Rapport Devoir 1

L’architecture du projet est basée une structure orientée objet respectant de bonnes pratiques de modularité. Chaque classe possède un rôle bien défini assurant la clarté et la réutilisabilité du code.

Les classes sont les suivantes :

* **Utilisateur** : Représentant et gère les informations des utilisateurs du système avec des attributs pour l’identifiant, le rôle, le nom, les pénalités ainsi que l’abonnement.
  + Dépendamment du rôle ils peuvent, rechercher et emprunter des ressources, gérer les ressources et les autres utilisateurs et générer des rapports sur l’utilisation des ressources.
  + La classe est associée à la classe **Emprunt**
* **Ressources** : définissant donc les ressources que l’on peut emprunter tel que les livres liés aux différentes informations comme le titre, l’auteur, le type et la disponibilité d’emprunt, en stockant les informations.
  + Ces dernières sont stockées dans une structure imbriquée permettant de simplifier leur organisation
  + La classe est associée à la classe **Emprunt**
* **Emprunt** : Récupérant les informations sur les emprunts de ressources tel que les utilisateurs, en liant ce dernier à la ressource en question, avec leur date d’emprunt et leur date prévue de retour.
* **GereEmprunt**: Responsable de la gestion des emprunts ainsi que des réservations.
* **GererRess**: Gère les ressources en permettant d’en ajouter, supprimer et/ou modifier.
  + Tout comme emprunt, il interagit avec **Ressources** et **Emprunt**
* **GestionUser**: Dans la même idée que « GererRess », mais pour les utilisateurs permettant d’en ajouter ou supprimer et gérer les possibles pénalités.
* **Rapport**: Permet de générer des rapports de statistiques sur les emprunts et l’utilisation des ressources.

Cette structure permet de séparer les responsabilités avec chaque classe dédiée à une fonction précise favorisant ainsi la lisibilité et la maintenance du code. Cette structure permet aussi une facilité pour la possible évolution du code, puisque le système est extensible ce qui permet d’ajouter de nouvelles fonctionnalités le tout sans impacter les autres modules. Pour finir cette structure permet de respecter les bonnes pratiques en développement logiciel avec l’utilisation de listes et de dictionnaires permettant de gérer les données facilement.

Les acteurs principaux du système proviennent principalement des différents rôles de la classe **Utilisateur**, pouvant être un étudiant, un professeur…, ainsi que des potentiels administrateurs et le système de manière global.

En plus de possiblement optimiser le système par la suite, on a rencontré des contraintes tel que le lien idéal entre les différentes classes, la mise en place de futures fonctionnalités tel que la notification des retards et de quoi gérer les pénalités automatiquement par le système.