Dans ce projet on est sensé à développer un logiciel qui pourra nous gérer une bibliothèque Alors pour cela on aura besoin des classes suivantes :

* **La classe Document :**

La classe doucement est une classe de base dans notre cas elle porteras les fonction et méthodes qui seront utilisé par les sous-classes elle va contenir :

1. **Propriétés :**

* Titre
* Résumé
* Code
* Prix
* Date d’expédition
* Nom d’étagère
* Numéro de case

1. **Les fonctions :**

* Un constructeur
* Redéfinition des fonctions Equals et ToString
* Emplacement : est une fonction qui vas nous retourner l’emplacement du document dans la bibliothèque (Nom d’étagère + Numéro de case)
* Disponibilité : c’est une fonction de type Boolean qui va nous retourner **TRUE** c’est le document est disponibleet **FALSE** c’est le document n’est pas disponible
* **Les sous-classes**

La classe documents va contenir 3 sous classe :

1. **Livre :**
2. Les attribues :

* Auteur
* Nombre de pages

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité

1. **Manuscrit :**
2. Les attribues :

* Auteur
* Langue originale

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité

1. **Magasine :**
2. Les attribues :

* Catégories
* Liste d’article

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité
* **La classe clients**

Une autre classe qui est indépendante de la classe doucement et des autres sous-classes elle aura ces propres méthodes et fonctions.

1. **Propriétés :**

* Nom
* Prénom
* Date De l’emprunt

1. **Les fonctions**

* Un constructeur
* Redéfinition de la fonction Equals et ToStrin
* **La classe Bibliothèque**

C’est une classe d’agrégation qui gère a la fois les clients et documents elle vas contenir :

1. **Une liste de Clients** : la liste va contenir les clients crée à partir de la classe clients
2. **Une liste des documents** : la liste va contenir les documents crée à partir de la classe document
3. Les fonctionnalités suivantes :

- Ajouter, supprimer, modifier et rechercher un document

- Ajouter, supprimer, modifier et rechercher un client

- Emprunter un document par un client.

- Restituer un document par un client.

- Afficher les documents de la bibliothèque.

* **Iresume**

Iresume est une interface adaptée pour tous les sous-classe (Livres, Manuscrit et Magasines) elle contient une fonctionnalité :

1. **AfficheResume :** qui retourne une idée générale sur le contenu de chaque document

* **Bibliothèque :**

1. Une liste de Clients
2. Une liste des documents
3. Les fonctionnalités suivantes :

- Ajouter, supprimer, modifier et rechercher un document

- Ajouter, supprimer, modifier et rechercher un client

- Emprunter un document par un client.

- Restituer un document par un client.

- Afficher les documents de la bibliothèque.

* **Clients :**

1. Propriétés :

* Nom
* Prénom
* Date De l’emprunt

1. Les fonctions

* Un constructeur
* Redéfinition de la fonction Equals
* Redéfinition de la fonction ToString
* **Documents :**

1. Propriétés :

* Titre
* Résume
* Code
* Prix
* Date d’expédition
* Nom d’étagère
* Numéro de case

1. Les fonctions :

* Un constructeur
* Redéfinition des fonctions Equals et ToString
* Emplacement
* Disponibilité

**Interface : Iresume**

1. Fonctionnalité :

* AfficheResume
* **Magasine** :

1. Les attribues :

* Catégories
* Liste d’article

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité
* **Manuscrit** :

1. Les attribues :

* Auteur
* Langue originale

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité
* **Livres :**

1. Les attribues :

* Auteur
* Nombre de pages

1. Les fonctions

* Un constructeur
* Redéfinition des fonctions Equals , ToString, Emplacement et Disponibilité