

Conception d'une application web de réservation

Laboratoire de technologie web

Christophe Simon
Guillaume de Moffarts

24 décembre 2017

Introduction

Dans le cadre du laboratoire de Technologie web, nous avons réaliser une application web dont le but est de réserver des billets d'avion. Cette application doit suivre le design pattern MVC (Model View Controler).

Nous avons choisis comme idée de réaliser une caricature de la compagnie aérienne Ryanair, que nous avons appeler BryanAir. Le style de notre application à été inspirer du site officiel de la compagnie.

Dans ce rapport, vous allez trouver des explications sur le fonctionnement normal et erroné de l'application. Vous y trouverez un diagramme de séquence pour imager cela.

1 Description du site

Il y a tout d'abord une page d'accueil (figure 1) qui permet de se rediriger vers une section pour effectuer la réservation d'un vol.

Sur la première page de réservation (figure 2), on peut réserver un ou plusieurs billets. Il faut donner l'aéroport de départ et d'arrivée, si on veut réserver un billet aller simple ou aller retour, le nombre de passager, l'adresse email et si on désire une assurance annulation ou pas.

La deuxième étape de réservation (figure 3) consiste a enregistrer les informations pour chaque passager tel que le nom, le prénom et l'âge.

Une fois tous les passagers enregistrés, on arrive sur une page de confirmation (figure 4) ou vous avez le choix de confirmer ou d'annuler.

FIGURE 1 – Page d'accueil

FIGURE 2 – Réservation étape 1

FIGURE 3 – Réservation étape 2

Une fois la réservation confirmée, on arrive sur une page récapitulant la commande (figure 5) avec le prix à payer.

Enfin, il y a une page d'administration (figure 6) où on peut voir, éditer ou supprimer les passagers pour chaque vol. Cette page ne devant être accessible du public, on y accède uniquement via `http://localhost/BryanAir/admin`.

2 Fonctionnement

Dans cette section, vous retrouvez les explications sur le fonctionnement des différentes parties de l'application, tel que la redirection d'URL, les réservations et la page d'administration et les templates HTML.

Redirection

La conception de l'application est basée sur le design pattern MVC. Nous avons donc un routeur, des contrôleurs et des vues. Le routeur redirige vers le contrôleur adéquat qui va vérifier les informations venant du formulaire, charger et enregistrer les modèles dans la base de données. Le contrôleur va ensuite afficher la page.

Lorsque qu'on veut accéder à une page, il suffit d'utiliser l'URL suivante `localhost/BryanAir/` suivi du nom de la page à accéder. Cette URL est redirigée vers `localhost/BryanAir?page=<nom_de_page>` car le routeur a besoin d'une variable `GET["page"]` pour pouvoir rediriger vers le bon contrôleur. Nous avons fait ce choix pour qu'il soit plus facile d'accéder aux différentes pages. Cette redirection est configurée dans le fichier `.htaccess`.

Templates

Pour faciliter la construction des vues, nous avons utilisé un système de templates HTML. Nous avons placé des balises comme suit `$balise$` dans nos pages HTML. Celle-ci vont être remplacées par le contrôleur. La fonction `buildHTML` dans le fichier `utils.php` construit d'abord la page en concaténant quatre fichiers HTML. Trois de ces fichiers sont communs pour toutes les vues, il y a le `head.html`, le `header.html` et le `footer.html`. Entre ces deux derniers, la vue propre à la page est ajoutée. On remplace ensuite toutes les balises, grâce à du regex, par des bouts d'HTML construits par le contrôleur. Cette fonction prend en paramètre

FIGURE 4 – Confirmation

FIGURE 5 – Récapitulatif

FIGURE 6 – Page admin

un liste de clef valeurs, les clefs correspond aux mots dans les balises de l'HTML et les valeur contienne donc le contenu a remplacer.

Cette méthode permet de faciliter grandement la création des vues. En effet, il ne sera pas nécessaire de copier dans tous les fichier html un bout de code commun a toute les pages. Cette méthodes permet aussi de bien séparer la partie vu de la partie controleur.

2.1 Réservation

Tous d'abord, en cliquant sur le bouton réservation de la page d'accueil, le controleur (controle_reservation.php) charge la liste de tous les aéroport présent dans la base de données. Le contrôleur va ensuite contruire la page de réservation grace a cette liste.

Lorsque l'utilisateur valide la première page de réservation (figure 2), un contrôleur (controleur_detail.php) vérifie tout d'abord les données venant du POST. En effet si on passe bien par le formulaire, il ne devrais pas avoir de problème mais on peut tout a fait utilisé un outil tel que curl pour envoyer n'importe quel requête POST au serveur ou tout simplement en changeant l'html dans l'inspecteur du navigateur, on peut envoyer du text au lieu d'un entier. Il faut donc bien vérifier que toutes les données du POST nécessaire au contrôleur on bien été définies et sont de type correcte. Par exemple, la variable contenant le nombre de passager doit pouvoir être convertit en un entier. Si un problème survient dans la vérification des variables , vous etes redirigez vers la page précédentes avec un message d'erreur sur le haut de celle ci (figure 7. Si la vérification a réussie, Le controleur cherche dans la base de donnée les numéros de vols (aller/retour) et le nombre de place restantes. Si il n'y a plus de place pour un des vols ou qu'il ne trouve pas de vol pour les aéroport sélectionner, on aura un message d'erreur comme décrit précédement. Toutes ces informations sont utilisées pour pour instanttier un objet de type **Reservation** et en définir l'état. Le controleur va ensuite construire la page HTML (detail.html) grace a la fonction expliqué dans les section précédente.

Nous somme maintenant a la deuxième étapes de la réservations (figure 3) où il faut encoder un a un les informations sur les passagers. Comme pour la première étapes, un contrôleurs (controleur_nextpassenger.php) vérifie les informations de la variable HTTP POST, le même système d'erreur est aussi d'application. Une fois la vérification passée, le contrôleur instancie un objet de type **Client** dont l'état est définit pas les informations entrées dans le formulaire. Cet objet client est ensuite ajouter dans la liste de client de l'objet réservation. Si il y a encore des passager a enregistrer, le contrôleur contruit la même page html que précédemment pour enregistrer un nouveau passager. Si tous les passager sont enregistré, le contrôleur construit

FIGURE 7 – Message d'erreur

FIGURE 8 – Erreur de majorité

la page de confirmation (confirmation.html).

Une fois arriver sur la page de confirmation (figure 4), on a le choix entre confirmer la réservation ou de l'annuler. Si la réservation est annulée, on est redirigé vers la page d'accueil et les l'objet réservation avec la liste des client est supprimé. Si on confirme la réservation, un contrôleur (controleur_confirmation.php) vérifie si au moins un des passagers est bien majeurs. Si ce n'est pas le cas, le système d'erreur affiche un message (figure 8), il faudra ré-encoder les informations des passagers. Si il y a bien au moins une personne majeur, le contrôleur enregistre les clients et réservations dans la basse de données et va ensuite nous rediriger vers une page résumant la commande(figure 5).

Le contrôleur (controleur_resumer.php) permettant d'afficher cette page vas tout d'abord construire un tableau html, avec tout les passagers de la réservation, grâce a l'objet réservation. Cet objet se charge aussi de calculer le prix total a payer en se basant sur le nombre de passager, si il on prix un vol aller retour ou aller simple et enfin, si une assurance annulation est prise. Le contrôleur vas ensuite construire la page avec toute ces information.

2.2 Admin

En allant sur la page admin, le contrôleur (controler_admin.php) charge depuis la base de donnée tout les vols ainsi que les passager pour chacun d'eux. Il en fait un tableau html représentant toute ces donnée puis construit la page html (admin.html).

Conclusion