# Projet d’analyse et de conception : Bibliothèque

## Descriptif :

La bibliothèque permet de gérer des livres, qui les empruntent, qui les réservent et qui les ramènent. Elle a également comme fonction d’avertir un utilisateur lorsqu’un livre est disponible ou lorsqu’un livre a été réservé. S’il est disponible, la bibliothèque averti l’utilisateur et/ou les autres bibliothèques qui l’avaient réservé via MQTT et par email.

## Logiciel et technologies utilisés :

OS : Debian 11

Eclipse : IDE permettant de coder

JAVA : openjdk-17-jdk et openjdk-17-jre

DB : limariadb-java

mariadb-client

mariadb-server

odbc-mariadb

Hibernate : 5.5.6.Final

Sun mail : javax.mail 1.6.2

MQTT : pahomqtt 1.2.0

# La solution : explication de la réalisation du programme

Pour réaliser l’application, j’ai dû recourir à plusieurs technologies vues en cours comme Hibernate ou MQTT. Il m’a aussi fallu comprendre ce que je voulais faire du programme et qu’elle était ma vision de l’application, alors j’ai pris l’énoncé donné au cours, et j’ai commencer à interpréter la façon dont le programme était décrit de ma manière.

Pour ça, je me suis basé sur les cas d’utilisation d’écrit dans l’énoncé, à savoir :

Emprunter un livre

Ramener un livre

Réserver un livre

Avertir un lecteur qu’un livre est réservé par email

Avertir un lecteur ou une autre bibliothèque qu’un livre réservé est disponible via MQTT.

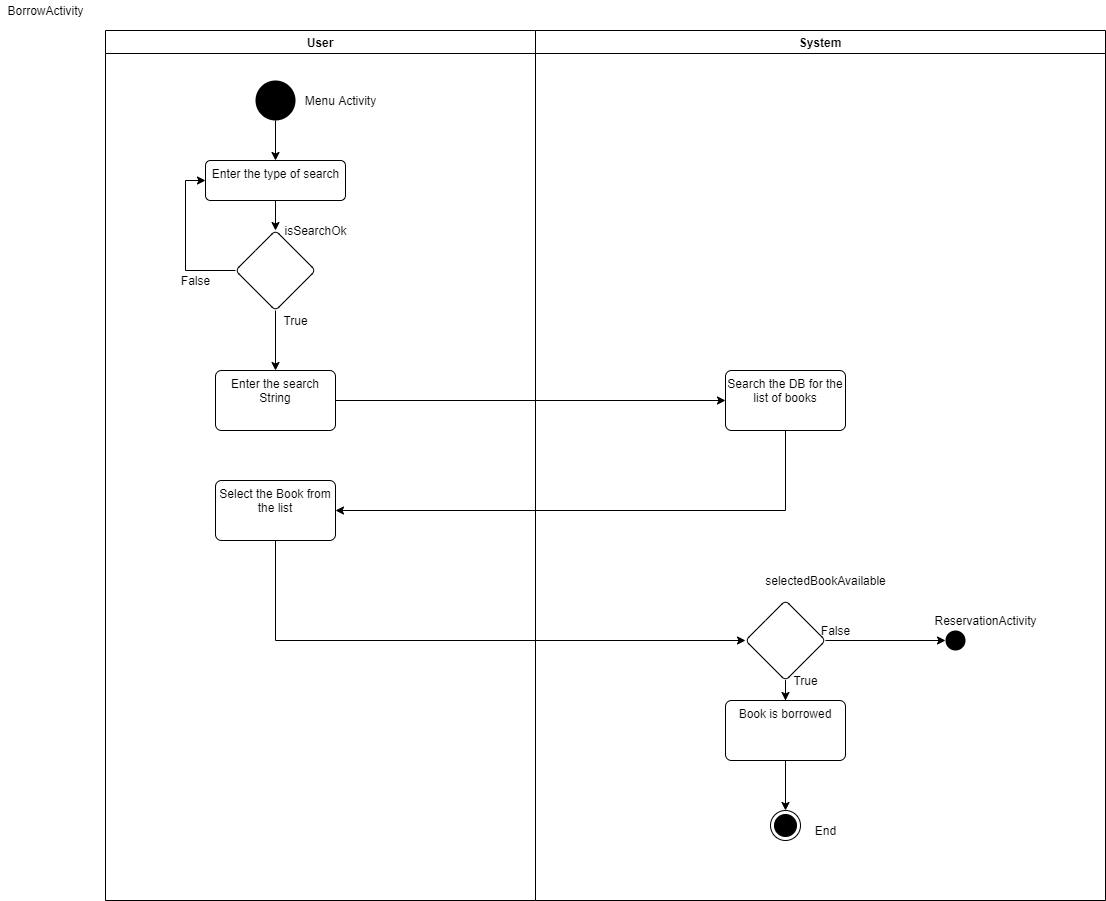
J’ai donc commencé à décrire chaque cas d’utilisation afin de pouvoir faire par la suite mon diagramme de classe, voici comment j’ai décrit mes cas d’utilisation :

D’une part, j’ai décomposé les cas en deux, ceux qui sont émis par la bibliothèque et ceux qui sont émis par le lecteur :

# Cas d’utilisation : Lecteur

# Lecteur :

## 1. Emprunter un livre

* **Le lecteur** se connecte à **la bibliothèque**
* **Le lecteur** sélectionne l’option de **recherche d’un livre**
* le lecteur saisit :
  + Le **titre** d’un **livre** OU
  + Le **nom** de **l’auteur** OU
  + **L’ISBN** du **livre** OU
  + **L’éditeur** d’un **livre**
* Le lecteur effectue la requête
* Le programme exécute la recherche
* Le programme affiche **le/les résultats** :
  + Liste des livres obtenu en fonction de la recherche
* Le lecteur choisit un **livre** parmi la liste
* Le programme affiche si celui-ci est disponible
* Le programme demande à l’utilisateur de :
  + Soit **d’emprunter** le livre
  + Soit De le **réserver** S’IL n’est plus disponible

## 2. Ramener un livre

* Le **lecteur** se connecte à la **bibliothèque**
* Le **lecteur** sélectionne l’option de **restitution d’un livre**
* Le programme cherche **la liste des livres empruntés**
* le programme affiche cette liste
* le lecteur choisit le **livre** qu’il veut rendre dans cette liste
* Le lecteur valide son choix
* Le programme valide la restitution d’un livre

## 3. Réserver un livre

* Le **lecteur** se connecte à la **bibliothèque**
* le **lecteur** sélectionne l’option de **réservation d’un livre**
* le **lecteur** saisit :
  + Le **titre** d’un **livre** OU
  + Le **nom** de **l’auteur** OU
  + **L’ISBN** du **livre** OU
  + **L’éditeur** d’un **livre**
* Le lecteur effectue la requête
* Le programme exécute la recherche
* Le programme affiche le/les résultats :
  + Liste des **livres** obtenu en fonction de la recherche
* Le lecteur choisit un **livre** parmi la liste
* Le programme réserve le **livre** et indique quand il sera **disponible**

# Cas d’utilisation : Bibliothèque

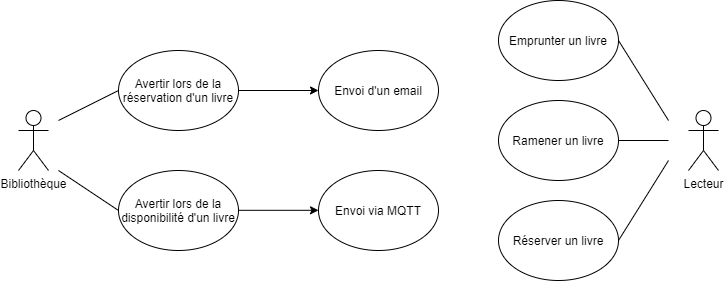
## 1. Avertir qu’un livre est réservé

* contrôle qu’une **réservation** a eu lieu
* envoi un **mail** au **lecteur** que le **livre** a été réservé

## 2. Avertir qu’un livre est disponible

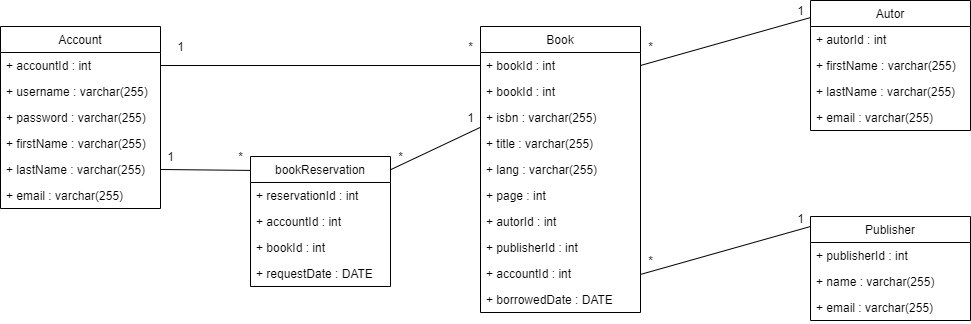
* une fois un **livre** ramener
* contrôle si ce l**ivre** est dans la **liste** des **réservations**
* si oui, averti le **lecteur** qu’il est disponible

## Schéma des cas d’utilisation :



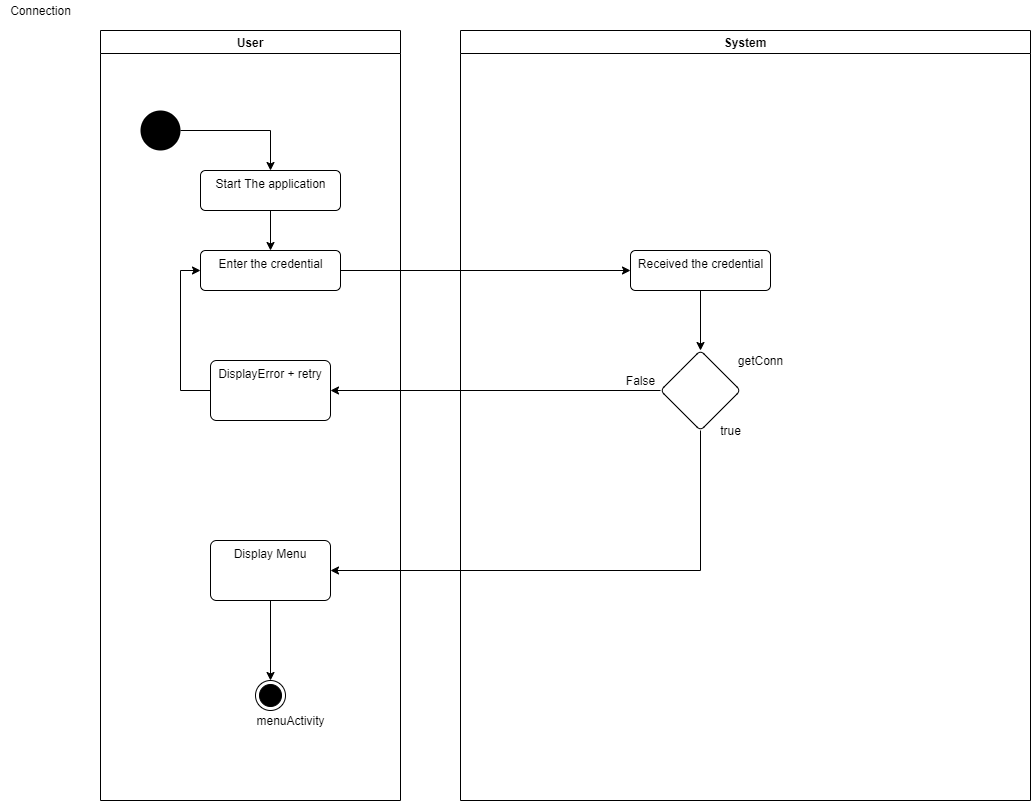
## Diagramme de classe :

À partir de toutes ses informations, j’ai commencé à regarder comment je pouvais construire la base de données et voici le schéma que ça a donné :

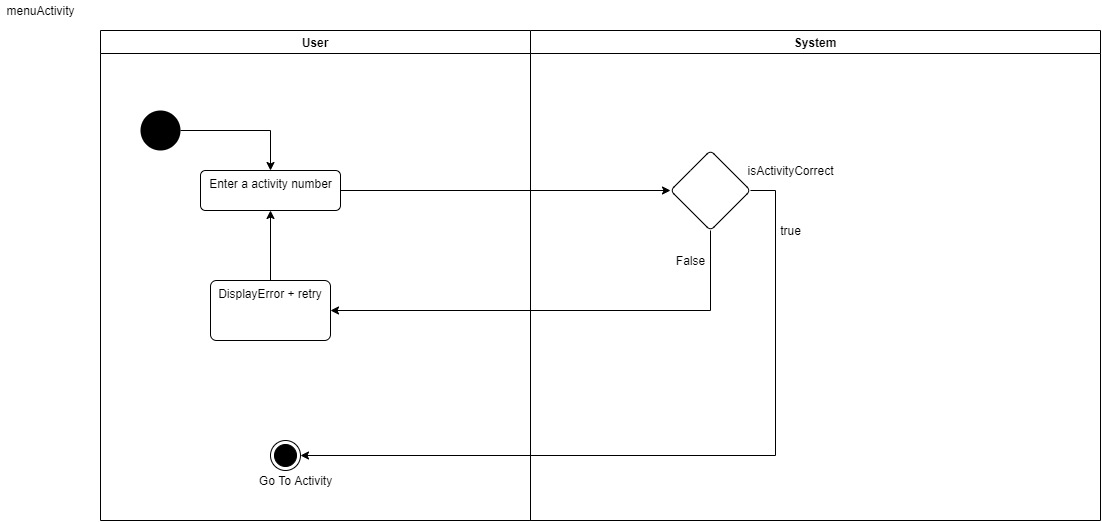


Une fois ces diagrammes fait, j’ai représenté ça avec des diagrammes d’activités :

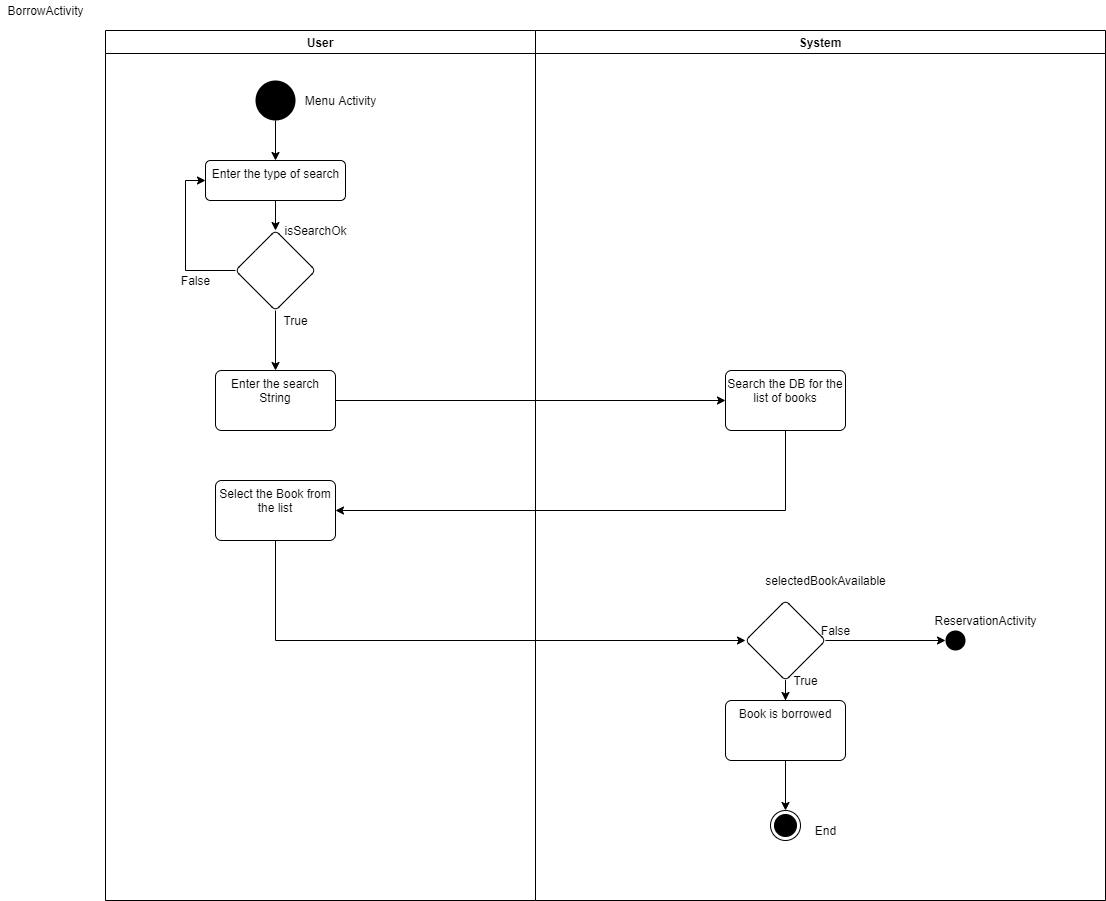
## 1.Démarrage de l’application :



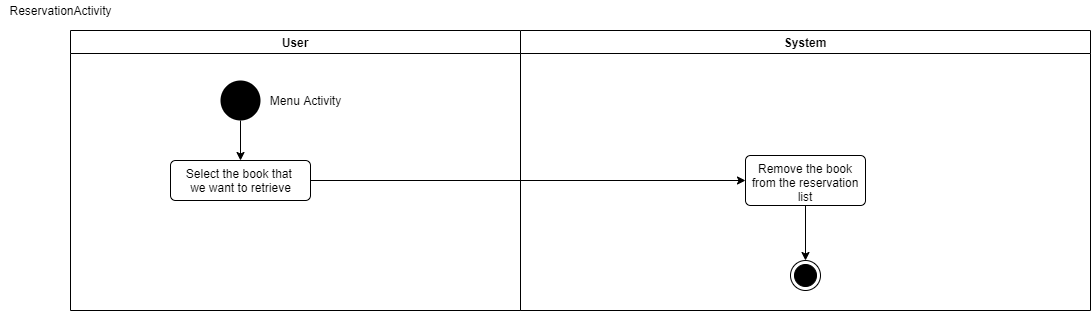
## 2.Sélection de l’activité



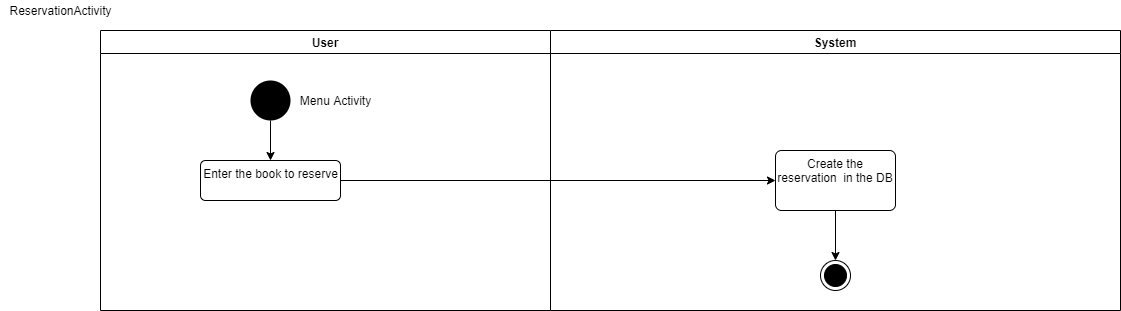
## 3.1 Emprunter un livre



## 3.2 Ramener un livre



## 3.3 Réserver un livre



# Final :

À partir de ça, j’ai essayé de développer l’application en usant de tout les outils vus au cours et en essayant de respecter au mieux le cahier de charges.