



SORBONNE  
UNIVERSITÉ

---

## Projet PC3R

Bouzarkouna Malek & Chetouani Younes

Mai 2024

---

# Table des matières

<b>Introduction.....</b>	<b>2</b>
<b>Cas d'utilisation pour l'application de manga.....</b>	<b>3</b>
Cas d'utilisation 1 : Consultation des mangas tendances.....	3
Cas d'utilisation 2 : Recherche de manga et lecture.....	3
Cas d'utilisation 3 : Personnalisation de compte.....	3
<b>Description de l'API Web et Contenu Utilisé.....</b>	<b>4</b>
Intégration et Utilisation.....	4
<b>Données stockées dans la base de données.....</b>	<b>5</b>
<b>Mise à jour des données et appel à l'API externe.....</b>	<b>6</b>
<b>Serveur.....</b>	<b>6</b>
REST API Endpoint.....	6
<b>Description du client.....</b>	<b>7</b>
<b>Plan du site.....</b>	<b>8</b>
<b>Requêtes et Réponses.....</b>	<b>10</b>
<b>Schéma d'architecture global du système.....</b>	<b>11</b>

## Application web : Manga reader

L'application de type MangaDex permet aux utilisateurs de consulter les mangas tendances, de faire des recherches spécifiques et de continuer la lecture de mangas qu'ils ont déjà commencée. La fonctionnalité de consultation des mangas tendances met en avant les titres les plus populaires et les plus discutés, en se basant sur des algorithmes de recommandation et des analyses de données pour déterminer quels mangas sont actuellement en vogue proposés par MangaDex.

Les utilisateurs peuvent également utiliser un moteur de recherche intégré pour trouver des mangas en fonction de divers critères, tels que le titre et le genre. Cette recherche avancée permet aux utilisateurs de découvrir facilement de nouveaux mangas correspondant à leurs intérêts.

Pour faciliter la continuité de la lecture, l'application garde une trace des mangas et des chapitres que chaque utilisateur a déjà lus. Lorsque l'utilisateur se reconnecte, il peut reprendre sa lecture exactement là où il l'avait laissée. Cette fonctionnalité est particulièrement utile pour les lecteurs qui suivent plusieurs séries en même temps, leur évitant ainsi de devoir se souvenir du dernier chapitre lu.

En plus de ces fonctionnalités, l'application offre la possibilité aux utilisateurs d'envoyer des commentaires pour chaque chapitre. Cela permet de créer une communauté interactive où les lecteurs peuvent partager leurs impressions, poser des questions et discuter des moments clés de chaque chapitre. Les commentaires peuvent être lus par d'autres utilisateurs, favorisant ainsi les échanges et les discussions autour des mangas.

L'application propose également un système de compte complet. Les utilisateurs peuvent créer un compte personnel, ce qui leur permet de sauvegarder leurs préférences et leur historique de lecture. Ils ont la possibilité de changer leur nom d'utilisateur, leur mot de passe, ainsi que de personnaliser leur profil avec une bannière et une photo de profil. Ces options de personnalisation permettent aux utilisateurs d'exprimer leur personnalité et de rendre leur expérience plus agréable et unique.

Initialement, nous souhaitions ajouter une fonctionnalité d'écoute de musique pendant la lecture via l'API de Spotify mais par manque de temps nous n'avons pas pu l'implémenter.

# Cas d'utilisation pour l'application de manga

## Cas d'utilisation 1 : Consultation des mangas tendances

Acteur principal : Alice

1. Alice se connecte à l'application.
2. Le système affiche la page d'accueil.
3. La page d'accueil suggère les mangas tendances.
4. Alice choisit un manga parmi ceux en tendances.
5. Le système affiche un écran listant les caractéristiques du manga sélectionné (titre, auteur, description, image de couverture, etc.).

## Cas d'utilisation 2 : Recherche de manga et lecture

Acteur principal : Bob

1. Bob se connecte à l'application.
2. Bob entre la chaîne "théière" dans un champ de recherche.
3. Le système affiche une liste de mangas correspondant au critère de recherche "théière".
4. Bob choisit un manga dans la liste des résultats de recherche.
5. Le système affiche un écran listant les caractéristiques du manga sélectionné.
6. Bob sélectionne un chapitre du manga.
7. Le système affiche le contenu du chapitre.
8. Bob peut laisser un commentaire pour ce chapitre.

## Cas d'utilisation 3 : Personnalisation de compte

Acteur principal : Alice

1. Alice se connecte à l'application.
2. Alice navigue vers les paramètres de son compte.
3. Alice modifie son nom d'utilisateur, son mot de passe, sa bannière et sa photo de profil.
4. Le système enregistre les modifications et met à jour le profil d'Alice.

## Cas d'utilisation 4 : Poste d'un commentaire sur un chapitre

Acteur principal : Claire

1. Claire se connecte et lit un chapitre d'un manga.
2. Claire choisit de laisser un commentaire pendant sa lecture.
3. Claire repère et clique sur le bouton "Nouveau commentaire"
4. Claire rédige son commentaire.
5. Elle le publie en cliquant sur le bouton "Poster".
6. Le commentaire est affiché sous le chapitre pour les autres utilisateurs.
7. Claire peut continuer sa lecture ou supprimer son commentaire.

# Description de l'API Web et Contenu Utilisé

Pour notre application de mangas avec des fonctionnalités étendues, nous avons choisi d'utiliser plusieurs API Web pour offrir une expérience utilisateur riche et interactive. Voici un aperçu des API sélectionnées, du contenu qu'elles proposent, et de la manière dont notre application les utilisera :

1. API de MangaDex (<https://api.mangadex.org/docs/>)
  - Contenu proposé : L'API MangaDex fournit des informations complètes sur les mangas, y compris les détails des titres, les chapitres, les auteurs, les genres, et les couvertures.
  - Utilisation dans l'application :
    - Affichage des mangas tendances sur la page d'accueil.
    - Recherche de mangas par titre et genre.
    - Consultation des caractéristiques détaillées de chaque manga sélectionné.
    - Accès au contenu des chapitres pour la lecture en ligne.
2. API Cloudinary (<https://cloudinary.com/documentation>)
  - Contenu proposé : Cloudinary fournit des services de gestion et de transformation d'images et de vidéos, incluant le stockage, la manipulation et la diffusion.
  - Utilisation dans l'application :
    - Stockage et affichage des images de couverture des mangas.
    - Personnalisation des profils utilisateurs avec des bannières et des photos de profil.
    - Optimisation des images pour une performance améliorée et un chargement rapide dans l'application.

## Intégration et Utilisation

Notre application intégrera ces API via des appels HTTP pour récupérer et envoyer des données. Voici un exemple de flux d'utilisation pour un utilisateur :

1. Connexion et Affichage de la Page d'Accueil :
  - L'utilisateur se connecte à l'application.
  - Un appel est effectué à l'API MangaDex pour récupérer les mangas tendances.
  - Les données récupérées sont affichées sur la page d'accueil.
2. Recherche de Manga :
  - L'utilisateur entre un mot-clé dans le champ de recherche.
  - Une requête est envoyée à l'API MangaDex pour rechercher les mangas correspondant au mot-clé.
  - Les résultats sont affichés à l'utilisateur.
3. Personnalisation de Profil :
  - L'utilisateur accède aux paramètres de son compte.
  - Les images de couverture et les photos de profil sont gérées via Cloudinary.

- Les modifications sont enregistrées et affichées immédiatement dans le profil de l'utilisateur.

Cette intégration harmonieuse des API permet à notre application de fournir une expérience utilisateur riche et interactive, en combinant la lecture de mangas et des fonctionnalités personnalisées.

## Données stockées dans la base de données

### User

On stocke les informations de compte :

<b>id</b>	"1714689781527805900"
<b>username</b>	"myusername"
<b>password</b>	"\$2a\$10\$HtLj1p4m8A6UuTjz2pZM4.JmnXjPath2Kshg2gXzfYY52dQWAyHUE"
<b>profilepicture</b>	" <a href="https://res.cloudinary.com/dhmp1kcxcd/image/upload/v1712792989/ScanGo/P...">https://res.cloudinary.com/dhmp1kcxcd/image/upload/v1712792989/ScanGo/P...</a> "
<b>banner</b>	" <a href="https://res.cloudinary.com/dhmp1kcxcd/image/upload/v1712793917/ScanGo/B...">https://res.cloudinary.com/dhmp1kcxcd/image/upload/v1712793917/ScanGo/B...</a> "
<b>theme</b>	"default"
<b>followedMangas</b>	null
<b>mangas</b>	null

Les mots de passe sont cryptés à l'aide de la fonction bcrypt.

### Comments

On stocke les informations et le contenu de commentaire :

<b>id</b>	"1716678342510422700"
<b>userid</b>	"1716580827430623000"
<b>chapterid</b>	"7e668010-d71c-4b76-96bd-0de39d5ddaf6"
<b>manga</b>	"Kaoru Hana wa Rin to Saku"
<b>text</b>	"nice chapter!"
<b>createdat</b>	2024-05-25T16:19:58.966+00:00

# Mise à jour des données et appel à l'API externe

Les données sont mises à jour en fonction des actions de l'utilisateur. Par exemple, lorsqu'un utilisateur poste un commentaire, ce commentaire est ajouté à la base de données. L'API externe est appelée lorsque l'utilisateur recherche un manga, affiche les détails d'un manga ou lit un chapitre. Nous souhaitions afficher une liste de mangas populaires stockés dans la bdd, qui serait actualisée tous les jours, mais encore une fois par manque de temps nous n'avons pas pu l'implémenter.

## Serveur

Le serveur suit une approche basée sur les ressources. Il comprend des ressources pour les mangas, les chapitres et les commentaires. Chaque ressource a des fonctionnalités associées, comme la récupération, la création et la suppression.

### REST API Endpoint

Voici une brève description des APIs principaux.

1. /api/Home
  - Méthode: GET
  - Description: Renvoie les informations générales sur la page d'accueil des mangas.
  - Gestionnaire: Controllers.HomeManga
2. /api/Manga
  - Méthode: GET
  - Description: Renvoie les informations sur les mangas disponibles.
  - Gestionnaire: Controllers.GetManga
3. /api/signup
  - Méthode: POST
  - Description: Permet à un utilisateur de s'inscrire.
  - Gestionnaire: Controllers.SignUp
4. /api/signin
  - Méthode: POST
  - Description: Permet à un utilisateur de se connecter.
  - Gestionnaire: Controllers.SignIn
5. /api/User
  - Méthode: GET
  - Description: Renvoie les informations de l'utilisateur connecté.
  - Gestionnaire: Controllers.GetUser
6. /api/user/info/
  - Méthode: GET
  - Description: Renvoie les détails des mangas de l'utilisateur.
  - Gestionnaire: Controllers.GetUserMangaDetails

#### 7. /api/user/info/comment

- Méthode: GET
- Description: Renvoie les commentaires de l'utilisateur.
- Gestionnaire: Controllers.GetUserComments

#### 8. /api/updateuser

- Méthode: PUT
- Description: Met à jour les informations de l'utilisateur.
- Gestionnaire: Controllers.UpdateUser

#### 9. /api/user/chapter/

- Méthode: PUT
- Description: Met à jour les informations du chapitre de l'utilisateur.
- Gestionnaire: Controllers.UpdateUserChapter

#### 10. /api/user/follow/

- Méthode: PUT
- Description: Met à jour les informations de suivi de l'utilisateur.
- Gestionnaire: Controllers.UpdateUserFollow

#### 11. /api/user/chapter/comment

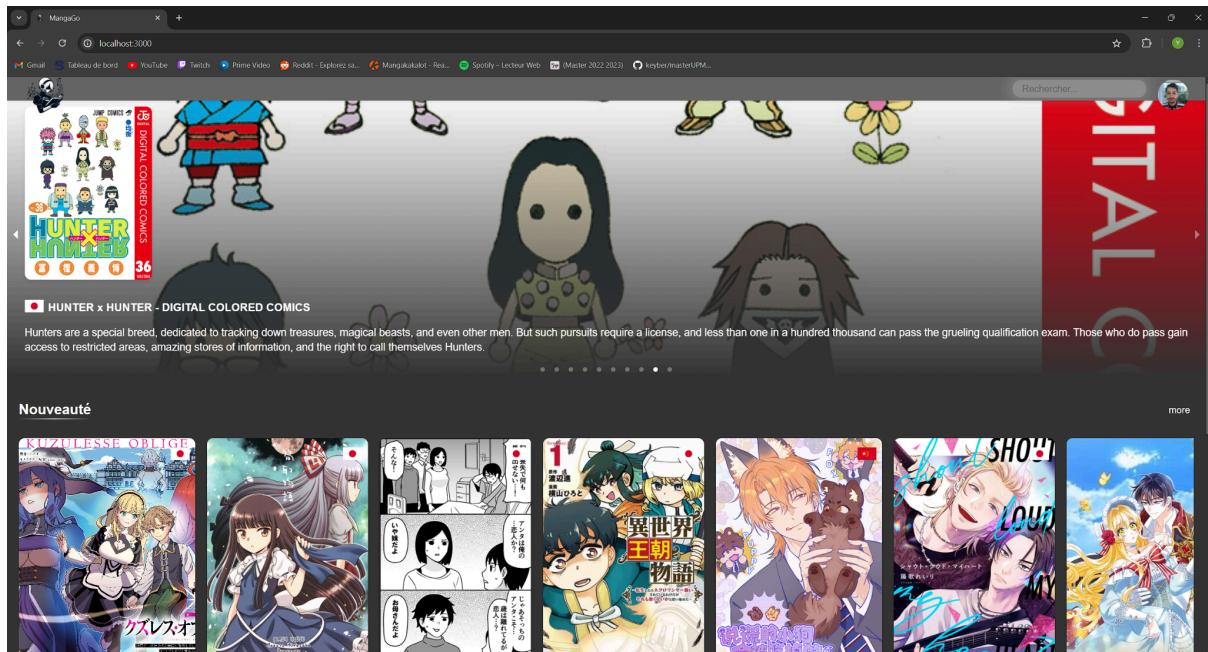
- Méthode: POST
- Description: Permet à l'utilisateur d'ajouter un commentaire à un chapitre.
- Gestionnaire: Controllers.HandleComment

## Description du client

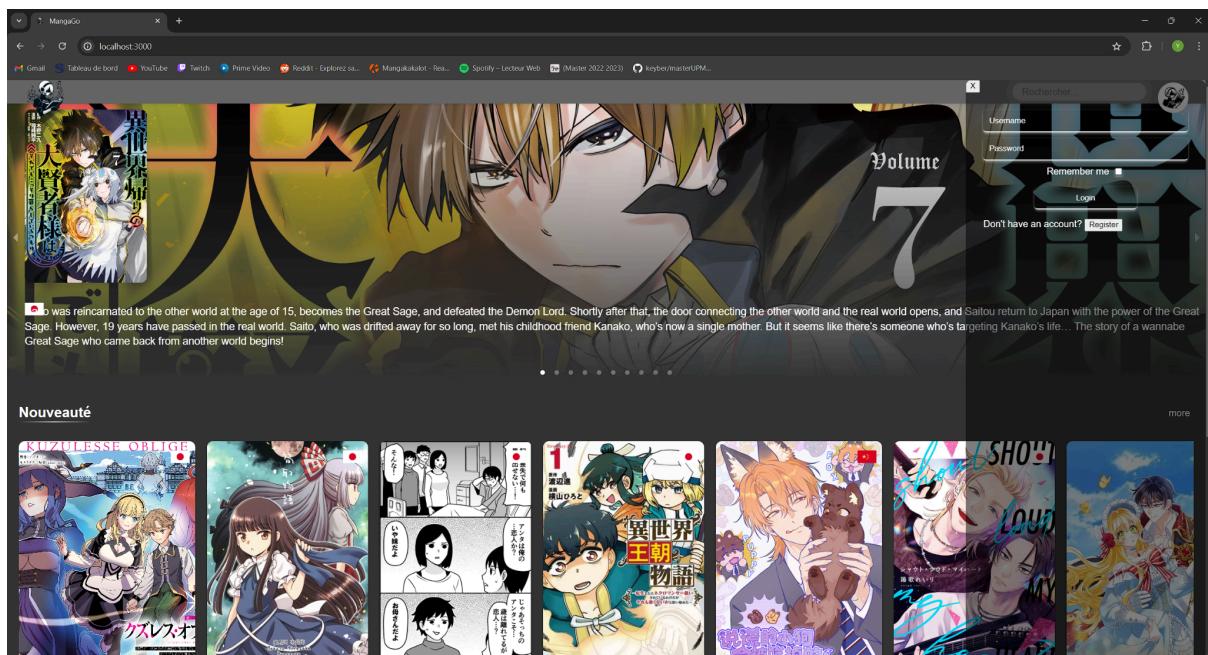
Le client est construit avec React. Il utilise react-router-dom pour gérer la navigation entre différentes pages. Voici une brève description des composants principaux :

- App.js : C'est le composant racine de l'application. Il définit les routes pour l'application et rend les composants correspondants en fonction de l'URL.
- Navbar : C'est le composant de la barre de navigation qui est présent sur toutes les pages.
- Home : C'est le composant de la page d'accueil de l'application.
- MangaDetails : Ce composant affiche les détails d'un manga spécifique.
- Chapter : Ce composant affiche les détails d'un chapitre spécifique.
- ProfilePage : Ce composant affiche la page de profil d'un utilisateur.
- EditProfile : Ce composant permet à l'utilisateur de modifier son profil.
- ShowSearch et ShowTag : Ces composants sont utilisés pour afficher les résultats de recherche et les tags respectivement.

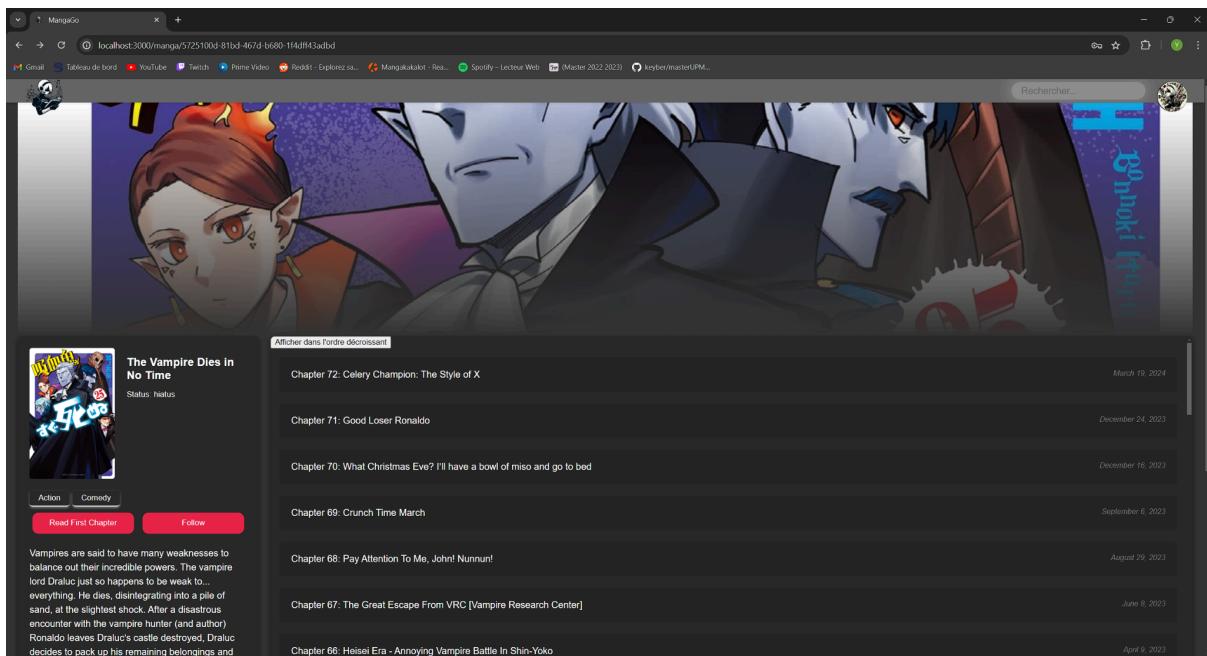
## Plan du site



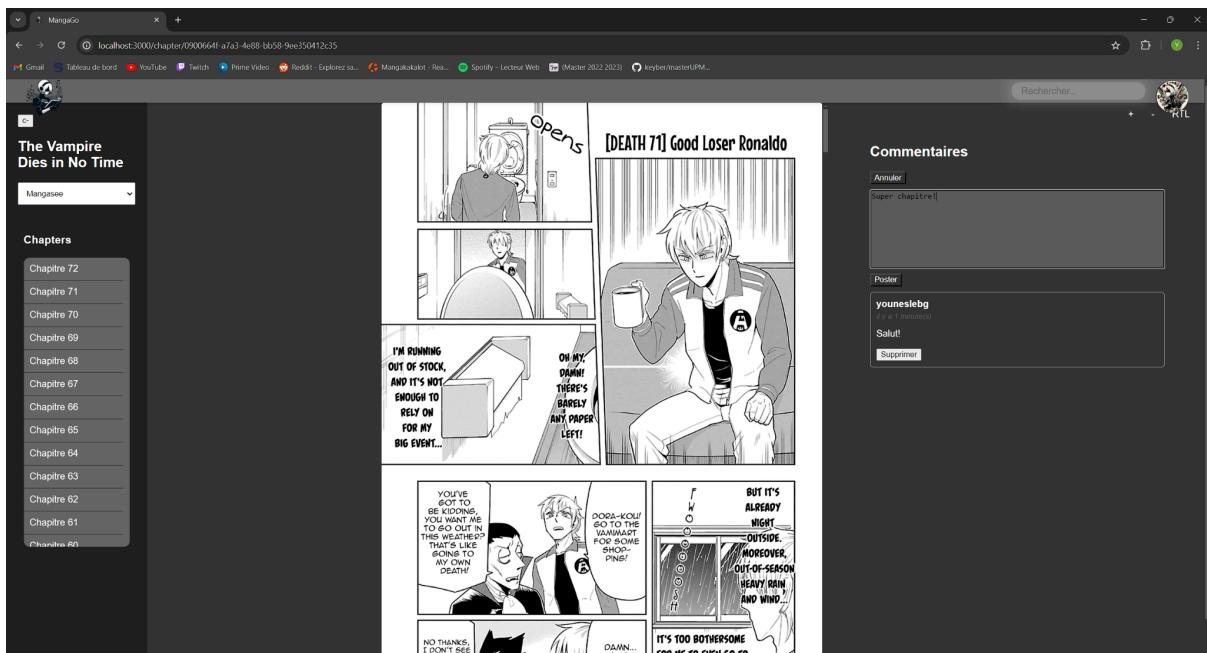
Sur la première vue de notre site, vous n'êtes pas obligé de vous connecter pour lire des mangas.



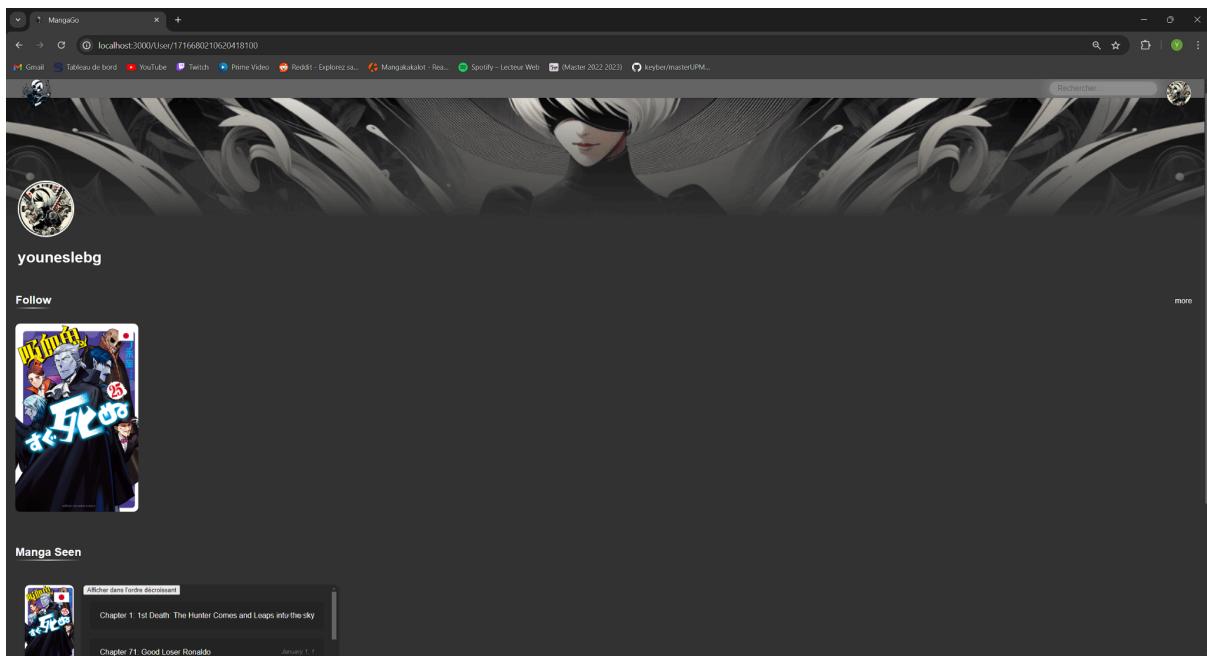
En cliquant sur l'icône en haut à droite un menu s'affiche, vous pouvez vous connecter en entrant votre identifiant et votre mot-de-passe. Vous pouvez également choisir de vous inscrire en cliquant sur le bouton Register.



En cliquant sur un manga, vous serez redirigé vers une page descriptive de celui-ci. Sur cette page, si vous êtes connecté, vous pourrez choisir de cliquer sur Follow pour ajouter ce manga à vos favoris. Vous pourrez également, même en étant connecté en tant qu'invité, lire des chapitres présentés dans la liste en milieu de page (liste que vous pouvez afficher dans l'ordre croissant ou décroissant). Vous pouvez également cliquer sur Read First Chapter pour lire le premier chapitre du manga.



Une fois un chapitre sélectionné vous serez redirigé vers une page contenant le scan du chapitre. Vous pouvez changer de chapitre en le sélectionnant dans le menu à gauche. Si vous êtes authentifié, vous pourrez également poster des commentaires. Vous pourrez également supprimer vos commentaires.



Sur votre page de profil, auquel vous pouvez accéder en cliquant sur votre photo de profil, vous pourrez voir la liste des mangas favoris, l'historique des chapitres lus et plus bas (pas visible sur cette capture) la liste des commentaires que vous avez posté.

## Requêtes et Réponses

La plupart des requêtes sont réalisées en HTTP car la plupart d'entre elles concernent l'authentification pour une demande de ressource. On ajoute des arguments dans la query string, ou pour les POST les données dans le body.

Voici une description d'une requête et de la réponse entre le client et le serveur pour une fonctionnalité de l'API :

Requête pour ajouter un commentaire (/api/user/info/comment)

Méthode HTTP : POST

Corps de la Requête : JSON contenant les détails du commentaire

Réponse (Contenu JSON) : {

```
id : "1716678342510422700"  
userid : "1716580827430623000"  
chapterid : "9e6a726e-28bd-43c2-8a4b-459e4d2acc12"  
manga : "Touhou - Nitori no Ashita wa Pororoca  
(Doujinshi)"  
text : "yo les gars"  
createdat : 2024-05-25T23:05:42.510+00:00  
}
```

## Schéma d'architecture globale du système

