

UNIVERSITÉ MOHAMMED V  
ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET  
D'ANALYSE DES SYSTÈMES

RAPPORT DE PROJET JAVA EE  
GÉNIE LOGICIEL

---

## Comparateur de Vol - Fligght

---

*Elèves :*

BOUGTIB HAFSA  
BOUCHRIT HAMZA  
CHICHI AMINE  
ABDELJALIL HEGGOR  
FAKHRI MOUAD

*Encadrant :*

M. Mahmoud EL HAMLAOUI

Année Universitaire : 2020-2021



## **Remerciements**

Avant tout développement sur ce projet, nous souhaitons adresser mes remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à la réalisation de ce stage . Nous ne pourrions commencer ce rapport sans présenter nos remerciements les plus loyales à :

**Mr. Hamlaoui** , qui nous a encadré et accompagné tout au long de mon travail avec beaucoup de patience et de pédagogie, et qui n'a cessé de nous guider.

Nous n'oublions pas nos **membres de team** pour leur contribution, leur soutien et leur patience.

## Résumé

Le présent document constitue le rapport de notre travail dans le cadre du Projet JEE, dont l'objectif est le développement d'un "**Comparateur de Vols**".

Ce projet a comme objectif de permettre la comparaison des offres de vols des compagnies aériennes pour présenter aux utilisateurs les meilleures offres et possibilités de flight qu'ils cherchent selon leurs besoins et conditions horaires et financières .

Afin de le réaliser nous sommes passés par plusieurs phases : une première phase d'étude du contexte théorique et aspect technique du sujet, une deuxième phase d'analyse et de conception dans laquelle on a pu tester les différentes approches permettant de concevoir et résoudre le problème, puis une troisième phase de finalisation du projet et de présentation des résultats. Nous présenterons donc tout au long de ce rapport, les étapes que nous avons suivis ainsi que les outils que nous avons utilisé pour réaliser ce projet.

Mots Clés :

JEE - API - Tests - Comparaison

## Abstract

This document is the report of our work on the JEE Project, whose objective is the development of a **Flight Comparator**.

The main objectif of this project is to enable the comparison of airline flight offers, to present users with the best flight deals and opportunities they seek according to their needs and schedule and financial conditions.

In order to achieve this, we went through several phases : a first phase of study of the theoretical context and technical aspects of the subject, a second phase of design in which we were able to test the different approaches to solving the problem, then a third phase of project finalisation and presentation of results. We will therefore present throughout this report, the steps we followed as well as the tools we used to carry out this project.

Keywords :

JEE - API - Tests - Comparaison

## Liste des abréviations

Abréviation	Signification
JEE	Java Entreprise Edition
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Javascript
UML	Unified Modeling Language

# Table des figures

2.1	Diagramme de cas d'utilisation . . . . .	13
2.2	Diagramme de classe . . . . .	14
3.1	Logo Java EE . . . . .	15
3.2	Logo Glassfish . . . . .	16
3.3	Logo HTML . . . . .	16
3.4	Logo CSS . . . . .	16
3.5	Logo JavaScript . . . . .	17
3.6	Logo Docker . . . . .	17
3.7	Logo Maven . . . . .	18
3.8	Git . . . . .	18
3.9	Google Meet . . . . .	18
3.10	Page Home . . . . .	19
3.11	About us . . . . .	19
3.12	Contact Us . . . . .	20
3.13	Search for flights . . . . .	20

# Table des matières

<b>Introduction Générale</b>	<b>8</b>
<b>1 Présentation de la plateforme Fligght</b>	<b>10</b>
1 Introduction . . . . .	10
2 Présentation de "Fligght" . . . . .	10
3 Conclusion . . . . .	11
<b>2 Analyse et Conception</b>	<b>12</b>
1 Introduction . . . . .	12
2 Besoin fonctionnels . . . . .	12
3 Besoin non fonctionnels . . . . .	12
3.1 Fiabilité : . . . . .	12
3.2 Les erreurs : . . . . .	13
3.3 Ergonomie et bonne interface : . . . . .	13
3.4 Sécurité : . . . . .	13
3.5 Aptitude à la maintenance et la réalisation : . . . . .	13
4 Diagramme de cas d'utilisation . . . . .	13
5 Diagramme de classe . . . . .	14
6 Conclusion . . . . .	14
<b>3 Réalisation</b>	<b>15</b>
1 Introduction . . . . .	15
2 Outils et technologies . . . . .	15
2.1 Langage de programmation . . . . .	15
2.2 Environnement de développement . . . . .	17
2.3 Gestion de version et de collaboration . . . . .	18
2.4 Outils de Communication . . . . .	18
3 Intérfaces . . . . .	19
<b>Conclusion</b>	<b>19</b>
<b>Bibliography</b>	<b>22</b>

# Introduction Générale

L'idée du projet est basé sur une plateforme de comparaison des vols des compagnies aériennes . En effet , ceci va regrouper plusieurs compagnies aériennes pour permettre à l'utilisateur d'avoir la meilleure option pour son vol , dépendamment des détails qu'il souhaite.

Ceci va être basé sur une comparaison des offres de ces compagnies aériennes , et lier par la suite ce client vers le site de cette dernière selon l'offre qu'il a choisi .

Notre document a pour objectif de présenter ce projet réalisé en équipe . Les phases de déroulement de ce projet sont structurées de la façon suivante :

Le **premier chapitre** donne une partie présentation de "Fligght" .

Le **deuxième chapitre** présente la partie Conception .

Le **troisième chapitre** est consacré à la réalisation de l'application , ainsi qu'une description de la démarche suivie pour la conduite du projet.

Enfin, la conclusion générale présente une récapitulation des principaux résultats obtenus et les perspectives du projet.



# Chapitre 1

## Présentation de la plateforme Fligght

### 1 Introduction

Ce chapitre introductif a pour objet de décrire la plateforme et son processus principal de comparaison.

### 2 Présentation de "Fligght"

La plateforme est basée sur une api pour pouvoir accéder aux informations nécessaires sur les vols et les afficher au utilisateur. Cette affichage est basé sur les filtres que ce dernier a choisi pour le vol désiré .

D'un point de vue utilisateur , ce dernier saisie en accédant à la plateforme toutes les informations qu'il souhaite a propos de son vol, pour que par la suite , toutes les vols qui sont convenables avec le résultat de recherche et comparaison s'affichent.

Ensuite , parmi ces résultats ce dernier présente le meilleur offre qui soit convenable avec ses critères , pour pouvoir lui lier à l'offre détaillée qui pourra par la suite la réserver sur le site officiel de la compagnie . Fligght inclue tous les vols que l'utilisateur cherchera , selon les possibilités horaires et budgétaires qu'il pourra chercher, tout dépendamment des offres annoncées par les compagnies responsables .

### **3 Conclusion**

A propos du cycle de l'application , on peut dire que la tache déclenchante c'est celle de la saisie des informations du vol , puis recherche et comparaison , pour avoir les résultats qui lient l'utilisateur aux sites officiels des compagnies pour la réservation . Notre plateforme est censée en production inclure plus qu'une compagnie aérienne , mais pour ce projet on se limitera à une seule .

# Chapitre 2

# Analyse et Conception

## 1 Introduction

Le but de ce chapitre est de faire un cadrage technique et fonctionnel de notre projet. Nous allons présenter la démarche de conception suivie, les besoins fonctionnels et non fonctionnels de notre projet, ainsi les diagrammes de cas d'utilisation réalisés.

## 2 Besoin fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du système.

Afin d'obtenir une vue globale sur les exigences de l'application et ainsi réussir une bonne spécification des besoins, ces derniers doivent être modélisés. Nous avons donc eu recours aux concepts d'UML (Unified Modeling Language) pour représenter l'architecture et le fonctionnement de différentes fonctionnalités grâce à un ensemble de diagrammes très explicites.

## 3 Besoin non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerter les contraintes d'implémentation. Ils agissent de façon indirecte sur le résultat et sur le rendement de l'application, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

### 3.1 Fiabilité :

La plateforme doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

### **3.2 Les erreurs :**

Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre application.

### **3.3 Ergonomie et bonne interface :**

L'application doit être adaptée à l'utilisateur dans toutes les résolution d'écran sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mises en textes utilisés.

### **3.4 Sécurité :**

Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans ces plateformes.

### **3.5 Aptitude à la maintenance et la réalisation :**

Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

## **4 Diagramme de cas d'utilisation**

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer ses exigences. Ils décrivent aussi les fonctions générales et la portée d'un système.

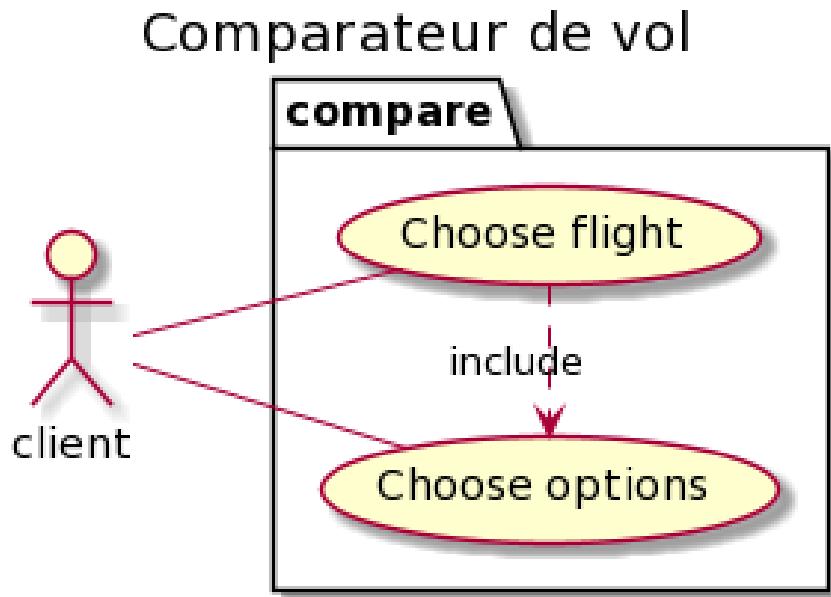


FIGURE 2.1 – Diagramme de cas d'utilisation

## 5 Diagramme de classe

Le diagramme de classes représente les classes intervenant dans le système. Le diagramme de classes est une représentation statique des éléments qui composent un système et de leurs relations

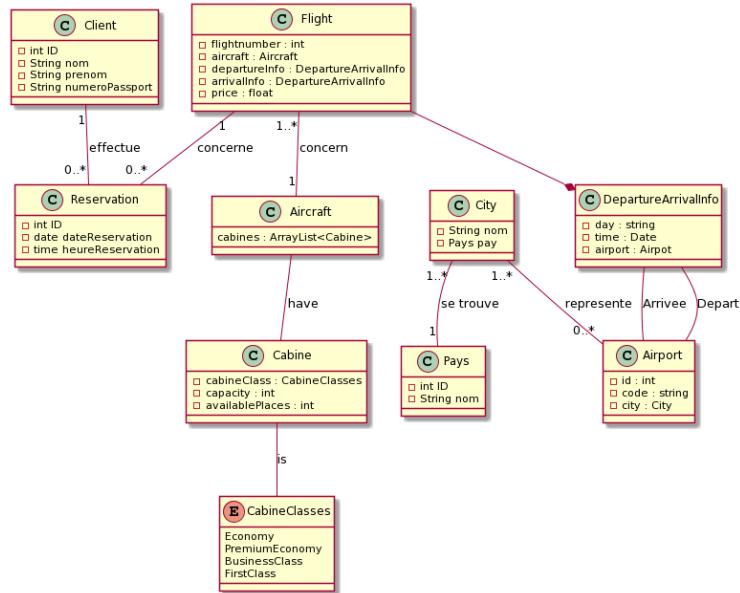


FIGURE 2.2 – Diagramme de classe

## 6 Conclusion

L’analyse fonctionnelle est une démarche qui consiste à rechercher et à caractériser les fonctions offertes par un produit pour satisfaire les besoins de son utilisateur. Ce chapitre nous a permis de bien décrire le comportement de notre projet du point de vue de l’utilisateur en étudiant les besoins fonctionnels et non fonctionnels ainsi les diagrammes Classes et Cas d’utilisation.

# Chapitre 3

## Réalisation

### 1 Introduction

Cette partie constitue le dernier volet de ce rapport. Après avoir terminé la phase de l'analyse et la conception, nous présentons les outils et l'environnement logiciel de développement utilisés ainsi que les différents captures du projet.

### 2 Outils et technologies

#### 2.1 Langage de programmation

Java Platform, Enterprise Edition ou Java EE, est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise. La plate-forme étend Java Platform, Standard Edition (Java SE) en fournissant une API de mapping objet-relationnel, des architectures distribuées et multitiers, et des services web<sup>3</sup>. La plate-forme se fonde principalement sur des composants modulaires exécutés sur un serveur d'applications.



FIGURE 3.1 – Logo Java EE

GlassFish est un projet de serveur d'applications de plateforme open source Jakarta EE lancé par Sun Microsystems, alors sponsorisé par Oracle Corporation, et résidant désormais à la Fondation Eclipse et soutenu par Payara, Oracle et Red Hat. La version prise en charge sous Oracle s'appelait Oracle GlassFish Server



FIGURE 3.2 – Logo Glassfish

HTML est un langage dit de « marquage », de « structuration » ou de « balisage » dont le rôle est de formaliser l’écriture d’un document avec des balises de formatage. Les balises permettent d’indiquer la façon dont doit être présenté le document et les liens qu’il établit avec d’autres documents.



FIGURE 3.3 – Logo HTML

Les feuilles de style en cascade, généralement appelées CSS de l’anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années.



FIGURE 3.4 – Logo CSS

JavaScript est un langage de programmation qui permet d’implémenter des mécanismes complexes sur une page web. À chaque fois qu’une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus

vidéo défilants, etc... — JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web, les deux premières (HTML et CSS).

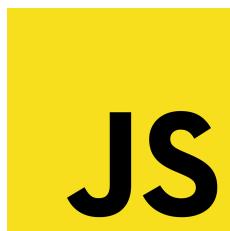


FIGURE 3.5 – Logo JavaScript

## 2.2 Environnement de développement

Docker est un ensemble de produits de plate-forme en tant que service (PaaS) qui utilisent la virtualisation au niveau du système d'exploitation pour fournir des logiciels dans des packages appelés conteneurs. Les conteneurs sont isolés les uns des autres et regroupent leurs propres logiciels, bibliothèques et fichiers de configuration ; ils peuvent communiquer entre eux via des canaux bien définis. Étant donné que tous les conteneurs partagent les services d'un seul noyau de système d'exploitation, ils utilisent moins de ressources que les machines virtuelles.

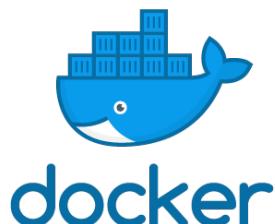


FIGURE 3.6 – Logo Docker

Maven est un puissant outil de gestion de projet basé sur POM (Project Object Model). Il est utilisé pour la construction de projets, les dépendances et la documentation. Cela simplifie le processus de construction comme ANT. Mais c'est trop avancé que ANT.

En bref, nous pouvons dire que maven est un outil qui peut être utilisé pour créer et gérer n'importe quel projet basé sur Java. maven facilite le travail quotidien des développeurs Java et aide généralement à la compréhension de tout projet basé sur Java.



FIGURE 3.7 – Logo Maven

### 2.3 Gestion de version et de collaboration

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvald, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.



FIGURE 3.8 – Git

### 2.4 Outils de Communication

Google Meet est un service de video conférence développé par Google. Il est utilisé au sein de l'entreprise Daba'Go pour effectuer des réunions avec des partenaires aussi bien une réunion pour tous les collaborateurs, elle est toujours programmée chaque vendredi après-midi pour se rencontrer tous et discuter les avancements dans chaque équipe.



FIGURE 3.9 – Google Meet

### 3 Intérfaces

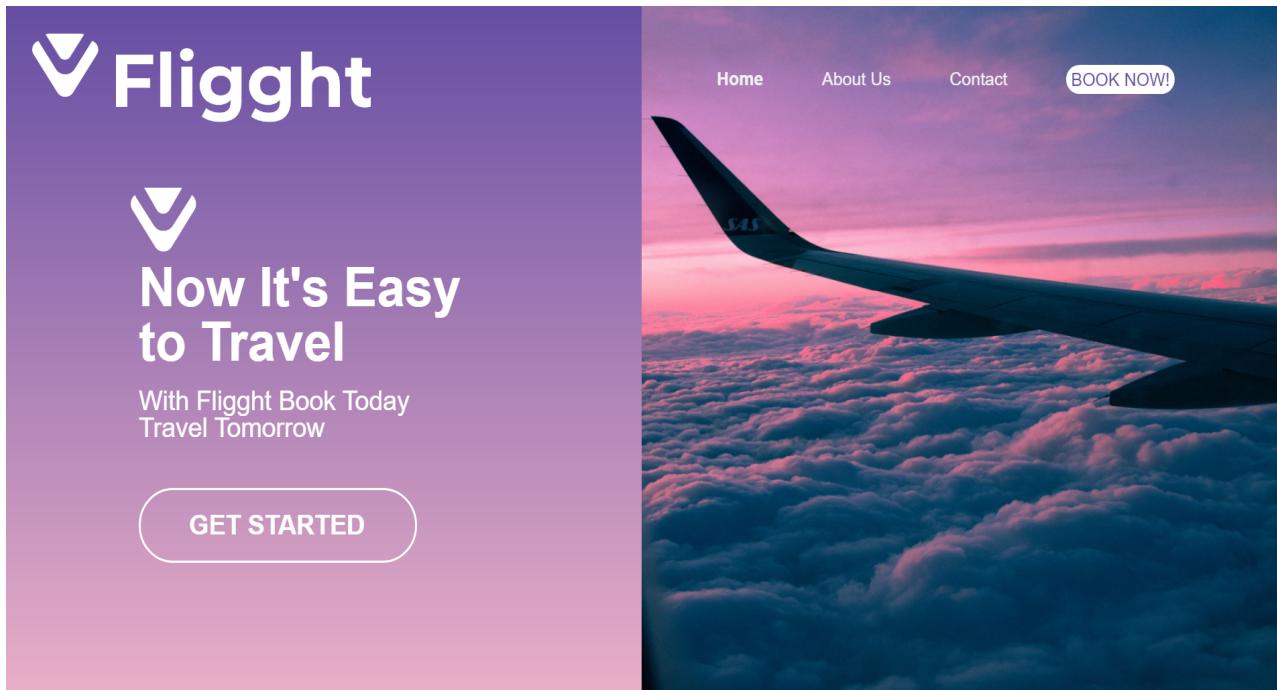


FIGURE 3.10 – Page Home

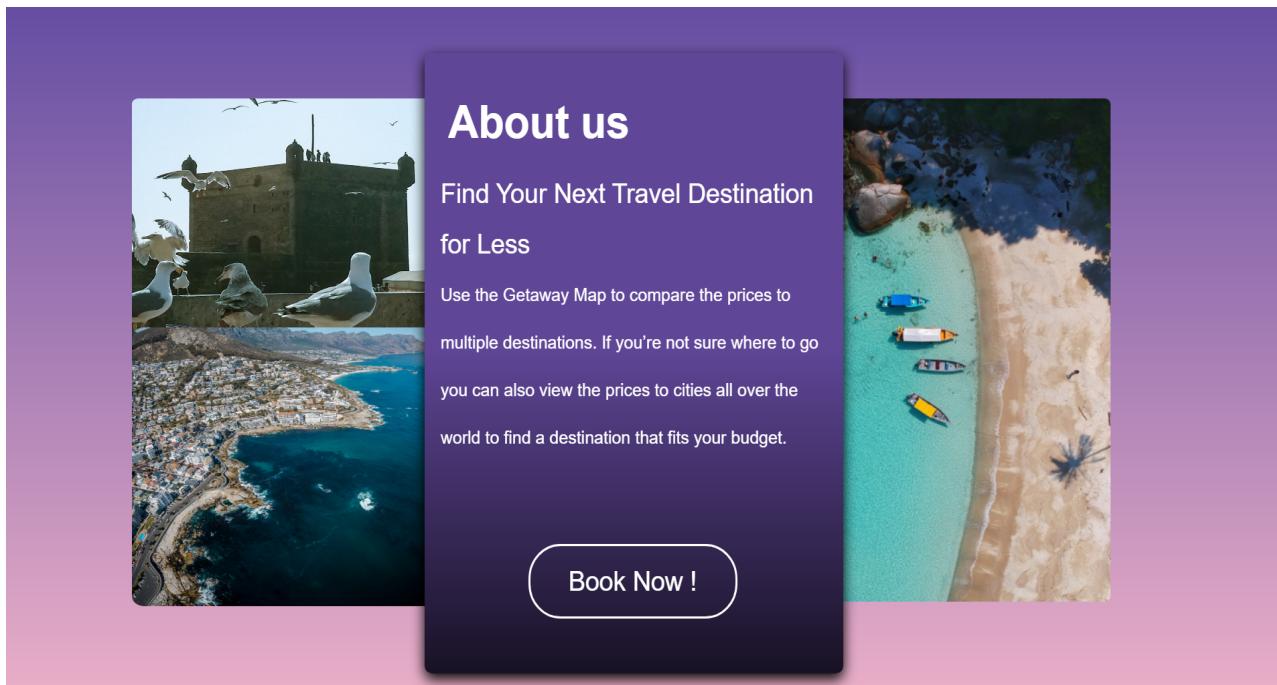


FIGURE 3.11 – About us

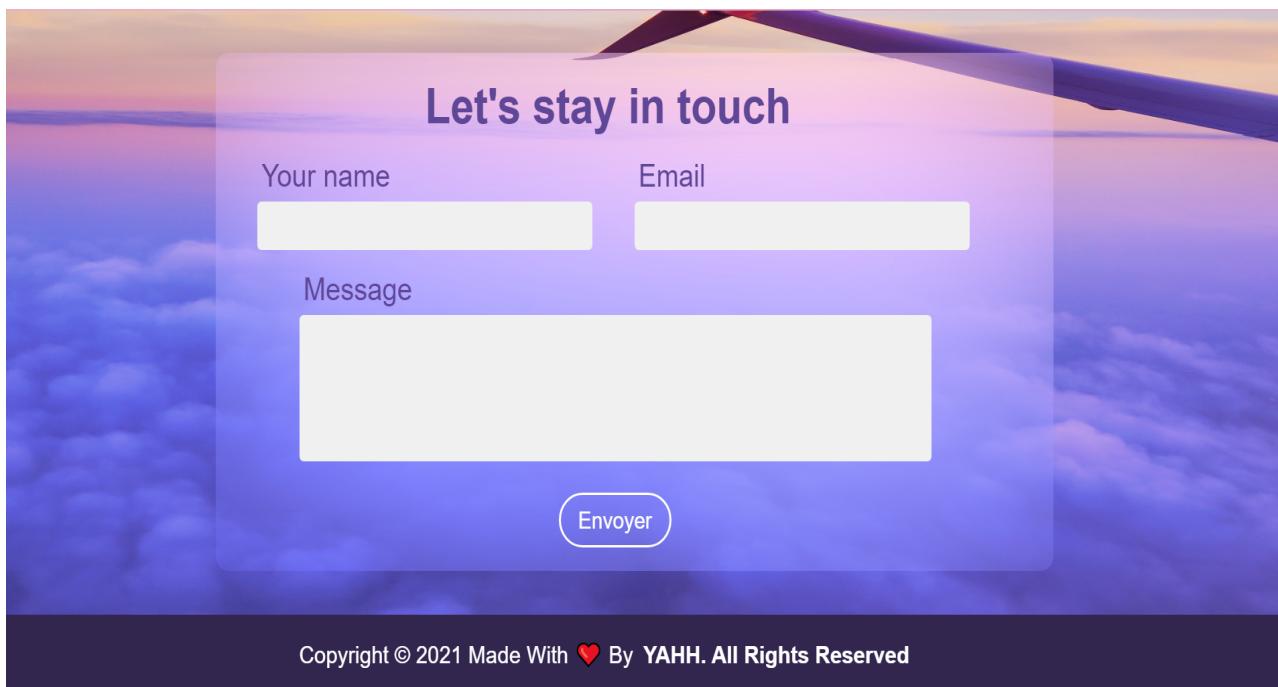


FIGURE 3.12 – Contact Us

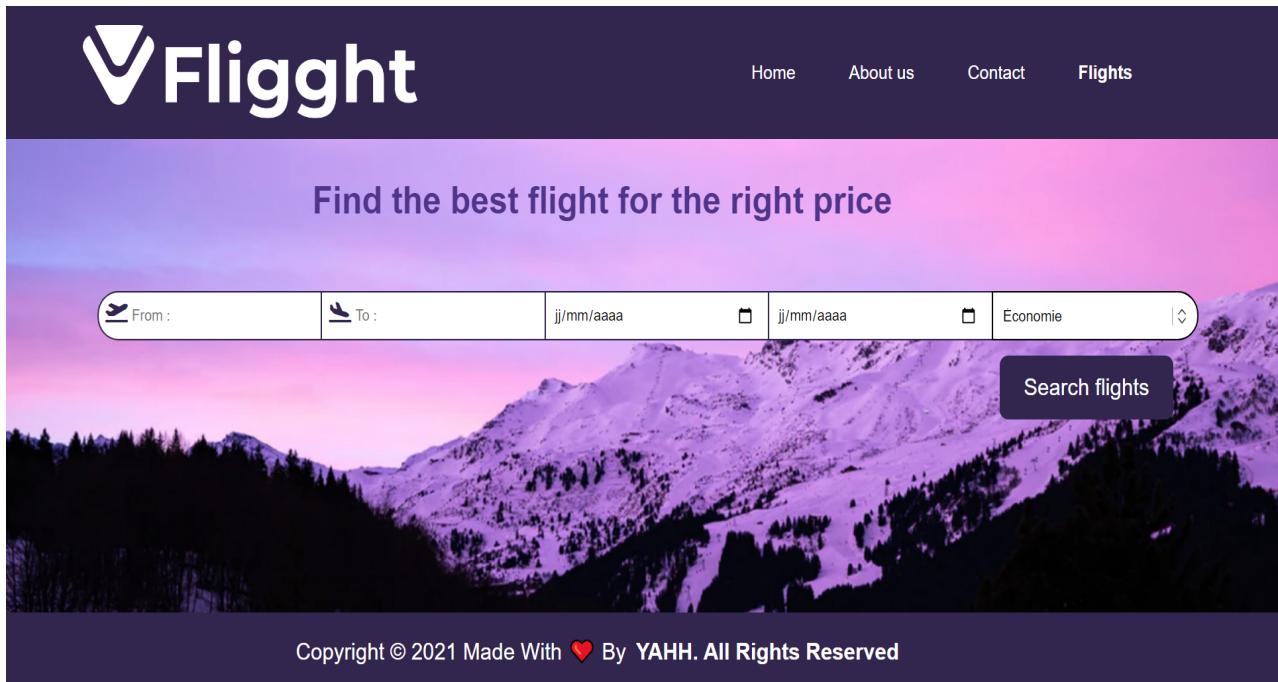


FIGURE 3.13 – Search for flights

# Conclusion

Tout au long de ce rapport, nous avons présenté la démarche que nous avons suivie pour mettre au point ce projet. Ainsi, l'analyse et la conception en étudiant les besoins et les scénarios de notre projet. Puis, la phase de développement qui nous a donné l'occasion de faire le lien entre les connaissances académiques, notamment la programmation Jee et découvrir de nouveaux outils tels que Docker, Junit .. etc

Ce projet nous a permis d'approfondir nos connaissances théoriques et pratiques notamment dans le domaine de développement des applications web. Ainsi, il a été aussi une belle occasion pour assurer un travail de groupe , même avec ces conditions non favorables.

Malgré les problèmes rencontrés durant la réalisation du stage notons le travail à distance comme étant l'obstacle majeur de notre projet puisque la mission était 100% en ligne et on n'a pas eu la possibilité de se rencontrer, nous avons pu gérer la situation et trouver des solutions citant par exemple les réunions de mise en point plusieurs fois par semaine ainsi pour se rencontrer tous et discuter les avancements dans chaque membre.

# Bibliographie

- [1] Documentation et Cours. <https://openclassrooms.com/fr/courses/626954-creez-votre-application-web-avec-java-ee>.
- [2] Tests et intégralité. <https://openclassrooms.com/fr/courses/6100311-testez-votre-code-java-pour-realiser-des-applications-de-qualite/6616481-decouvrez-les-tests-dintegration-et-les-tests-fonctionnels>.
- [3] Project. <https://github.com/>.
- [4] Problem solving. <https://stackoverflow.com/>.
- [5] Junit. <https://www.jmdoudoux.fr/java/dej/chap-junit.htm>.