de la bibliothèque CodeIgniter\shield

Dans le projet qui nous a été donné, il ne dispose pas de codeigniter4\shield.

Pour gérer les droits il a fallu installer la bibliothèque pour se faire, et en suivant la documentation, j'ai réalisé diverses étapes en invite de commande, l'installation se fait via Composer.

1. **composer require codeigniter4/shield** : est utilisée pour installer la bibliothèque "Shield" dans le projet CodeIgniter 4. "Shield" est une bibliothèque officielle de CodeIgniter 4 qui fournit des fonctionnalités de sécurité avancées pour votre application web.

```
composer require codeigniter4/shield
```

2 php spark shield:setup : est une commande artisanale fournie par la bibliothèque "Shield" dans CodeIgniter 4. Elle est utilisée pour configurer et initialiser les fonctionnalités de sécurité avancées de "Shield" dans l'application.

```
php spark shield:setup
```

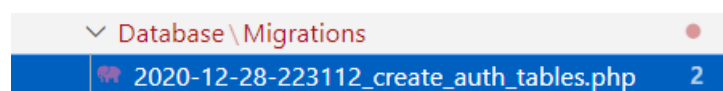
3 php spark migrate --all : est utilisée dans CodeIgniter 4 pour exécuter toutes les migrations disponibles dans l'application. Les migrations sont un mécanisme intégré de CodeIgniter 4 qui permet de gérer facilement les modifications de structure de la base de données de manière contrôlée.

Lors de l'exécution de la commande elle va rechercher tous les fichiers de migration non exécutés dans l'application et les exécuter dans l'ordre chronologique. Chaque fichier de migration représente une modification de structure de la base de données, telle que la création de tables, l'ajout de colonnes, les clés étrangères

```
php spark migrate --all
```

A l'exécution de la commande, il va chercher dans la solution du projet, dans le dossier Database\Migrations et exécuter les fichiers de migrations en .php.



exemple du dossier DataBase



```
// Users Table
$this->forge->addField([
    'id'           => ['type' => 'int', 'constraint' => 11, 'unsigned' => true, 'auto_increment' => true],
    'username'     => ['type' => 'varchar', 'constraint' => 30, 'null' => true],
    'status'       => ['type' => 'varchar', 'constraint' => 255, 'null' => true],
    'status_message' => ['type' => 'varchar', 'constraint' => 255, 'null' => true],
    'active'       => ['type' => 'tinyint', 'constraint' => 1, 'null' => 0, 'default' => 0],
    'last_active'  => ['type' => 'datetime', 'null' => true],
    'created_at'   => ['type' => 'datetime', 'null' => true],
    'updated_at'   => ['type' => 'datetime', 'null' => true],
    'deleted_at'   => ['type' => 'datetime', 'null' => true],
]);
$this->forge->addPrimaryKey('id');
$this->forge->addUniqueKey('username');
$this->forge->createTable($this->tables['users']);
```






















Dans l'extrait de code ci-dessus, à la migration il va créer la table "Users" dans la base de données, avec ses contraintes et les clés primaires et étrangères.

Dans la base on retrouve bien toute les tables de créé avec les contraintes de clé. Ici dans la table "Users" je constate que la requête de création qui a été exécutée à la migration a bien fonctionnée.

#	Nom	Type de données
 1	id	INT
 2	username	VARCHAR
3	status	VARCHAR
4	status_message	VARCHAR
5	active	TINYINT
6	last_active	DATETIME
7	created_at	DATETIME
8	updated_at	DATETIME
9	deleted_at	DATETIME

la cle **jaune** représente la cle primaire.

la cle **rouge** représente un champ qui doit être unique

	adresse
	auth_groups_users
	auth_identities
	auth_logins
	auth_permissions_users
	auth_remember_tokens
	auth_token_logins
	civilite
	client
	client_adresse
	contact
	editeur
	editeur_adresse
	menu
	migrations
	pays
	settings
	type_adresse
	type_contact
	users
	users_editeur

représentation de la base

5.4 Fonctionnalités Attendues

Il est demandé que l'application ait comme fonctionnalités :

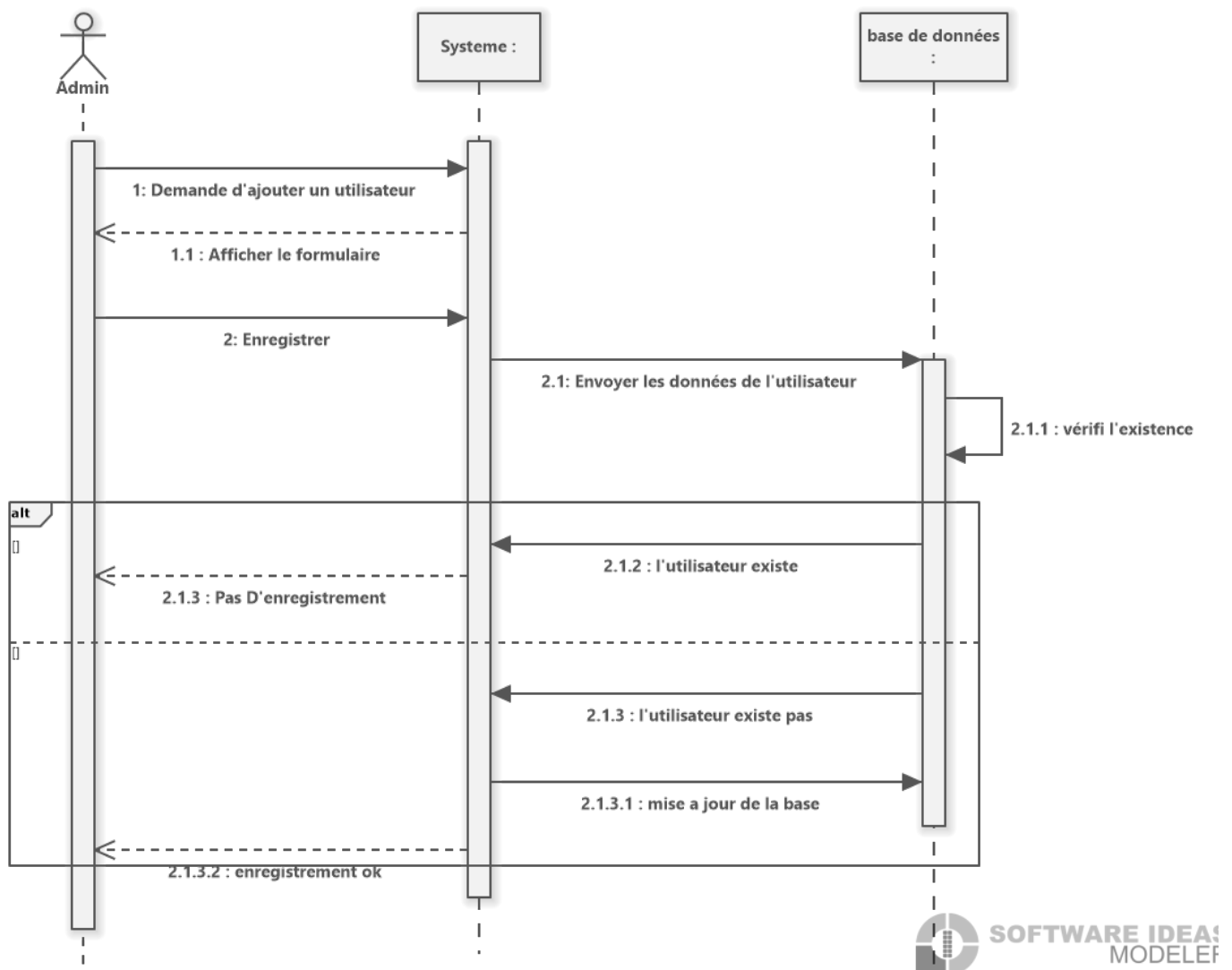
- Créer une page d'accueil proposant la connexion par session
 1. Réinitialiser le mot de passe à la première connexion
- Ajouter des utilisateurs
- Ajouter des groupes utilisateurs
- Bloquer les accès à des fonctionnalités

5.4.1 Présentation de la Fonctionnalité "créer un utilisateur"

Pour commencer je crée un diagramme de séquence.

Un diagramme de séquence est un outil de modélisation qui permet de représenter de manière séquentielle l'interaction entre les différents objets ou composants d'un système.

Je crée un diagrammes de séquence pour voir comment créer un utilisateur et savoir si les différentes couches interagissent correctement pour fournir le résultat attendu.



1. l'administrateur va demander au système d'enregistrer un utilisateur.
2. le système va lui afficher la page d'enregistrement,
3. l'utilisateur rentre les données à envoyer au système.
4. le système va faire une vérification si ces données existent.
5. à ce moment il y a deux possibilité:
 - si l'utilisateur existe alors pas d'enregistrement, le système informe l'administrateur.
 - si l'utilisateur existe pas alors un enregistrement sur la base sera effectué et le succès est retourné au système qui informe l'administrateur que le nouvel utilisateur eet enregistré.

5.5 Le Model

Pour faire appel à la base de données, je crée un model en suivant la documentation de codeigniter4.

Je crée la classe UserModel qui va étendre de Model. Pour se faire je lui rajoute un use "CodeIgniter\model" pour lui dire où aller chercher le Model.

```
use CodeIgniter\Model;

class UserModel extends Model
{
```

1. use Code\Model : Cette ligne importe la classe de base Model du namespace Code pour être utilisée par le modèle.

2. class UserModel extends Model : Définit la classe du modèle, qui étend la classe de base Model.

Par la suite je crée les variables protected qui vont servir au model.

Vue qu'elles seront utilisées uniquement dans cette classe et les classes qui hériteront de celle si.

```
class UserModel extends Model
{
    protected $DBGroup          = 'default';
    protected $returnType       = 'object';
    protected $table             = 'users';
    protected $primaryKey       = 'id';
    protected $useAutoIncrement = true;

    protected $allowedFields    = [
        'username',
        'user_id',
    ];
}
```

1. `protected $DBGroup = 'default'` : Spécifie le groupe de base de données à utiliser pour le modèle. La valeur par défaut est 'default'.

2. `protected $returnType = 'object'` : Définit le type de retour pour les résultats des requêtes. Dans cet exemple, les résultats seront retournés sous forme d'objets.

3. `protected $table = 'users'` : Indique le nom de la table de base de données associée à ce modèle.

4. `protected $primaryKey = 'id'` : Spécifie la clé primaire de la table.

5. `protected $useAutoincrement = true;` : Indique si la clé primaire utilise l'auto-incrément.

6. `protected $allowedFields = ['username', 'user-id'];` : Définit les champs de la table qui sont autorisés à être manipulés par les méthodes du modèle.

À la suite j'ai créé les variables qui permettront d'exécuter n'importe quelle fonction du CRUD.

```
protected $useSoftDeletes = true;
```

```
protected $createdField = 'created_at';
protected $updatedField = 'updated_at';
protected $deletedField = 'deleted_at';
```

```
// Validation
protected $validationRules = [];
```

```
// Callbacks
protected $afterFind = [];
```

1. `protected $useSoftDeletes = true;` : Active la suppression douce (soft delete) pour ce modèle, ce qui signifie que les enregistrements ne sont pas supprimés physiquement mais marqués comme supprimés.

2. `protected $createdField = 'created_at';` Spécifie le nom du champ pour stocker la date de création d'un enregistrement.

3. `protected $updatedField = 'updated_at';` Spécifie le nom du champ pour stocker la date de mise à jour d'un enregistrement.

4. `protected $deletedField = 'deleted_at';` : Indique le nom du champ pour stocker la date de suppression d'un enregistrement (en cas d'utilisation de la suppression douce).

5. `protected $validationRules = [];` : Définit les règles de validation des données pour les opérations de création et de mise à jour des enregistrements. On peut spécifier les règles de validation ici.

6. `protected $afterFind = [];` : Permet de spécifier les fonctions de rappel (callbacks) qui seront exécutées après la recherche d'un enregistrement.

Une fois le modèle de base réalisé, je crée une méthode pour récupérer les informations d'un utilisateur qui sont réparties sur différentes tables de la base de données.

```
public function getUser($id)
{
    $this->table('users');

    $this->select('users.*, auth_groups_users.*, auth_identities.secret');
    $this->join('auth_groups_users', 'auth_groups_users.user_id = users.id');
    $this->join('auth_identities', 'auth_identities.user_id=users.id');
    $this->where('users.id', $id);

    $users = $this->first();

    return $users;
}
```

`public function getUser($id)` : Définit la méthode `getUser` qui prend un paramètre `$id` représentant l'identifiant de l'utilisateur à récupérer.

`$this->table('users')` : Spécifie la table principale à utiliser, dans ce cas, la table "users".

`$this->select('users.*, auth_groups_users.*, auth_identities.secret')` : Indique les colonnes à sélectionner dans la requête. Dans cet exemple, toutes les colonnes de la table "users", "auth_groups_users" et "auth_identities" sont sélectionnées, ainsi que la colonne "secret" de la table "auth_identities".

`$this->join('auth_groups_users', 'auth_groups_users.user_id = users.id')` : Effectue une jointure entre la table "users" et la table "auth_groups_users" sur la clé étrangère "user_id".

`$this->join('auth_identities', 'auth_identities.user_id=users.id')` : Effectue une jointure entre la table "users" et la table "auth_identities" sur la clé étrangère "user_id".

`$this->where('users.id', $id)` : Spécifie une condition de filtrage pour récupérer uniquement l'utilisateur avec l'identifiant spécifié.

`$users = $this->first()` : Exécute la requête et récupère le premier résultat retourné.

`return $users` : Retourne les informations de l'utilisateur récupérées.

5.6 Le Contrôleur

Une fois les données de création d'un utilisateur inscrit dans la vue, quand on appuie sur enregistrer le contrôleur va exécuter la fonction add qui va vérifier les informations et les transmettre au model.

```
public function add()
{
    if ($this->user->can('users.add')) {
        if ($this->request->getPost('add')) {
```

En premier je vérifie si l'utilisateur qui demande l'appel de cette fonction est autorisé. Puis je récupère les données qui ont été envoyées.

```
$users = auth()->getProvider();
$rules = $this->getValidationRules();
```

Puis je récupère le fournisseur d'authentification actuel, qui est utilisé pour interagir avec le système d'authentification et d'autorisation.

Le `getValidationRules()` pour obtenir les règles de validation des données, ces règles de validation sont créées par codeigniter\shield et doivent être utilisées pour enregistrer un utilisateur.

Ci-dessous un extrait de règle de validation.

```
return setting('Validation.registration') ?? [
    'username' => [
        'label' => 'Auth.username',
        'rules' => $registrationUsernameRules,
    ],
    'email' => [
        'label' => 'Auth.email',
        'rules' => $registrationEmailRules,
    ],
    'password' => [
        'label' => 'Auth.password',
        'rules' => 'required|' . Passwords::getMaxLenghtRule() . '|strong_password',
        'errors' => [
            'max_byte' => 'Auth.errorPasswordTooLongBytes',
        ],
    ],
];
```

On constate que le password doit répondre à des règles:

- 1 une longueur minimum de 8 caractères
- 2 il ne doit pas être commun (c'est vérifié dans un DictionaryValidator qui comparera le mot de passe à un fichier fourni avec environ 600 000 mots de passe fréquemment utilisés)

A la suite des règles des validations, je vérifie si les données sont bonnes, si c'est pas le cas je renvoie sur la vue et informe l'utilisateur des erreurs.

```
if (!$this->validateData($this->request->getPost(), $rules)) {
    return redirect()->back()->withInput()->with('errors', $this->validator->getErrors());
}
```

Si les règles sont ok, je commence l'enregistrement des valeurs données.

```
$allowedPostFields = array_keys($rules);
$user               = $this->getUserEntity();
$user->fill($this->request->getPost($allowedPostFields));

$users->save($user);

$user = $users->findById($users->getInsertID());
$user->forcePasswordReset();

// ajout de l'utilisateur au groupe par défaut
$users->addToDefaultGroup($user);

return redirect()->to('/users');
```

En premier je prépare et enregistre un nouvel utilisateur. Les champs autorisés pour l'ajout sont récupérés à partir des règles de validation. Ensuite, une instance de l'entité utilisateur est créée et remplie avec les données postées correspondantes. Enfin, l'utilisateur est enregistré dans la base de données.

La méthode "forcePasswordReset()" pour forcer la réinitialisation du mot de passe de l'utilisateur quand il se connectera pour la première fois.

Puis j'ajoute l'utilisateur au groupe par défaut.

Une fois enregistrer l'administrateur est redirigé vers la vue où il pourra voir tous les utilisateurs et constater que le nouvel utilisateur est bien enregistré.

5.7 La vue Twig

Twig est un moteur de template, c'est-à-dire un langage de balisage utilisé pour générer des pages web ou d'autres types de documents. Il est souvent utilisé dans les frameworks PHP tels que Symfony.

Il permet de définir des templates avec des balises et des instructions spécifiques qui seront ensuite traitées pour générer du contenu dynamique.

Twig utilise une syntaxe claire et lisible qui facilite la création et la maintenance des templates. Il utilise des balises délimitées par des accolades doubles (`{{ }}`) pour les variables et des balises délimitées par des pourcentages (`{% %}`) pour les instructions.

```
{% for row in profil_row %}
```

exemple de balise où l'instruction est un "for" pour lire un tableau

```
<td>{{ row.id }}</td>
<td>
    {{ row.username }}
</td>
<td> {{ row.group }}</td>
```

exemple de balise de variable du tableau lu par l'instruction "for"

Pour la vue qui permet d'ajouter un utilisateur je commence par lui attribuer son template en lui précisant quel extends est le nom du fichier.

```
% extends "Layouts/main.twig" %}
```

Puis je lui dit avec un block content le contenu qui sera pour cette vue ce qui permet de pouvoir réutiliser le templates sur d'autre vues.

```
% block content %}
```

Pour faire un formulaire pour ajouter un utilisateur je commence par déclarer une variable form_open, je lui précise pour quelle route il sera utilisé, le 'class="mt-5"' pour le css.

```
{{ form_open(base_url('users/add'), 'class="mt-5"') }}
```

Pour les champs du formulaire je crée un form_input.

```
Email -->
class="mb-2">
{{ form_input("email", set_value('Auth.email'), {class: "form-control", name:"Auth.email", placeholder: "Adresse mail"}}
```

```
{{ form_input("email",
```

1. Indique que l'entrée de formulaire concerne l'adresse e-mail.

```
set_value('Auth.email'),
```

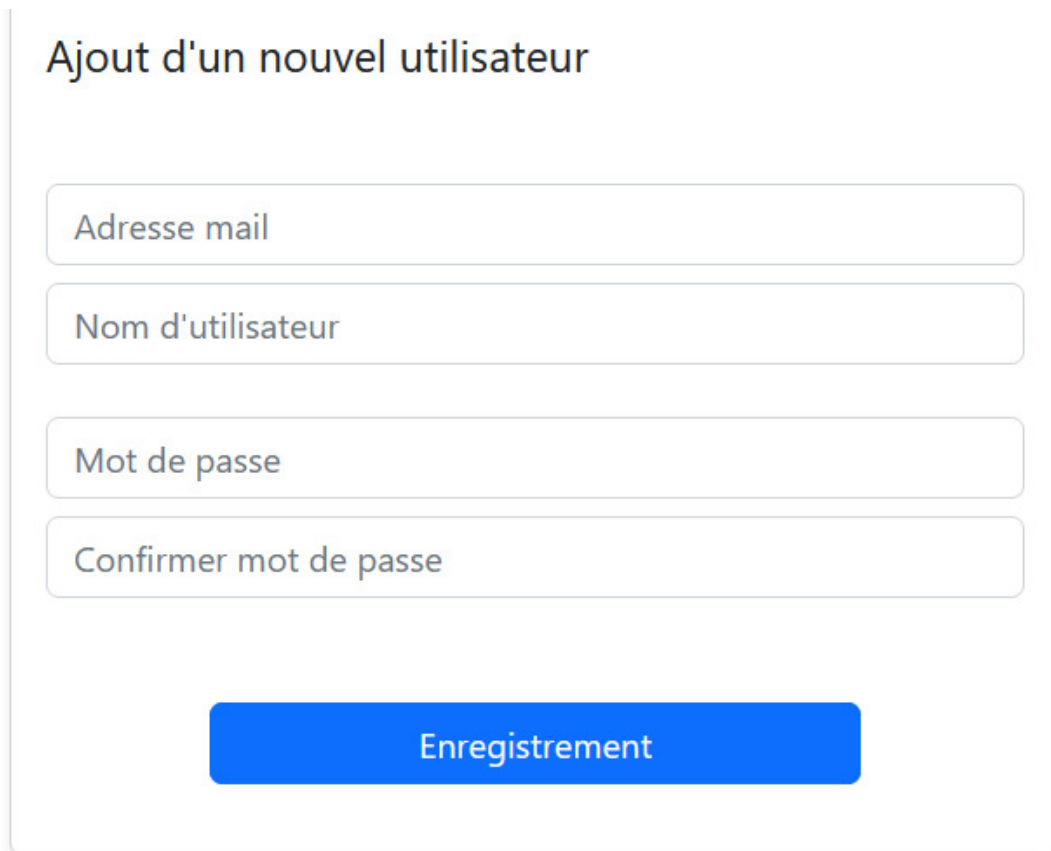
2. Est une fonction pour récupérer une valeur préremplie pour l'entrée du formulaire. Elle utilise la clé 'Auth. email' pour récupérer cette valeur.

```
{class: "form-control", name:"Auth.email", placeholder: "Adresse mail"}) }}
```

3. Est un tableau d'options supplémentaires pour personnaliser l'apparence et le comportement de l'entrée du formulaire. Il définit la classe CSS "form-control" pour l'entrée, le nom "Auth. email" pour l'élément de formulaire et le texte "Adresse mail" comme indicateur de champ.

Je fait la même chose pour chaque champs du formulaire puis je crée le bouton de validation avec le form_submit. Je referme ma form avec form_close.

```
<div class="d-grid col-12 col-md-8 mx-auto m-3">
|   {{ form_submit("add", "Enregistrement", {class: "btn btn-primary btn-block"})
</div>
form_close() }}
```



Ajout d'un nouvel utilisateur

Adresse mail

Nom d'utilisateur

Mot de passe

Confirmer mot de passe

Enregistrement

Résultat du formulaire créé

5.8 Le CSS

Pour le CSS j'utilise Bootstrap qui est un framework de développement web. Il fournit un ensemble de composants et d'outils front-end préconçus pour faciliter la création d'interfaces utilisateur attrayantes, réactives et adaptatives.

Pour l'utiliser je recherche sur la documentation de Bootstrap l'élément dont j'ai besoin. Par exemple pour le bouton du formulaire je vais chercher sur la documentation le bouton et je regarde lequel je veux.

Capture d'écran de la documentation de Bootstrap



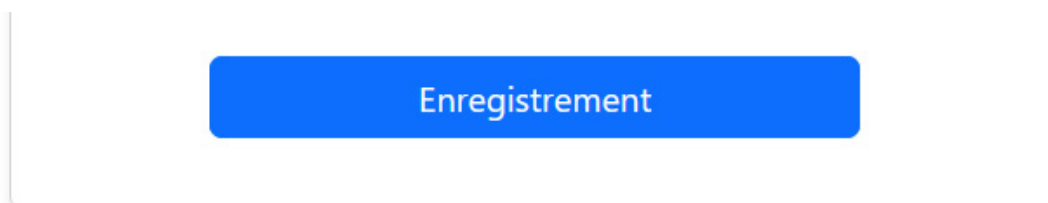
```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
```

On va prendre en exemple le premier bouton.

`class="btn btn-primary"` : est une classe CSS qui est appliquée au bouton. La classe "btn" est une classe de base de Bootstrap pour styliser les boutons, et la classe "btn-primary" est une classe spécifique de Bootstrap qui donne au bouton un style de bouton principal. Cela se traduit généralement par un fond bleu et une couleur de texte blanche.

Dans mon cas j'ai utilisé le css existant de Bootstrap pour créer le bouton enregistré de ma vue. Je précise qu'il est quand même possible de créer soi-même son CSS.

```
<div class="d-grid col-12 col-md-8 mx-auto m-3">
|   {{ form_submit("add", "Enregistrement", {class: "btn btn-primary btn-block"})
</div>
form_close() }}
```



5.9 La création d'un groupe

Un groupe représente un ensemble de permissions auxquels sont affectés des utilisateurs. Ils se définissent par une Clé qui est le libellé du groupe et sa description (stockés en BDD dans AuthGroups.groups).

A cette clé correspond aussi un ensemble de permissions accessibles au groupe (stockés en BDD dans AuthGroups.matrix).

5.9.1 Scénario Textuel « Créer un groupe utilisateur »



CU : Créer un groupe d'utilisateurs

Portée : système de gestion des droits

Acteur principal : L'administrateur principal

Intervenants et intérêts : Administrateur principal, les utilisateurs du groupe

Préconditions : Administrateur principal authentifié et sur la page "Groupes" accessible à partir du Menu

Garanties minimales : rien ne se passe

Garanties en cas de succès : le groupe d'utilisateur a été créé et un utilisateur peut être défini comme appartenant à ce groupe
L'utilisateur accède à tous les contrôleurs et les méthodes prévues pour son groupe

Scénario nominal :

1. L'administrateur principal appuie sur le bouton "Ajouter un groupe".
2. Le système vérifie le token utilisateur et les droits d'accès de celui-ci à la méthode appelée.
3. L'administrateur principal saisit le nom du groupe, la description de celui-ci, et l'ensemble des permissions qu'il souhaite attribuer à celui-ci.
4. Le système vérifie que ce groupe et cet ensemble de permissions n'existent pas déjà.
5. Le nouveau groupe est créé et enregistré dans le système.

5.9.2 Le Model Group

Il a été inutile de créer un model Group car ce dernier est géré par le framework SHIELD de CodeIgniter4.

5.9.3 Le Controller Group

```
namespace App\Controllers;

use App\Libraries\CiReflection;
use CodeIgniter\Shield\Authorization\Groups as AuthorizeGroup ;
use CodeIgniter\Shield\Entities\Group as GroupBase ;
```

Le contrôleur Group utilise des classes directement implémentées dans le framework de CodeIgniter\Shield.

Je choisis de leur donner un alias par soucis de clarté et pour éviter de confondre celles-ci. Je dois ensuite les instancier dans le constructeur

```
public function __construct(){
    $this->ciReflection = new CiReflection();
    $this->authorizeGroup = new AuthorizeGroup();
    $this->groupBase = new GroupBase();
}
```

La bibliothèque ciReflection a pour rôle de récupérer l'ensemble des contrôleurs et méthodes d'un projet.

A partir de l'input 'permissions' présent dans le formulaire, j'ajoute une à une (explode) ces permissions au groupe créé à partir de la chaîne de caractère du champ.

```
$perm = explode(',',(string)$this->request->getPost('permissions'));
foreach ($perm as $permission) {
    $this->groupBase->addPermission($permission);
}
$this->authorizeGroup->save($this->groupBase);
```

Le groupe ainsi défini par ses permissions peut être sauvegardé dans la table settings avec pour références :

- La class Config\AuthGroups
- La key matrix

class	key	value
Config\AuthGroups	matrix	a:9:{s:10:"superadmin";a:5:{i:0;s:7...
Config\AuthGroups	groups	a:9:{s:10:"superadmin";a:2:{s:5:"ti...

La création du groupe se fait de la manière suivante :

```
if ($this->request->getPost('ajouter')) {
    $this->groupBase->title = $this->request->getPost('group_name');
    $this->groupBase->alias = $this->groupBase->title;
    $this->groupBase->description = $this->request->getPost('group_description');
    //sauvegarde du nouveau groupe
}
```

Et est enregistré dans la même table mais à la clé groups.

5.9.4 Conception du formulaire

Le formulaire s'ouvre à l'appel de l'URL base_url('group/add') précisé dans le form_open. Le groupe se définit par son nom, je crée donc un input qui est chargé de le récupérer. Le input pour la description de ce dernier.

```
<div class="form-floating mb-1">
    {{ form_input('group_name', set_value('group_name'), {class: 'form-control'}) }}
    <label for="group_name">Nom de groupe</label>
</div>
```

La suite requière de pouvoir définir un ensemble de permissions et de les récupérer.

```
<h4>Veuillez définir les permissions</h4>
<table class="table table-bordered border-primary">
    {{ form_input('permissions', set_value('permissions'),
        {class: 'form-control', id:'permissions', 'readonly': 'readonly'}) }}
    <label for="permissions">Permissions</label>
```

Je décide donc de créer un champ dans lequel je vais récupérer les permissions qui seront cochées dans le tableau.

Celui-ci est bien entendu en readonly de manière à ne pas permettre l'ajout de permissions non existantes.

L'ajout de ce champ permet également d'avoir un retour visuel immédiat pour s'assurer que les méthodes cochées dans le tableau sont bien prises en compte.

L'ensemble des permissions est géré par un tableau qui répertorie tous les contrôleurs par leur nom, définis par leur chemin d'accès relatif. Cela permet de les référencer.

Seuls les contrôleurs qui ont des méthodes sont affichés

```
{% for controllerName, methods in controller.methodes %}
    <!-- Boucle sur les contrôleurs et méthodes -->
    {% if methods|length > 0 %}
        <!-- Vérifier si le contrôleur a des méthodes -->
        <tr>
            <td>{{ controllerName }}</td>
```

Les méthodes des contrôleurs sont nommées de manière standard et suivent généralement une structure CRUD classique :

```
Create = add,
Read = index,
Update = edit,
Delete = delete.
```

Je décide donc d'organiser l'affichage de celle-ci selon l'organisation suivante.

Si le nom de la méthode correspond à un des 4 cas ci-dessus la check-box est créée dans cette colonne.

```
{% for column in ['index', 'edit', 'add', 'delete', 'other'] %}
    <!-- Boucle sur les colonnes -->
    <td>
        {% if column in methods %}
            <!-- Vérifier si la méthode existe -->
            <div class="form-check form-check-inline">
                <input class="form-check-input" type="checkbox"
                    value="{{ controllerName ~ '.' ~ column }}" name="selected_methods[]"
                >
                <label class="form-check-label" for="{{ controllerName ~ '_' ~ column }}"
                    {{ column|capitalize }}
                </label>
            </div>
```

Dans le cas contraire elle est mise dans la colonne autre.

En prenant en compte toutes ces considérations, le rendu final obtenu est le suivant.

Création d'un nouveau groupe

Veuillez définir les permissions

Tableau des permissions

Contrôleur	Afficher	Mettre à jour	Ajouter	Supprimer	Autres
<input type="checkbox"/> Account					<input type="checkbox"/> Login <input type="checkbox"/> Disconnect <input type="checkbox"/> Resetpassword
<input type="checkbox"/> Adresses		<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Group	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Publishers	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	<input type="checkbox"/> Detail
<input type="checkbox"/> Settings\Civilites	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Settings\iteration2\iter3\Troisiemeiteration	<input type="checkbox"/> Index		<input type="checkbox"/> Add		<input type="checkbox"/> Getlist
<input type="checkbox"/> Settings\Menu	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Settings\Pays	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Settings\Profils	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	<input type="checkbox"/> Detail
<input type="checkbox"/> Settings\Tva	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Settings\TypesAdresses	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	
<input type="checkbox"/> Users	<input type="checkbox"/> Index	<input type="checkbox"/> Edit	<input type="checkbox"/> Add	<input type="checkbox"/> Delete	

Valider

Afin de faciliter le travail de l'administrateur je prévois une check-box au niveau du nom du contrôleur qui me permet d'ajouter toutes les méthodes de celui-ci par défaut.

6 Sécurité

6.1 Le CSRF

`csrf_field()` est une fonction utilisée dans certains frameworks de développement web, pour générer un champ de formulaire caché contenant un jeton CSRF. Ce jeton CSRF est utilisé pour protéger les formulaires contre les attaques CSRF.

Lors de l'utilisation de `csrf_field()`, le framework génère automatiquement un champ de formulaire caché contenant le jeton CSRF approprié. Ce champ est généralement inclus dans les formulaires en utilisant la syntaxe du framework.

```
{{ form_open(base_url('users/add'), 'class="mt-5"') }}  
    {{ csrf_field() }}
```

Les attaques CSRF (Cross-Site Request Forgery) sont des attaques de sécurité qui exploitent la confiance entre un site web et ses utilisateurs authentifiés. L'attaque CSRF se produit lorsque des actions non autorisées sont effectuées au nom d'un utilisateur sans son consentement.

6.2 La session

Dans le projet pour l'imiter l'accès au site j'utilise une authentification par session.

Une session utilisateur est un mécanisme utilisé pour suivre et maintenir l'état d'un utilisateur lorsqu'il interagit avec un site web. Elle permet de stocker temporairement des informations spécifiques à cet utilisateur pendant une période de temps définie.

Lorsqu'un utilisateur accède à un site web, une session utilisateur est créée pour lui.

6.2.1 Création de la session

Quand un utilisateur se connecte au site une méthode va vérifier si cet utilisateur est autorisé à accéder au site.

La méthode login se lance et va

```
public function login()
{
    if (auth()->loggedIn()) {
        // Utilisateur déjà connecté
        return $this->twig->display('app/index', $this->data);
    }
}
```

Dans un premier temps je vérifie si l'utilisateur est déjà connecté.

Le `auth()->loggedIn()` est une méthode de shield qui permet de voir si un utilisateur est connecté. Ça renvoie un booléen.

```
if ($this->request->getPost('valider')) {
    //creation de la session de l'utilisateur
    $credentials = [
        'email' => $this->request->getPost('email'),
        'password' => $this->request->getPost('password'),
    ];
    //creation du token session
    $loginAttempt = auth()->remember()->attempt($credentials);
}
```

S'il n'est pas connecté je récupère son email et son mot de passe, et je tente de créer une session utilisateur avec `auth()->remember()->attempt($credentials)` en utilisant les informations d'identification fournies.

```

if (!$loginAttempt->isOK()) {
    return redirect()->back()->with('error', $loginAttempt->
}
if ($loginAttempt->isOK()) {
    //la session est ok

```

Après si la tentative de connexion a échoué avec (`!$loginAttempt->isOK()`), l'utilisateur est redirigé vers la page précédente avec un message d'erreur.

Si la tentative de connexion a réussi avec (`$loginAttempt->isOK()`), l'utilisateur est redirigé vers la page 'app/index'.

6.2.2 Ce que contient une session

Une session contient des informations spécifiques à un utilisateur pendant sa visite sur un site web. Ces informations sont stockées temporairement et peuvent être utilisées pour personnaliser l'expérience de l'utilisateur, maintenir son état et assurer une interaction cohérente entre les différentes pages du site.

`public string $cookieName = 'ci_session';` la variable qui stocke le nom du cookie de session.

`public int $expiration = 7200;` la variable qui représente la durée de la session.

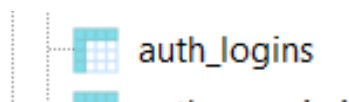
`public bool $matchIP = true;` la variable qui stocke une valeur booléenne indiquant si la correspondance d'adresse IP est activée ou non

`public ?string $DBGGroup = null;` la variable qui représente le groupe auquel appartient l'utilisateur.

6.3 Session avec BDD

Quand un utilisateur se connecte, sur la base de données la tentative de connexion est stockée. Cela permet de voir les tentatives de connexion qui ont été faites sur le site, et permet de contrôler si une personne ou un logiciel malveillant tente de se connecter sans être autorisé.

L'enregistrement se fait sur la table auth_logins.



Sur cette table on retrouve différentes colonnes, un id, ip_address, user_agent, id_type, identifier (qui est l'adresse mail), user_id (l'id de l'utilisateur), date (date de tentative de connexion), success (qui est 1 ou 0)

La colonne user_agent qui va être une chaîne de texte qui identifie le logiciel client, tel qu'un navigateur web, utilisé par un utilisateur pour accéder à un site web.

user_agent
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv...

La colonne id_type qui saura comment il a tenté de se connecter : ici par email et mot de passe

id_type
email_password

La colonne identifiée qui est l'adresse email

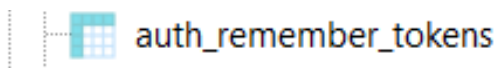
identifier
dimitriguerout@gmail.com

Les colonnes user id et date et success, si le user est pas dans la base ou qu'il s'est trompé de mot de passe alors l'id_user sera null et le success sera 0

user_id	date	success
(NULL)	2023-05-26 13:49:40	0

Si la connexion s'est bien passée alors sur la table `auth_remember_token` une ligne sera insérée. Cette ligne permettra à l'utilisateur de pouvoir rester connecté temps que sa session n'a pas expiré ou qu'il s'est pas déconnecté.

Même s'il décide de fermer son navigateur web et de revenir dessus sa session restera ouverte. Une fois qu'il a fermé sa session cette ligne est automatiquement supprimée.



selector	hashedValidator	user_id	expires	created_at	updated_at
18066f92b37bd961310db2e5	72f95395ddcf6439b40e7a544410eabfb1ecf3a9...	1	2023-06-28 13:08:18	2023-05-29 13:08:18	2023-05-2

6.4 Auth-Rates Filtre

Pour aider à protéger les formulaires d'authentification contre le spam par des bots, il est recommandé d'utiliser le `auth-rates` filtre sur toutes les routes d'authentification.

```
public array $filters = [
    'auth-rates' => [
        'before' => [
            'login*', 'register', 'auth/*'
        ]
    ],
]
```

Cela permet d'appliquer des mécanismes de protection contre des attaques. On peut limiter le nombre de tentatives de connexion autorisées dans un certain laps de temps, bloquer une adresse IP après un certain nombre d'échecs de connexion.

7 Gestion du projet

7.1 GitLab

Un environnement sur gitlab m'a été créé.

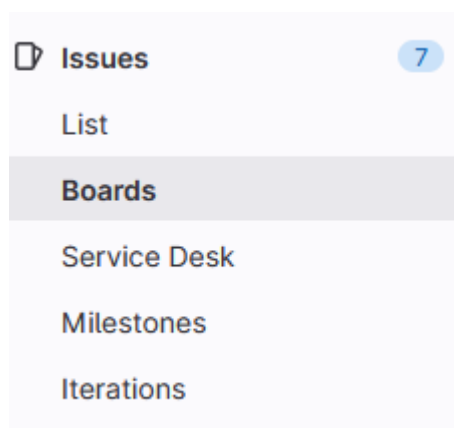
Une fois connecté à gitlab, différents onglets sont présents, ils ont tous un lien avec le projet.

Le dossier Issues qui permet le suivi du projet qui est divisé en ticket.

Les tickets sont utilisés pour suivre, gérer et organiser le travail. Ils permettent de signaler les bogues, les demandes de fonctionnalités, les tâches à effectuer.

Chaque ticket représente une unité de travail qui va être affectée à un membre de l'équipe.

l'onglet Issues



sur l'onglet Boards on va trouver un tableau avec les tickets à traiter, ceux en cours et le ou les membres qui travaillent dessus, ceux qui sont finis qui doivent être vérifiés et ceux qui sont clos.

Pour le projet j'ai créé un ticket pour chaque fonctionnalité à rajouter.

Par exemple pour modifier un utilisateur

Je crée un ticket "edit user" que je m'assigne. Une fois créé, je crée un merge request pour créer une nouvelle branche sur le projet qui portera un numéro pour l'identifier rapidement.

Cette branche sera utilisée pour ce ticket puis une fois terminée et revérifiée, sera ajoutée à la branche principale.

Exemple de création d'un ticket

edit user

0
 0

Create merge request ▾

Drag your designs here or [click to upload](#).

Tasks 0
 Add ▾ ^

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0
 Add ^

Link issues together to show that they're related or that one is blocking others. [Learn more](#).

Related merge requests 1

When this merge request is accepted, this issue will be closed automatically.

Draft: Resolve "edit user" !27

Exemple du Boards où le ticket est en cours

à faire

1 0

en cours

4 0

edit user

#28

La récupération de la branche se fait avec le logiciel fork qui permet de voir les branches, de faire des commit du code que l'on a effectué et de pouvoir tirer les modifications qui ont été effectuées sur le projet. Il permet aussi de voir quelles modifications on été effectuées sur le projet et de choisir si on veut les appliquer ou pas.

Branches	
1-mise-en-place-des-droits-profils	
24-reset-password	
✓ 28-edit-user	21
master	11

7.2 Méthode scrum

La méthode Scrum est un cadre de gestion de projet Agile, principalement utilisé dans le développement logiciel, qui permet de gérer efficacement des projets complexes et en évolution rapide. Voici les principaux éléments de la méthode Scrum :

1. Le Product Owner est responsable de la vision du produit et représente les intérêts des parties prenantes (clients, utilisateurs, etc.). Il définit les objectifs du projet, priorise les fonctionnalités à développer et gère le backlog du produit, qui est une liste ordonnée des éléments à réaliser.
2. Le Scrum Master est responsable de la mise en œuvre et de la promotion des pratiques Scrum. Son rôle principal est de faciliter le processus Scrum, d'aider l'équipe à être autonome et d'éliminer les obstacles qui peuvent entraver la progression du projet.
3. L'équipe de développement est responsable de la réalisation du travail. Elle est auto-organisée, pluridisciplinaire et généralement composée de 5 à 9 membres. L'équipe estime la complexité des tâches à réaliser et s'engage à livrer des fonctionnalités de qualité à la fin de chaque sprint.
4. Les sprints sont des itérations de durée fixe, généralement de deux à quatre semaines, au cours desquelles l'équipe de développement réalise les fonctionnalités du produit. Au début de chaque sprint, l'équipe sélectionne les éléments du backlog du produit à réaliser et établit un objectif de sprint.
5. Plusieurs réunions rythment le processus Scrum. La réunion quotidienne de synchronisation (daily Scrum) est une brève réunion quotidienne où chaque membre de l'équipe partage son avancement, identifie les obstacles et coordonne son travail.

L'approche Scrum se caractérise par une forte collaboration, une adaptation aux changements et une transparence entre les membres de l'équipe et les parties prenantes. Le processus itératif et incrémental de Scrum permet de fournir des résultats rapidement et d'obtenir des feedbacks réguliers pour orienter le développement du produit.

8 Bilan

8.1 Apports et difficultés rencontrées

Ce projet m'a apporté beaucoup de connaissances sur la programmation avec de nouvelles technologies, que j'ai dû apprendre rapidement.

Il m'a permis de travailler en équipe et d'échanger avec elle sur l'avancement du projet au travers de réunions chaque matin.

J'ai approfondi mes connaissances et l'utilisation de gitlab.

J'ai dû faire des recherches supplémentaires pour comprendre certains concepts de programmation notamment sur codeigniter4\shield. Ceci m'a permis de débloquer certaines situations et d'approfondir la compréhension du langage.

Les difficultés que j'ai pu rencontrer ont été au début sur l'utilisation de codeigniter4. J'ai passé beaucoup de temps pour comprendre et chercher beaucoup d'informations.

Je sais dorénavant comment fonctionne le travail en équipe avec la méthode Agile\scrum qui je trouve est un bon moyen de voir les projets avancer et une bonne technique pour avancer plus vite.

Pour moi, j'ai su gérer au mieux le projet.

8.2 Perspectives

Après avoir suivi une formation de Concepteur Développeur d'Applications (CDA) et réalisé deux périodes en entreprise (PAE). Cette expérience a non seulement consolidé mes connaissances théoriques, mais elle m'a également permis d'acquérir une expérience pratique précieuse dans le domaine du développement.

Au cours de ma formation, j'ai pu explorer en profondeur les différents aspects du développement, de la conception à la mise en œuvre, en passant par la gestion de projet. De plus, j'ai appris à travailler en équipe.

Les PAE m'ont offert l'opportunité de mettre en pratique tout ce que j'avais appris lors de ma formation. J'ai été impliqué dans des projets concrets, ce qui m'a permis de perfectionner mes compétences et de découvrir de nouvelles technologies. J'ai travaillé en collaboration avec des professionnels expérimentés, ce qui m'a donné une perspective réelle du monde du travail en tant que développeur.

Fort de cette expérience réussie, mon objectif principal est maintenant de poursuivre mes études en alternance. Je suis convaincu que l'alternance me permettra de combiner les connaissances théoriques acquises lors de ma formation avec une expérience professionnelle continue. Cela me permettra de me familiariser davantage avec les exigences réelles du marché du travail et de développer des compétences polyvalentes et adaptées aux besoins des entreprises.

9 Partie fil rouge

Le projet fil rouge est une mise en situation professionnelle qui nous a été remis en début de formation. Dans cette partie je vais parler de la partie concevoir et développer la persistance des données en intégrant les recommandations de sécurité.

9.1 Liste des compétences du référentiel couverte

Cette partie montre l'acquisition et les compétences professionnelles qui font partie de l'activité 2 « concevoir et développer la persistance des données ». Il s'agit de concevoir une base de données, de mettre en place la base de données et de développer des composants dans le langage de la base de données.

9.2 Les outils utilisés



- Looping est un logiciel qui vous permet d'organiser et de structurer vos données en diagrammes et représentations graphiques ordonnées



- Le SGBD Oracle est un système de gestion de base de données relationnelle



- L'IDE Oracle SQL developer est l'outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL (Structured Query Language) qui est utilisé pour créer et administrer des bases de données relationnelles.



Visual Paradigm

- Visual Paradigm permet de réaliser des UML

9.3 Contexte du projet

Le projet fil rouge « La Fourchette » nous a été donné en parallèle de notre enseignement, il sera donc divisé en plusieurs étapes pour mettre en évidence les acquis de la formation.

Pour une mise en situation professionnelle d'un développeur d'application, un cahier des charges nous a été remis le 16 septembre 2021, date à laquelle le projet est lancé. Les formateurs jouent le rôle du client pour toutes questions et besoins de précision.

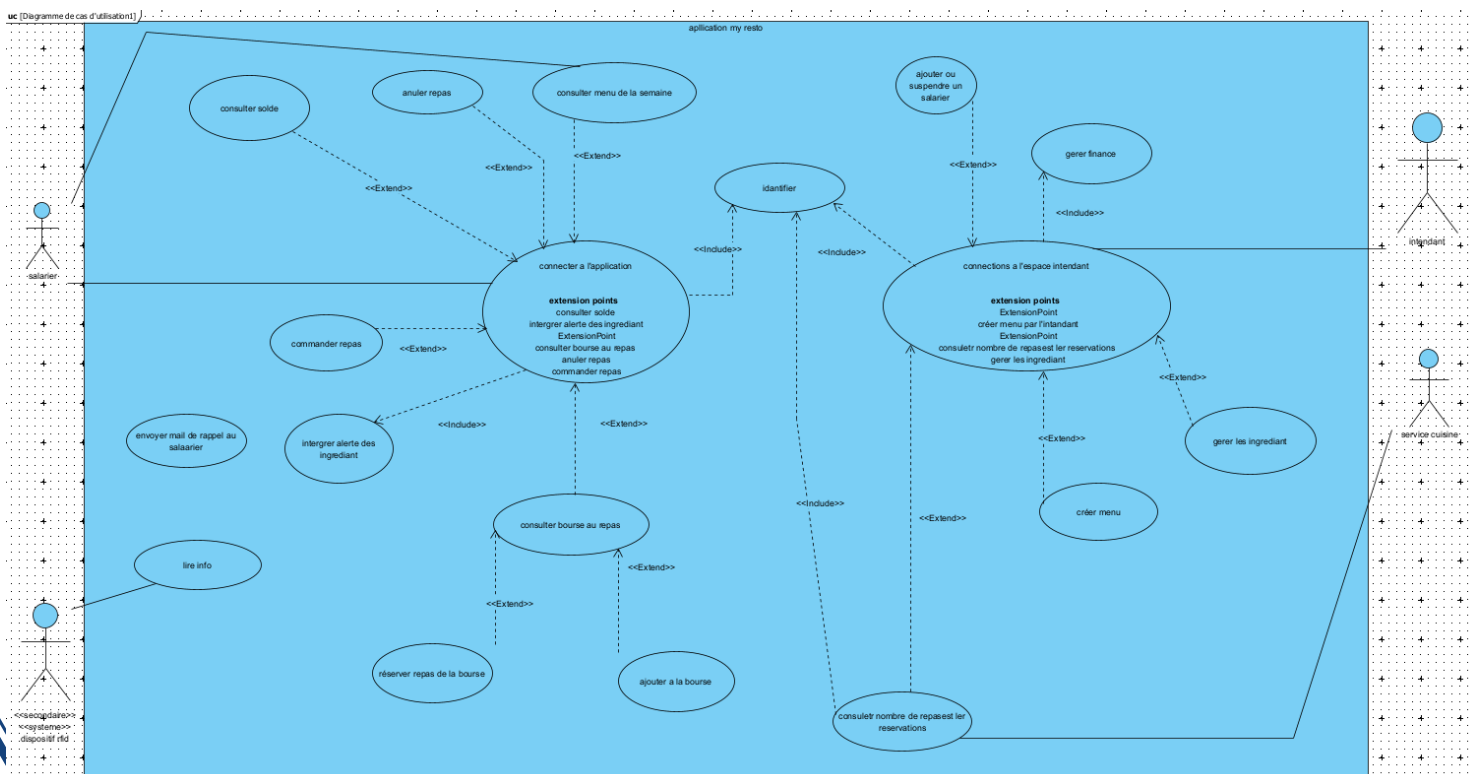
9.4 Contexte et besoin de l'entreprise

L'entreprise « La Fourchette » spécialisée dans le domaine de la restauration collective possède son propre restaurant d'entreprise. Elle nous a remis un cahier des charges de son projet. Elle nous demande d'informatiser par une application web «My Resto» et une Desktop «our Resto», la prise d'inscription pour connaître le nombre de personnes qui mangent et ainsi éviter le gaspillage alimentaire. Les repas sont du lundi au vendredi midi et soir avec une annulation jusqu'au jeudi midi de la semaine précédente. La revente du repas doit être possible. Dans cette parti je traite uniquement la partie base de données».

10 Analyse du besoin

10.1 Cas d'utilisation

Pour commencer, je crée un diagramme de cas d'utilisation avec les informations du cahier des charges pour présenter le fonctionnement du système et les acteurs. Ce diagramme est très utile pour un développeur.



Sur le diagramme, quatre utilisateurs sont identifiés :

1. Le salarié

Il aura accès à :

- Son espace avec son login et son mot de passe.
- Il aura accès à des fonctions : réserver son repas, dire quels sont ses allergènes, consulter le menu de la semaine en cours et celle de la semaine prochaine, consulter son solde et d'autres actions.

2. Le service cuisine

-Il devra aussi s'identifier. Il a les mêmes droits que les salariés, et en plus, disposera d'un accès pour consulter le nombre de repas et les réservations.

3. L'intendant

Il devra s'identifier, il dispose de droits supplémentaires : gérer les finances, ajouter ou suspendre un salarié, créer les menus et gérer les ingrédients.

4 le système secondaire

Qui est là pour lire les informations de la carte RFID au passage du client.

11 MCD, MLD

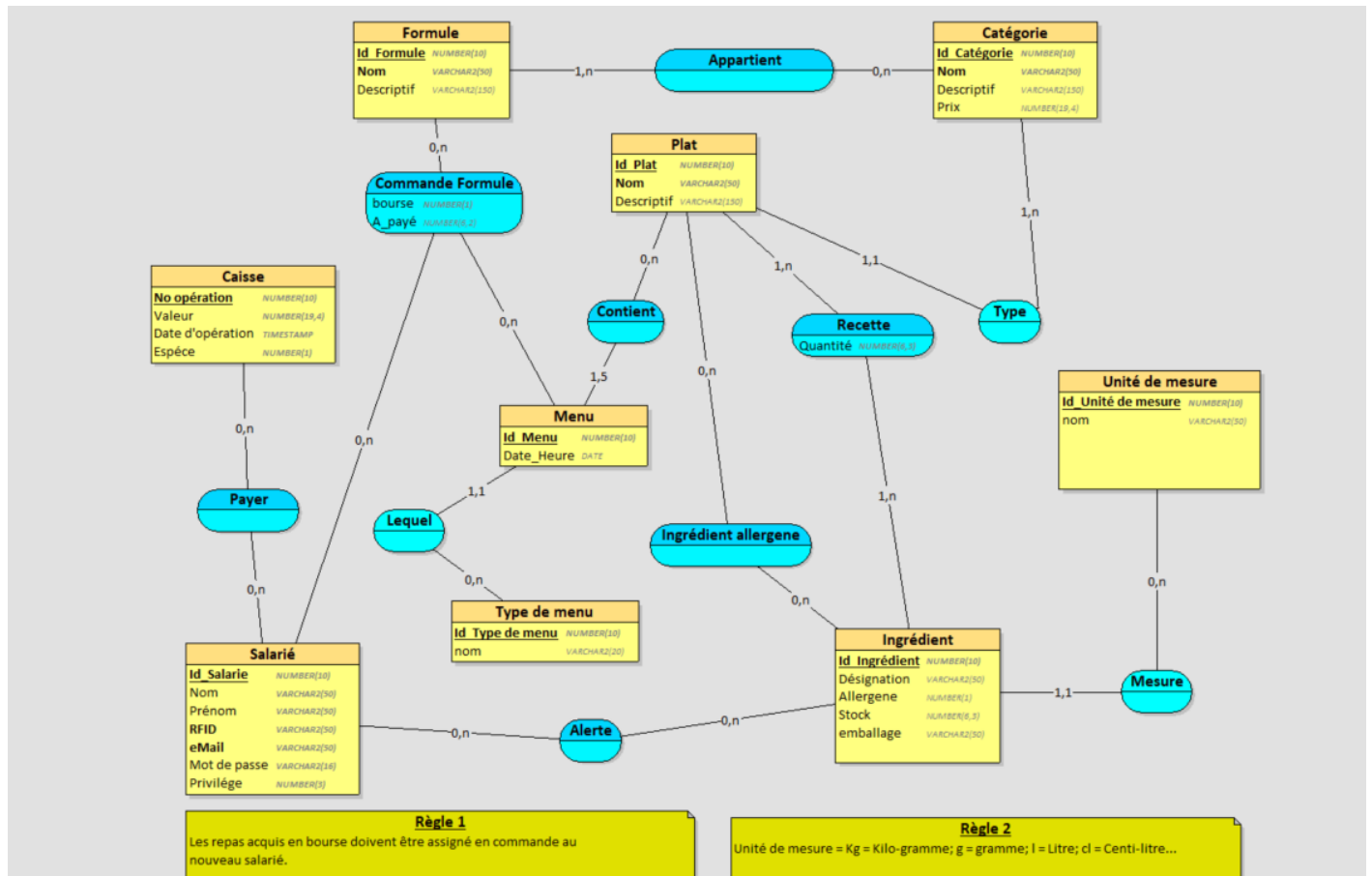
11.1 Le MCD (Modèle Conceptuel des Données)

Il permet de spécifier tout ce qui est nécessaire pour construire une base de données relationnelle normalisée, avec la méthode Merise.

Pour réaliser un MCD il faut éviter les redondances d'attribut, les attributs doivent être non décomposables et dépendre directement de l'identifiant de l'entité.

On y trouve :

- Des entités qui ont un identifiant unique naturel
- Des attributs qui dépendent directement de l'identifiant
- Des associations qui sont en lien entre des occurrences d'entités reliées.
- Les cardinalités de chaque coté expriment le nombre minimum et le nombre maximum d'occurrences pouvant exister pour une entité.



11.2 Le MLD (Modèle Logique des Données)

À partir du MCD on en déduit le schéma de la base de données relationnelle.

Toute entité devient une table et chaque attribut devient une colonne, l'identifiant devient une clé primaire. Dans les tables on peut trouver une ou plusieurs clés étrangères qui sont l'identifiant d'autres tables. Une association va créer une nouvelle table avec une Clé primaire qui est composée de plusieurs clés primaires des tables mises en relation. Le MLD est représenté à l'écrit.

On y retrouve :

- Les tables
- Les attributs
- Les clés primaires
- Les clés étrangères

Les clés primaires sont représentées en premier dans chaque table.

Les clés étrangères sont représentées avec un #.

Pour les tables où il n'y a pas de clés primaires au début, ce sont toutes les clés étrangères qui font la clé primaire.

12 Mise en place de la base de données

12.1 Création des tables

Par la suite je crée chaque table, pour se faire j'utilise l'IDE Oracle SQL developer.

Pour créer une table je lui donne son nom, son identifiant unique et ses attributs ainsi que leur type et leur longueur. Je leur applique leurs contraintes en même temps. Il faut créer les tables dans un ordre précis à cause des contraintes de clés primaires qui deviennent Clé étrangère dans d'autres tables.

Toutes les tables ne possèdent pas encore de données.

Avant de créer les tables je les drop pour être sûr que ma base de données soit vide.

```
DROP TABLE IngedAllergene;
```

```
DROP TABLE Appartient;
```

```
DROP TABLE Payer;
```

```
DROP TABLE Recette;
```

```
DROP TABLE Alerte;
```

```
DROP TABLE Contient;
```

```
CREATE TABLE Salarie(
```

```
  Id_Salarie NUMBER(10),
```

```
  Nom VARCHAR2(50) NOT NULL,
```

```
  Prenom VARCHAR2(50) NOT NULL,
```

```
  RFID VARCHAR2(50),
```

```
  eMail VARCHAR2(50) NOT NULL,
```

```
  Mot_de_passe VARCHAR2(16) NOT NULL,
```

```
  Privilege NUMBER(3) NOT NULL,
```

```
  CONSTRAINT Salarie_PK PRIMARY KEY(Id_Salarie),
```

```
  CONSTRAINT Salarie_AK UNIQUE(RFID),
```

```
  CONSTRAINT Salarie_1_AK UNIQUE(eMail)
```

12.2 Élaboration du jeu d'essai

Afin de tester la base et le bon fonctionnement du modèle physique des données

Je crée un jeu d'essai qui consiste à remplir les tables. Pour se faire je fais un script avec « des insert » il ne faut pas oublier de remplir tous les attributs s'ils ne peuvent être nuls.

Table des salarie

```
--
```

```
INSERT INTO SALARIE (ID_SALARIE, PRENOM, NOM, RFID, EMAIL, MOT_DE_PASSE, PRIVILEGE)
VALUES (1, 'SMAIL', 'ABARDOUZ', '76011664', 'abardouz7992@gmail.com', 'YHLPbmhk', 1);
```

```
INSERT INTO SALARIE (ID_SALARIE, PRENOM, NOM, RFID, EMAIL, MOT_DE_PASSE, PRIVILEGE)
VALUES (2, 'ADRIEN', 'ADOLPHE', '134079582', 'adrien229@gmail.com', 'Jl@P^OWs', 1);
```

13 Réalisation

13.1 Requêtes les plus significatives et résultats obtenus

Dans le cahier des charges, on me demande différentes requêtes à effectuer (je les ai toutes effectuées à la conception), je choisis la requête 8 qui est de lister les 5 salariés ayant mangé le plus souvent au restaurant d'entreprise en 2021.

-- 8. Lister les 5 salariés ayant mangé le plus souvent au restaurant d'entreprise en 2021.

```
SELECT s.id_salarie, s.nom, s.prenom, count( s.nom) As "Nombre de resa"
FROM salarie s join cmd_form cmd
on s.id_salarie = cmd.id_salarie
join menu m
on m.id_menu = cmd.id_menu
where TO_CHAR (m.date_heure,'DD,MM,YYYY') BETWEEN TO_DATE ('01/01/21') and ('31/12/21')
group by s.nom,s.id_salarie, s.prenom
ORDER by count(s.nom) desc
FETCH FIRST 5 ROWS ONLY;
```

ID_SALARIE	NOM	PRENOM	Nombre de resa
10	DES PLACES	CHARLOTTE	6
17	MARQUET	LUDWIG	4
15	JOSSAN	QUENTIN	3
8	CHEN	YIJIONG	3
24	RAFFIER	NICOLAS	3

FETCH FIRST 5 ROWS ONLY (permet de sélectionner uniquement les 5 premiers de la liste)

On remarque que le résultat nous donne bien que les cinq personnes qui ont mangé le plus. Après avoir écrit et testé chaque requête je constate que le base de données fonctionne correctement.

13.1.1 en PL/SQL

Le PL/SQL est un langage de programmation procédurale et structurée, créé par Oracle et utilisé dans le cadre de bases de données relationnelles.

Différences entre procédures et fonctions :

Procédures : Une procédure stockée est un bloc de code PL/SQL nommé et défini par l'utilisateur. Elle est composée d'instructions compilées et enregistrées dans la base de données. Elle peut être exécutée à partir des applications ou d'autres procédures stockées.

Fonctions : Les fonctions sont semblables aux procédures, mais retournent une valeur résultat. Une fonction diffère d'une procédure par le fait qu'on peut l'utiliser dans une expression.

On me demande de programmer des procédures stockées et des fonctions (les autres requêtes en annexe)

Je choisis une fonction pour valider le format d'une adresse mail avec un retour en Boolean. Dans cette fonction on retrouve « REGEXP_LIKE » qui est une expression régulière par laquelle on lui dit que les mails doivent avoir une seule possibilité d'être valide.

J'effectue des recherches pour trouver comment l'utiliser.

```
-- Créer une fonction pour valider le format d'une adresse mail.
create or replace FUNCTION valid_email (
  email IN salarie.email%TYPE)
RETURN NUMBER -- Retournera 0 ou 1 en fonction de la validité de l'email
IS
BEGIN
  if REGEXP_LIKE (email, '^([a-zA-Z0-9._%~]+@[a-zA-Z0-9._%~]+\.[a-zA-Z]{2,4})$')
  THEN
    RETURN 0; -- L'email est correct
  ELSE
    RETURN 1; -- L'email est incorrect
  END IF;
END;
```

```
CLEAR SCR;
SELECT *
FROM salarie
WHERE valid_email(salarie.email)=1;
```

Élément Function VALID_EMAIL compilé

Une fois la fonction créée je la teste, pour ce faire je crée une requête qui va appeler cette fonction et je lui demande de me retourner toutes les adresses mail qui ne sont pas au bon format.

PRENOM	RFID	EMAIL
ODILE	76012181	972odile@gmail.,com

14 Gestion de la sécurité

14.1 Sauvegarde de la base de test

Pour maintenir une sécurité je crée une sauvegarde qui serre :

- Pour sauvegarder l'état d'une base avant de tester une procédure qui met à jour les données par exemple
- Pour rejouer des scénarios de tests
- Pour migrer un jeu de données d'une base de données à une autre ou d'un serveur Oracle à un autre.
- En cas de problème sur le serveur pour récupérer au plus vite une base de données correcte

```
-----
-- Fichier créé - mardi-novembre-02-2021
-----
-- DDL for Table SALARIE
-----
CREATE TABLE "CDA6_GUER_FILROUGE"."SALARIE"
(
  "ID_SALARIE" NUMBER(10,0),
  "NOM" VARCHAR2(50 BYTE),
  "PRENOM" VARCHAR2(50 BYTE),
  "RFID" VARCHAR2(50 BYTE),
  "EMAIL" VARCHAR2(50 BYTE),
  "MOT_DE_PASSE" VARCHAR2(16 BYTE),
  "PRIVILEGE" NUMBER(3,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
-----
REM INSERTING into CDA6_GUER_FILROUGE.SALARIE
SET DEFINE OFF;
Insert into CDA6_GUER_FILROUGE.SALARIE (ID_SALARIE,NOM,PRENOM,RFID,EMAIL,MOT_DE_PASSE,PRIVILEGE) values ('1','ABARDOUZ','SMAIL','76011664','abardouz7992@gmail.com','YHLPbmhk','1');
Insert into CDA6_GUER_FILROUGE.SALARIE (ID_SALARIE,NOM,PRENOM,RFID,EMAIL,MOT_DE_PASSE,PRIVILEGE) values ('2','ADOLPHE','ADRIEN','134079582','adrien229@gmail.com','Jl@P^OWs','1');
Insert into CDA6_GUER_FILROUGE.SALARIE (ID_SALARIE,NOM,PRENOM,RFID,EMAIL,MOT_DE_PASSE,PRIVILEGE) values
-----
-- DDL for Index SALARIE_PK
-----

CREATE UNIQUE INDEX "CDA6_GUER_FILROUGE"."SALARIE_PK" ON "CDA6_GUER_FILROUGE"."SALARIE" ("ID_SALARIE")
PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

Pour faire la sauvegarde je passe par SQL Developer :

- J'exporte la base de données
- Je sélectionne toutes les données à exporter et je donne le chemin de la sauvegarde
- Le fichier se trouve sur le disque C dans le dossier Backup avec la date du jour.

15 Bilan

15.1 Apports et difficultés rencontrées

Ce projet m'a apporté beaucoup de connaissances sur la base de données, une technique de travail différente, où tout se fait par étape, et un langage SQL qui m'était inconnu auparavant. J'ai dû faire des recherches supplémentaires sur ce langage pour certaines requêtes. Ceci m'a permis de débloquer certaines situations et d'approfondir la compréhension du langage.

Les difficultés que j'ai pu rencontrer ont été au niveau de la mise en application de certaines requêtes. Après plusieurs recherches et avec l'aide de mes collègues de promotion, j'ai réussi à les finaliser.

Au départ, je m'étais fixé un calendrier précis des différentes étapes à réaliser. Je n'ai pas toujours réussi à tenir les délais par manque de connaissances et d'expériences.

Je sais dorénavant le temps et le travail pour le développement d'une base de données. Aussi, je saurai mieux gérer mon organisation.

