

# RAPPORT 2

***M1 Informatique BDIA***

Projet Open Data Métropole  
**2024-2025**

**Travail réalisé par :**

Anthony MICHAUD	Kévin PRADIER
Aichetou N'DIAYE	Ahmed KAIDI
Wassim ENNAJI	Alan GUILLEMINOT
Imane CHOUKRI	Aya ELHASSANI
Trinite MOMBOULI	Rafik RHARMAOUI

**Encadré par :**

Marinette SAVONNET	Nadine CULLOT
Claire BOURGEOIS-REPUBLIQUE	



## Table des matières

I.	Contexte et objectifs du projet .....	3
1.	Projet.....	3
2.	Objectifs .....	3
3.	Solution proposée.....	4
II.	L'équipe de développement .....	4
1.	Organigramme.....	4
2.	Présentation de l'équipe .....	4
3.	Moyens de communication.....	5
III.	Choix techniques .....	6
IV.	Fonctionnalités .....	8
V.	Dépendances .....	10
VI.	Fonctionnalités implémentées .....	10
1.	Création de compte .....	11
2.	Connexion.....	13
3.	Modification des informations du compte .....	15
4.	Affichage des données .....	17
5.	Recherche des données .....	20
6.	Filtrage des données.....	21
7.	Téléchargement des données .....	22
8.	Importation des données .....	23
9.	Choix de la licence.....	24
VII.	Normes et tests unitaires .....	25
VIII.	Organisation du groupe .....	26
IX.	Conclusion .....	27
X.	Liens .....	28
XI.	Bibliographie.....	29
XII.	Annexes .....	30

## Table des illustrations

Figure 1 : Organigramme de l'équipe du projet .....	4
Figure 2 : Logo de mattermost .....	5
Figure 3 : Logo de GitLab .....	5
Figure 4 : Logo de OpenProject .....	6
Figure 5 : Présente les interactions entre les technologies choisis .....	7
Figure 6 : Diagramme d'utilisation .....	9
Figure 7 : Présente le MCD de la partie utilisateur .....	12
Figure 8 : Tâches création de compte .....	13
Figure 9 : Tâches connexion .....	15
Figure 10 : Tâches modification des informations du compte .....	17
Figure 11 : MCD partie données .....	18
Figure 12 : Tâches affichage des données .....	19
Figure 13 : Diagramme séquence recherche .....	20
Figure 14 : Tâche recherche des données .....	21
Figure 15 : Diagramme séquence filtre .....	21
Figure 16 : Tâche filtre des données .....	22
Figure 17 : Tâches télécharger des données .....	23
Figure 18 : Tâches importation des données .....	24
Figure 19 : Tâches choix de la licence .....	24
Figure 20 : Tests unitaires validé .....	25
Figure 21 : Tableau de répartition des fonctionnalités par membre .....	27
 Tableau 1 : Fonctionnalités du portail .....	8
Tableau 2 : Table des fonctionnalités et leurs dépendances .....	10
Tableau 3 : Table des fonctionnalités et leurs rôles .....	26

# I. Contexte et objectifs du projet

## 1. Projet

Le projet **Open Data Métropole** répond à une initiative visant à se conformer à la loi pour une République numérique, qui depuis le 7 octobre 2016 impose aux collectivités de plus de 3 500 habitants et ayant 50 agents de publier leurs données publiques en ligne : **”Des données ouvertes pour tous, des droits numériques renforcés, une neutralité du Net garantie et une inclusion numérique pour chacun.”**.

L'objectif de cette loi est d'améliorer l'accès libre aux données publiques, tout en garantissant la protection des informations personnelles. Les données disponibles peuvent être utilisées par les citoyens, les entreprises et les développeurs pour des analyses, le développement d'applications ou encore la création de services innovants.

Parmi ces données figurent entre autres des informations temporelles (données historiques, mises à jour régulières, etc.) et des données géographiques numériques (cartes interactives, coordonnées GPS), qui offrent un potentiel important pour des usages variés dans la planification urbaine, la mobilité ou encore l'environnement.

## 2. Objectifs

Le projet **Open Data Métropole** vise à développer un portail Web qui facilite l'accès aux données publiques en conformité avec les obligations légales pour les collectivités. Il a plusieurs objectifs, comme dit précédemment notre métropole fictive doit répondre aux obligations imposées par la loi du 7 octobre 2016.

Les données doivent être accessibles pour tous les citoyens, entreprises et/ou développeurs afin d'encourager les initiatives et l'utilisation des ressources publiques. Cela permettra également aux collectivités locales de mieux comprendre les besoins de leur territoire grâce aux retours d'utilisation des données et aux analyses générées.

L'interface intuitive du portail offrira aux citoyens, entreprises et développeurs la possibilité d'explorer, consulter, télécharger et utiliser une variété de données géographiques, temporelles et statistiques, favorisant ainsi l'innovation et l'analyse. Les utilisateurs pourront créer un compte, se connecter et modifier leurs informations personnelles à tout moment. Ils pourront également consulter les données importées et les télécharger dans différents formats.

À la suite de ce document, le détail du projet est présenté ainsi que notre organisation. Nous parlerons des différentes phases du projet, des technologies, en précisant les fonctionnalités, les étapes clés et les délais prévus. Vous trouverez également une présentation de l'équipe ainsi que les outils d'organisation incluant les rôles et responsabilités de chaque membre.

### 3. Solution proposée

Pour répondre aux objectifs du projet Open Data Métropole et aux exigences de la loi sur la République numérique, nous proposons la mise en place d'un portail Web. Ce projet respecte différentes méthodes de travail ainsi que les aspects techniques, organisationnels et fonctionnels nécessaires à sa réalisation. Le portail est construit en utilisant une architecture robuste et évolutive (MVC), les données ainsi que l'application sont hébergées en local. Le projet est réalisé en suivant une méthode de travail organisée avec des rôles clairement définis pour les membres de l'équipe.

## II. L'équipe de développement

### 1. Organigramme

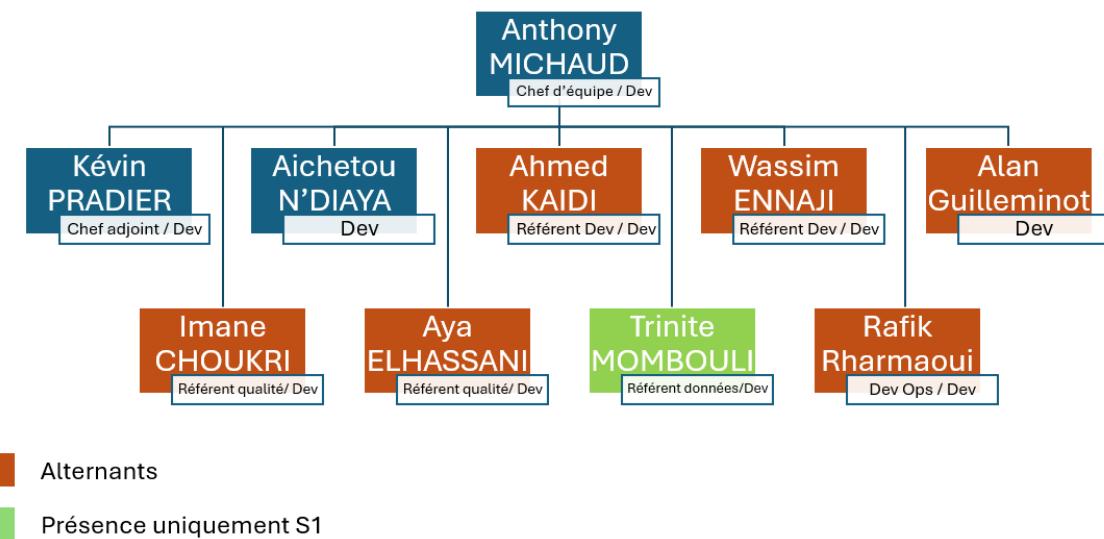


Figure 1 : Organigramme de l'équipe du projet

Dans notre équipe de développement, 70% des membres ne seront pas présents lors du 2ème semestre. Par conséquent nous devons produire un travail précis et organisé pour assurer la bonne continuité du projet. Nous nous appuierons également sur l'expérience des personnes en alternance pour améliorer la communication au sein de l'équipe.

### 2. Présentation de l'équipe

- **Chef d'équipe / Responsable d'équipe :** Le chef d'équipe est **Anthony MICHAUD**, il est responsable de la coordination et l'organisation générale du projet. Il gère la répartition des tâches, assure le respect des délais en maintenant la communication entre les membres de l'équipe et les encadrants.

- **Chef Adjoint** : Le chef adjoint est **Kévin PRADIER**. Il soutient l'organisation de l'équipe et remplace le chef d'équipe en cas d'absence.
- **Référent Développeur** : Le référent développeur est **Ahmed KAIIDI** et **Wassim ENNAJI**. Ils vérifient que le développement réalisé est globalement cohérent avec le projet et les fonctionnalités, s'assurant ainsi que chaque aspect technique respecte les spécifications et les objectifs définis.
- **Référent Données** : Le référent données est **Trinité MOMBOULI**. Il a pour mission d'assurer la cohérence des données.
- **Référent Qualité** : Les référents qualité sont **Imane CHOUKRI** et **Aya ELHASSANI**. Elles vérifient les tests réalisés et valident les fonctionnalités développées pour garantir qu'elles répondent aux normes fixées par l'équipe au début du projet.
- **Dev Ops** : Le DevOps est **Rafik RHARMAOUI**. Il s'occupe de l'installation des différents éléments de développement sur les machines virtuelles mises à disposition et veille à la cohérence des choix techniques tout au long du processus de développement.
- **Développeur** : Tous les membres du groupe sont développeurs.

### 3. Moyens de communication

- **Mattermost**



*Figure 2 : Logo de mattermost*

L'équipe de projet utilise le logiciel [Mattermost](#) : Logiciel de messagerie interne à une structure afin d'échanger sur le projet, de partager des fichiers, de garder une trace des réunions et des choix réalisés.

- **GitLab**



*Figure 3 : Logo de GitLab*

[GitLab](#) est une plateforme complète de développement DevOps qui nous permet de gérer l'ensemble du développement du projet. Nous l'utiliserons afin d'héberger notre code, avoir

un contrôle des versions et réaliser les « merges »\* des différentes fonctionnalités réalisées. Il possède également un onglet Wiki qui sera utilisé afin de fournir la documentation sur le projet (installation/configuration/etc...) et les comptes-rendus des réunions réalisées (en groupe seulement ou avec les encadrants).

- **OpenProject**



Figure 4 : Logo de OpenProject

[OpenProject](#) est une plateforme de gestion de projet open-source qui est utilisé pour organiser et suivre notre projet. Le diagramme Gantt réalisé en début de projet nous permet de nous organiser, créer les différentes phases du projet, les fonctionnalités à réaliser ainsi que les tâches des fonctionnalités. Les tâches sont ensuite assignées aux membres, qui peuvent suivre leurs avancements et prévenir s'il y a des blocages.

### III. Choix techniques

Après plusieurs échanges concernant les langages de programmation, les différents frameworks et les différents outils, nous avons finalement choisi un framework de Java : **Spring Boot**. Spring Boot permet principalement de générer et gérer des projets orientés web, en full stack. Il est adapté à des projets évolutifs grâce à ses dépendances il nous permet, entre autres, de définir une sécurité d'authentification acceptable, de générer des API rest d'une application et surtout, de gérer et de fluidifier plusieurs bases de données. En effet, nous aurons besoin, au cours de ce projet de différents types de base de données :

- Relationnelles
- Graphes
- Documents
- Ou encore temporelles

Spring Boot possède plusieurs composants et permet le développement rapide et structuré d'un projet, notamment grâce à son architecture *Model View Controller* (MVC - Modèle - Vue - Contrôleur). Ce Framework a été proposé par des membres du groupe qui possédaient déjà de l'expérience, quant aux membres n'ayant jamais eu l'occasion de l'utiliser, ils souhaitaient découvrir de nouveaux outils de travail. La prise en main de Spring Boot étant assez simple, personne n'a été laissé pour compte lors du développement du projet.

\*merges : opération qui consiste à fusionner les modifications d'une branche source avec une branche cible. Cette opération permet d'intégrer le code réalisé sur la branche source dans la branche cible.

Concernant le "Front-end" de l'interface Web nous avons utilisé **Spring Web** qui est le module pour Spring Boot le plus utilisé concernant le développement d'une application web et d'une API. Il est également couplé avec le module **Thymeleaf** qui est un moteur de template permettant de traiter de l'HTML standard. De cette manière, les membres du projet pourront coder les interfaces Web en HTML.

Concernant le "Back-end" l'API de Spring Boot sera utilisée afin de réaliser les GET/POST sur les bases de données. Nous avons décidé d'avoir 3 bases de données différentes dans le projet :

- **MySQL** (BDD relationnelle) est utilisé pour le stockage des données structurées ainsi que les fichiers de données (csv,pdf,xml,...) en utilisant des BLOB
- **Neo4J** (BDD Graphe) est utilisé pour le stockage des données n'ayant pas de structure propre.
- **PostgreSQL + TimescaleDB** est utilisé pour le stockage des données temporel (données collectées au fil du temps), comme les relevés de capteur ou le trafic routier.

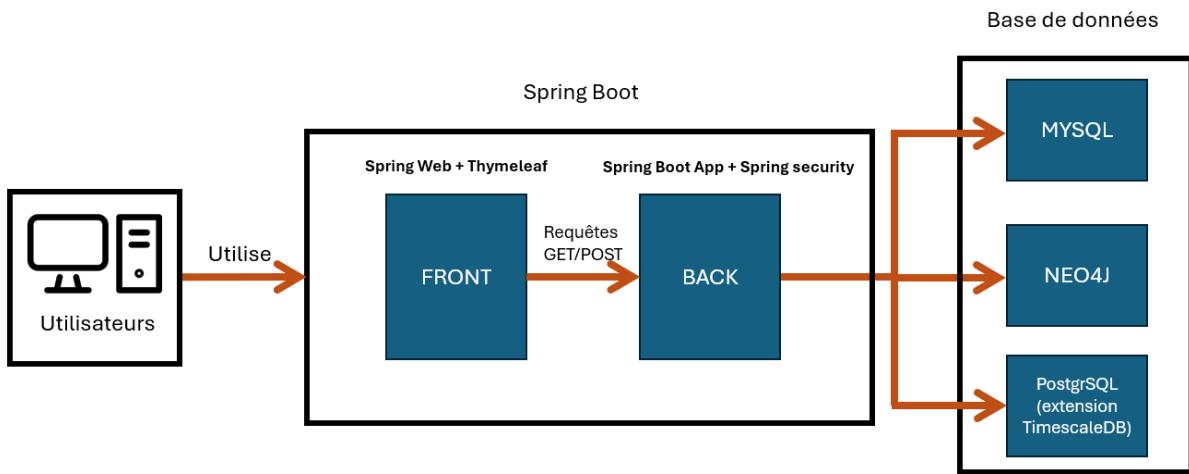


Figure 5 : Présente les interactions entre les technologies choisis

## IV. Fonctionnalités

Le tableau 1 décrit les fonctionnalités offertes par notre portail Web d'open data pour le premier semestre, un utilisateur non-admin représente les citoyens, entreprise, DSI, etc.

Catégories	Fonctionnalités	Description
Gestion des utilisateurs	Création de compte	L'utilisateur non connecté doit fournir des informations (adresse email et un mot de passe) pour créer un compte sur le portail.
	Connexion	L'utilisateur non connecté ou admin entre son email et son mot de passe pour se connecter.
	Modifications des informations du compte (mot de passe, adresse email)	L'utilisateur connecté ou admin peut modifier ses informations personnelles, comme son mot de passe ou son adresse email depuis les paramètres de son compte.
Gestion des données	Importation des données	L'utilisateur admin (agent de la métropole) peut téléverser (uploader) des fichiers de données vers le portail.
	Affichage des données	Une fois les données importées, l'utilisateur connecté, non connecté et admin peut les consulter directement sur le portail.
	Téléchargement des données	L'utilisateur non connecté, connecté et admin peut télécharger les données dans différents formats (CSV, PDF, Excel, etc.).
	Affichage / Choix politique de licence	L'utilisateur non connecté peut sélectionner ou consulter les licences (comme Creative Commons) sous lesquelles les données sont partagées ou accessibles.
	Recherche des données	L'utilisateur non connecté, connecté et admin peut rechercher des données spécifiques sur le portail en utilisant une barre de recherche et en saisissant le nom de la donnée.
	Filtrage des données	L'utilisateur non connecté, connecté et admin peut trier ou filtrer les données en fonction de critères comme le thème, la date, ou le type de données pour faciliter la recherche et l'affichage.

Tableau 1 : Fonctionnalités du portail

Ces fonctionnalités sont conçues pour atteindre nos objectifs en facilitant l'accès aux données publiques tout en respectant les obligations légales. Le diagramme ci-dessous montre les principaux cas d'utilisation du portail.

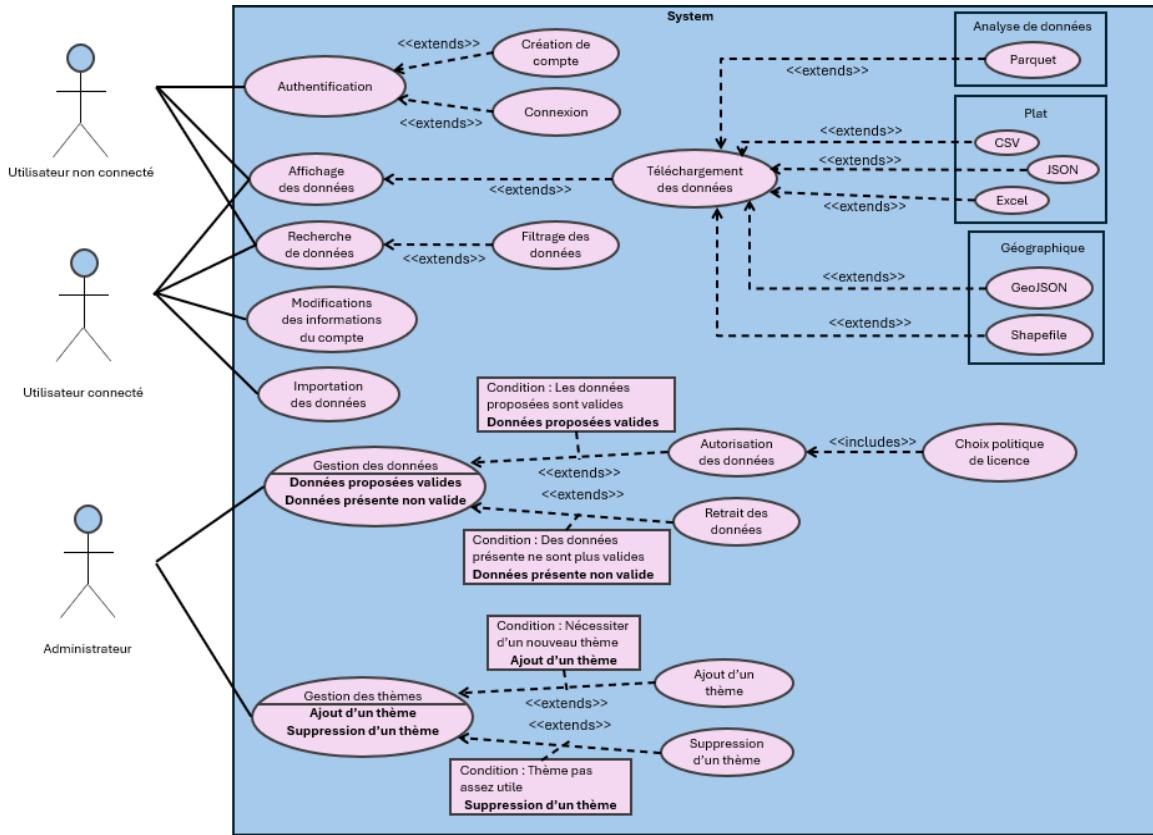


Figure 6 : Diagramme d'utilisation

Notre open data va compter 3 types d'utilisateur :

- **L'utilisateur non connecté** peut accéder aux données publiques et les télécharger.
- **L'utilisateur connecté** peut accéder aux données publiques, les télécharger et importer des données pour les partager.
- **L'administrateur** peut autoriser les données importées, associer la politique de licence, importer des nouvelles données et créer de nouvelles catégories.

## V. Dépendances

Pour que le développement du projet soit efficace, nous avons établi un tableau de dépendances pour l'ensemble de nos fonctionnalités. Cela nous permet d'avancer tous ensemble de manière continue et de s'assurer que notre prévision des tâches est cohérente.

Numéro	Fonctionnalités	Dépendance	Pourquoi ?
A	Création de compte	/	La création de compte ne dépend d'aucune autre fonctionnalité
B	Connexion	A	Pour pouvoir se connecter, il faut que le système de compte soit mis en place et fonctionnel
C	Modifications des informations	A	La modification des informations sur le compte nécessite d'abord la création de compte
D	Importation des données	B	L'import de données ne peut se faire que si l'utilisateur est connecté, et qu'il s'agit d'un administrateur
E	Affichage/ Visualisation des données	/ (nous partons du principe qu'il y a déjà des données dans la bdd)	L'affichage des données ne dépend d'aucune autre fonctionnalité
F	Téléchargement des données	E	Le téléchargement des données ne peut se faire que si l'utilisateur y a accès
G	Affichage / Choix politique de licence	D	Le choix de la licence des données est réalisé lors de l'importation des données
H	Recherche de données	E	La recherche des données à besoins de données pour être réalisé.
I	Filtrage des données	H	Le filtrage des données se fait en fonction des données qui peuvent être affichées à l'utilisateur et après la prise en compte de la recherche

Tableau 2 : Table des fonctionnalités et leurs dépendances

## VI. Fonctionnalités implémentées

La première étape après le démarrage de la phase de développement a été de découvrir Spring Boot ainsi que de rechercher différents styles graphiques d'open Data existant.

Après avoir exploré plusieurs options nous avons sélectionné trois styles graphiques comme sources d'inspiration :

- Open Data Londre [\(2\)](#) 
- Open Data Strasbourg [\(3\)](#) 
- Open Data Paris [\(4\)](#) 

Ces choix nous ont permis d'analyser et de comparer différentes approches graphiques pour concevoir notre propre style de site. Nous avons remarqué que les trois sites intègrent un logo, cela nous a inspiré à concevoir un logo pour notre projet car il s'agit d'un élément essentiel pour renforcer l'identité visuelle d'un site Open Data. Nous avons utilisé une IA qui génère des images afin d'avoir un logo, ensuite l'image a été amélioré par un membre de l'équipe.

Nous avons également choisi 2 thèmes (ou catégories) existant sur l'open data de Dijon [\(1\)](#) afin de pouvoir travailler et récupérer des données concrètes. Nous nous sommes restreints avec 2 thèmes (Démographie et Analyse du territoire, Déplacements et Mobilités) pour cette première phase du projet afin de nous concentrer sur les bases du projet. Ces 2 thèmes sont présents sur la page d'accueil de notre portail.

#### L'Annexe 4 : Page du site

### 1. Crédation de compte

L'objectif principal de cette fonctionnalité est de permettre aux utilisateurs de créer un compte dans l'application. Ce compte est stocké dans une base de données MySQL et la sécurité des données est réalisée à l'aide de Spring Security.

#### a. Descriptif fonctionnel

Un utilisateur non connecté peut créer un compte en cliquant sur le bouton "Création de compte". Une fois cela fait, il lui suffit ensuite de renseigner son nom, prénom, adresse mail et mot de passe et de cliquer sur "Créer". Le système vérifie que l'adresse email n'existe pas déjà dans la base de données, le mot de passe est ensuite encodé avant d'être enregistré pour garantir la confidentialité. Si les données sont correctes, un compte utilisateur est créé et stocké dans MySQL.

#### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application, **Spring Security** pour gérer l'authentification et la sécurité des comptes, **Thymeleaf** pour le rendu des vues côté serveur, **SweetAlert2** pour afficher des pop-ups interactifs et des messages de confirmation ou d'erreur à l'utilisateur et la bdd **MySQL** pour stocker les informations utilisateur.

Spring Boot est basé sur MVC, donc nous avons réalisé les différentes classes nécessaires :

- **UserController** : contrôleur principal gérant les requêtes d'inscription, de connexion, et de déconnexion :
  - `@GetMapping("/register")` : affiche la page de création de compte
  - `@PostMapping("/register")` : gérer la création d'un compte utilisateur.
- **UserServices** : Service entre le contrôleur et le repository, pour appliquer les traitements nécessaires avant d'enregistrer les données :
  - La méthode `registerUser(User user)` fait appel au repository pour sauvegarder l'utilisateur.
- **UserRepository** : Interface de type repository qui étend JpaRepository, permettant de communiquer avec la base de données MySQL.
- **User** : Entité représentant un utilisateur dans la base de données MySQL.

#### Gestion des erreurs de création de compte :

En cas d'erreur de saisie (email déjà utilisé, mots de passe non correspondants, ou erreur de connexion à la base de données), une pop-up SweetAlert2 indique à l'utilisateur l'erreur.

Concernant l'encodage du mot de passe Spring Security est configuré pour encoder les mots de passe avec la classe **BCryptPasswordEncoder** et restreindre l'accès aux pages en fonction des droits de l'utilisateur.

La figure 7 présente la partie utilisateur du MCD de la base de données.

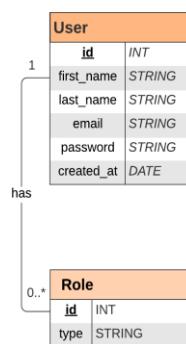


Figure 7 : Présente le MCD de la partie utilisateur

#### c. Organisation

La figure 8 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

## Tous les ouverts

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Création de la vue pour la page de création	Clôturé	14-10-2024	17-10-2024	4 j	Kevin Pradier
Créer la classe UserService	Clôturé	14-10-2024	14-10-2024	1 j	Anthony Michaud
Créer la classe UserController	Clôturé	14-10-2024	14-10-2024	1 j	Kevin Pradier
Créer la classe Model User	Clôturé	14-10-2024	14-10-2024	1 j	Anthony Michaud
Réalisation des tests unitaires	Clôturé	15-10-2024	16-10-2024	2 j	Anthony Michaud

Figure 8 : Tâches création de compte

[L'Annexe 5 : présente les diagrammes des classes et séquences partie utilisateur](#)

## 2. Connexion

La fonctionnalité de connexion permet aux utilisateurs d'accéder de manière sécurisée à l'application en saisissant leur email et mot de passe. Elle valide les informations fournies, encode le mot de passe pour protéger les données et crée une session sécurisée. En cas de succès l'utilisateur est redirigé vers l'accueil, sinon, un message d'erreur s'affiche et l'invite à se reconnecter.

### a. Descriptif fonctionnel

La fonctionnalité de connexion permet à un utilisateur de se connecter à l'application en saisissant son adresse email et son mot de passe. Voici les étapes du processus :

1. **Accès à la page de connexion** : L'utilisateur clique sur le bouton "Connexion", ce qui le redirige vers le formulaire de connexion.
2. **Remplissage du formulaire** : L'utilisateur saisit son adresse email et son mot de passe dans les champs appropriés du formulaire.
3. **Validation des informations** :
  - o Lorsque l'utilisateur clique sur le bouton "Se connecter", le système vérifie que l'adresse email correspond à un compte existant.
  - o Ensuite, le mot de passe est vérifié pour s'assurer qu'il correspond au mot de passe enregistré dans la base de données pour cet email.
4. **Connexion réussie** : Si l'email et le mot de passe sont corrects, l'utilisateur est authentifié et redirigé vers la page d'accueil ou vers la dernière page visitée avant la demande de connexion. Un message de bienvenue personnalisé peut être affiché à l'utilisateur.
5. **Gestion des erreurs** : Si l'email n'existe pas ou si le mot de passe est incorrect, un message d'erreur s'affiche pour indiquer à l'utilisateur que les informations saisies sont invalides. L'utilisateur reste sur la page de connexion et peut réessayer avec les bonnes informations.

## b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour l'architecture de l'application et la gestion des requêtes, **Spring Security** pour l'authentification des utilisateurs et la gestion de la session de sécurité, **Thymeleaf** pour afficher le formulaire de connexion et les messages de retour à l'utilisateur et **SweetAlert2** pour afficher des notifications ou messages d'erreur interactifs si la connexion échoue.

### Structure et composants de la fonctionnalité

#### Contrôleur UserController :

- Le contrôleur UserController expose deux endpoints pour la connexion :
  - **@GetMapping("/login")** : Affiche la page de connexion avec le formulaire.
  - **@PostMapping("/login")** : Gère l'envoi du formulaire et vérifie les informations de connexion.

#### Service UserServices :

- La méthode **getUserByEmail(String email, String password)** vérifie l'existence de l'utilisateur et valide le mot de passe.
- La méthode **getUserByEmail** utilise **PasswordEncoder.matches()** pour comparer le mot de passe saisi par l'utilisateur avec le mot de passe encodé stocké dans la base de données.

#### Gestion de la Session Utilisateur :

- La méthode **connectUser(User user, HttpServletRequest req)** dans UserController initialise une session utilisateur sécurisée.

Elle utilise **UsernamePasswordAuthenticationToken** pour créer un jeton d'authentification avec les informations de l'utilisateur. **AuthenticationManager** utilise et authentifie le jeton et **SecurityContextHolder** sauvegarde l'authentification dans le contexte de sécurité. Une session est ensuite créée pour l'utilisateur avec un délai d'expiration configurable (nous avons choisi 300 secondes d'inactivité).

#### Gestion des erreurs de connexion :

Si les informations sont invalides, le contrôleur renvoie l'utilisateur vers la page de connexion avec un paramètre d'erreur (redirect:/login?error=invalid) et l'invite à se reconnecter. SweetAlert2 peut afficher des messages personnalisés pour informer l'utilisateur de l'erreur.

### Sécurité et cryptage des mots de passe :

Spring Security et BCryptPasswordEncoder assurent la sécurité des mots de passe lors de la comparaison dans UserServices. Le mot de passe de l'utilisateur n'est jamais stocké en texte clair mais est toujours encodé, pour des raisons de sécurité évidentes.

#### c. Organisation

La figure 9 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

 OpenProject

##### Lots de Travaux

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	ASSIGNÉ À	DURÉE
Ajout de la méthode dans la classe UserController	Clôturé	15-10-2024	18-10-2024	Rafik Rhmaoui	4 j
Ajout de la méthode dans la classe UserService	Clôturé	15-10-2024	18-10-2024	Ahmed-El-Aziz Kaidi	4 j
Création de la vue pour la page connexion	Clôturé	15-10-2024	18-10-2024	Wassim Ennaji	4 j

Figure 9 : Tâches connexion

L'Annexe 5 : présente les diagrammes des classes et séquences partie utilisateur

## 3. Modification des informations du compte

#### a. Descriptif fonctionnel

La fonctionnalité de modification des informations du compte permet aux utilisateurs de mettre à jour leur adresse email liée au compte ou leur mot de passe. Lorsqu'un utilisateur est connecté et clique sur "Mon Profil", son nom, prénom et adresse email sont affichées. Ces données ne peuvent pas être modifiées directement dans les champs affichés. Lors de la modification de l'adresse email, celle-ci doit respecter un format valide (présence d'un @ et d'un nom de domaine) et l'utilisateur doit s'assurer que l'adresse e-mail saisie n'existe pas déjà dans la base de données. Si l'adresse email est invalide ou déjà utilisée une pop-up d'erreur s'affiche pour avertir l'utilisateur. Pour valider le changement d'adresse email l'utilisateur doit obligatoirement entrer son mot de passe actuel.

En ce qui concerne la modification du mot de passe celle-ci se déroule à droite de la section de modification de l'email. L'utilisateur doit saisir son ancien mot de passe, puis entrer un nouveau mot de passe qu'il devra confirmer. Le système vérifie que l'ancien mot de passe est correct, que le nouveau mot de passe est différent de l'ancien et que le champ de confirmation du nouveau mot de passe correspond. Si l'une de ces conditions n'est pas remplie des pop-ups d'erreur s'affichent pour signaler le problème. L'utilisateur devra ensuite cliquer sur le bouton "Modifier" pour valider la modification de son mot de passe.

#### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application, **Spring Security** pour gérer l'authentification et la sécurité des comptes, **Thymeleaf** pour le rendu des vues côté serveur, **SweetAlert2** pour afficher des pop-ups interactifs et des messages de

confirmation ou d'erreur à l'utilisateur et la bdd **MySQL** pour stocker les informations utilisateur.

### Structure et composants de la fonctionnalité

#### 1. Contrôleur UserController :

- o Le contrôleur UserController expose deux endpoints pour la modification des informations du compte :
  - **@PostMapping("/account/update\_email")** : Gère la mise à jour de l'adresse email de l'utilisateur. Le mot de passe actuel est vérifié et le format de l'adresse email doit être correct. Suivant le succès ou l'échec de la modification l'utilisateur est redirigé vers la page de profil avec un message de succès ou d'erreur.
  - **@PostMapping("/account/update\_password")** : Gère la mise à jour du mot de passe de l'utilisateur en vérifiant que le nouveau mot de passe soit différent de l'ancien et vérifie les informations de connexion avant de procéder à la mise à jour du mot de passe, redirigeant l'utilisateur avec un message de succès ou d'erreur en fonction du résultat.

#### 2. Service UserServices :

- o La méthode **updateEmail(String currentEmail, String newEmail, String currentPassword)** :  
**Cette méthode permet de mettre à jour l'adresse e-mail d'un utilisateur.**
- o La méthode **updatePassword(String email, String oldPassword, String newPassword)** : Cette méthode permet de changer le mot de passe d'un utilisateur.

#### 3. Gestion des erreurs de connexion :

Si les informations sont invalides le contrôleur renvoie l'utilisateur vers la page de connexion avec un paramètre d'erreur (**redirect:/login?error=invalid**). SweetAlert2 affiche des messages personnalisés pour informer l'utilisateur de l'erreur.

#### 4. Sécurité et cryptage des mots de passe :

Spring Security et BCryptPasswordEncoder assurent la sécurité des mots de passe lors de la comparaison dans UserServices. Le mot de passe de l'utilisateur n'est jamais stocké en texte clair mais est toujours encodé, ce qui protège contre les attaques par vol de données en cas de compromission.

#### c. Organisation

La figure 10 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

#### Lots de Travaux

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Modifications des informations du compte	Clôturé	16-10-2024	23-10-2024	6 j	Aya El-Hassani
Ajout de la méthode dans la classe UserController	Clôturé	16-10-2024	23-10-2024	6 j	Aya El-Hassani
Ajout de la méthode dans la classe UserService	Clôturé	16-10-2024	23-10-2024	6 j	Imane Choukri
Création de la vue de la page	Clôturé	16-10-2024	23-10-2024	6 j	Alan Guilleminot
Ajout d'une notification pour informer l'utilisateur du succès / échec	Clôturé	16-10-2024	23-10-2024	6 j	Imane Choukri
Réalisation des tests unitaires	Clôturé	17-10-2024	23-10-2024	5 j	Alan Guilleminot

Figure 10 : Tâches modification des informations du compte

[L'Annexe 5 : présente les diagrammes des classes et séquences partie utilisateur](#)

## 4. Affichage des données

### a. Descriptif fonctionnel

L'affichage des données permet aux utilisateurs de visualiser les données stockées dans la base de données de manière claire et structurée. Lorsqu'un utilisateur (connecté ou non) accède à la page des données, une liste des éléments lui est présentée avec des informations clés telles que le nom, le thème, la description, la licence, la langue et les fichiers (PDF, CSV, XML). Chaque donnée est présentée sous forme d'un élément, un bloc affichant son titre et sa description. Afin de gérer le cas où le nombre de données est important (+100) un système de pagination est mis en place. Cela permet d'afficher un nombre limité d'éléments par page, ce qui améliore la lisibilité et la performance. La pagination est dynamique et s'ajuste automatiquement en fonction du nombre d'éléments disponibles et des filtres appliqués par l'utilisateur. Ce système de pagination aide à maintenir une navigation fluide et à éviter une surcharge visuelle. En regardant le détail d'une donnée l'utilisateur peut visualiser en direct les méta-données sous forme d'un tableau et le pdf.

Des boutons de téléchargement sont présents pour chaque type de fichier, permettant de récupérer les fichiers dans le format souhaité. En outre, la présentation de chaque fichier peut inclure une petite description ou une icône pour identifier son contenu. L'objectif principal de cette fonctionnalité est d'offrir une interface claire, fonctionnelle et performante, permettant à l'utilisateur de naviguer, trier, filtrer et télécharger les données sans difficulté, tout en maintenant une charge minimale sur le serveur.

### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application backend, **Thymeleaf** pour le rendu des pages côté serveur permettant d'afficher dynamiquement les données sur l'interface utilisateur, **JavaScript** pour améliorer l'interactivité de l'application en gérant la pagination côté client et la bdd **MySQL** avec **Spring Data JPA** pour gérer les données, y compris les relations entre entités et le stockage des fichiers volumineux (PDF, CSV, XML) sous forme de blobs.

Nous avons choisi les blobs pour le stockage des fichiers car ils permettent de stocker directement les fichiers dans la base de données sans avoir besoin de système de fichiers externe, cela simplifie la gestion des données car tout est centralisé dans une seule base. Les fichiers étant intégrés à la base ils bénéficient des mécanismes de sécurité de MySQL. Les

blobs permettent de gérer différents formats de fichiers (PDF, CSV, XML, etc) ce qui dans notre cas est très important car nous stockons plusieurs types de fichier.

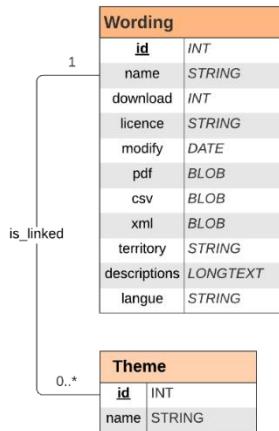


Figure 11 : MCD partie données

## Structure et composants de la fonctionnalité

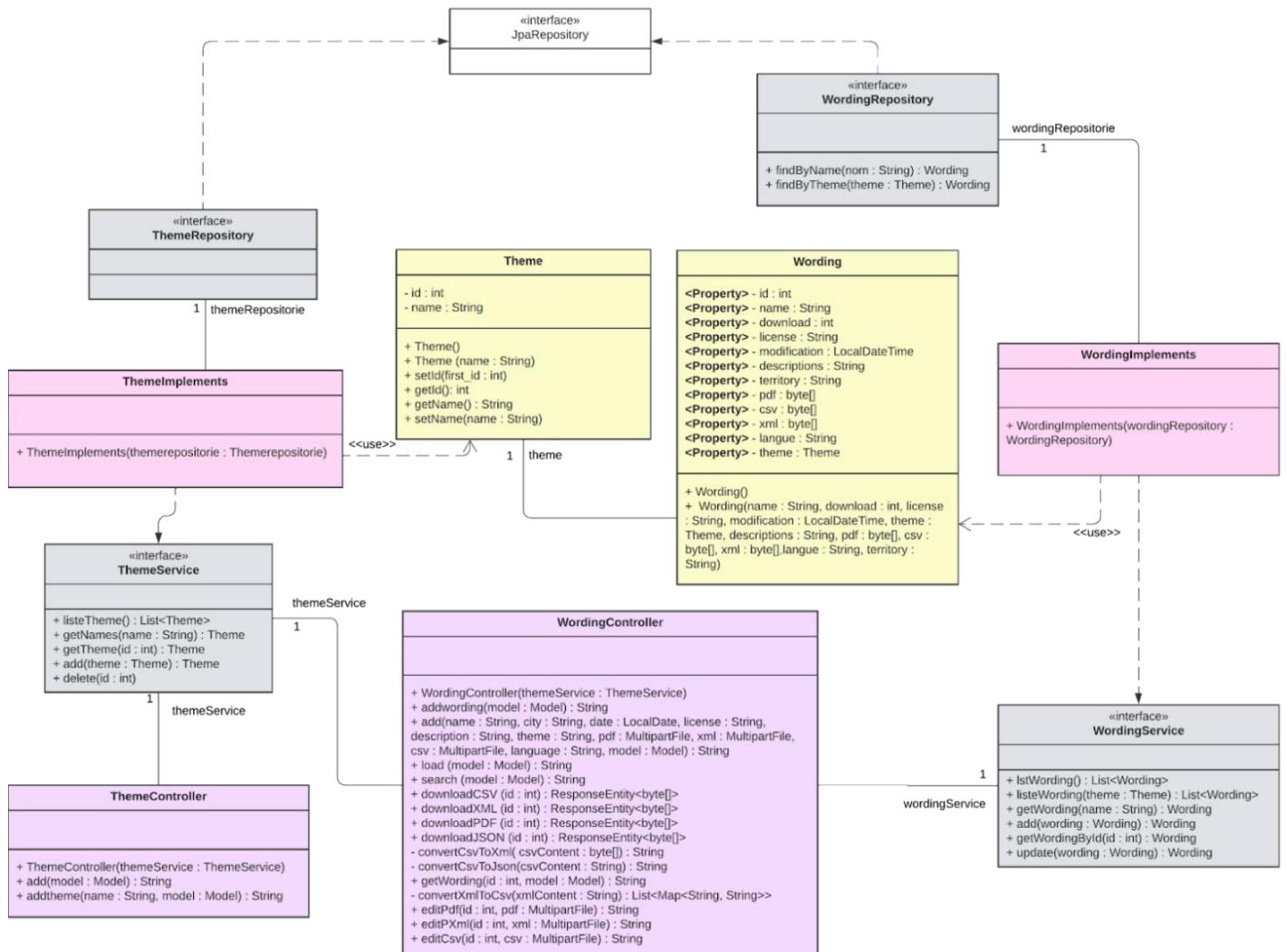


Figure 22 : présente le diagramme de classe de la partie donnée

La structure de notre projet repose sur une architecture bien définie basée sur le principe de séparation des responsabilités afin d'avoir une organisation claire et évolutive. Nous utilisons des interfaces (comme ThemeService, WordingService, ThemeRepository, etc.) afin de faciliter le remplacement ou l'ajout de nouvelles implémentations sans perturber les autres composants. Les classes Theme et Wording représentent les entités de la base de données assurant une correspondance avec les données persistantes. Les contrôleurs (ThemeController, WordingController) servent d'interfaces entre les utilisateurs et les services et les classes implémentant les services (ThemeImplements, WordingImplements) contiennent la logique métier afin de séparer la logique métier du reste du projet. Les repositories JPA (ThemeRepository, WordingRepository) permettent une interaction directe avec la base de données via Spring Data JPA grâce aux opérations CRUD.

### **1. Contrôleur WordingController :**

Le contrôleur WordingController expose deux endpoints pour l'affichage de toutes les données et l'affichage des détails d'une donnée :

**@GetMapping("/search")** : gère l'affichage de toutes les données sous forme de bloc.

**@GetMapping("/wording/{id}")** : gère l'affichage des informations d'une donnée.

### **2. Service WordingService :**

- La fonction **IstWording()** est utilisée afin de récupérer toutes les données disponibles dans la base de données.
- La fonction **getWordingById(int id)** permet de récupérer les informations détaillées d'un jeu de données en fonction de son identifiant unique (ID).

### **c. Organisation**

La figure 12 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Créer la vue pour l'affichage des données sur le portail	Clôturé	15-10-2024	25-10-2024	9 j	Aichetou N-diaye
Créer la méthode dans la classe service pour récupérer les données	Clôturé	15-10-2024	25-10-2024	9 j	Trinite Mombouli
Créer la méthode dans la classe contrôleur pour afficher les données	Clôturé	15-10-2024	25-10-2024	9 j	Trinite Mombouli
Réaliser les tests unitaires	Clôturé	17-10-2024	25-10-2024	7 j	Trinite Mombouli
Implémenter la pagination pour la visualisation	Clôturé	21-10-2024	25-10-2024	5 j	Aichetou N-diaye

Figure 12 : Tâches affichage des données

L'Annexe 5 : présente les diagrammes des classes et séquences partie utilisateur

## 5. Recherche des données

### a. Descriptif fonctionnel

Un utilisateur connecté ou non, peut rechercher des données en fonction de leur titre à l'aide d'une barre de recherche, cela permet d'affiner ses résultats. La recherche est gérée côté client en JavaScript, pour offrir une expérience utilisateur fluide et réactive. Dès que l'utilisateur tape dans la barre de recherche et valide, un script JavaScript filtre dynamiquement les données déjà chargées dans le navigateur sans interroger la base de données, cela permet de réduire la charge sur le serveur et améliore les temps de réponse. Cette option peut également avoir des défauts s'il y a beaucoup de données à charger en amont, mais après recherche sur les différents open data existant le nombre de données varient entre 20 -100 c'est pour cette raison que la solution dynamique a été choisie.

Pour éviter de surcharger le visuel de l'utilisateur, un système de pagination est intégré à la page des données. La pagination permet d'afficher les données par nombre de blocs limité, facilitant la lecture et la navigation, en particulier lorsque le nombre de résultats est important. Cette pagination est dynamique, s'adaptant en temps réel aux filtres de recherche appliqués par l'utilisateur.

### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application, **Thymeleaf** pour le rendu des vues côté serveur et **Javascript** qui gère la logique de recherche dynamique et la pagination côté client pour améliorer l'interactivité et la réactivité de l'application.

La figure 13 présente le diagramme de séquence lorsqu'un utilisateur réalise une recherche sur le portail.

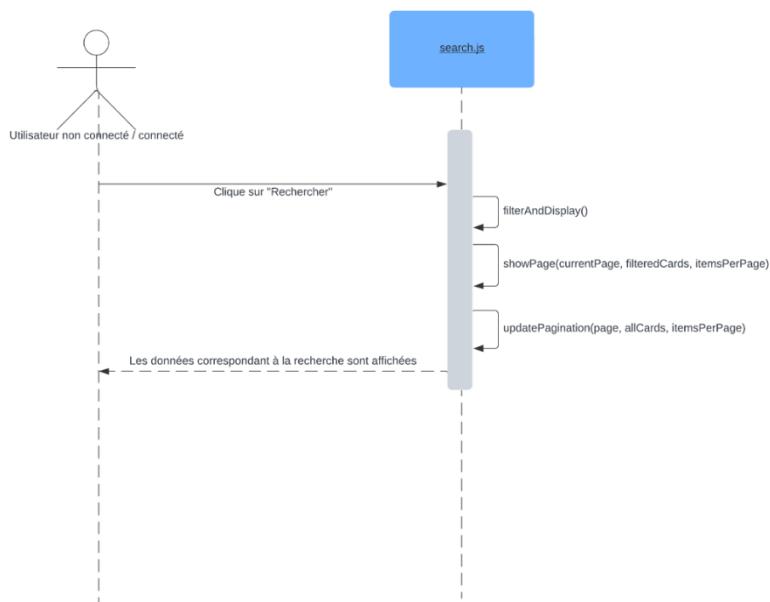


Figure 13 : Diagramme séquence recherche

### c. Organisation

La figure 14 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Création fonction js pour gérer l'affichage en fonction de la barre de recherche	Clôturé	04-11-2024	07-11-2024	4 j	Anthony Michaud
Modification de la vue pour l'ajout de la barre de recherche	Clôturé	07-11-2024	08-11-2024	2 j	Anthony Michaud
Vérification et prise en compte des injections SQL	Clôturé	08-11-2024	08-11-2024	1 j	Anthony Michaud

Figure 14 : Tâche recherche des données

## 6. Filtrage des données

### a. Descriptif fonctionnel

Un utilisateur connecté ou non peut filtrer des données en fonction du nombre de téléchargement, ce qui permet de connaître les plus populaires. On peut filtrer en fonction de la licence associée, du thème et du territoire. On peut également filtrer par rapport à la date de modification ce qui permet de récupérer les données les plus récentes. On peut aussi filtrer par langue ou par possibilités d'export (CSV, XML, PDF...). Les filtres sont cumulatifs, c'est-à-dire que les données affichées devront satisfaire tous les filtres renseignés. Le filtrage se fait de manière dynamique, donc les filtres sont appliqués sans recharger la page.

### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application, **Thymeleaf** pour le rendu des vues côté serveur et **Javascript** qui gère la logique de recherche dynamique et la pagination côté client pour améliorer l'interactivité et la réactivité de l'application.

La figure 15 présente le diagramme de séquence lorsqu'un utilisateur réalise un filtre sur les données du portail.

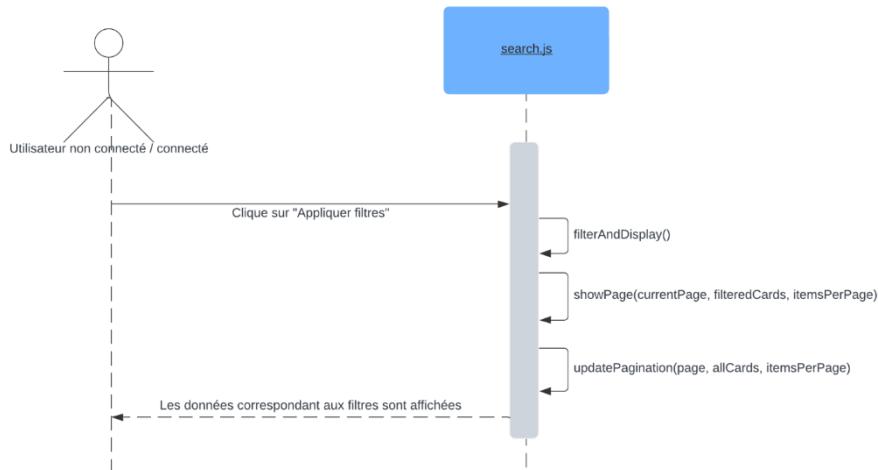


Figure 15 : Diagramme séquence filtre

### c. Organisation

La figure 16 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Création fonction js pour gérer l'affichage en fonction des filtres	Clôturé	14-11-2024	15-11-2024	2 j	Kevin Pradier
Modification de l'affichage pour ajouter les filtres	Clôturé	15-11-2024	18-11-2024	2 j	Kevin Pradier
Vérification prise en compte injection SQL	Clôturé	18-11-2024	19-11-2024	2 j	Kevin Pradier

Figure 16 : Tâche filtre des données

## 7. Téléchargement des données

### a. Descriptif fonctionnel

La fonctionnalité de téléchargement des données permet aux utilisateurs de récupérer les fichiers associés aux jeux de données stockés dans la base de données. Chaque entrée de la liste des données contient des boutons de téléchargement pour différents types de fichier (CSV, XML, JSON, PDF). Lorsqu'un utilisateur clique sur un bouton de téléchargement, le fichier est automatiquement téléchargé dans le format choisi.

### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour gérer les requêtes HTTP côté serveur, **Spring Data JPA** pour les fichiers stockés dans la base de données sous forme de blobs **Thymeleaf** pour le rendu des vues côté serveur, la bdd **MySQL** pour les fichiers pdf, csv et xml qui sont stockés dans la table Wording. Le format JSON est automatiquement convertie depuis l'un des autres formats. Les autorisations d'accès aux fichiers sont vérifiées avec **Spring Security**, empêchant le téléchargement non autorisé.

### Structure et composants de la fonctionnalité

#### Contrôleur WordingController:

- **@PostMapping("/wording/download/csv")** gère le téléchargement du fichier csv pour une donnée.
- **@PostMapping("/wording/download/xml")** gère le téléchargement du fichier xml pour une donnée.
- **@PostMapping("/wording/download/pdf")** gère le téléchargement du fichier pdf pour une donnée.
- **@GetMapping("/wording/download/json/{id}")** gère le téléchargement des données au format JSON

### Service UserServices :

- La fonction **convertCsvToJson(String csvContent)** permet de convertir les données sous format CSV au format JSON
- La méthode **update(Wording wording)** : Cette méthode permet de mettre à jour une données, elle est utilisée pour mettre à jour les fichiers ou augmenté de 1 le nombre de téléchargement

#### c. Organisation

La figure 17 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Créer la méthode dans la classe service	Clôturé	21-10-2024	08-11-2024	15 j	Trinite Mombouli
Création de la méthode dans la classe contrôleur	Clôturé	21-10-2024	08-11-2024	15 j	Trinite Mombouli
Créer une vue avec de option de téléchargement pour chaque format	Clôturé	22-10-2024	08-11-2024	14 j	Trinite Mombouli
Notification (Pop-up) succès ou échec téléchargement	Clôturé	23-10-2024	08-11-2024	13 j	Trinite Mombouli
Gestion des droits d'accès selon les licences	Clôturé	24-10-2024	08-11-2024	12 j	Trinite Mombouli
Réaliser les tests unitaires	Clôturé	25-10-2024	08-11-2024	11 j	Trinite Mombouli

Figure 17 : Tâches télécharger des données

[L'Annexe 6 : présente les diagramme classes données et diagrammes séquences](#)

## 8. Importation des données

#### a. Descriptif fonctionnel

La fonctionnalité d'importation des données permet à l'administrateur d'importer de nouvelles données dans la base de données. L'administrateur indique le titre, le thème, la langue, la description, la date, la ville et fournit les fichiers. L'administrateur peut également ajouter une nouvelle catégorie de données et modifier les fichiers (pdf,csv,xml) d'une données existante.

#### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour gérer les requêtes HTTP côté serveur, **Spring Data JPA** pour les fichiers qui sont stockés dans la base de données sous forme de blobs, **Thymeleaf** pour le rendu des vues côté serveur, la bdd **MySQL** pour les fichiers pdf, csv et xml qui sont stockés dans la table Wording. Concernant la sécurité seulement un compte administrateur peut importer des données.

**Contrôleurs ThemeController et WordingController:**

- **@GetMapping("/add")** gère l'affichage de la page d'ajout d'un thème
- **@PostMapping("/add")** gère l'ajout du nouveau thème dans la bdd
- **@GetMapping("/wording")** gère l'affichage de la page d'ajout d'une donnée

- **@PostMapping("/wording")** : gère l'ajout d'une nouvelle donnée dans la bdd

### Service ThemeService et WordingService :

- La fonction **add(Theme theme)** permet d'ajouter dans la bdd un nouveau thème
- La fonction **add(Wording wording)** permet d'ajouter dans la bdd une nouvelle donnée

#### c. Organisation

La figure 18 présente une copie d'écran d'open project détaillant les tâches de cette fonctionnalité.

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Création de la classe DataController	Clôturé	21-11-2024	22-11-2024	2 j	Trinite Mombouli
Créer la classe DataService	Clôturé	25-11-2024	25-11-2024	1 j	Aichetou N-diaye
Création de la vue pour permettre à l'administrateur de téléverser les données	Clôturé	26-11-2024	27-11-2024	2 j	Trinite Mombouli

Figure 18 : Tâches importation des données

### L'Annexe 6 : présente les diagramme classes données et diagrammes séquences

## 9. Choix de la licence

#### a. Descriptif fonctionnel

Un utilisateur connecté ou non peut voir quelle licence est liée à une donnée ainsi que le détail des droits cédés par cette licence, ce qui permet aux utilisateurs de savoir les limites de l'utilisation d'une donnée. On peut voir ces informations en affichant une donnée en détail et en cliquant sur le "?", un lien est également disponible menant vers une page explicative dédiée à la licence. Cette page fournit des informations complètes sur les termes et conditions pour la licence, les droits d'utilisation, de modification, de redistribution et les restrictions. Cela permet de garantir une transparence totale pour les utilisateurs.

#### b. Descriptif technique

Les technologies utilisées sont **Spring Boot** pour la structure de l'application, **Thymeleaf** pour le rendu des vues côté serveur et **Javascript** qui gère la logique d'affichage des informations pour la licence. Les détails d'une licence sont dans un fichier JSON qui est récupéré et utilisé lors de l'affichage d'une page de détail d'une donnée.

#### c. Organisation

SUJET	STATUT	DATE DE DÉBUT	DATE DE FIN	DURÉE	ASSIGNÉ À
Récupérer des informations sur les licences	Clôturé	27-11-2024	28-11-2024	2 j	Rafik Rhamaoui
Ajouter la prise en compte des licences dans l'affichage	Clôturé	28-11-2024	29-11-2024	2 j	Rafik Rhamaoui
Stockage dans un fichier json du détail d'une licence	Clôturé	28-11-2024	29-11-2024	2 j	Wassim Ennaji

Figure 19 : Tâches choix de la licence

## VII. Normes et tests unitaires

Au début du projet, nous nous sommes mis d'accord sur les normes de codage à respecter. Étant donné que nous développons en Java, nous avons décidé de suivre les conventions standard de codage Java.

- Noms méthodes, fonctions, attributs en anglais avec la première lettre du premier mot en minuscule ensuite en majuscule.
- Noms des classes en anglais et première lettre en majuscule
- Commentaires de commit, documentation et commentaire de code en français

Pour les tests unitaires, un test est réalisé pour chaque méthode et fonction cela s'applique aux couches controller, service et model. En parallèle, des tests d'intégration sont également réalisés contrairement aux tests unitaires qui sont réalisé sur de fausse données, les tests d'intégration utilisent des données réelles pour vérifier le bon fonctionnement global du système. Les référents qualité s'assurent que la couverture des tests unitaires est complète. Nous avons également mis en place un *runner GitLab\** afin d'automatiser la validation des tests unitaires à chaque nouveau commit sur la branche principale, cela permet d'assurer une qualité constante du code et de détecter rapidement les régressions. Ce pipeline CI/CD (Intégration Continue et Déploiement Continu) est configuré pour exécuter automatiquement les tests unitaires.

La figure suivante est une capture d'écran présentant les tests unitaires exécutés ainsi que le niveau de couverture du code.

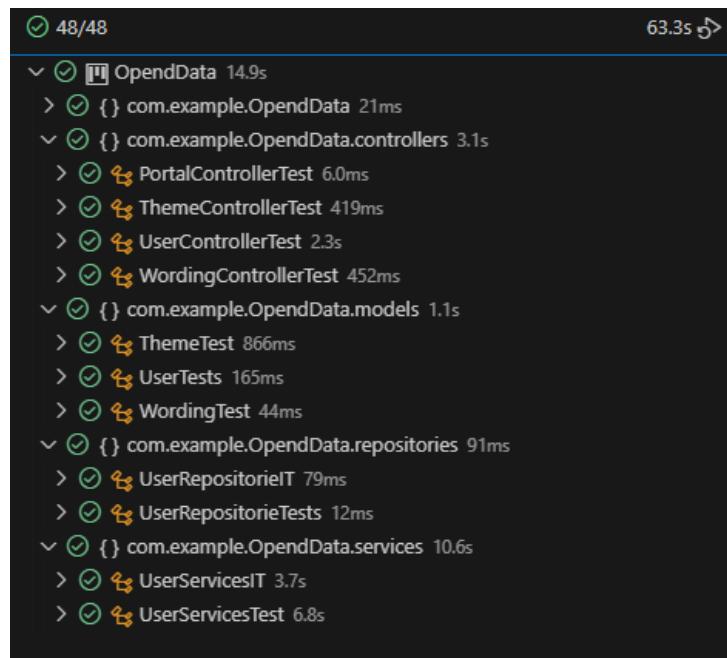


Figure 20 : Tests unitaires validé

Pour la validation de chaque fonctionnalité, plusieurs étapes sont suivies :

1. Les référents développement vérifient que les normes de codage sont respectées.

\*runner Gitlab : opération qui consiste à exécuter des tâches automatisées comme des tests, des builds ou des déploiements à l'aide d'un pipeline CI/CD. Cette opération permet de vérifier l'intégrité et la qualité du code.

2. Les référents qualité s'assurent que tous les tests unitaires passent et que la couverture est totale.
3. Pour terminer, le chef de projet valide la fonctionnalité puis intègre la branche de la fonctionnalité dans la branche principale (main).

[Annexe 7 : Classe de tests unitaires](#)

## VIII. Organisation du groupe

Fonctionnalités	Rôle principal	Rôles secondaires	Explications du rôle principal
Création de compte	Référent développement	Référent qualité	Toutes les parties liées à la création, modification de compte et connexion seront gérées par le référent développement, qui assistera les développeurs pour un avancement conjoint.
Connexion	Référent développement	Référent données	
Modifications des informations	Référent développement	Référent données / Référent qualité	
Importation des données	Référent données	Référent qualité	Le référent des données, responsable de la cohérence, s'assurera qu'il n'y a pas d'incohérence avec les bases de données.
Téléchargement des données	Référent données	Référent qualité	
Affichage des données	Référent qualité	Référent données	Le référent qualité sera principalement en charge de vérifier que l'affichage des données se fait correctement et que celles-ci soient cohérentes.
Affichage / Choix politique de licence	Référent qualité	Référent développement	Le référent qualité en lien avec le référent développement devront s'assurer que les données soient bien en correspondance avec leur licence attribuée.
Recherche de données	Référent développement	Référent Données / Réf qualité	Le référent développement devra s'assurer que le moyen de recherche et de filtrage soit, au moyen de requêtes, efficaces.
Filtrage des données	Référent développement	Référent données	

*Tableau 3 : Table des fonctionnalités et leurs rôles*

Voici un aperçu des tâches de fonctionnalités que nous avons ajoutées sur OpenProject. Nous avons réparti les tâches de manière logique entre les alternants et les non-alternants, tout en estimant le temps nécessaire pour chacune, afin de respecter les délais du projet.

Certaines fonctionnalités possèdent beaucoup de membres car ce sont des alternants. Les alternants possèdent moins de temps de travail sur le projet donc nous les avons regroupés pour qu'il soit plusieurs et avons également ajouté un membre en formation initiale pour les aider. Chaque membre ne travaillera pas sur toutes les fonctionnalités, mais en cas de problème des membres ayant fini leurs tâches ou n'en possédant plus pourront aller aider les autres.

L'image 21 présente le tableau des répartitions des fonctionnalités entre les membres de l'équipe.

Fonctionnalités	Membres assignés
Création de compte	Anthony MICHAUD, Kévin PRADIER
Connexion	Rafik RHARMAOUI, Ahmed KAIDI, Wassim ENNAJI
Modifications des informations	Aya ELHASSANI, Imane CHOUKRI, Alan GUILLELINOT
Importer des données	Aichetou N'DIAYA, Trinité MOMBOULI
Affichage des données	Trinité MOMBOULI
Télécharger les données	Trinité MOMBOULI
Affichage / Choix politique de licence	Rafik RHARMAOUI, Ahmed KAIDI, Wassim ENNAJI
Filtrage des données	Kévin PRADIER
Recherche de données	Anthony MICHAUD

Figure 21 : Tableau de répartition des fonctionnalités par membre

#### Annexe 1, 2 et 3 : Gantt + Détails des tâches + Organisation des réunions

Chaque fonctionnalité a été réalisée en suivant la même procédure. D'abord la réalisation d'une maquette fil de fer que ce soit pour l'ajout ou la modification d'éléments sur l'interface. Une fois la maquette validée, le développement de la fonctionnalité est réalisé en suivant les normes de codage définies en début de projet. L'implémentation respecte l'architecture du projet notamment le modèle MVC (Modèle-Vue-Contrôleur). Après le développement, des tests unitaires sont réalisés. Il n'y a pas de différence entre le planning initial et final, nous avons réussi à réaliser toutes les tâches. Grâce à une bonne organisation nous n'avons pas pris de retard sur notre travail.

## IX. Conclusion

Ce projet nous a permis de découvrir et de travailler dans toutes les étapes du développement d'une application Web, de la conception à l'implémentation. Nous avons amélioré nos compétences techniques, notamment en Java, HTML, CSS, et JavaScript, tout en explorant des outils et des frameworks comme Spring Boot et Thymeleaf. Ce projet nous a également permis d'améliorer nos méthodes de travail en équipe. Nous avons appris à mieux

planifier nos tâches, à respecter les délais, et à collaborer de manière plus efficace en tirant parti des retours et des validations de nos référents et du chef de projet.

La prise en main de Spring Boot qui était un framework nouveau pour la majorité de l'équipe a été un défi nécessitant une autoformation et une période d'adaptation. Travailler à dix personnes sur un projet a également complexifié la coordination et ralenti certaines décisions, cependant cela a renforcé nos compétences collaboratives. Les outils mis à disposition nous ont permis de nous organiser mais certains outils étaient limités, comme open project qui ne nous permet pas de créer des groupes pour les tâches (du moins facilement), mattermost ne possède pas de channel vocal et l'utilisation du VPN pour l'accès aux outils est contraignant.

Concernant l'organisation les différentes tâches ont toutes été réalisé dans les temps. La fonctionnalité Affichage des données qui est un élément important du projet a été réalisé plus vite que prévu ce qui nous a permis de prendre de l'avance ensuite sur le développement des autres fonctionnalités.

Lors de la deuxième partie du projet au semestre 2, nous nous concentrerons sur la partie statistique du projet.

## X. Liens

- Lien du rapport sur google drive :

<https://docs.google.com/document/d/1brRBeXUFYW26YZNqscM9c04Kw5B8FnxG/edit?usp=sharing&ouid=114233599138467094838&rtpof=true&sd=true>

- Lien du wiki GitLab compte rendu des réunions :

<https://gitlabvigan.iem/groupe6/opendatametropole/-/wikis/Comptes-Rendus-de-R%C3%A9union>

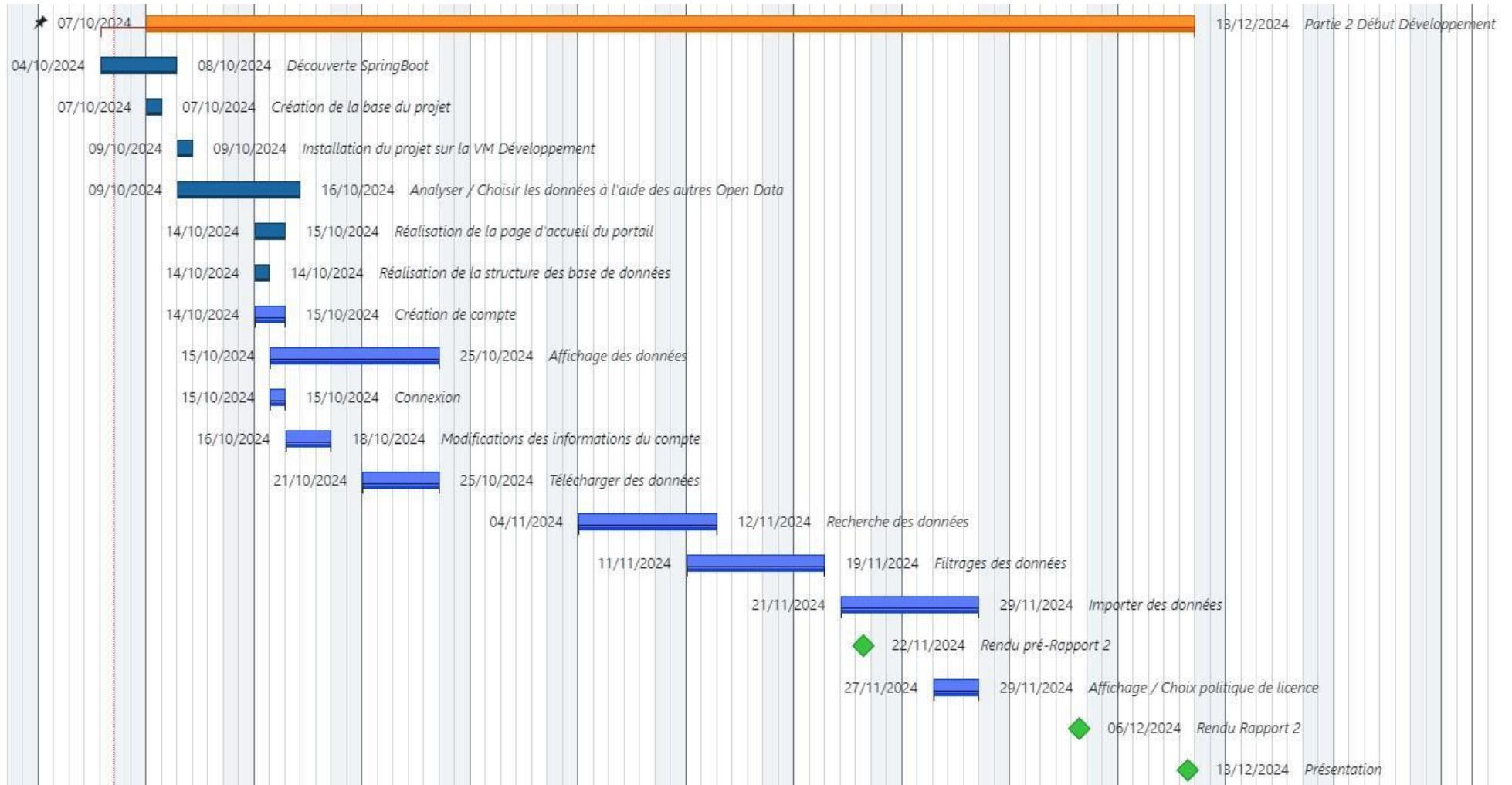
-Lien vidéo de présentation du projet : <https://youtu.be/yxjQ14NPUF8>

## XI. Bibliographie

- OpenData Dijon. (2024). *Open dijon, la métropole ouverte et innovante*. Consulté le 30/10/2024, sur <https://dijon-metropole.opendatasoft.com>
- OpenData Londre. (2024). *City of London Open Data*. Consulté le 30/10/2024, sur <https://opendata.london.ca/>
- OpenData Starsbourg. (2019). *Open Data Strasbourg.eu*. Consulté le 30/10/2024, sur <https://data.strasbourg.eu/pages/accueil/>
- OpenData Paris. (2024). *Paris Data met à disposition les données produites et collectées par la Ville de Paris*. Consulté le 30/10/2024, sur <https://opendata.paris.fr/pages/home/>

## XII. Annexes

### Annexe 1 : Gantt d'origine du projet



Annexe 2 : Détails des tâches du projet

### Lots de Travaux

SUJET	STATUT	RESPONSABLE	ASSIGNÉ À	PRIORITÉ	DATE DE DÉBUT	DATE DE FIN	DURÉE	TRAVAIL
Rendu pré-Rapport 2	Clôturé	Anthony Michaud	Grp6-Développeur	Haute	23-11-2024	23-11-2024	1 j	
Partie 1 Analyse	Clôturé	Grp6-Développeur	Grp6-Développeur	Normal	17-09-2024	05-10-2024	19 j	· Σ 20h
Définir les différentes fonctionnalités	Clôturé	Anthony Michaud	Grp6-Développeur	Normal	17-09-2024	24-09-2024	6 j	6h
Rendu Rapport 1	Clôturé	Anthony Michaud	Grp6-Développeur	Haute	05-10-2024	05-10-2024	1 j	10h
Définir les dépendances des fonctionnalités	Clôturé	Anthony Michaud	Grp6-Développeur	Normal	24-09-2024	26-09-2024	3 j	2h
Planning prévisionnel des fonctionnalités	Clôturé	Anthony Michaud	Grp6-Développeur	Normal	27-09-2024	03-10-2024	5 j	2h
Découverte SpringBoot	Clôturé	Grp6-Développeur	Grp6-Développeur	Normal	04-10-2024	08-10-2024	3 j	10h
Partie 0 Découverte Projet	Clôturé	Grp6-Développeur	Grp6-Développeur	Normal	10-09-2024	11-09-2024	2 j	4h · Σ 8h
Analyse des différents sujets / Liste de préférence	Clôturé	Grp6-Développeur	Grp6-Développeur	Normal	10-09-2024	11-09-2024	2 j	4h
Rendu Liste des projets	Clôturé	Grp6-Développeur	Grp6-Développeur	Normal	11-09-2024	11-09-2024	1 j	
Création de la base du projet	Clôturé	Grp6-Développeur	Trinite Mombouli	Normal	07-10-2024	07-10-2024	1 j	1h
Création de compte	Clôturé	Imane Choukri	Ahmed-El-Aziz Kaidi	Normal	14-10-2024	17-10-2024	4 j	· Σ 4.9h
Créer la classe Model User	Clôturé	Ahmed-El-Aziz Kaidi	Anthony Michaud	Normal	14-10-2024	14-10-2024	1 j	0.3h
Créer la classe UserController	Clôturé	Ahmed-El-Aziz Kaidi	Kevin Pradier	Normal	14-10-2024	14-10-2024	1 j	0.3h
Créer la classe UserService	Clôturé	Ahmed-El-Aziz Kaidi	Anthony Michaud	Normal	14-10-2024	14-10-2024	1 j	0.3h
Création de la vue pour la page de création	Clôturé	Ahmed-El-Aziz Kaidi	Kevin Pradier	Normal	14-10-2024	17-10-2024	4 j	3h
Réalisation des tests unitaires	Clôturé	Imane Choukri	Anthony Michaud	Normal	15-10-2024	16-10-2024	2 j	1h
Connexion	Clôturé	Wassim Ennaji	Trinite Mombouli	Normal	15-10-2024	18-10-2024	4 j	· Σ 4h
Ajout de la méthode dans la classe UserController	Clôturé	Wassim Ennaji	Rafik Rharmaoui	Normal	15-10-2024	18-10-2024	4 j	1h

SUJET	STATUT	RESPONSABLE	ASSIGNÉ À	PRIORITÉ	DATE DE DÉBUT	DATE DE FIN	DURÉE	TRAVAIL
Ajout de la méthode dans la classe UserService	Clôturé	Wassim Ennaji	Ahmed-El-Aziz Kaidi	Normal	15-10-2024	18-10-2024	4 j	1h
Création de la vue pour la page connexion	Clôturé	Wassim Ennaji	Wassim Ennaji	Normal	15-10-2024	18-10-2024	4 j	1h
Réalisation des tests unitaires	Clôturé	Imane Choukri	Rafik Rhmaoui	Normal	15-10-2024	18-10-2024	4 j	1h
Modifications des informations du compte	Clôturé	Rafik Rhmaoui	Aya El-Hassani	Normal	16-10-2024	23-10-2024	6 j	· Σ 6h
Ajout de la méthode dans la classe UserController	Clôturé	Wassim Ennaji	Aya El-Hassani	Normal	16-10-2024	23-10-2024	6 j	1h
Ajout de la méthode dans la classe UserService	Clôturé	Wassim Ennaji	Imane Choukri	Normal	16-10-2024	23-10-2024	6 j	1h
Création de la vue de la page	Clôturé	Wassim Ennaji	Alan Guilleminot	Normal	16-10-2024	23-10-2024	6 j	2h
Ajout d'une notification pour informer l'utilisateur du succès / échec	Clôturé	Wassim Ennaji	Imane Choukri	Normal	16-10-2024	23-10-2024	6 j	1h
Réalisation des tests unitaires	Clôturé	Aya El-Hassani	Aya El-Hassani	Normal	17-10-2024	23-10-2024	5 j	1h
Importer des données	Clôturé	Ahmed-El-Aziz Kaidi	Trinite Mombouli	Normal	21-11-2024	29-11-2024	7 j	· Σ 6h
Création de la classe DataController	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	21-11-2024	22-11-2024	2 j	1h
Créer la classe DataService	Clôturé	Aya El-Hassani	Aichetou N-diaye	Normal	25-11-2024	25-11-2024	1 j	2h
Création de la vue pour permettre à l'administrateur de téléverser les données	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	26-11-2024	27-11-2024	2 j	3h
Affichage des données	Clôturé	Ahmed-El-Aziz Kaidi	Imane Choukri	Normal	15-10-2024	25-10-2024	9 j	· Σ 17h
Créer la méthode dans la classe contrôleur pour afficher les données	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	15-10-2024	25-10-2024	9 j	1h
Créer la méthode dans la classe service pour récupérer les données	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	15-10-2024	25-10-2024	9 j	1h
Créer la vue pour l'affichage des données sur le portail	Clôturé	Imane Choukri	Aichetou N-diaye	Normal	15-10-2024	25-10-2024	9 j	12h
Implémenter la pagination pour la visualisation	Clôturé	Imane Choukri	Aichetou N-diaye	Normal	21-10-2024	25-10-2024	5 j	2h
Réaliser les tests unitaires	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	17-10-2024	25-10-2024	7 j	1h
Télécharger des données	Clôturé	Trinite Mombouli	Aya El-Hassani	Normal	21-10-2024	08-11-2024	15 j	· Σ 8h
Création de la méthode dans la classe contrôleur	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	21-10-2024	08-11-2024	15 j	1h
Créer la méthode dans la classe service	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	21-10-2024	08-11-2024	15 j	1h

SUJET	STATUT	RESPONSABLE	ASSIGNÉ À	PRIORITÉ	DATE DE DÉBUT	DATE DE FIN	DURÉE	TRAVAIL
Créer une vue avec de option de téléchargement pour chaque format	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	22-10-2024	08-11-2024	14 j	3h
Notification (Pop-up) succès ou échec téléchargement	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	23-10-2024	08-11-2024	13 j	1h
Gestion des droits d'accès selon les licences	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	24-10-2024	08-11-2024	12 j	1h
Réaliser les tests unitaires	Clôturé	Aya El-Hassani	Trinite Mombouli	Normal	25-10-2024	08-11-2024	11 j	1h
Affichage / Choix politique de licence	Clôturé	Wassim Ennaji	Imane Choukri	Normal	27-11-2024	29-11-2024	3 j	· Σ 7h
Récupérer des informations sur les licences	Clôturé	Aya El-Hassani	Rafik Rharmaoui	Normal	27-11-2024	28-11-2024	2 j	2h
Stockage dans un fichier json du détail d'une licence	Clôturé	Aya El-Hassani	Wassim Ennaji	Normal	28-11-2024	29-11-2024	2 j	2h
Ajouter la prise en compte des licences dans l'affichage	Clôturé	Aya El-Hassani	Rafik Rharmaoui	Normal	28-11-2024	29-11-2024	2 j	3h
Recherche des données	Clôturé	Trinite Mombouli	Wassim Ennaji	Normal	04-11-2024	08-11-2024	5 j	· Σ 7h
Modification de la vue pour l'ajout de la barre de recherche	Clôturé	Wassim Ennaji	Anthony Michaud	Normal	07-11-2024	08-11-2024	2 j	4h
Vérification et prise en compte des injections SQL	Clôturé	Imane Choukri	Anthony Michaud	Normal	08-11-2024	08-11-2024	1 j	1h
Création fonction js pour gérer l'affichage en fonction de la barre de recherche	Clôturé	Anthony Michaud	Anthony Michaud	Normal	04-11-2024	07-11-2024	4 j	2h
Filtrages des données	Clôturé	Ahmed-El-Aziz Kaidi	Trinite Mombouli	Normal	11-11-2024	19-11-2024	7 j	· Σ 14h
Création fonction js pour gérer l'affichage en fonction des filtres	Clôturé	Ahmed-El-Aziz Kaidi	Kevin Pradier	Normal	14-11-2024	15-11-2024	2 j	6h
Modification de l'affichage pour ajouter les filtres	Clôturé	Imane Choukri	Kevin Pradier	Normal	15-11-2024	18-11-2024	2 j	6h
Vérification prise en compte injection SQL	Clôturé	Imane Choukri	Kevin Pradier	Normal	18-11-2024	19-11-2024	2 j	2h
Analyser / Choisir les données à l'aide des autres Open Data	Clôturé	Trinite Mombouli	Grp6-Développeur	Normal	09-10-2024	16-10-2024	6 j	6h
Réalisation de la structure des base de données	Clôturé	Trinite Mombouli	Trinite Mombouli	Normal	14-10-2024	18-10-2024	5 j	2h
Réalisation de la page d'accueil du portail	Clôturé	Aya El-Hassani	Aichetou N-diaye	Normal	14-10-2024	15-10-2024	2 j	4h
Installation du projet sur la VM Développement	Clôturé	Rafik Rharmaoui	Rafik Rharmaoui	Normal	23-10-2024	23-10-2024	1 j	2h

### Annexe 3 : Organisation réunion

## Annexe 4 : Pages du projet OpenData

The screenshot shows the homepage of the "Portail Open Data de la Métropole". At the top, there is a navigation bar with links: Accueil, Jeux de Données, API, Ville en chiffres, Actualités, S'inscrire, and Se connecter. To the left of the navigation is a logo featuring a stylized cityscape with the words "OPEN DATA" below it. Below the navigation, the main heading "Bienvenue sur le Portail Open Data de la Métropole" is displayed, followed by the subtext "Accédez, téléchargez, et explorez les données ouvertes de notre métropole.". A blue button labeled "Explorez les données" is visible. To the right of the main content area, there is a small illustration of a person sitting at a desk with a computer monitor. Below the main heading, there is a search bar with the placeholder "Rechercher des données..." and a "Rechercher" button. The central part of the page features a section titled "Jeux de données disponibles" with two cards. The first card, titled "Démographie et Analyse du territoire", contains the subtext "Données sur la population et l'analyse du territoire." and a "Voir les données" button. The second card, titled "Déplacements et Mobilités", contains the subtext "Données sur les transports et les mobilités dans la ville." and a "Voir les données" button.



Accueil Jeux de Données API Ville en chiffres Actualités S'inscrire Se connecter

Nom:

Prénom:

Adresse mail:

Mot de passe:

Confirmer le mot de passe:

[S'inscrire](#)

Page création d'un compte



Conditions d'utilisation Politique de confidentialité Contact

© 2024 Métropole - Portail Open Data

Page connexion



Conditions d'utilisation Politique de confidentialité Contact

© 2024 Métropole - Portail Open Data



Ajout d'un thème

Ajouter

Page ajout d'un thème



Page ajout d'une donnée

Accueil Jeux de Données API Ville en chiffres Actualités Ajout d'un thème Ajout d'un libelle Mon profil Se déconnecter

Nom des données

Descriptions

Thème

Sélectionnez un thème

Langue

License

Sélectionnez une Licence

Ville

Ajout de Fichier

PDF XML CSV

Date de modification

jj/mm/aaaa

Ajouter données

## Page jeux de données



Accueil Jeux de Données API Ville en chiffres Actualités S'inscrire Se connecter

Rechercher des données...

**Filtres**

Téléchargements minimum  
Ex. : 100

Licence  
Toutes les licences

Date de modification  
jj/mm/aaaa

Thème  
Tous les thèmes

Territoire  
Tous les territoires

Langue  
Toutes les langues

Exporter en CSV  Exporter en XML  Exporter en PDF

**Aménagements cyclables**  
Dispositif de voirie destiné à organiser la circulation des vélos. Ces aménagements peuvent prendre la forme de chaussées dédiées aux cyclistes, restréntes à certains usagers dont les cyclistes, de panneaux indicateurs spécifiques ou de facilités de circulation.

**Alertes Waze**  
Données des incidents de circulation en temps réel sur Dijon Métropole relevés par les utilisateurs de l'application Waze.

**Nom de l'élément 1**  
Description de l'élément 1 download : 50, date : 01/01/2021, theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 2**  
Description de l'élément 2,download : 0,date : 01/01/2022, territoire : Dijon Métropole

**Nom de l'élément 3**  
Description de l'élément 3, download : 100,date : 01/01/2023,theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 4**  
Description de l'élément 4 download : 15, date : 01/11/2021, theme : Démographie et Analyse du territoire, territoire : Dijon Métropole

**Nom de l'élément 5**  
Description de l'élément 5,download : 30,date : 01/01/2022, territoire : Dijon Métropole

**Nom de l'élément 6**  
Description de l'élément 6, download : 100,date : 01/01/2024,theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 7**  
Description de l'élément 7 download : 50, date : 01/01/2021, theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 8**  
Description de l'élément 8,download : 0,date : 01/01/2022, territoire : Dijon Métropole

**Nom de l'élément 9**  
Description de l'élément 9, download : 100,date : 01/01/2023,theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 10**  
Description de l'élément 10 download : 15, date : 01/11/2021, theme : Démographie et Analyse du territoire, territoire : Dijon Métropole

**Nom de l'élément 11**  
Description de l'élément 11,download : 30,date : 01/01/2022, territoire : Dijon Métropole

**Nom de l'élément 12**  
Description de l'élément 12, download : 100,date : 01/01/2024,theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 13**  
Description de l'élément 13 download : 50, date : 01/01/2021, theme : Démographie et Analyse du territoire, territoire : Longvic

**Nom de l'élément 14**  
Description de l'élément 14,download : 0,date : 01/01/2022, territoire : Dijon Métropole

Page 1 sur 2

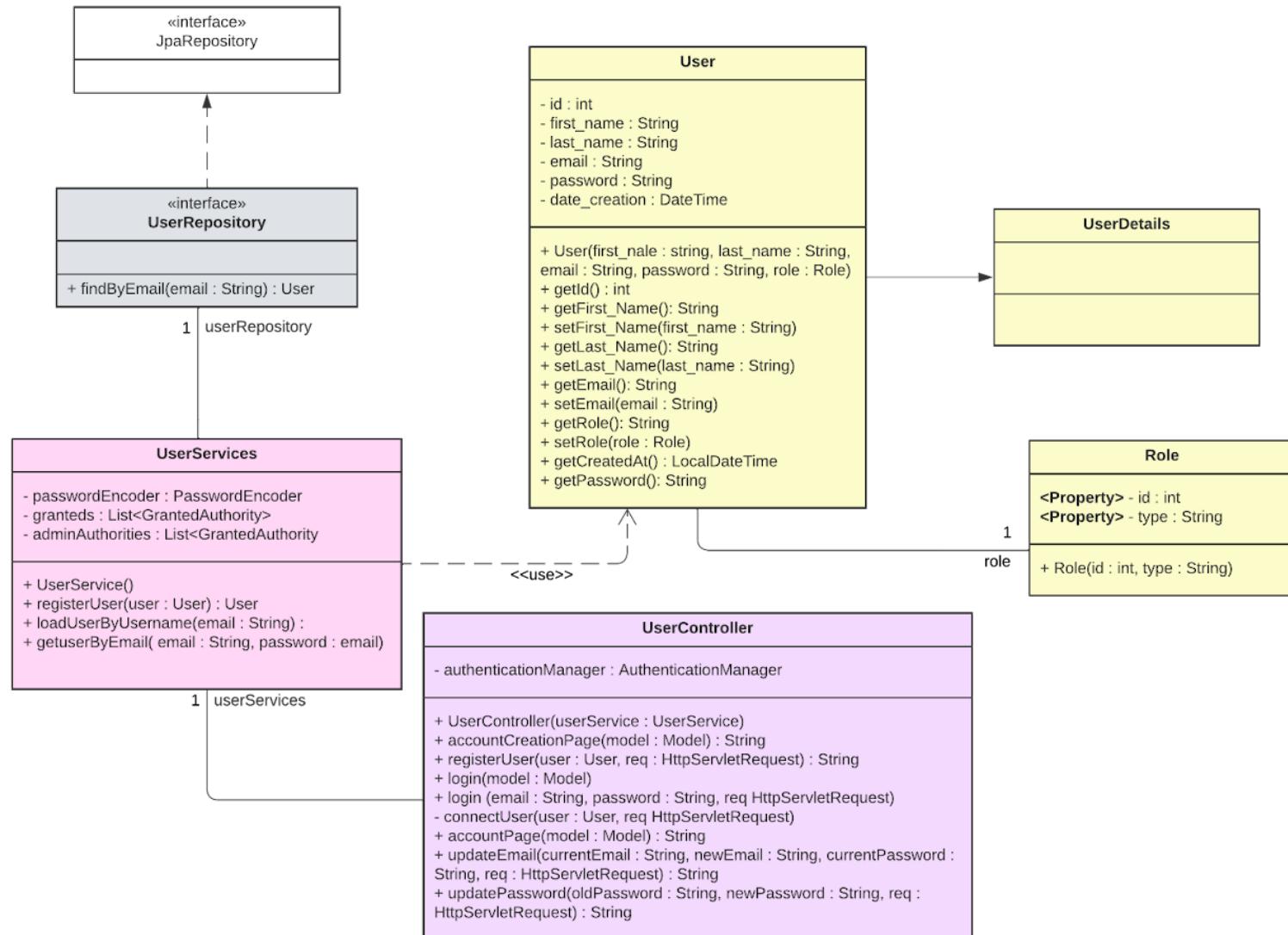
Conditions d'utilisation Politique de confidentialité Contact

## Page détail d'une donnée

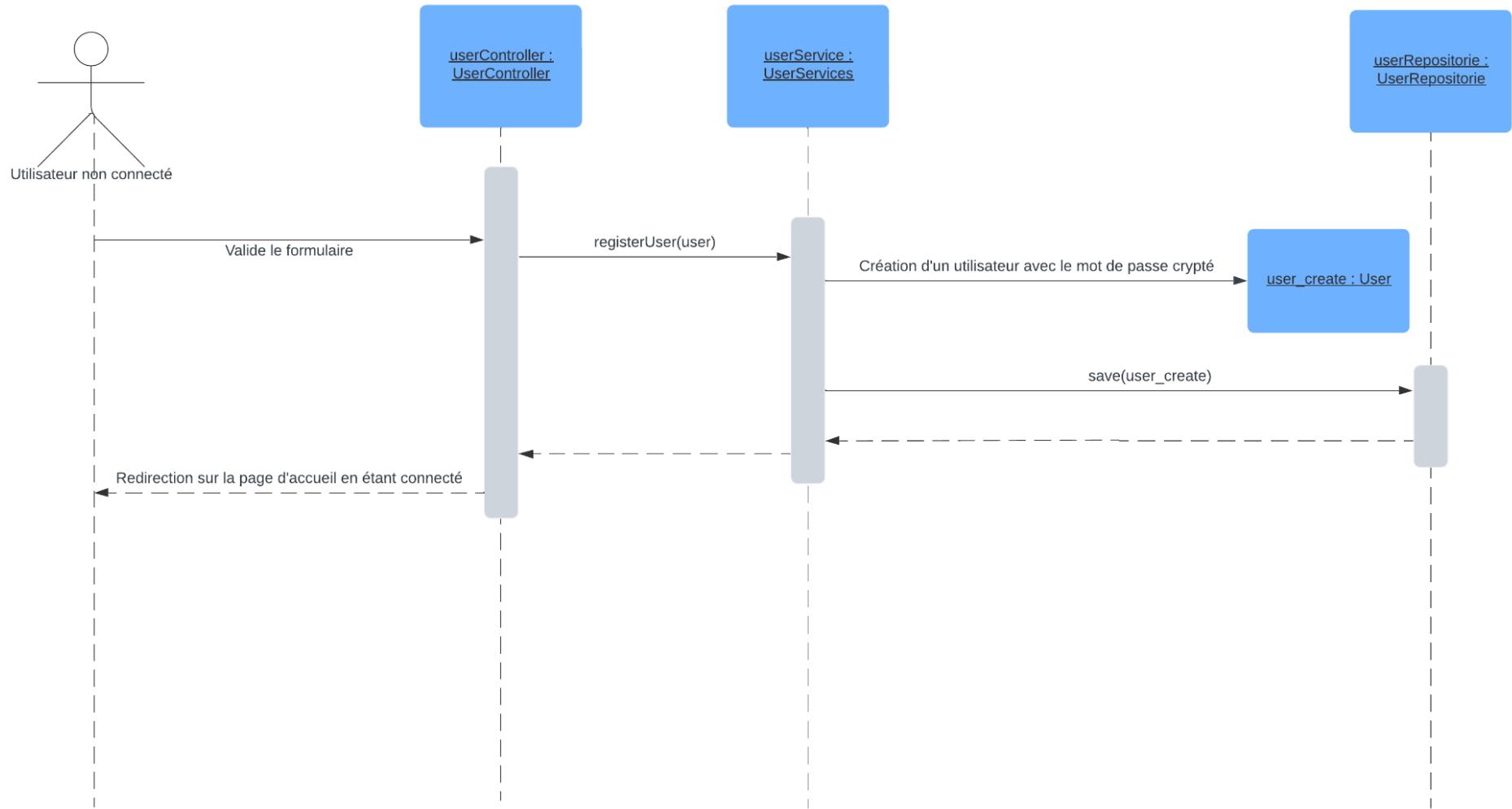
The screenshot shows a detailed view of a dataset on an open data platform. At the top, there's a navigation bar with links: Accueil, Jeux de Données, API, Ville en chiffres, Actualités, Ajout d'un thème, Ajout d'un libelle, Mon profil, and Se déconnecter. On the left, there's a sidebar with fields for Nom (testt), Thème (test), Licence (t), Télécharger (0), Modifié (02/02/2023), Langue (t), and Description (t). Below these are buttons for Télécharger (PDF, CSV, XML, JSON) and Modifier PDF, Modifier CSV, Modifier XML. At the top right, there are buttons for Afficher le csv and Afficher le pdf. To the right of the sidebar is a table with columns: OBJECTID, Nom de la commune, Code identifiant unique, ID, "Indique si oui ou non la zone est une zone communautaire dans le sens", and Code INSEE. The table contains two rows of data. At the bottom, there's a footer with links: Conditions d'utilisation, Politique de confidentialité, and Contact.

OBJECTID	Nom de la commune	Code identifiant unique	ID	"Indique si oui ou non la zone est une zone communautaire dans le sens"	Code INSEE
d'un transfert à l'EPCI dans le cadre de la loi NOTRE"	Sous type au regard de la loi Notre	Numéro SIREN de l'EPCI d'affectation du site économique	Lettres de classement du site économique	Surface en hectares	Nom
appellation de la ZAE"	Date de l'arrêté de création ou	Année de création de la zone d'activité	URL ressource		Type de maître d'ouvrage

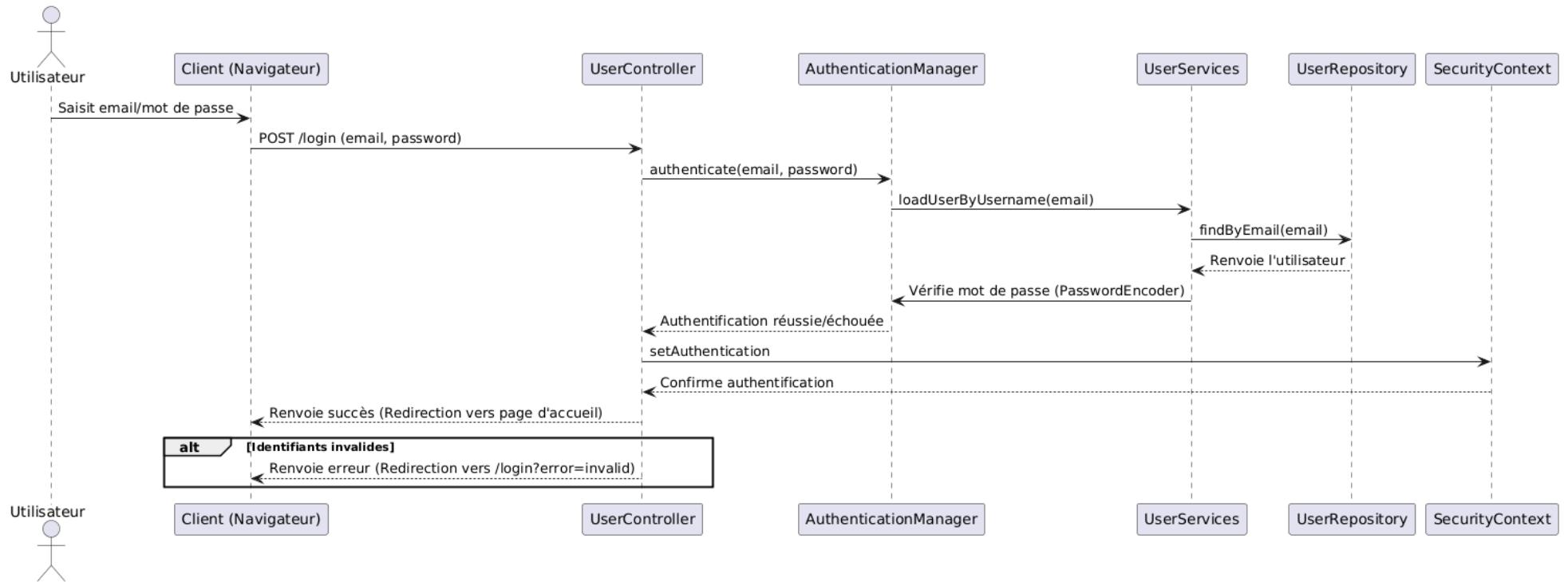
## Annexe 5 : Diagrammes des classes et séquences partie utilisateur



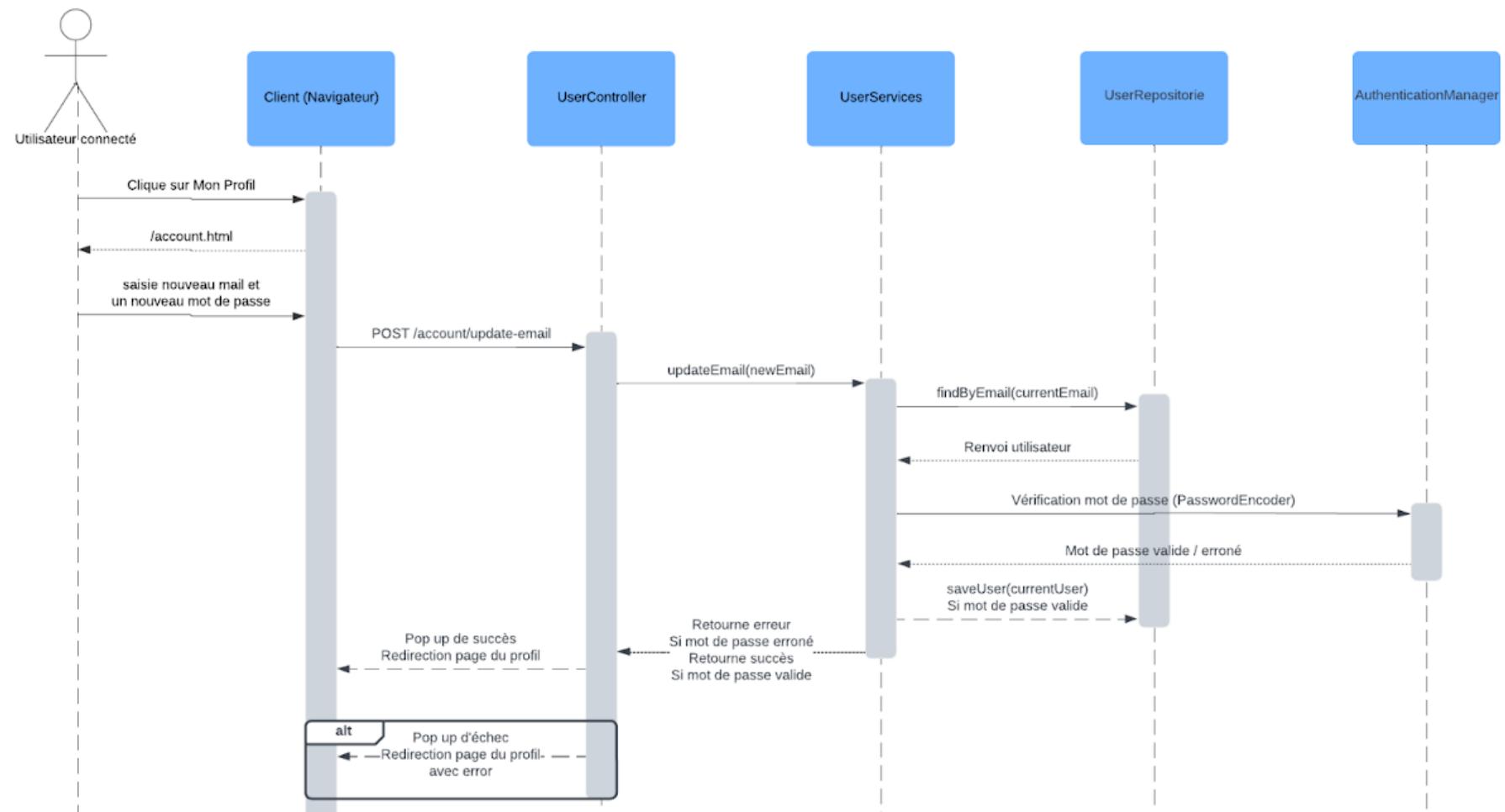
## Création d'un compte :



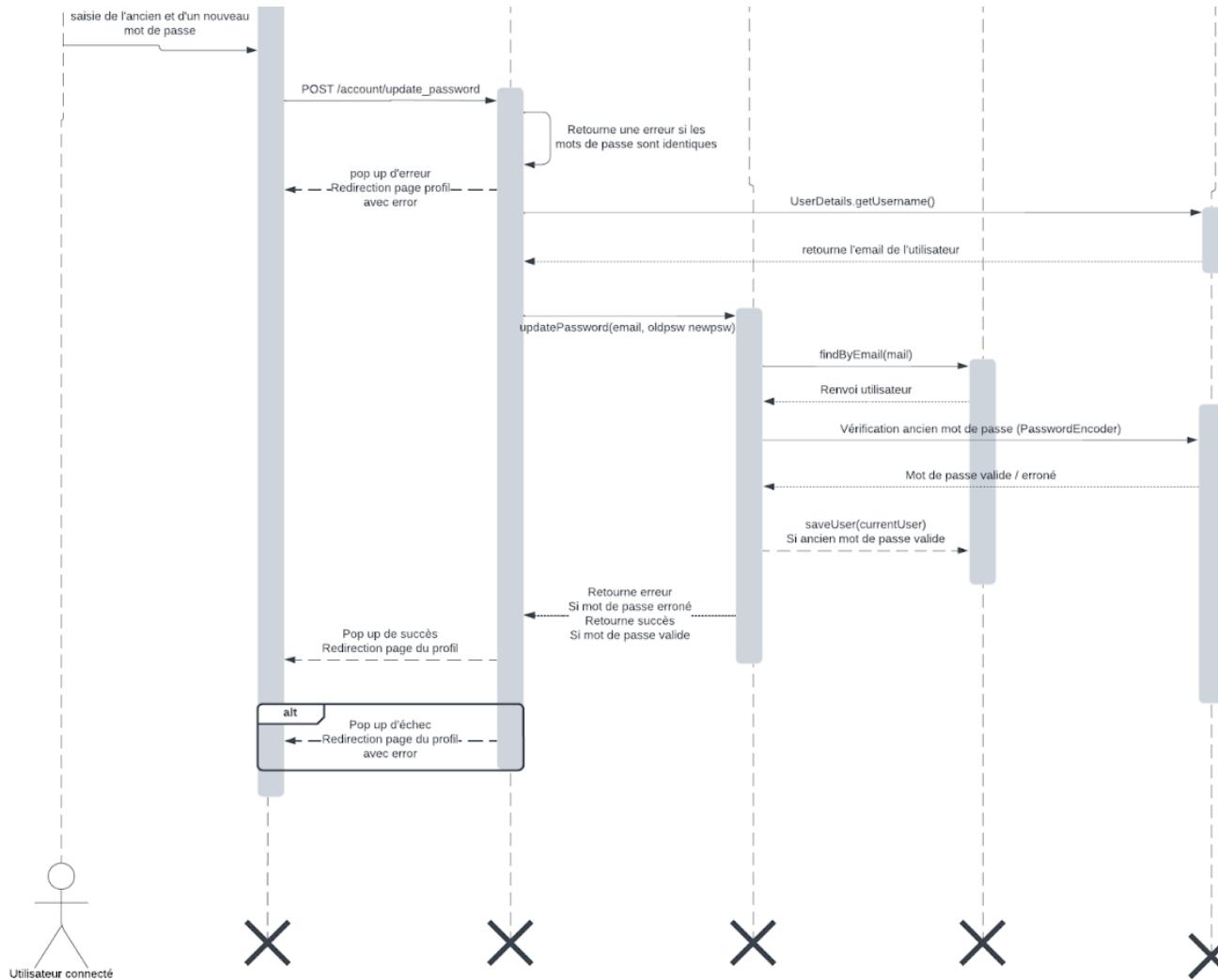
## Connexion d'un utilisateur :



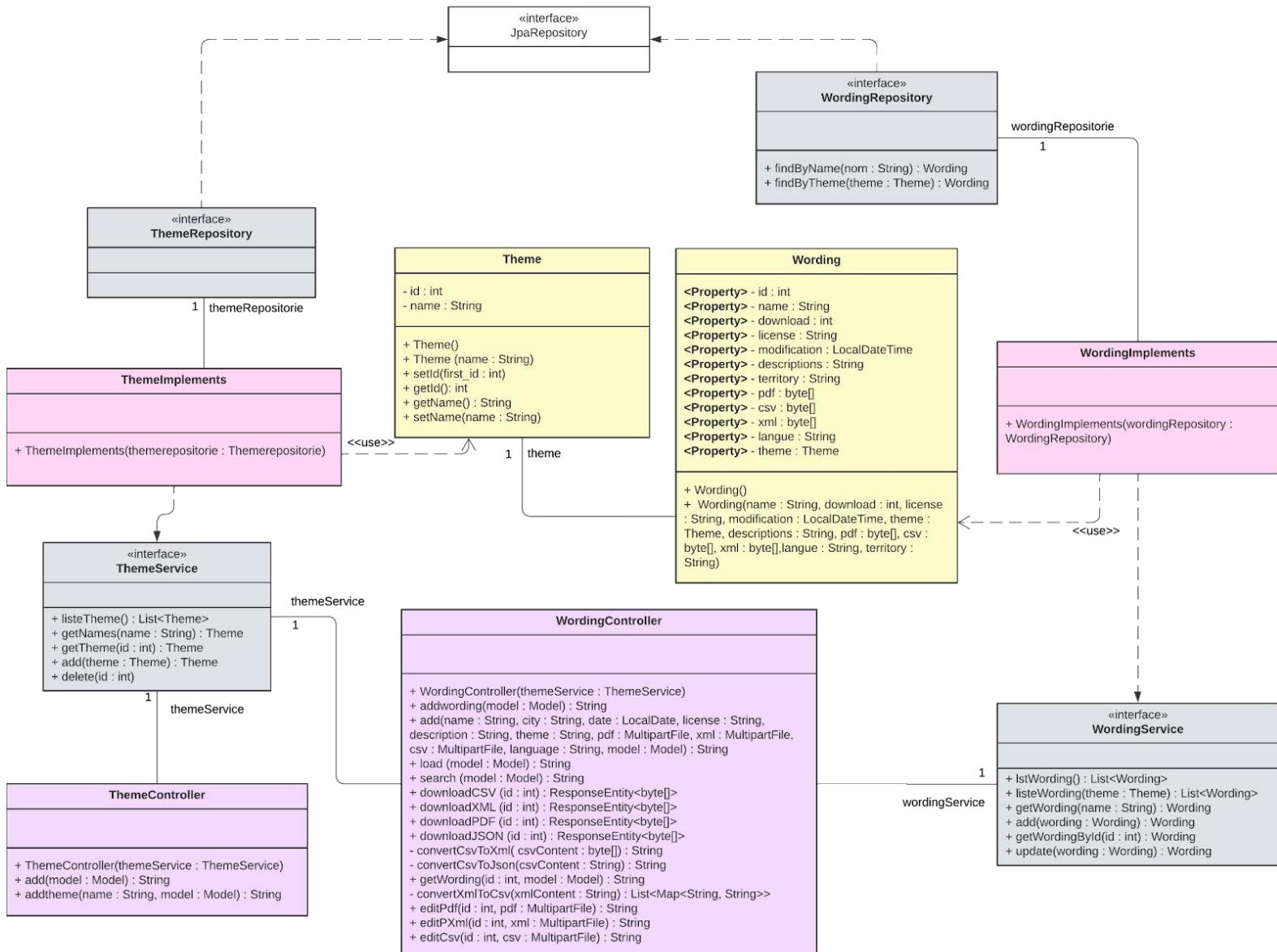
## Modification de l'adresse mail du compte :



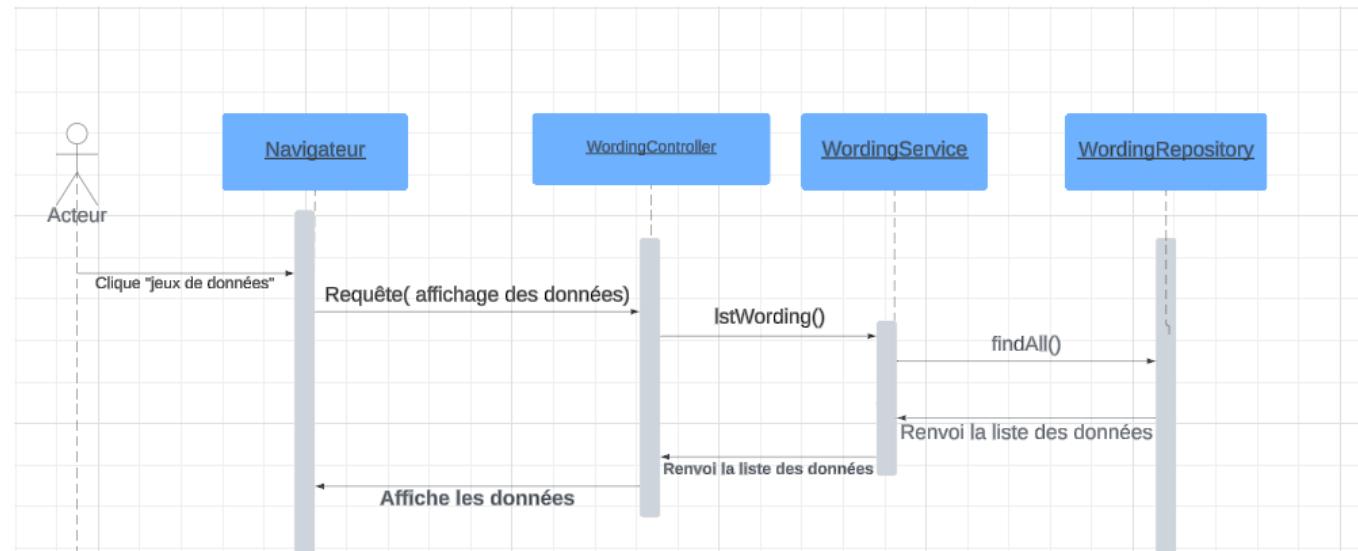
## Modification du mot de passe :



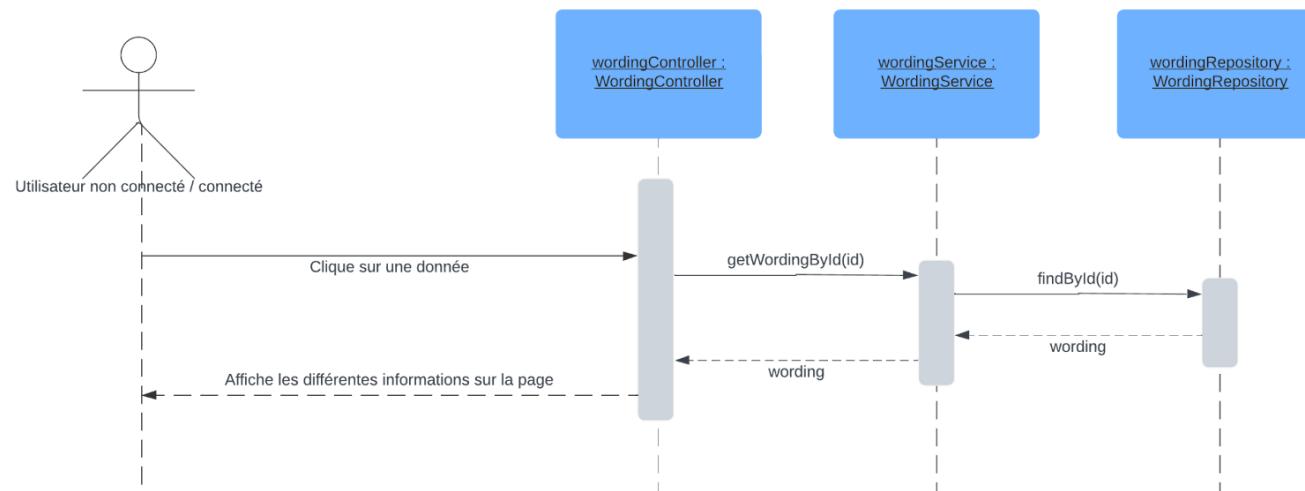
Annexe 6 : Diagrammes des classes et séquences parties données



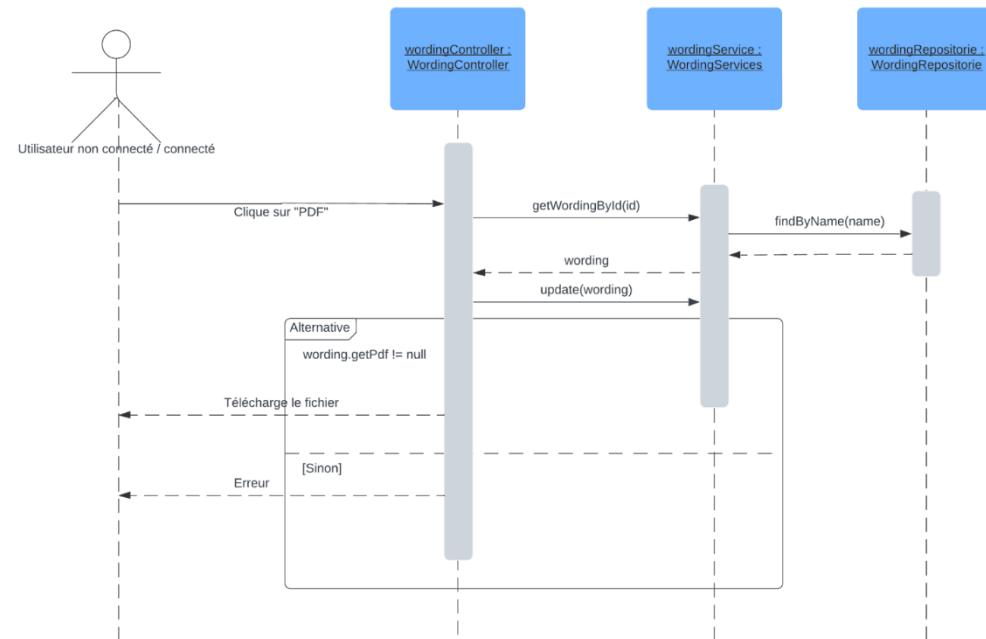
## Affichage des données :



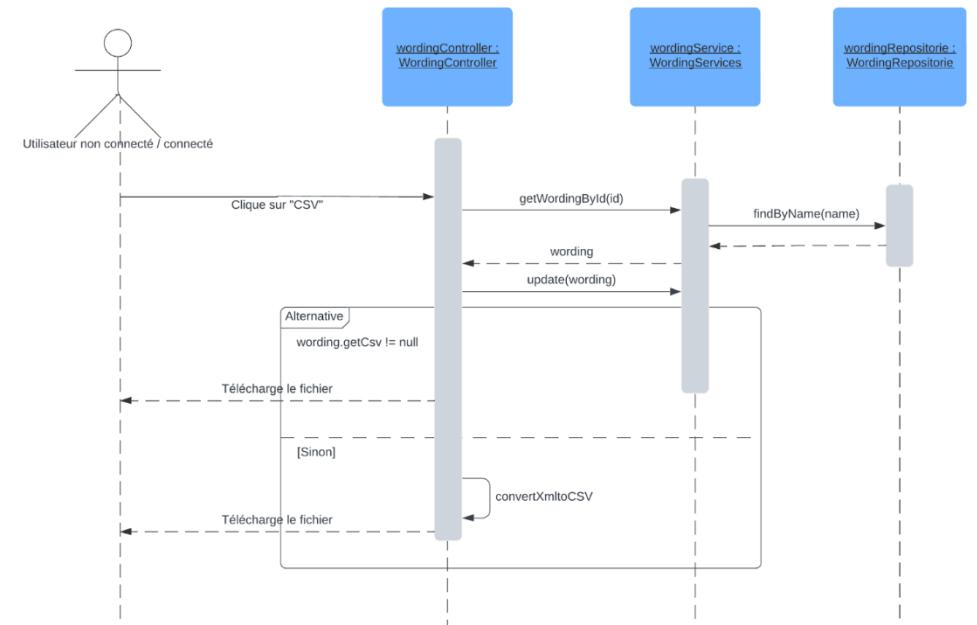
## Affichage détail d'une donnée :



Téléchargement d'un pdf:



Téléchargement d'un csv:



## Annexe 7 : Classe de tests unitaires

