Instructions pour récupérer et faire fonctionner le projet

Le projet est sur le repository GitHub suivant :

https://github.com/HuntTheShunt11/BurgerQueen

Pour faire fonctionner le projet, il ne faut pas oublier de redéfinir le chemin du fichier de base de données dans le fichier *persistence.xml* présent dans le dossier *src/main/resources/META-INF*.

Il est possible de créer et de pré remplir la base de données en exécutant le programme de la classe *FillDatabase.java* qui se trouve dans le dossier *src/main/java/fr/tp/isima/BurgerQueen*.

Structure

Pour ce projet, nous sommes partis de la base du TP6 de J2EE. Nous avons tout abord récupéré le TP6 que nous avons adapté à nos besoins.

Pour ce qui est de la structure du projet, comme nous sommes partis de la base du TP6, nous avons également repris la stack présentation/business/data access layer de ce TP. Pour la base de données, nous sommes partis sur 4 tables :

- Une table *Burger* qui contient un identifiant unique, le nom du burger, sa description, ses notes (son goût, son originalité, la qualité et la présentation) et une liste d'identifiants d'ingrédients.
- Une table *Ingrédients* qui contient un identifiant unique et un nom d'ingrédient.
- Une table de liaison *Burger_Ingrédient* fait le lien entre la liste d'id d'ingrédients dans la table Burger et les ingrédients de la table ingrédient.
- Une table *User* qui contient un identifiant unique et un nom d'utilisateur.

Au niveau du découpage de l'application, cette dernière est composée de 5 packages :

- *fr.tp.isima.BurgerQueen. business* qui correspond à la couche business avec nos burgers, ingrédients et users
- *fr.tp.isima.BurgerQueen. persistance* qui correspond à la couche data access layer avec nos DAO et nos Beans pour les burgers, les ingrédients et les users
- *fr.tp.isima.BurgerQueen. common* pour gérer les objets au niveau de l'application
- *fr.tp.isima.BurgerQueen. presentation* qui correspond à la couche presentation avec les éléments de base de la présentation (servlet de base etc.)
- *fr.tp.isima.BurgerQueen. presentation.burger* qui correspond à la couche presentation avec des servlets et des ViewBean qui nous permettent de faciliter l'affichage des données dans les pages jsp

Il y a 8 servlets importantes dans notre couche présentation :

- *ListBurgersServlet* qui récupère tous les burgers en base pour les afficher sur la page d'accueil du site
- *ConnectionServlet* et *DeconnectionServlet* pour se connecter et se déconnecter du site
- *CreateBurgerServlet* qui permet de créer un burger
- *SeeBurgerServlet* pour récupérer les détails d'un burger
- *RateBurgerServlet* qui permet de récupérer un burger que l'on veut noter en base de données
- *SaveRateServlet* qui enregistre les notes d'un burger en base de données
- *SaveBurgerServlet* qui enregistre un burger en base de données

Dans notre couche présentation, nous avons également un filtre SaveFilter qui permet de filtrer l'accès des utilisateurs à des servlet de sauvegarde s'ils ne sont pas identifiés sur le site.

Les servlets interagissent avec des pages JSP qui se trouvent dans le dossier *src/main/webapp/jsp* :

- connection.jsp qui correspond à la page de connexion du site
- createBurger.jsp qui correspond à la page de création de burger
- listBurger.jsp qui permet d'afficher les burgers en base de données
- SeeBurger.jsp qui affiche les détails d'un burger
- RateBurger qui permet de noter un burger donné
- head.jsp qui fait tous les includes de fichiers jasvaScript, CSS etc. pour les pages
- banner.jsp qui correspond à la bannière du site

Développement

Nous gérions au départ seulement les burgers et leurs ingrédients en ayant modifié les articles et les catégories du TP6.

Nous avons tout d'abord créé la page d'affichage des burgers qui est la page d'accueil du site, la base étant pré-remplie à des fins de tests à l'aide de la classe FillDatabase. Par la suite, nous avons ajouté la page de notation des burgers. Chaque burger dispose d'un lien « noter » qui mène à une page où l'on peut les noter sur leur originalité, leur goût etc. à l'aide de boutons radios qui correspondent à une note entre 1 et 5.

Lorsque la note est enregistrée dans la base de données, on effectue également la moyenne des critères qui est affichée pour chaque burger sur la liste de ces derniers. Pour la gestion d'erreur de cette page de notation, on prend en compte le fait que l'utilisateur peut modifier la page html et mettre des valeurs non valides pour les différents boutons. Ainsi si la note n'est pas comprise entre 1 et 5, l'erreur est affichée sur la page de notation. Le nombre de notation de chaque burger est affiché à droite du bouton permettant d'accéder à la notation de chaque burger.

Pour améliorer le design et l'ergonomie du site, nous avons ajouté une bannière pour chaque page afin que les utilisateurs puissent se connecter et revenir à la page d'accueil du site (bouton home à gauche). Pour la connexion, un volet déroulant se trouve sur la droite de la bannière. Si l'utilisateur n'est pas connecté, le nom du volet est « Menu » et la seule option proposée dans le menu est la connexion. Pour cette dernière, l'utilisateur choisit un nom d'utilisateur qui sera enregistré en base de données et en session. Une fois ce dernier connecté, le menu de la bannière change : il affiche désormais le pseudonyme choisi et il propose comme option de se déconnecter à la place de la connexion.

Une fois ces éléments ajoutés, nous avons pu mettre en place la page d'ajout des burgers. Cette dernière permet de définir le nom du burger, sa description, ses ingrédients et de le noter. Pour les ingrédients, une liste prédéfinie d'ingrédients que l'on peut ajouter à la recette est proposée (si des ingrédients sont dans la base). Chaque ingrédient rajouté au burger s'affiche sur la page à l'aide d'un script JavaScript. Au niveau de la gestion des erreurs, on ne peut pas rajouter deux burgers avec le même nom par exemple.

Pour éviter que des utilisateurs puissent modifier la base de données sans être connectés, nous avons mis en place un filtre. Ce dernier, nommé *SaveFilter*, s'applique sur les servlets avec une url commençant par */save/*. Si l'utilisateur essaye de sauvegarder des informations sans qu'il soit connecté, il est automatiquement redirigé vers la page de connexion pour se connecter.

Au niveau de l'organisation du développement, nous avons tout d'abord travaillé ensemble sur un même PC à l'ISIMA lors des deux derniers TP, puis en venant dans la salle de TP pendant une semaine après les partiels. Par la suite, comme nous ne pouvions pas être tous les deux sur Clermont-Ferrand pour les derniers jours, nous avons travaillé à distance en utilisant Github.

Pour les tests de l'application, nous nous sommes basés sur les dernières versions des navigateurs avec notamment des tests sur Firefox 45.0.1 et Internet Explorer 11.