# Informations générales

**Noms des coéquipiers** : Nabil H. & Anthony B.

**URL du référentiel sur GitHub** : (optionnel si vous travaillez seul)

## Description du prototype

(Copiez la description de votre prototype ici)

Pour notre projet complet, nous aimerions créer une base de données pour les librairies à partir de python. Cette base de données permettrait d’ajouter des livres, de mettre une date d’emprunt avec frais de retard et de classifier les multiples objets de la bibliothèque concernée. Nous avons l’intention d’utiliser l’ORM SQLAlchemy comme nouvelle technologie.

## Table des matières

[Semaine #1 : 9 octobre au 15 octobre 2](#_Toc148011688)

[Semaine #2 : 16 octobre au 22 octobre 4](#_Toc148011689)

[Semaine #3 : 23 octobre au 30 octobre 5](#_Toc148011690)

# Semaine #1 : 9 octobre au 15 octobre

*Optionnel : Si vous aviez déjà commencé à apprendre/tester des nouvelles technologies tout en faisant votre analyse préliminaire, vous pouvez les indiquer ici.*

## Objectifs \*\*Recherche de tutoriels et configuration initiale\*\*

Bien qu’il soit difficile de prédire le temps que prendra le développement des différentes fonctionnalités lorsqu’on utilise des technologies inconnues, il est important de se fixer des objectifs concrets à atteindre afin de guider l’avancement du projet. Si vous ne parvenez pas à atteindre tous vos objectifs au cours d’un semaine donnée, vous pourrez les repousser au prochain.

**Listez vos objectifs pour cette semaine (en ordre de priorité) :**

1. Trouver des tutoriels et des ressources sur Flask, SQLAlchemy et Jinja2.
2. Installer Flask et SQLAlchemy, configurer la base de données SQLite, et créer une application Flask de base.

## Apprentissages effectués

Quelles ressources (sites web, vidéos, forums, etc.) avez-vous consultées au cours de cette semaine pour apprendre à utiliser les technologies choisies?  
Résumez en quelques points ce que vous avez appris (les éléments clés), puis assignez un score entre 1 et 5 à chaque ressource que vous avez consultée (et justifiez brièvement pourquoi).  
Vous pouvez regrouper plusieurs apprentissages connexes sur une seule ligne pour alléger le tableau.

| Résumé des éléments clés appris | Technologie(s) concernée(s) | Ressources consultées (URL) et évaluation de leur utilité (score de 1 à 5) |
| --- | --- | --- |
| (EXEMPLES)   * Comment organiser les widgets qui sont déjà dans un *frame* en utilisant le *layout* *grid()* * Comment changer la police (*font*) et la taille des caractères d’un champ texte * Comment utiliser l’attribut *ANCHOR* pour centrer les widgets (cependant cela s’applique à tous les widgets, ce qui est problématique, et je n’ai pas trouvé de moyen de le faire pour un seul élément) | Librairie **Tkinter** pour faire des GUI en python | **(EXEMPLE)**   * <https://www.youtube.com/watch?v=CWehUVhfGXU> * *(Autres ressources connexes ici si c’est réparti sur plusieurs pages ou vidéos)* * 3/5 : Le programmeur a montré comment centrer les widgets, mais il montre aussi qu’il n’arrive à le faire qu’en l’appliquant à tous les widgets, ce qui n’est pas vraiment le résultat que je voulais obtenir. |
| Configuration de base de l'application Flask. | **Flask** : Framework web minimaliste pour Python |  |
| Configuration de la base de données et des modèles de données | **SQLAlchemy** : Bibliothèque pour l'interaction avec la base de données SQL. | [**https://docs.sqlalchemy.org/en/14/orm/tutorial.html**](https://docs.sqlalchemy.org/en/14/orm/tutorial.html)  **3/5** nous à aider à comprendre comment marche SQLAlchemy, mais pas à faire exactement ce qu’on veut |
| Comment rendre notre page html dynamique | **Jinja2** : Moteur de modèle pour générer des pages HTML dynamiques. | [**https://www.youtube.com/watch?v=OraYXEr0Irg**](https://www.youtube.com/watch?v=OraYXEr0Irg)  **4/5** Cette vidéo nous a donné une idée générale de comment utiliser jinja2 pour notre projet |

## Fonctionnalités développées

Qu’avez-vous réussi à accomplir dans votre projet au cours de cette semaine?  
Mentionnez les fonctionnalités que vous avez implémentées même si elles ne sont pas encore entièrement fonctionnelles ou intégrées au reste du prototype.   
Vous pouvez regrouper plusieurs fonctionnalités connexes sur une seule ligne pour alléger le tableau.

| Fonctionnalité(s) développée(s) | Technologie(s) utilisée(s) | Difficultés rencontrées |
| --- | --- | --- |
| (EXEMPLE)   * Ajout d’un *label* qui indique l’endroit pour entrer des commandes * Ajout d’une *textbox* pour lire les commandes entrées par l’utilisateur * Ajout d’un bouton pour fermer l’application (il est toujours en commentaire donc non visible, mais fonctionnel) * Ajout de styles au bouton pour quitter en utilisant un *font*, une couleur de fond, et une couleur de caractère différente | Librairie **Tkinter** pour faire des GUI en python | * On peut seulement changer la largeur d’un *Entry*. * L’attribut *height* n’est pas modifiable, donc pour réussir à le changer il faut utiliser la méthode *place()* avec les coordonnées indiqués, MAIS cette solution n’est pas possible si on utilise le système de *grid* pour placer les widgets, car tkinter ne peut tenir compte que d’un seul système de placement des widgets à la fois. * Il faut utiliser la méthode *place()* dans les *canvas*, car sinon en utilisant *pack()* le canvas devient dynamique et il perd les dimensions qu’on l’indique lors de sa création. |

## Changements ou ajouts aux technologies choisies

*Si au cours de cette semaine vous avez décidé de changer vos choix de technologies initiaux, ou si vous avez réalisé qu’il vous manquait des technologies utiles, listez-les ici :*

* **Jinja2**
  + **Intégration Naturelle avec Flask** : Jinja2 est le moteur de modèle par défaut de Flask, ce qui signifie qu'il est directement pris en charge et recommandé pour la création de modèles dans des applications Flask.
  + **Syntaxe lisible** : utilise une syntaxe simple et lisible, similaire à HTML.
  + **Performance** : Connu pour sa performance pour la génération de contenu dynamique.
* …

# Semaine #2 : 16 octobre au 22 octobre

## Objectifs \*\* Développement des fonctionnalités \*\*

**Listez vos objectifs pour ce sprint (en ordre de priorité) :**

1. Définir les modèles de données (Book, Classification, Loan) à l'aide de SQLAlchemy.
2. Créer des vues et des routes pour l'ajout de livres, la gestion des emprunts et l'affichage des livres et des emprunts.
3. Créer des templates HTML pour l'affichage des données.

## Apprentissages effectués

Quelles ressources (sites web, vidéos, forums, etc.) avez-vous consultées au cours de cette semaine pour apprendre à utiliser les technologies choisies?  
Résumez en quelques points ce que vous avez appris (les éléments clés), puis assignez un score entre 1 et 5 à chaque ressource que vous avez consultée (et justifiez brièvement pourquoi).  
Vous pouvez regrouper plusieurs apprentissages connexes sur une seule ligne pour alléger le tableau.

| Résumé des éléments clés appris | Technologie(s) concernée(s) | Ressources consultées (URL) et évaluation de leur utilité (score de 1 à 5) |
| --- | --- | --- |
|  |  |  |

## Fonctionnalités développées

Qu’avez-vous réussi à accomplir dans votre projet au cours de cette semaine?  
Mentionnez les fonctionnalités que vous avez implémentées même si elles ne sont pas encore entièrement fonctionnelles ou intégrées au reste du prototype.   
Vous pouvez regrouper plusieurs fonctionnalités connexes sur une seule ligne pour alléger le tableau.

| Fonctionnalité(s) développée(s) | Technologie(s) utilisée(s) | Difficultés rencontrées |
| --- | --- | --- |
| * Définition des modèles de données | SQLAlchemy | * Aucune. (grâce a internet.) |
| * Création de routes et de vues pour les fonctionnalités de l'application. | Flask | * À un moment le lien vers notre page menait à un chargement infini. Un redémarrage a permis de régler ce problème. |
| * Utilisation de templates HTML pour afficher des données dynamiques. | Jinja2 | * Aucune. (Notions similaire à ce que nous apprenons en classe actuellement) |

## Changements ou ajouts aux technologies choisies

*Si au cours de cette semaine vous avez décidé de changer vos choix de technologies initiaux, ou si vous avez réalisé qu’il vous manquait des technologies utiles, listez-les ici :*

* **Nouvelle technologie choisie 1** : Raison de votre changement ou ajout

# Semaine #3 : 23 octobre au 30 octobre

## Objectifs \*\* Validation, styles CSS et fonctionnalité de retard \*\*

**Listez vos objectifs pour ce sprint (en ordre de priorité) :**

1. Ajouter une validation pour la date de retour minimale dans le formulaire d'emprunt.
2. Appliquer des styles CSS à l'application pour l'améliorer visuellement.
3. Implémenter la fonctionnalité de calcul des frais de retard en fonction de la date de retour dans la route d'emprunt.

## Apprentissages effectués

Quelles ressources (sites web, vidéos, forums, etc.) avez-vous consultées au cours de cette semaine pour apprendre à utiliser les technologies choisies?  
Résumez en quelques points ce que vous avez appris (les éléments clés), puis assignez un score entre 1 et 5 à chaque ressource que vous avez consultée (et justifiez brièvement pourquoi).  
Vous pouvez regrouper plusieurs apprentissages connexes sur une seule ligne pour alléger le tableau.

| Résumé des éléments clés appris | Technologie(s) concernée(s) | Ressources consultées (URL) et évaluation de leur utilité (score de 1 à 5) |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## Fonctionnalités développées

Qu’avez-vous réussi à accomplir dans votre projet au cours de cette semaine?  
Mentionnez les fonctionnalités que vous avez implémentées même si elles ne sont pas encore entièrement fonctionnelles ou intégrées au reste du prototype.   
Vous pouvez regrouper plusieurs fonctionnalités connexes sur une seule ligne pour alléger le tableau.

| Fonctionnalité(s) développée(s) | Technologie(s) utilisée(s) | Difficultés rencontrées |
| --- | --- | --- |
| * Validation de formulaires. | Flask | * **jinja2.exceptions.TemplateAssertionError: No filter named 'date'**. Jinja2 ne fournit pas de filtre natif pour obtenir la date actuelle. Du a cette erreur, nous avons du obtenir la date actuelle directement dans votre application Flask et la transmettre à votre modèle HTML. |
| * Utilisation de CSS pour styliser l'application web. | CSS | * Aucune, nous avons utilisé bootstrap pour modifier l’apparence de notre page web et nous étions déjà familier avec celui-ci. |
| * Calcul des frais de retard en fonction de la date de retour. | Notre génie | * Aucune |

## Changements ou ajouts aux technologies choisies

*Si au cours de cette semaine vous avez décidé de changer vos choix de technologies initiaux, ou si vous avez réalisé qu’il vous manquait des technologies utiles, listez-les ici :*

* **Nouvelle technologie choisie 1** : Raison de votre changement ou ajout