Rapport du projet d'architecture logiciel

13.06.2022

—

Eckson Glitho
Licence 3 Génie Logiciel
ESITEC
Sup de Co Dakar

SOMMAIRE

SOMMAIRE **2**

Conception **2**

- Package models 3
- Package dao 3
- Package controllers 3
- Package views 3

Descriptif technique et fonctionnel **4**

Comment lancer le projet **4**

A. Conception

Afin de mettre en pratique les connaissances acquises au cours du module d'architecture logiciel, il nous à été demandé de mettre en place un blog avec le langage Java.

Ce dernier devait être réalisé en utilisant l'architecture monolithique. C'est donc sur ce modèle de conception et du motif d'architecture logiciel MVC (Model View Controller) que se base l'entièreté du projet.

Afin de respecter les contraintes du projet, nous l'avons structuré en en quatre (4) packages: controllers, dao, models, views.

a. Package models

En ce qui concerne ce package, il contient les modèles relatifs au différentes entités de notre base de données en l'occurrence Post et User. Dans chacun de ces modèles, on y trouve les structures de données correspondant à nos entités accompagnées de leurs constructeurs, getters et setters.

b. Package dao

Le package DAO (Data Access Object) quant à lui contient la classe DBConnect. A travers cette dernière nous faisons le lien entre la couche métier, en l'occurrence les controllers et la couche de persistance, qui fait référence à la base de données. Grâce à lui, pour récupérer un objet stocké, nous faisons appel à la couche dao qui se chargera d'effectuer la connexion à la base données et d'exécuter les requêtes SQL.

c. Package controllers

Le package Controllers contient les différents controllers relatifs aux modèles créés plus tôt. Ainsi le PostModel possède le PostController qui lui est dédié et le UserModel est en possession du Usercontroller. Ces controllers détiennent la logique de traitement de toutes les intéractions et entrées des Vues et les requêtes de mise à jour et d'accès de la base de données en utilisant les modèles.

d. Package views

Enfin, ce package représente l'ensemble des interfaces graphiques de l'application (UI). A travers eux on affiche les données à l'utilisateur, à l'aide des modèles et lui permet également de modifier les données.

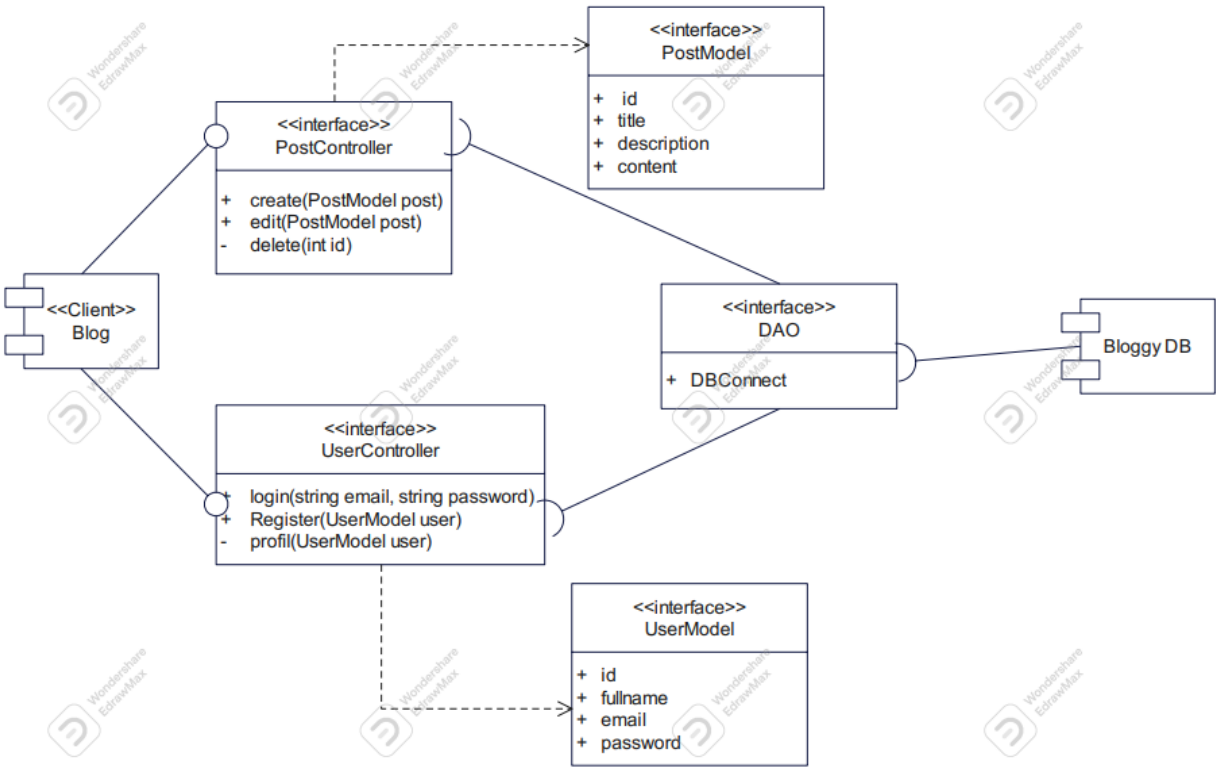

Diagramme de composant du blog

B. Descriptif technique et fonctionnel

En ce qui concerne l'utilisation de l'application, il est nécessaire pour l'utilisateur de s'authentifier avant de commencer. Ce blog est un blog privé où nous n'avons besoin que d'un seul utilisateur.

Lors de l'authentification, si l'utilisateur a un compte, il renseignera ses informations. Derrière, le UserController avec la fonction login(), vérifiera les informations saisies en faisant appel à la couche DAO par le biais de la classe DBConnect pour accéder à la base de données. Si les informations sont correctes il sera redirigé vers la page d'accueil ; et dans le cas contraire il sera invité à renseigner de nouveau ses informations de connexion.

Dans le cas où il n'aurait pas de compte, il s'inscrira sur la plateforme en remplissant un formulaire. La vue récupérera les données saisies qu'elle enverra vers le UserController qui, de son côté, grâce à la méthode register() et le model, UserModel créera un nouvel utilisateur dans la base de données.

Une fois l' authentification réussie, l'utilisateur voudra sûrement faire une publication sur son blog. Sur sa page d'accueil se trouve un bouton de navigation vers l'interface qui lui permettra de réaliser un post. Sur cet espace, il lui suffira de remplir les champs nécessaires pour le post.

En arrière-plan, la vue renverra les données vers le PostController. Là bas les données seront traitées par la méthode create() et le model PostModel. Une fois les posts créés, il pourra les voir en navigant vers l'interface derrière le bouton "BROWSE POSTS" de la page d'accueil.

Sur cette dernière, il pourra voir ses posts, faire une recherche par *id* d'un post afin de le modifier ou de le supprimer. Ses fonctionnalités sont respectivement traitées par les méthodes searchId(), edit() et delete() du PostController et par le modèle PostModel.

Pour chacune des méthodes développées, la classe DBConnect fut incontournable pour interagir avec la base de données et l'utilisation du design pattern DAO fut d'une grande aide. En effet, sans ce dernier, pour chaque fonctionnalité il aurait fallu refaire à chaque fois la connexion à la base de données pour faire exécuter les requêtes. Mais grâce à elle nous avons pu rendre le code flexible et faciliter l'interaction avec la base de données.

C. Comment lancer le projet

Pour lancer le projet, vous devez commencer par importer la base de données jointe dans le projet et la nommer bloggy. Il s'agit d'une base de données mysql. Vous aurez par conséquent besoin de xampp ou de wamp. Nous vous conseillons l'éditeur de texte NetBeans pour pouvoir lancer le projet.

Pour lancer, rendez vous dans \Source Packages\Views\. Faites un clic droit sur Login.java puis sur "Run file".

Afin de voir certains changements s'effectuer, des réactualisations des pages sont nécessaires. Pour ce faire, il suffira d'utiliser le bouton back pour revenir en arrière et rouvrir la page. Vous aurez à le faire notamment sur la page de "BROWSE POST".