



Rapport du projet d'Applications Web « My Travel »

Andrei PAKHMUTOV , Cyril Brandon TCHOFFO,
Thiago SOTORIVA LERMEN, Nihal BELMAKHFI

Département Sciences du Numérique - Filière HPC et Big data - 2A
UE Applications concurrentes et communicantes - Bases de données
2022-2023

1 Introduction

L'application web « My Travel » est une plateforme conviviale conçue pour offrir un moyen pratique et efficace aux utilisateurs de réserver des billets d'avion et de voyager vers différentes destinations dans le monde. L'application permet aux utilisateurs de créer un compte, de se connecter, de rechercher des vols disponibles, de générer les billets correspondants aux vols choisis, de consulter les détails de leur compte et de leurs billets.

Le développement de « My Travel » a impliqué l'utilisation de plusieurs technologies pour garantir une expérience utilisateur fluide et robuste. Java, associé à Java Persistence API (JPA), a servi de langage de programmation principal pour le développement du backend. Cette combinaison offre un cadre puissant et flexible pour la gestion de la persistance des données et de la base de données. Le backend a été intégré à JBoss, un serveur d'applications qui facilite le déploiement et la gestion des applications basées sur Java.

Pour le développement frontend, une combinaison de HTML, CSS, JavaScript et AngularJS a été utilisée. HTML et CSS ont été utilisés pour respectivement structurer et styler les pages web, en fournissant ainsi une interface utilisateur esthétique et intuitive. JavaScript a joué un rôle essentiel car il a assuré la communication et les interactions dynamiques entre les composants frontend et backend de l'application. Enfin, AngularJS - un framework JavaScript populaire, a fourni des fonctionnalités supplémentaires pour la création de pages web dynamiques.

2 Architecture du système

2.1 Architecture de haut niveau

L'application web « My Travel » utilise une architecture à trois niveaux, composée de l'interface utilisateur (UI), de la façade et de la base de données. Cette conception permet la modularité et la scalabilité, en séparant efficacement les préoccupations et en garantissant la maintenabilité.

2.2 Modèle client-serveur et interaction des composants

Conformément au modèle client-serveur, le client de l'application web (le navigateur web de l'utilisateur) communique avec le serveur hébergeant l'application. Le client envoie des requêtes HTTP au serveur, qui répond avec les données demandées ou effectue les actions souhaitées.

Les composants du système interagissent de la manière suivante :

Les utilisateurs interagissent avec le composant de l'interface utilisateur (UI), construit avec HTML, CSS et Bootstrap 5.0. L'UI offre une interface intuitive pour différentes fonctionnalités telles que la recherche de vols, la connexion et la création de compte, les détails de compte et la liste des billets associés à un compte donné.

Le composant UI se connecte au composant de la façade en transmettant des requêtes HTTP, principalement en utilisant AngularJS côté frontend. Ces requêtes déclenchent des méthodes correspondantes au sein de la façade.

Agissant en tant qu'intermédiaire entre l'UI et la base de données, le composant de la façade traite les requêtes HTTP de l'UI, exécute des méthodes POST et GET, et accède aux données ou les enregistre dans la base de données.

Le backend est développé en utilisant Java, en exploitant l'API Java Persistence (JPA) pour les opérations sur la base de données. Les entités de données sont représentées sous forme d'objets Java, facilitant la communication entre la façade et la base de données.

La base de données stocke les données relatives aux vols, aux avions, aux aéroports, aux informations sur les utilisateurs et aux billets. Lors du premier déploiement de l'application, la base de données est remplie d'informations provenant de fichiers JSON contenant des données de vol, d'avion et d'aéroport. La base de données est mise à jour à chaque création de billets ou d'informations utilisateur. Des requêtes SQL sont utilisées pour accéder aux tables de la base de données, correspondants aux entités créées en Java.

2.3 Rôle des technologies

L'intégration de ces technologies permet à l'application web « My Travel » d'offrir une expérience utilisateur fluide et intuitive, tout en garantissant une gestion efficace des données et une communication entre les différents composants du système. Les technologies sont listées ci-dessous.

1. Java : En tant que langage principal pour le développement du backend, Java permet la mise en œuvre de la logique métier et facilite la communication avec la base de données grâce à JPA.
2. JBoss : Ce serveur d'applications fournit une plateforme pour l'hébergement et la gestion de l'application web basée sur Java, garantissant sa disponibilité et sa scalabilité.
3. HTML, CSS et Bootstrap 5.0 : Ces technologies sont utilisées dans le développement de l'interface utilisateur pour créer des pages web visuellement attrayantes et réactives, améliorant l'expérience utilisateur.
4. JavaScript : JavaScript est crucial pour assurer la communication et les interactions dynamiques entre les composants frontend et backend de l'application.
5. AngularJS : AngularJS est utilisé pour créer des pages web dynamiques et interactives, permettant une communication fluide entre l'UI et les composants backend de l'application.

3 Conception de l'interface utilisateur

3.1 Importance de la conception de l'interface utilisateur

La conception de l'interface utilisateur (UI) joue un rôle essentiel dans les applications web, car elle a un impact direct sur l'expérience utilisateur. Une interface intuitive et visuellement attrayante améliore la convivialité, engage les utilisateurs et favorise les interactions positives. Il est essentiel de créer une interface facile à naviguer, visuellement cohérente et qui correspond à l'objectif et au public cible de l'application.

3.2 Principes de conception

Dans l'application web « My Travel », une attention particulière a été accordée à la création d'une interface conviviale et esthétiquement plaisante. Les principes de conception et les considérations suivants ont été pris en compte :

- Identité visuelle : Une identité visuelle unique a été établie pour « My Travel », comprenant le développement d'un logo distinctif. Cette identité visuelle est maintenue de manière cohérente sur toutes les pages, offrant une présence de marque cohésive et reconnaissable.
- Palette de couleurs : Une palette de couleurs bleu clair a été choisie pour l'application afin d'évoquer une sensation de tranquillité et de refléter l'expérience immersive des voyages. Les couleurs ont été soigneusement sélectionnées pour créer une interface harmonieuse et visuellement attrayante.
- Focus sur le bureau : L'accent principal de la conception de l'interface utilisateur était mis sur une utilisation sur ordinateur. La mise en page et les éléments de design ont été optimisés pour les écrans d'ordinateur, garantissant une expérience fluide et immersive pour les utilisateurs accédant à l'application depuis leur ordinateur.

3.3 Principales fonctionnalités de l'interface utilisateur

L'interface utilisateur de « My Travel » propose une variété de fonctionnalités visant à améliorer l'expérience utilisateur. Voici quelques éléments remarquables :

- Recherche de vols : L'interface comprend une fonction de recherche permettant aux utilisateurs d'entrer leurs destinations souhaitées, leurs dates de voyage et leurs préférences pour explorer les options de vols disponibles. Les utilisateurs peuvent filtrer et trier les résultats en fonction de différents critères.
- Gestion de compte : Les utilisateurs peuvent créer des comptes, se connecter et accéder à leurs informations personnelles. L'interface permet aux utilisateurs de consulter et de mettre à jour leurs coordonnées, y compris les informations personnelles et les préférences.
- Génération de billets : Après avoir réservé des billets d'avion, l'interface permet aux utilisateurs de générer et de consulter leurs billets. Les billets sont affichés dans un format attrayant, fournissant aux utilisateurs toutes les informations de voyage nécessaires.
- Immersion visuelle : L'interface présente des images de lieux célèbres et maintient une identité visuelle cohérente tout au long, dans le but d'immerger les utilisateurs dans l'expérience du voyage. La conception de l'interface évoque un sentiment de désir de voyager et d'excitation, capturant l'essence de l'exploration.

La conception de l'interface utilisateur de « My Travel » privilégie une expérience engageante et conviviale en incorporant des éléments visuellement attrayants, une navigation intuitive et une fonctionnalité fluide. En respectant les principes de conception et les considérations, l'interface vise à créer un parcours positif et immersif pour les utilisateurs lorsqu'ils réservent des billets d'avion et se lancent dans leurs aventures de voyage.

4 Développement Backend

4.1 Authentification des utilisateurs

Lorsqu'un utilisateur crée un compte ou se connecte, le backend vérifie les informations d'identification de l'utilisateur, telles que le nom d'utilisateur et le mot de passe.

4.2 Gestion des données de vol

Le backend gère les informations liées aux vols à l'aide d'entités Java. Des entités telles que Aircraft (avion), Airport (aéroport), Flight (vol), Reservation (réservation), Ticket (billet), Passenger (passager) et User (utilisateur) sont définies, chacune représentant une table dans la base de données. Ces entités possèdent des attributs qui correspondent aux colonnes de leurs tables respectives, et des relations sont établies à l'aide de clés primaires et étrangères.

4.3 Initialisation de la base de données

Le backend utilise les annotations @Singleton et @Startup dans la classe DatabaseInit pour initialiser la base de données lors du démarrage de l'application. Un gestionnaire d'entités spécifique est créé pour lire un fichier JSON contenant des données obtenues à partir de sources en ligne. Les entités Aircraft, Airport et Flight sont remplies en fonction de ces données, et les dates de vol (dates de départ) sont générées de manière aléatoire dans une plage spécifiée.

4.4 Classe Facade

Le backend comprend une classe Facade annotée @Singleton, servant de composant central responsable de la gestion métier et de la communication avec la base de données. La Facade reçoit des appels HTTP POST et GET depuis le frontend, permettant l'exécution de diverses opérations. Ces méthodes gèrent la récupération, la modification et le stockage des données, fournissant les fonctionnalités nécessaires pour l'interface utilisateur.

Le backend met en œuvre des méthodes POST pour gérer la soumission et la modification des données reçues depuis le frontend. Ces méthodes traitent les entrées de l'utilisateur, telles que les réservations de vols et les mises à jour de compte, et stockent les informations pertinentes dans la base de données. Les méthodes GET sont utilisées pour récupérer des données à partir de la base de données et fournir les informations demandées au frontend pour affichage.

5 Développement Frontend

5.1 Mécanisme de recherche

La page principale comprend un formulaire où les utilisateurs peuvent spécifier leurs préférences de vol, telles que aller-retour ou aller simple, les aéroports de départ et d'arrivée (pays) et les

dates du voyage. Des éléments interactifs tels que des menus déroulants et des sélecteurs de date améliorent l'utilisabilité du formulaire.

5.2 Authentification des utilisateurs

Les utilisateurs peuvent se connecter ou s'inscrire en utilisant leur adresse e-mail et leur mot de passe. Le frontend effectue des validations de formulaire pour s'assurer que les champs requis sont correctement remplis. Si le formulaire n'est pas correctement rempli ou si les informations de connexion ne sont pas authentifiées, des messages d'alerte sont affichés sur le frontend pour informer l'utilisateur. Lorsque l'utilisateur se connecte au système, il reste connecté en parcourant les pages. Cela est rendu possible par l'utilisation de cookies de navigateur qui maintiennent les informations de l'utilisateur connecté dans le frontend.

5.3 Liste et réservation des vols

Après qu'un utilisateur effectue une recherche de vol, une page de liste affiche les options de vol disponibles. Après avoir sélectionné un vol souhaité, les utilisateurs sont dirigés vers une page de réservation où ils remplissent un formulaire avec leurs informations personnelles, telles que leur nom, les détails de leur passeport et leurs préférences de vol. Les validations de formulaire garantissent l'exactitude et l'exhaustivité des informations fournies.

5.4 Gestion du compte

Les utilisateurs peuvent accéder aux informations de leur compte, y compris les détails personnels et les billets de vol réservés. Ils ont la possibilité de se déconnecter et de supprimer leur compte. Les interfaces conviviales et la navigation claire permettent une gestion facile du compte.

5.5 Génération de PDF

Lorsqu'un utilisateur réserve un vol, un PDF peut être automatiquement généré avec les informations de vol pertinentes. Cette fonctionnalité facilite l'expérience utilisateur en fournissant un support téléchargeable et imprimable des vols réservés.

6 Conclusion

Ce projet nous a permis de mettre en pratique la quasi totalité des notions apprises lors du cours magistral, et ce sur quelque chose d'aussi concret qu'un site web. Par ailleurs, la grande quantité de code et de fichiers, impliquée par les modalités du projet, a posé un défi d'organisation.

Dans l'ensemble nous sommes satisfaits de notre travail, car il constitue, à notre sens, un ensemble cohérent et assez ambitieux.

Enfin, en ce qui concerne les améliorations possibles de l'application, on pourrait implanter la possibilité de générer un fichier pdf avec les billets achetées par tous les clients. Nous en avons

d'ailleurs préparé le terrain, avec notamment l'introduction du concept d'admin, car cela faisait partie des objectifs initialement prévus. Mais nous avons finalement dû renoncer à cela pour des raisons de la charge globale de travail. En outre, notre application fonctionne bien de manière attendue sur Chrome, mais sur Firefox toutes les données ne s'affichent pas toujours. On pourrait donc également explorer et améliorer cet aspect compatibilité avec les navigateurs.