

Présentation du contexte

1) Domaines fonctionnels

Nous allons illustrer l'analyse, la conception et le développement d'une application Web, en réalisant une partie d'un site Web existant : celui de la compagnie aérienne française : Air France : <http://www.airfrance.fr/>

Nous n'aurons malheureusement pas le temps en cours de tout finir.

Mais tous les Etudiants / Stagiaires qui souhaitent continuer le projet sont encouragés à le faire. Je les aiderai par mes conseils et par des réponses à leurs questions.

2) Cahier des charges

2.1) Définition des besoins

Vous allez observer le site Web d'Air France et observer plus particulièrement ceci :

- ➔ L'entête ou les entêtes qui sont communs à toutes les pages,
- ➔ Le pied ou les pieds de pages qui sont communs à toutes les pages,
 - Comment consulte-t-on **les horaires des vols** : pages, données saisies, affichées, contrôles de données,....
 - Comment **achète-t-on un billet** : pages, données saisies, affichées, contrôles de données,....
 - Comment **s'enregistrer et obtenir la carte d'embarquement** : pages, données saisies, affichées, contrôles de données,....

2.2) Architecture de la future application

L'**application Web** sera complétée par une **application Mobile** qui offre les mêmes fonctionnalités,

Une **application fenêtrée** (sous Windows ou Unix) sera réservée au personnel d'Air France et utilisera les mêmes données et les mêmes classes que l'application Web.

Ces applications seront **construites au fur et à mesure** (on dit par itérations) : on analyse, on conçoit, on code et on livre une partie de l'application et on recommence avec une autre partie,etc

Donc **nous avons intérêt** à bien penser l'architecture de notre application Web afin qu'elle :

- ➔ Partage les accès aux données : couche **Modèles**,
- ➔ Soit facile à maintenir (erreurs d'exécution et nouvelles fonctionnalités) : séparer en deux couches : **Vues** et **Contrôles**
- ➔

2.3) Outils à utiliser

Concepteur Développeur Informatique

Projet Cours : Application Web AirFrance

- ➔ Pour modéliser, nous utiliserons l'outil « **Visual Paradigm** » qui permet de faire les diagrammes UML,
- ➔ Pour développer les applications, nous utiliserons :
 - cible « Monde Windows » : **l'IDE Visual Studio avec C#**,
 - Cible « Monde Open Source » : **l'IDE Eclipse avec Java**,
- ➔ Pour stocker les données, nous utiliserons une Base de données relationnelle :
 - Soit SQL Server 2012 ou ultérieure,
 - Soit MySql,
 - Soit Oracle 11g ou ultérieure,
 - Autre SGBDR,
 - Notre application permettra :
 - De basculer facilement d'un SGBDR à un autre,
 - De changer le type de middleware utilisé pour accéder à la BD : « JDBC », « JPA »,

3) Travail à faire

3.1) dossier parent à rendre

Créez un dossier sur votre ordinateur et nommez le « **ProjetAirFrance_PromoX_201504** »
Ou X est le numéro de votre promo.

Tous vos travaux seront créés dans dossiers fils de ce dossier parent.

3.2) itération « Consulter les horaires »

- ➔ Analyse et conception :
 - Créez un dossier « **AnalyseEtConception** » dans le dossier parent,
 - Créez, dans Visual Paradigm (VP), un projet nommé « **prjvp_AirFrance** »,
 - Conception :
 - Réfléchissez et dessinez sur du brouillon le diagramme de use cases qui représente les 3 besoins cités plus haut. Validez avec le professeur.
 - Créez ce diagramme dans le projet sous VP, sous le nom « **duc_AirFrance** »,
 - Réfléchissez et dessinez sur brouillon un diagramme de classes pour représenter les données et les traitements du besoin « **consulter les horaires** ». Validez avec le professeur.
 - Créez ce diagramme dans le projet sous VP, sous le nom « **dc_AirFrance** »,
 - Réfléchissez et dessinez sur brouillon un MCD à partir du diagramme de classes pour représenter les données sous forme de tables SQL avec identifiants, relations entre tables. Validez avec le professeur,
 - Réfléchissez et dessinez sur brouillon une architecture MVC pour représenter la réalisation de ce besoin « consulter les vols ». Validez avec le professeur,
 - Traduisez l'archi. MVC en un diagramme de séquences pour représenter les pages, les classes modèles, les servlets et leur communication pour représenter le code à fabriquer pour consulter les horaires,

Concepteur Développeur Informatique

Projet Cours : Application Web AirFrance

→ Codage :

- Créez un script SQL, sous SQL Server 2012, nommé «**script_Creation_BDAirFrance**» :
 - Supprime la BD si elle existe,
 - Crée les tables dans l'ordre avec uniquement les clés primaires sans les clés étrangères,
 - Ajoute les clés étrangères aux différentes clés,
 - Insère des lignes dans les tables pour fabriquer un jeu de test (s'inspirer des vraies données trouvées sur le site Air France),
- Les classes modèles seront utilisées dans plusieurs applications (Web, Mobile,),
- Créez un projet nommé « **BiblioClassesAirFrance** de type Plug-in dans Eclipse ou de type bibliothèque de classes dans Visual Studio :
 - Ajoutez la classe « **ParametresBD** » et configurez-la selon votre serveur et votre BD,
 - Créez dedans, les différentes classes du diagramme de classes et codez les méthodes nécessaires pour le besoin actuel,
 - Générez le plug-in (une archive Java) ou la bibliothèque de classe (fichier .dll),
- Testez l'utilisation du plug-in / dll dans un projet de type console avec méthode main :
 - Testez l'appel à chaque méthode et testez les résultats reçus
- Créez le Projet Web nommé «**WebAirFrance_nomPrenom**» dans Eclipse ou Visual Studio (remplacez nom et prénom par les vôtres),
 - Pour les styles **CSS3** de mise en forme : fabriquez des fichiers externes aux pages,
 - Si vous utilisez du **JavaScript** ou **JQuery** dans vos pages : fabriquez des fichiers externes aux pages,
 - Créez des **entêtes** et **les pieds de pages** communs aux autres pages (selon technologie utilisée : Faire des inclusions de fragments ou fabriquer des modèles de pages)
 - Fabriquez vos pages nommées : **page_xxxxxx** ou xxxxx doit être un nom ou quelques mots qui reflètent ce que fait la page. Page de Page_1, Page_2, !!!!
La première page de l'application doit être nommée « **index** » et doit être publique, les autres pages doivent être privées (non accessibles par leur url à partir d'un navigateur extérieur)
- Testez cette itération de l'application. Vérifiez bien que toute la mécanique en dessous respecte bien :
 - Ce que vous avez fait dans les différents diagrammes et schémas,
 - Sinon : corrigez.
- Faites une démo au professeur et si tout va bien : passez à l'itération suivante.