

# CLÔTURE DU TRAVAIL

## **Synthèse :**

Nous avons développé une application utilisant Java, simulant un système de réservation de vol.

Avant de nous lancer dans le développement de l'application, nous devions, dans un premier temps, avoir une bonne compréhension des concepts de réservation de vol et des différents aspects d'une application de vol. Pour ce faire, nous avons analysé des sites web de réservation de vol tels que *"Ryanair"*, *"Air France"*.

L'application fonctionne ainsi :

Il y a le client ainsi que l'admin. L'admin rentre les informations du vol (destination, aéroport de départ, prix du billet etc...). Puis le client peut réserver un vol, trier les réservations possibles par prix etc...

Ce projet était notre vrai premier projet en Java, il nous a permis de nous améliorer en programmation orienté objet, tout en faisant une application qui nous plaisait.

## **Les problèmes rencontrés :**

Nous avons eu diverses difficultés avant de commencer à développer notre application.

En effet, dans un premier temps, nous n'avions que très peu d'idées sur le thème de notre application, sur ce qu'elle devait faire.

Une fois ce problème réglé, nous devions faire l'étude de conception. Nous avons mis du temps avant de trouver un modèle de conception qui correspondait avec le thème choisi.

Le développement commençait, puis vint les premières difficultés, notamment la connexion à une "base de données", afin de mémoriser le choix de réservation des clients. Nous avons trouvé une solution qui était de stocker dans une hashmap lors de la création d'un compte, le nom, prénom, ainsi que numéro de téléphone du client. S'il souhaite se reconnecter, il doit taper ses informations. Cependant, il y a une chance infime que deux personnes aient le même numéro de téléphone et la même identité.

Pendant que l'on développe il est important de faire des tests unitaires afin de s'assurer que notre application fonctionne normalement. C'était très difficile puisque nous n'en avons quasiment jamais fait. Il fallait s'acclimater avec la bibliothèque JUnit.

Dans la consigne, il était indiqué que nous devions utiliser au moins un algorithme de tri.

Il fallait réfléchir pour savoir quel algorithme de tri ainsi que comment le développer. Nous avons opté pour un quicksort ainsi qu'un mergesort afin de trier les réservations de vol disponibles par prix.

Notre application ne nous permettait pas de développer de thread.

Enfin, il y avait quelques difficultés de logique dans l'application, par exemple la date de départ d'un avion doit être antérieure à celle d'atterrissage.

### **Les mesures d'amélioration :**

Notre application n'a pas d'interface graphique. Nous souhaitons à la base en faire une mais nous avons un gros manque de temps pour la faire. C'est pour cela que pour améliorer notre application il faudrait que l'on fasse une interface graphique en utilisant FXML.

Nous n'avons pas de vraie base de données reliée à notre application étant donné que les données sont stockées dans des listes ainsi que des hashmaps.

Notre application a énormément de redondance. Il faudrait trouver un moyen de l'éliminer.

Il faudrait faire en sorte que lorsque l'administrateur crée une nouvelle réservation, la ville de départ doit être automatiquement la même que la ville de départ choisie pour la réservation.

Un point important est l'ajout de fonctionnalités pour le client ainsi que l'admin.

Dans le futur, nous pouvons implémenter la vérification de la disponibilité ou non de chaque réservation ainsi que la vérification de l'entrée du client avant de créer de nouveaux objets.

Nous pouvons aussi améliorer nos tests qui ne sont pas parfaitement corrects afin d'avoir une application plus robuste, dont on est sûr qu'elle fonctionne.

### **Écarts avec les prévisions :**

Le projet de base devait contenir une interface graphique comme je l'ai dit plus haut mais manque de temps. De plus, nous voulions ajouter une API de météo afin de voir sur l'application si le temps est propice pour un vol ou non. De plus, il devait y avoir une base de données que l'on aurait reliée à notre application grâce au JPA mais c'était trop compliqué.