



# Systeme de Gestion de Bibliothèque



- **EL MOULOUDI HICHAM**

JUIN 2025

# Table des matières

- 1- Introduction Du Projet :**.....2
  - 1.1- Problématique :.....2
  - 1.2- Solution Proposée :.....2
- 2- Modélisation du Système :**.....2
  - 2.1- Diagramme de class des Views :.....2
  - 2.2- Diagramme de class des Models :.....3
- 3- Les Algorithmes Clés :**.....4
  - 3.1- Diagramme circulaire - Livres par genre: .....4
  - 3.2- Histogramme : Top 10 des auteurs les plus populaires : .....4
- 4- Architecture du Système et Documentation Technique:** .....6
- 5- Les Visualisations :** .....6
- 6- Difficultés rencontrées et solutions:** .....6

# 1- Introduction Du Projet :

## 1.1- Problématique :

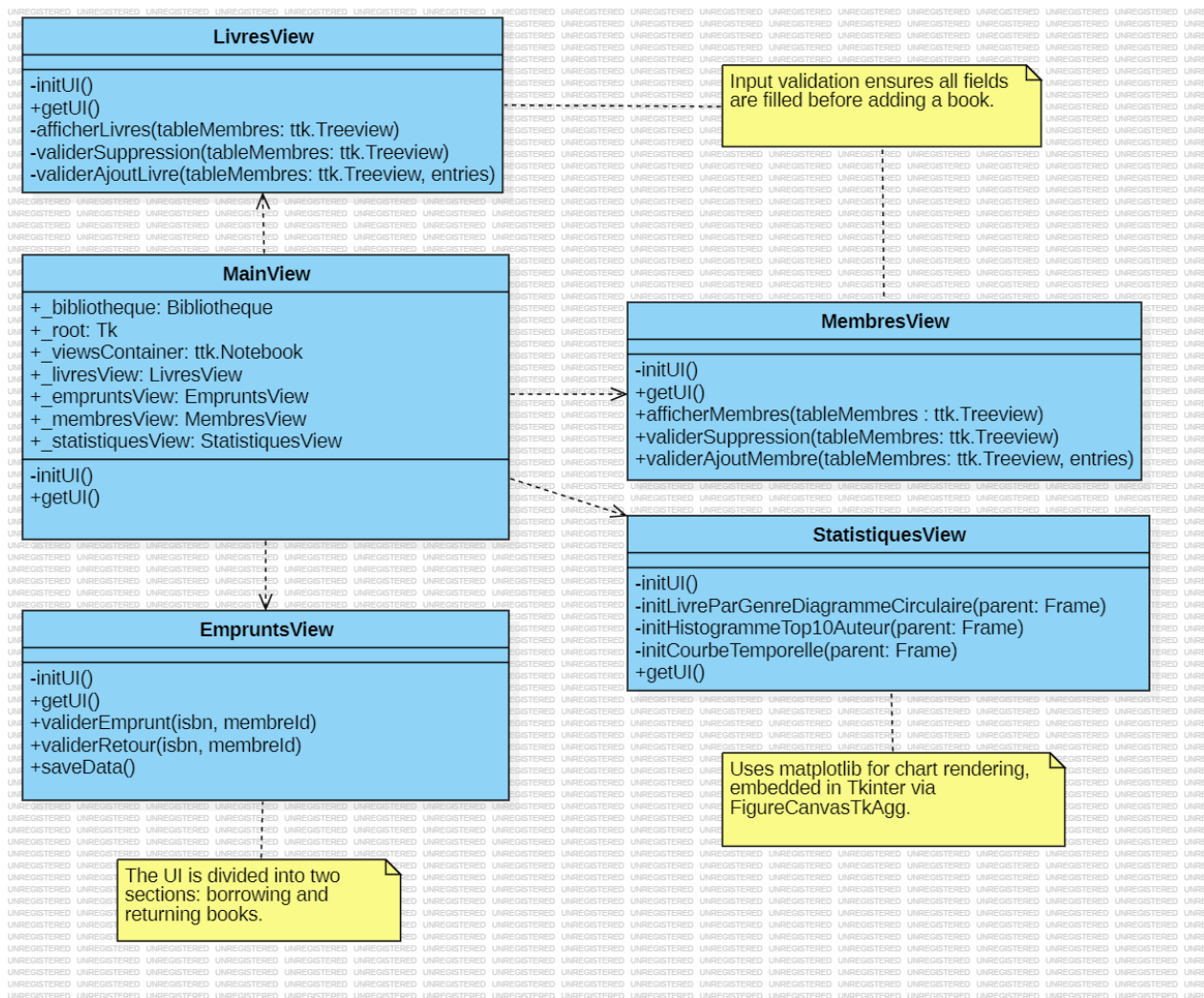
La gestion manuelle des livres, des membres et des emprunts dans une bibliothèque peut être fastidieuse et sujette à des erreurs. Ce projet vise à automatiser ces processus en offrant une solution logicielle complète et intuitive.

## 1.2- Solution Proposée :

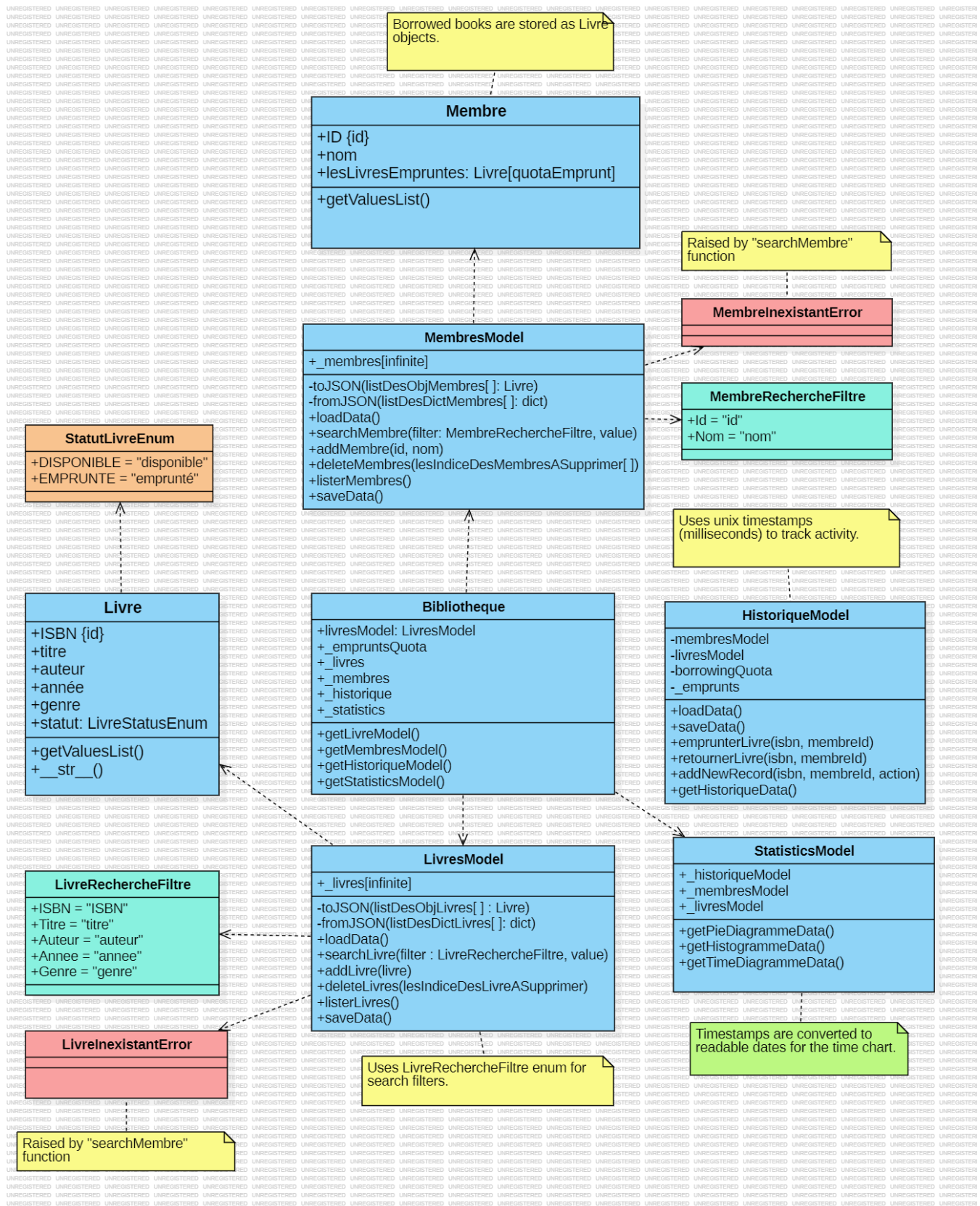
Une application Python avec une interface graphique Tkinter, permettant de gérer les livres, les membres, les emprunts, et de générer des statistiques visuelles. Les données sont persistées dans des fichiers JSON pour une gestion durable. Avec une architecture M/V.

# 2- Modélisation du Système :

## 2.1- Diagramme de class des Views :



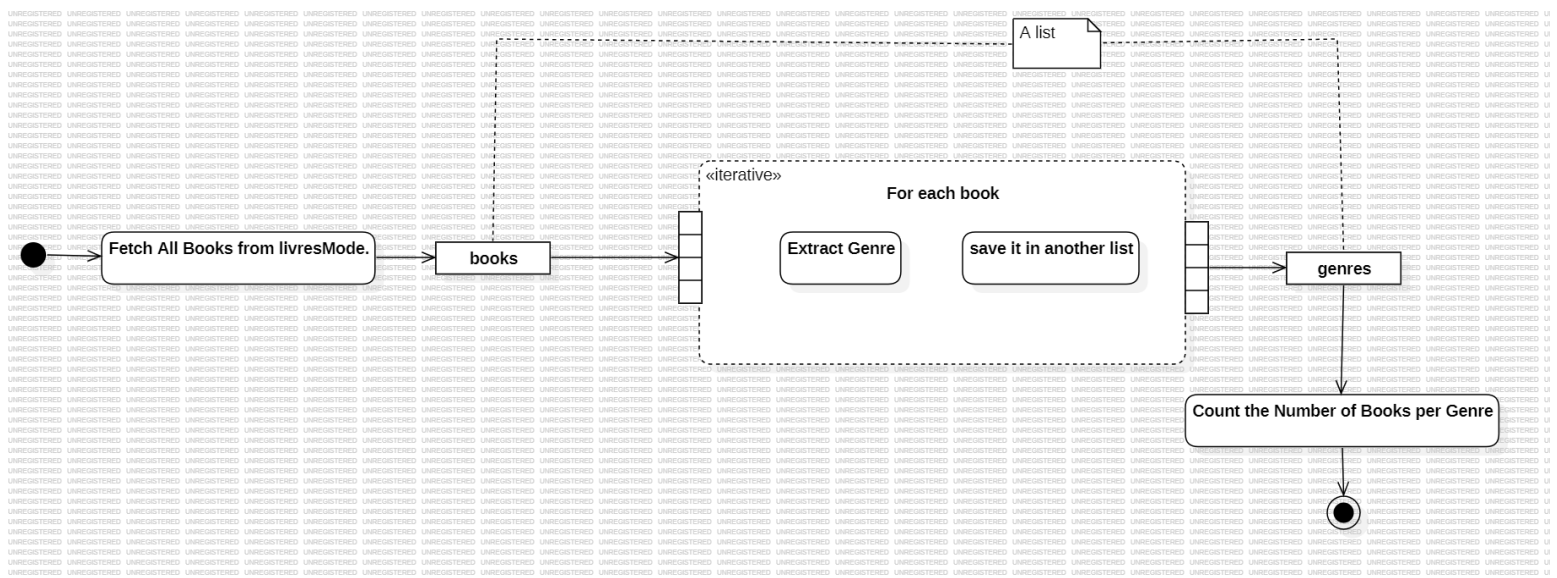
## 2.2- Diagramme de class des Models :



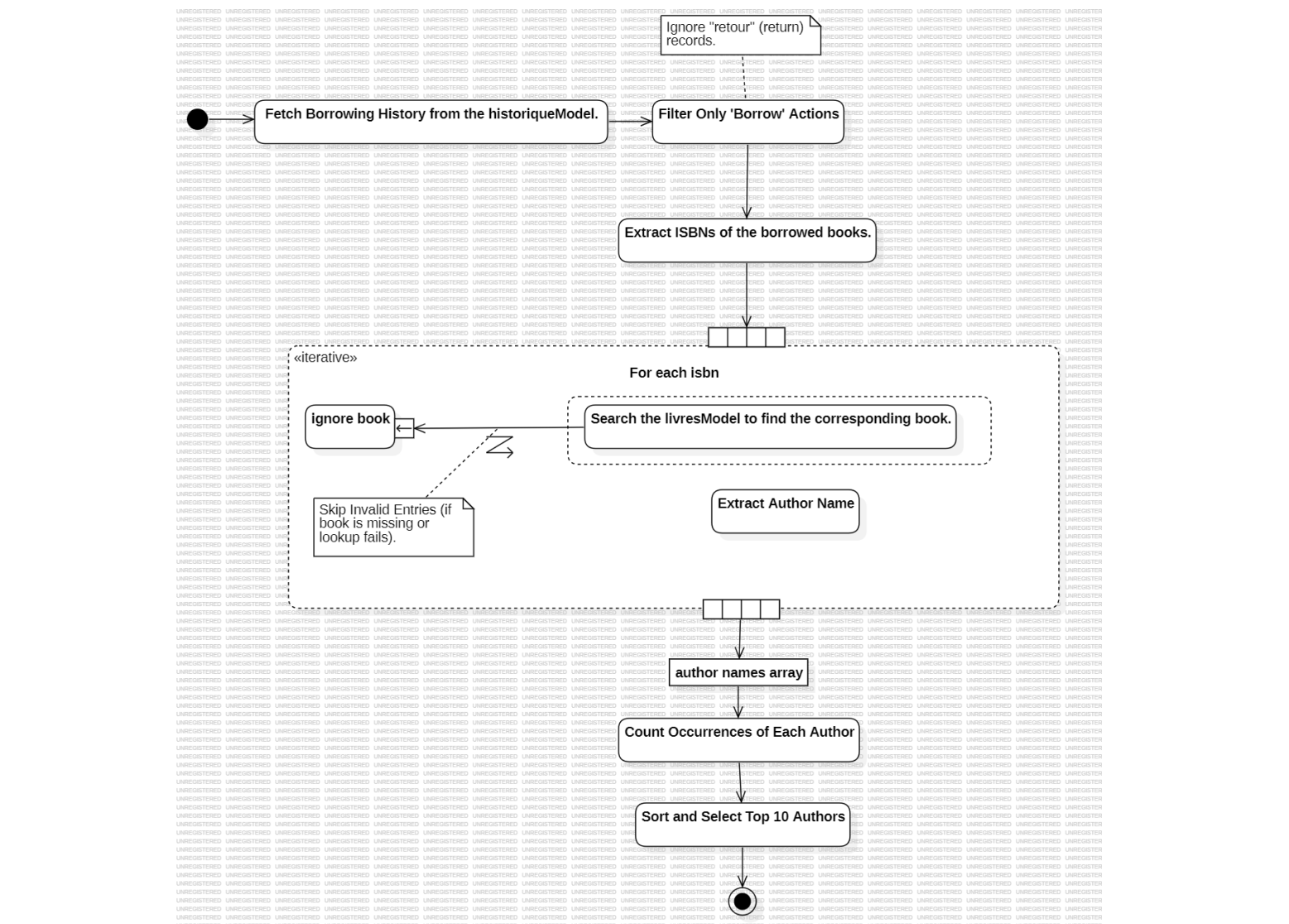


# 3- Les Algorithmes Clés :

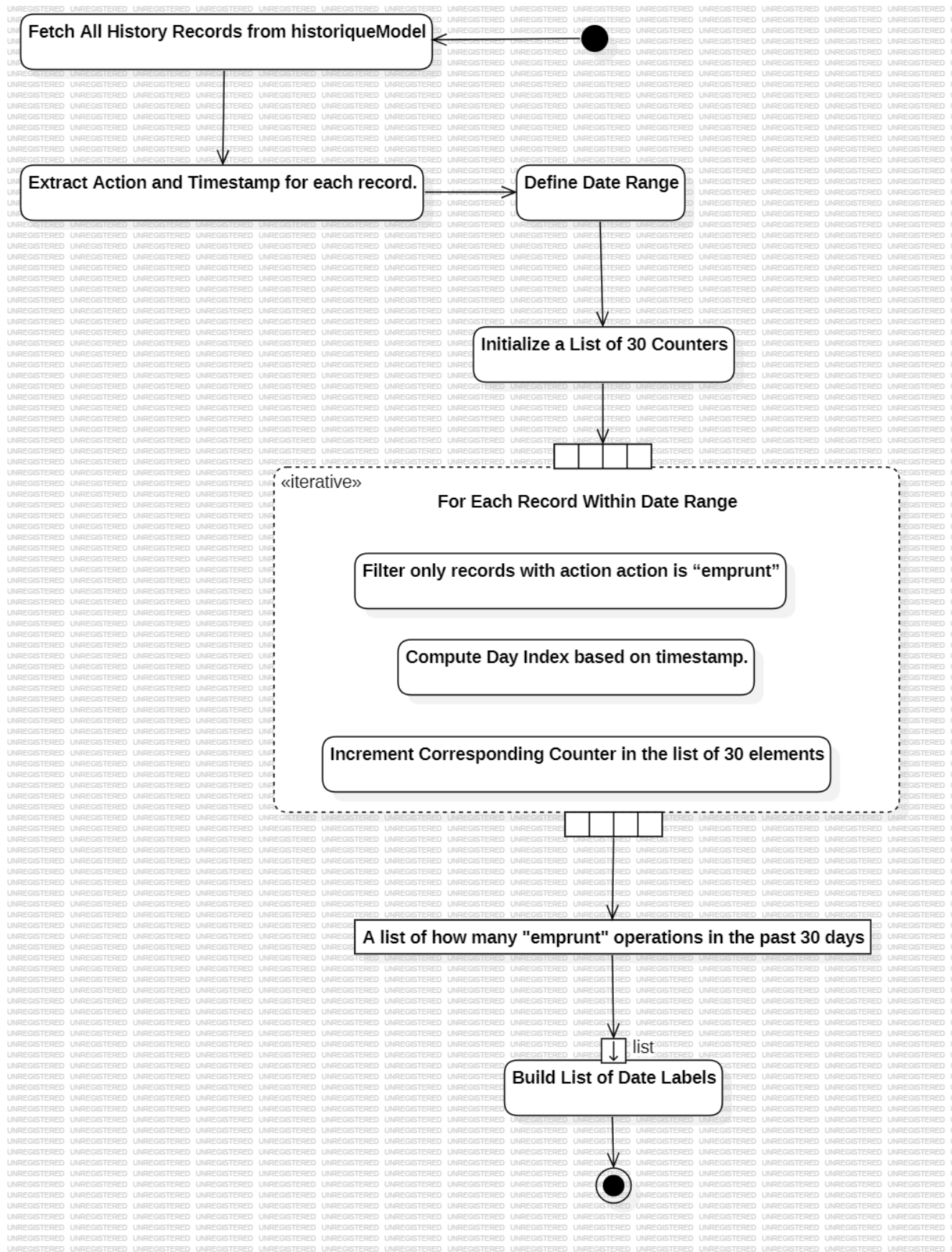
## 3.1- Diagramme circulaire - Livres par genre:



## 3.2- histogramme : Top 10 des auteurs les plus populaires :



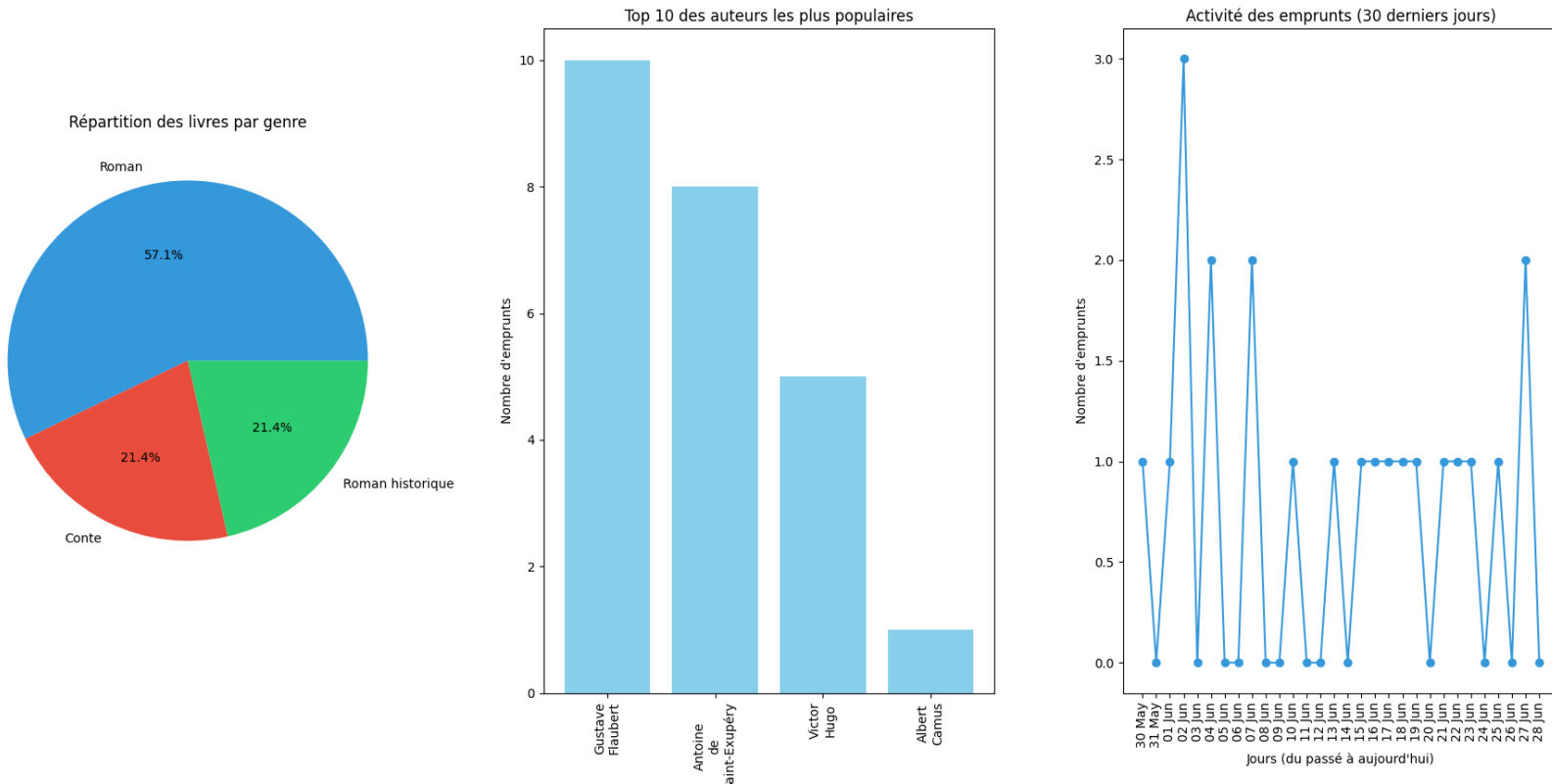
### 3.3- Histogramme temporelle : Activité des emprunts (30 derniers jours):



## 4- Architecture du Système et Documentation Technique:

Se trouve sur : <https://hicham-el-mouloudi.github.io/Projects-Documentations/>

## 5- Les Visualisations :



## 6- Difficultés rencontrées et solutions:

### 1- Problème : Structure maintenable du projet

Difficulté à organiser le code pour qu'il soit lisible, évolutif et facile à maintenir.

#### Solution :

- Adopter une architecture M/V (Modèle-Vue).

#### Explication :

- La séparation des responsabilités (modèles pour la logique métier et la gestion des données, vues pour l'interface graphique) permet de modifier ou d'étendre une partie du code sans impacter le reste.

### 2- Problème : Transformation des données entre JSON et objets Python

Comment convertir facilement les données lues depuis un fichier JSON en objets Python (Livre, Membre) et inversement.

#### Solution :

- Utiliser des méthodes de transformation (fromJSON, toJSON) et l'attribut spécial `__dict__` pour sérialiser/désérialiser les objets.

### 3- Problème : Historique des emprunts/retours

Comment identifier de façon unique chaque action d'emprunt ou de retour et garder un historique fiable.

#### Solution :

- Utiliser le timestamp Unix (en millisecondes) comme identifiant unique pour chaque action.

### 4- Problème : Chargement et sauvegarde des données

Éviter de recharger ou de sauvegarder les données inutilement, ce qui ralentirait l'application.

#### Solution :

- Charger et sauvegarder les données une seule fois au démarrage/à la fermeture, ou lors d'une action explicite de l'utilisateur.

### 5- Problème : Chargement indépendant des données pour chaque vue

Éviter que l'ouverture d'une vue ralentisse l'application à cause du chargement de données inutiles.

#### Solution :

- Chaque vue charge uniquement les données dont elle a besoin.

### 6- Problème : Suppression multiple dans la vue des livres

Supprimer plusieurs livres sélectionnés sans provoquer d'erreur d'indice.

#### Solution :

- Trier la liste des indices à supprimer en ordre décroissant avant la suppression.

### 7- Problème : Gestion des formulaires et des champs

Comment regrouper et manipuler facilement les champs d'un formulaire.

#### Solution :

- Utiliser un dictionnaire pour stocker les widgets d'entrée par nom de champ.

### 8- Problème : Gestion du statut des livres

Comment représenter de façon fiable le statut d'un livre (disponible/emprunté).

#### Solution :

- Utiliser une énumération (Enum) pour les statuts, mais stocker la valeur sous forme de chaîne de caractères.