

Projet AWBD

23 mars 2015

1 Objectif

Ce projet vise à construire une application Web de gestion simplifiée des réservations pour un Théâtre. Cette application permet, par exemple, de consulter les spectacles à l’affiche et le programme des représentations associées, d’effectuer des réservations, d’ajouter des représentations, etc.

L’application suit une architecture multi-tiers constituée des niveaux suivants :

- un premier niveau web permet aux utilisateurs de l’application d’accéder aux diverses fonctionnalités fournies via des requêtes web,
- un second niveau données permet de stocker, de manière persistante et cohérente, les informations relatives à l’application.

1.1 Technologies

Les technologies utilisées sont les suivantes :

- les données sont stockées dans une base de données gérée par Oracle
- la programmation côté serveur sera réalisée en Java. Elle respectera le modèle d’architecture Modèle Vue Contrôle. L’accès aux données se fera à travers un pattern DAO. Le déploiement sera réalisé sur un serveur Tomcat.
- le contrôle de la présentation reposera sur l’utilisation des CSS

2 Présentation des informations

Un directeur de théâtre désire informatiser son système de réservation de places pour les spectacles qui se déroulent dans son théâtre au cours d’une même saison (de septembre à juin). A la fin d’une saison, le contenu de la base de données est archivé et vidé. Pour les besoins du projet les données de l’application décrites ci-après ont été simplifiées.

Les spectacles.

Un spectacle est identifié par un numéro unique et on connaît son nom. Un spectacle fait généralement l’objet de plusieurs représentations dans des salles à des moments différents. Une représentation d’un

spectacle dans une salle débute à un moment donné exprimé à la granularité de l'heure (par exemple 28/11/2007 20H).

La salle. Une salle regroupe un ensemble de rang qui lui même regroupe un ensemble de places. Un rang possède un numéro de rang unique par salle. Une place est identifiée par un numéro de rang, et un numéro de place dans le rang (par exemple pour une chaque salle nous avons les rangs de 1 à 10 et chaque rang possède les places numérotées de 1 à 20) Un rang est associé à une catégorie tarifaire (orchestre, balcon, poulailler, etc). Toutes les places du même rang sont dans la même catégorie. Le tarif de base associé à chaque catégorie est fixé pour l'ensemble de tous les spectacles.

Les utilisateurs. On distingue le responsable de la programmation et les clients. Le responsable programme les spectacles. Le responsable et les clients sont identifiés par leur *login*. Ils sont décrits par leur nom, leur prénom, leur adresse électronique (facultatif) et leur mot de passe.

Les ventes. Une personne peut acheter n places ($n \geq 1$) pour une représentation d'un spectacle donné (pour rappel représentation étant le déroulement du spectacle dans une salle à une certaine date et heure). Tout achat de place(s) se traduit par la création d'un dossier (identifié par un numéro) et par l'émission d'un ticket correspondant estampillé par un numéro de série unique.

Les réservations. Un client peut réserver des places avant de procéder à leur achat. Une place réservée ne peut être achetée que par l'utilisateur qui a effectué la réservation. Une place réservée peut être rendue disponible soit par l'administrateur de l'application soit par le client qui l'a réservée.

Hypothèses supplémentaires. Le théâtre conserve toujours 70 places (toutes catégories confondues) qui seront vendues au guichet, dans l'heure qui précède le début du spectacle. D'autre part, il arrive que le responsable de la programmation annule une représentation. Dans ce cas, il n'est plus possible ni de réserver, ni d'acheter des places pour cette représentation.

3 Travail à faire

3.1 Spécification de l'interface

Avant de commencer tout travail de développement, il est nécessaire de réfléchir à la structuration de votre site. Cette structuration vous permettra entre autre de préciser les modèles que vous utiliserez ainsi que les contrôleurs nécessaires pour les animer.

De manière à guider cette réflexion, nous vous proposons de recourir aux scénarios d'usage. Un scénario d'usage est une description narrative d'une interaction d'un usager avec le site. Il décrit tout d'abord la personne concernée (un utilisateur habitué du site, découvrant le site etc), la tâche qu'il souhaite réaliser et les différentes interaction pour y parvenir.

Exemple de début de scénario : Paul est un utilisateur enregistré. Il se connecte sur le site web et choisit directement la page d'identification. Il est alors redirigé vers sa page d'accueil affichant son agenda des spectacles (liste des représentations à venir pour lesquelles il a acheté des places). Il décide alors de consulter les liste des spectacles à venir pour la saison en cours. Il choisit le spectacle Star Mania. Il accède alors à la liste des représentations et pour chacune, etc.

3.2 Tiers données

1. Proposer un diagramme de classes UML permettant de modéliser ces informations.
2. Proposer un schéma relationnel en faisant tout d'abord une analyse de votre problème (DF, contraintes de multiplicité, valeurs) ainsi qu'une conception entité association notation UML.
3. Implanter la base de données obtenues avec Oracle et remplir vos tables. Identifier les contraintes d'intégrité qui restent à la charge de l'application.
4. Proposer les transactions qui correspondent à vos cas d'usages.

4 Travail à rendre

Le projet est à réaliser en équipe de 4. La constitution des équipes est libre et sera réalisée sous TEIDE. Les scénarios d'usage devront être déposés sur TEIDE et validés par l'enseignant WEB. Il se terminera par une soutenance durant laquelle vous devrez présenter votre architecture logicielle (découpage en contrôleurs, modèles, DAO), le diagramme UML de vos données et le schéma relationnel en découlant. Vous devrez également réaliser une démonstration de votre application, illustrant au moins un des scénarios d'usage proposé.