

PROJET WEB

AirAdvisor



Table des matières

Présentation du site.....	2
Base de données.....	2
Une base de données cohérente.....	3
Les différents niveaux de privilèges des utilisateurs.....	4
La sécurité sur le site.....	4
Les mails automatiques.....	5
L'utilité des \$_GET.....	5
Avec Ajax, on garde le contact.....	6
Mais trop de contact tue le contact.....	6
Les fichiers images uploadés.....	7
A propos de AirAdvisor.....	8
AirAdvisor est facilement configurable.....	8
AirAdvisor est un site valide aux normes HTML5/CSS3.....	8
Le template XHTML/CSS.....	8
CONCLUSION.....	9

Présentation du site

AirAdvisor est un site d'avis en ligne à l'instar de TripAdvisor.com ou encore FIA-NET centré sur le transport aérien. Le but étant de présenter la popularité des aéroports et des compagnies aériennes à l'aide d'avis, de critiques ou de témoignages de nos inscrits. Son équivalent qui s'en rapproche le plus, car je me suis aperçu bien après que AirAdvisor n'est pas une invention, est le site whattheflight.com (bien plus complet, propose aussi de réserver les billets d'avions directement en ligne ainsi que pleins d'autres fonctionnalités plus ou moins utiles).

AirAdvisor est un site de génération 2.0. Son contenu ne dépend pas d'une ou un groupe de personnes mais de ses membres qui ont la possibilité de s'exprimer. Ainsi les internautes qui postent des messages sur un aéroport par exemple, lui attribue automatiquement une note sur 10. On peut alors par ce moyen établir un classement avec les meilleurs aéroports ou compagnies choisis par la communauté du site.

AirAdvisorDB : La base de données du site

Je voulais faire depuis longtemps une base de données référençant tout les aéroports du monde. Mais je ne maîtrisais pas assez bien les outils voir même ne les connaissais pas du tout. Ce semestre m'a beaucoup appris et aidé à réaliser mon idée. J'ai commencé donc tout en premier par concevoir la base de données de la base. La conception s'est faite à la suite de l'élaboration du cahier des charges. Il me fallait une table qui contiendrait mes aéroports, une autre les compagnies aériennes, une pour les utilisateurs et une pour les messages. Puis voyant que les compagnies, aéroports et utilisateurs ont tous une nationalité, j'ai fait une table avec la liste de tout les pays et le code pays comme clé primaire. Les clés primaires n'ont pas été difficiles à identifier : un aéroport a un code OACI unique tout comme une compagnie. Pour la clé primaire des messages, j'ai choisi le pseudo et la date de publication qui est en fait un timestamp précise à la seconde. Ainsi on considère qu'un utilisateur ne peut pas publier plus de un message au même instant précis (c'est logique). Ce choix de clé primaire permet aussi d'éviter l'insertion de doublons accidentels dans la base de données.

Je me suis documenté sur l'organisation administrative des aéroports et compagnies avant de concevoir AirAdvisorDB et voir aussi comment je pourrai relier les tables entre elles. Par exemple une compagnie a une plate-forme de correspondance principale (appelée base). Ce qui permet d'avoir une clé étrangère faisant le lien entre compagnies et aéroports. Les clés étrangères permettent d'avoir une navigation facile et efficace sur le site. Elles sont pour la plupart utilisés pour la création d'hyperliens.

La table messages est un peu particulière. C'est une table associative liant les utilisateurs et les compagnies mais aussi les aéroports. Cette table comprend au total trois clés étrangères : le pseudo permettant de lier la table utilisateur et le code OACI de l'aéroport OU le code OACI de la compagnie. Si un utilisateur poste par exemple un message au sujet d'une compagnie, ce message aura comme valeur pour ses clés étrangères le pseudo de l'auteur, le code OACI de la compagnie correspondante et NULL pour le code OACI de l'aéroport. Un schéma UML pour expliquer tout cela, vous comprendrez certainement mieux si ce n'était pas très explicite :

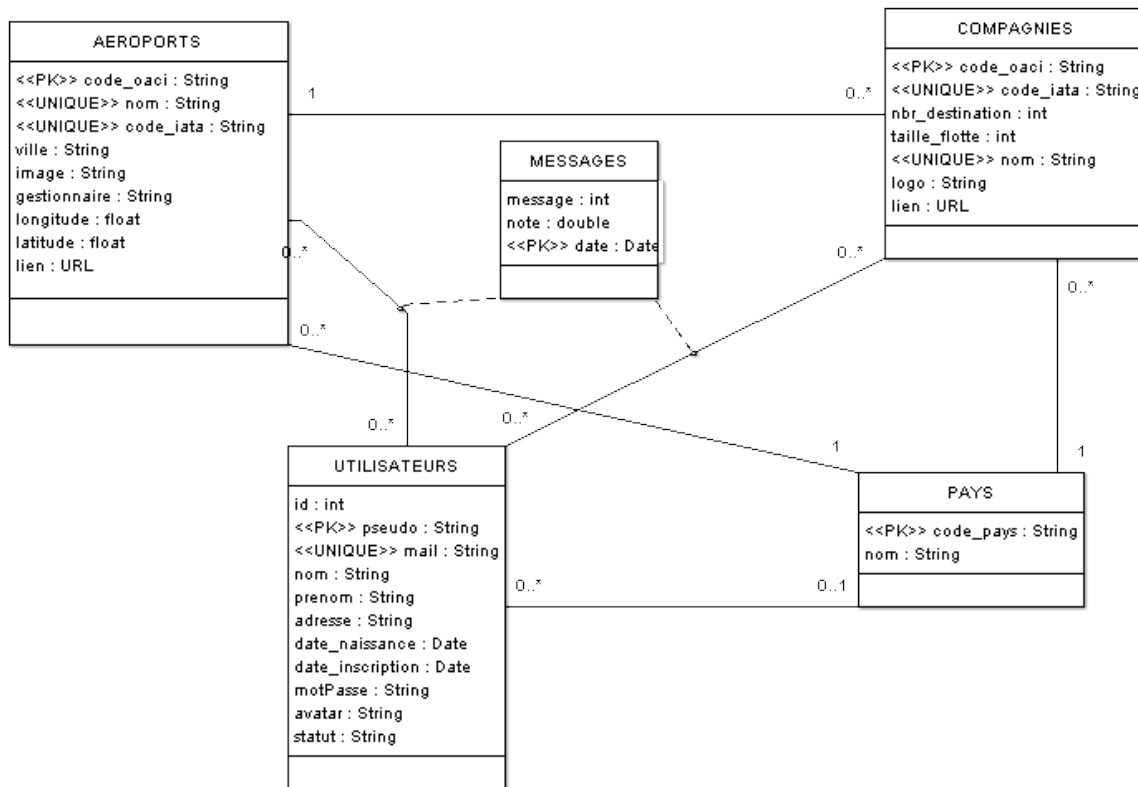


Illustration 1 - Modélisation UML de AirAdvisorDB

Une base de données cohérente

Des « on update cascade/on delete cascade » sont présents et permettent d'avoir une base de données qui se met à jour lors d'un changement de clé primaire sur un tuple de donnée ou de la suppression d'un tuple. Exemple : si on change le code OACI d'un aéroport particulier, tout les messages concernant cet aéroport se verront leur clé étrangère changé par le nouveau code. Ceci évite à avoir des requêtes à faire sur le développement PHP et des incohérences dans la base.

Les différents niveaux de privilèges des utilisateurs

Il existe 4 types de visiteurs sur AirAdvisor qui ont pour chaque type des droits plus ou moins importants :

- les anonymes
- les membres
- les modérateurs
- les administrateurs

Le statut anonyme est défini par défaut. On est anonyme si on est pas inscrit sur le site ou tout simplement pas authentifié. Les anonymes ont juste les droits de lecture. Ils peuvent consulter les informations recensées sur les aéroports et compagnies, lire les messages des membres, voir le classement.

Les membres ont en plus des droits de lecture, le droit de commenter et de supprimer ses propres messages. Ils peuvent consulter et modifier leur profil.

Les modérateurs ont en plus des droits des membres, le droit de supprimer n'importe quel message posté même si il n'en est pas l'auteur.

Les administrateurs ont tout les droits cités plus haut. Ils peuvent en plus accéder à l'espace administration du site. On peut y insérer, supprimer, modifier des aéroports et compagnies. L'espace donne aussi la possibilité de supprimer un utilisateur, de lui envoyer un message personnel ou de lui changer de statut.

La sécurité sur le site

Les mots de passe de nos inscrits sont cryptés dans la base de données avec l'algorithme de hachage Whirlpool. Une fonction très puissante équivalente à SHA-512 laissant des empreintes codés sur 512 bits. Les mots de passe choisis doivent au moins faire 8 caractères.

Le site donne l'impossibilité d'accéder aux fonctionnalités si on a pas les droits nécessaires. Ceci par un jeu de redirection et de conditions exécutées avant l'envoi de la page ou de formulaires par le serveur.

L'injection de code HTML ou JavaScript est difficile pour l'internaute grâce à l'emploi de fonctions tel que htmlentities. Les guillemets sont acceptées en tant que texte dans la base de données grâce à la fonction addslashes permettant ainsi d'éviter toute injection SQL et aussi de publier des informations comprenant ce caractère sans que ce ne soit tronqué.

Les mails automatiques

AirAdvisor envoie des mails dits « automatiques » à la suite de diverses actions. Ses actions sont :

- quand un internaute (pas forcément membre) envoie un message depuis le formulaire de contact (peut envoyer aussi une copie à l'expéditeur si il le désire)
- quand un internaute vient de s'inscrire. AirAdvisor lui souhaite la bienvenue et lui rappelle ses identifiants de connexion (mail et mot de passe) qu'il a demandé
- quand un inscrit change de mot de passe. AirAdvisor lui transmet son nouveau mot de passe. Ce mail est une mesure de prévention au piratage en même temps d'être un message de confirmation
- quand un inscrit change d'adresse mail. AirAdvisor lui envoie dans ce cas là deux e-mails : une à l'ancienne, l'autre à la nouvelle. Ce double envoi est aussi par mesure de prévention au piratage.
- quand un inscrit ne se souvient plus de son mot de passe. AirAdvisor lui envoie un nouveau mot de passe de 8 caractères générés pseudo-aléatoirement.
- quand un inscrit vient de supprimer son profil. AirAdvisor lui dit au revoir et espère le revoir bientôt.
- quand un administrateur envoie un message à un inscrit. AirAdvisor fait la transmission du message par mail.
- quand un administrateur change le statut à un inscrit. AirAdvisor lui indique que ses droits ont changé.
- quand un administrateur a supprimé un inscrit. AirAdvisor le tient au courant de cette suppression.

Des redirections sont présentes afin d'éviter le renvoi de formulaire par le rafraîchissement de la page et avoir des mails intempestifs. On a tout de même des messages de confirmation. L'indication du statut est envoyé via les variables `$_GET`.

L'utilité des `$_GET`

J'ai trouvé beaucoup d'utilité aux variables `$_GET`. C'est en effet une information que l'on peut envoyer simplement par un hyperlien ce qui facilite énormément le développement. De plus, je pense aussi au référencement. L'information étant placée dans l'URL, on peut facilement accéder au contenu voulu

sans passer par des formulaires de recherche ou autres pages. On peut donc être facilement cité dans un forum, sur un site ou sur un moteur de recherche. Et il y a comme cité plus haut, lors d'une redirection évitant le renvoi de formulaire, avoir la possibilité d'afficher un message de confirmation.

Avec Ajax, on garde le contact

Du JavaScript est présent à divers endroits afin de donner un certain dynamisme et une certaine attractivité au site. Mais surtout l'emploi de la technologie AJAX dans le site tient à jour le client sur les nouveautés. Le classement en page d'accueil est mis à jour toutes les secondes grâce à Ajax. Et si vous êtes sur la page d'un aéroport ou d'une compagnie, il rafraîchit de manière dynamique toutes les 5 secondes les messages. On peut presque tchatter en ligne. La recherche est aussi effectuée de manière dynamique et filtrée au fur et à mesure de la saisie de l'internaute dans la barre de texte. Je n'ai pas utilisé le plugin autocomplete de jQueryUI pour le faire. L'ajout de commentaire se fait aussi avec AJAX. Si l'utilisateur aurait désactivé JavaScript de son navigateur, le message est inséré de manière standard avec rechargement de la page. Sur le formulaire d'inscription et de profil, il donne directement la disponibilité du pseudonyme.

Mais trop de contact tue le contact

AJAX n'est pas du tout la meilleure solution pour mettre à jour les informations. L'inconvénient est que c'est le client qui demande constamment au serveur toutes les X secondes. Du coup si il y a beaucoup d'utilisateurs en ligne, le serveur peut vite saturer. De l'autre côté le client télécharge des données depuis le serveur et fait croître la quantité de données transférées au fur et à mesure qui reste sur la page. Vraiment pas idéal si on a une connexion limitée en Mo. Pire, on pourra avoir une facture lourde si on l'utilise régulièrement sur un portable équipé d'un forfait non bloqué. Pour la solution, je pense à l'utilisation de Node.js qui permet de faire du JavaScript côté serveur. En effet, cette fois le client ne demande plus et c'est le serveur qui envoie de l'information dès qu'il y a du nouveau. C'est donc une solution beaucoup plus réactive et surtout plus économe en bits transférés. J'ai consulté très brièvement le cours sur le site openclassrooms.com (anciennement site du zéro) et j'ai vu qu'avec la bibliothèque websocket.io, il est possible de le faire.

Les fichiers images uploadés

Les images uploadées sont soit un logo pour une compagnie aérienne, soit une photo pour un aéroport ou soit un avatar pour l'utilisateur. Si il n'y a pas de photos, images et avatars, une image par défaut est mise à l'affichage. Les images uploadées sont placées dans des dossiers adéquats toujours sous la forme `cle_primaire.extension`. La taille du fichier ne doit pas excéder 2 Mo pour l'avatar et 10 Mo pour l'aéroport et la compagnie. Dans le cas où la taille n'est pas respectée, un message d'erreur est affiché. De même, l'extension doit être obligatoirement du `.png`, `.jpg/.jpeg` et `.gif`. Des conditions en PHP sont exécutées pour vérifier le format.

Pour ce faire, j'ai créé un objet PHP de la classe `SplFileInfo` avec comme attribut les données du fichier uploadé contenues dans la variable `$_FILES`. Cet objet a une méthode très pratique qui est `.getExtension()` et qui permet de récupérer l'extension.

Dans le cas d'un changement d'image, on suit cette procédure :

- je vérifie si le nouveau fichier correspond aux conditions (taille et format)
- je fais la mise à jour dans la base de données. Le nom du fichier sera le même mais pas forcément l'extension. C'est pourquoi la mise à jour est nécessaire
- je supprime l'ancien fichier avec la fonction PHP `unlink`
- je déplace le nouveau fichier au même endroit que l'ancien avec `move_uploaded_file`

Noter que le déplacement seul ne suffit pas. Il est obligatoire de supprimer d'abord l'ancien fichier. En effet, les extensions peuvent être différentes et du coup deux fichiers différents (même nom mais pas même format) pourraient être stockés dans le serveur.

A propos de AirAdvisor

AirAdvisor est facilement configurable

J'ai eu l'idée de faire un peu comme on trouve dans les CMS, un fichier de configuration « configuration.php ». Dans le cas où on migrerait AirAdvisor sur un autre serveur, on aura qu'un seul fichier à modifier. On change simplement les valeurs des variables qui ont un nom assez explicite par les valeurs propres à la configuration de la base de données et du serveur. Ces variables sont ensuite utilisées dans les autres fichiers .php du site. Le fait de n'avoir qu'un seul fichier à modifier rend la configuration plus simple.

AirAdvisor est un site valide aux normes HTML5/CSS3

Toutes les pages ont été validées avec succès par le validateur en ligne de W3C. AirAdvisor n'est en revanche pas responsive mais reste adaptable sur de nombreux écrans même tablette. Pour les écrans de petite taille comme smartphone, le site n'est pas agréable à naviguer.

Le template XHTML/CSS

La maquette graphique utilisée est une maquette graphique libre de droit téléchargée sur le site de son auteur, Luka Cvrk, www.solucija.com. Je le félicite de ses créations qui sont variés et sympas et le remercie de les mettre à la disposition de la communauté du web.

CONCLUSION

Je suis content de moi sur la réalisation de ce site qui me tenait à cœur. Cela faisait un moment que je voulais un site associé à une base de donnée pour pouvoir faire de l'affichage d'information notamment de l'information sur les aéroports. AirAdvisor référence en plus les compagnies et donne la possibilité de commenter. C'est mon tout premier site 2.0.

Comme dit en page d'index, je compte améliorer encore le site, continuer à remplir ma base de données d'aéroports et de compagnies et intégrer quand j'aurai plus le temps Node.js.

AirAdvisor est disponible en ligne à cette adresse :

<http://airadvisor.totodu.net/>