Documentation technique

Application ScrumTool

Contenu

Version : 1.0.0.1

Présentation de l'application :

L'application ScrumTool est une application, de type client lourd (SWT + JFace), de gestion de projets SCRUM. Les données de l'application sont enregistrées dans une base de données, permettant ainsi d'assurer leur mémorisation au fil du temps.

I Caractéristiques techniques :

- Langage de programmation : Java
- Type de développement : programmation orientée objet
- Utilisation de l'ORM (Object Relational Mapper) Hibernate, pour le mappage relationnel/objet
- application de type client lourd : utilisation de la librairie de composants graphiques SWT (Standard Widget Toolkit) et de JFace (framework permettant l'implémentation de MVC au niveau de l'interfaçage)

- Plateforme de développement : WAMP (version 2.2)
- Système de Gestion de Base de Données : MySQL (version 5.5.24)
- Serveur Web : Apache (version 2.2.22)
- Navigateurs web : Google Chrome (version 34.0.1847.116 m); pour l'accès à l'interface d'administration (PHPMyAdmin) de la base de données sous MySQL

- Environnement de Développement Intégré : Eclipse (version Kepler)

II MVC :

L'application développée utilise le patron de conception MVC (Modèles, Vues, Contrôleurs) afin de séparer les traitements, les données, et la présentation.

III Arborescence du projet :

Etude des packages de l'application :

Nom du package	Rôle
net.controllers	contient les contrôleurs de l'application
net.db	contient le script de création de la base de données
net.images	contient les images utilisées dans l'application
net.models	contient les classes métier générées par Hibernate
net.technics	contient des classes techniques (classes utilitaires, providers, classes passerelles d'accès aux données)
net.vues	contient les vues de l'application

1. Classes métier :

Nom	Descriptif
Collaborator	Classe permettant l'instanciation d'un collaborateur
Comment	Classe permettant l'instanciation d'un commentaire
Commenttype	Classe permettant l'instanciation d'un type de commentaire
Event	Classe permettant l'instanciation d'un événement
Eventtype	Classe permettant l'instanciation d'un type d'événement
Paticipate	Classe permettant l'instanciation d'un objet Participate
ParticipateId	Classe générée par Hibernate pour donner un id à une instance de Participate
Playrole	Classe permettant l'instanciation d'un objet Playrole
PlayroleId	Classe générée par Hibernate pour donner un id à une instance de Playrole
Product	Classe permettant l'instanciation d'un produit
Realize	Classe permettant l'instanciation d'un objet Realize
RealizeId	Classe générée par Hibernate pour donner un id à une instance de Realize
Role	Classe permettant l'instanciation d'un rôle
Sprint	Classe permettant l'instanciation d'un sprint
Status	Classe permettant l'instanciation d'un statut
Userstory	Classe permettant l'instanciation d'une user story

2. Classes techniques :

Classes utilitaires :

Nom	Descriptif
HibernateUtil	Méthodes utilitaires liées à la session
Utils	Méthodes utilitaires (chargement d'images, vérification de la connexion à l'application...)

Classes d'accès aux données (DAO) :

Nom	Descriptif
DAOCollaborator	Classe passerelle : accès aux données concernant les collaborateurs
DAOProduct	Classe passerelle : accès aux données concernant les produits
DAOSprint	Classe passerelle : accès aux données concernant les sprints
DAOUserStory	Classe passerelle : accès aux données concernant les user stories

Providers :

Nom	Descriptif
CollaboratorTvProvider	Définit comment les collaborateurs doivent s'afficher dans une liste
ProductTvProvider	Définit comment les produits doivent s'afficher dans une liste
UserStoryTvProvider	Définit comment les user stories doivent s'afficher dans une liste

3. Contrôleurs :

Nom	Descriptif
AccueilController	Contrôleur lié à l'accueil
ActionProduitController	Contrôleur lié à la gestion des produits
AffectationCollaboratorsController	Contrôleur lié à l'affectation de collaborateurs sur des user stories
AppController	Classe utilisée lors du lancement de l'application
CollaboratorController	Contrôleur lié à la gestion des collaborateurs
CreateProjectController	Contrôleur lié à la création de produits
CreateUserStoryController	Contrôleur lié à la création de user story
LoginController	Contrôleur lié à l'authentification à l'application

MdpModificationController	Contrôleur lié à la modification du mot de passe
ModifyProjectController	Contrôleur lié à la modification de produits
ModifyUserStoryController	Contrôleur lié à la modification de user stories
MyProfileController	Contrôleur lié à la gestion du profil
ProductController	Classe de sauvegarde du produit et de la user story sélectionnés
Prog	Programme de lancement de l'application

IV Bonnes pratiques et normes suivies

Structure de l'application :

L'application ScrumTool, respectant le patron de conception MVC, on respectera la structure suivante :

- les classes métiers devront être stockées dans le package *net.models*
- le script de création de la base de données est stocké dans le package *net.db*
- les classes techniques devront être stockées dans le package *net.technics*
- les contrôleurs devront être stockés dans le package *net.controllers*
- les vues dans le package *net.vues*
- les images dans le package *net.images*

Classes métier :

- leur nom doit correspondre à une table de la base de données

Classes techniques :

- leur nom commence par une majuscule

Classes d'accès aux données :

- leur nom commence par le préfixe "DAO", suivi du nom de la classe métier concernée

Contrôleurs :

- leur nom sera toujours suffixé par l'extension "Controller", pour montrer qu'il s'agit d'un contrôleur

Vues :

- leur nom commence toujours par un "V" majuscule, permettant ainsi de déterminer rapidement qu'il s'agit d'une vue

- on s'efforcera de donner un nom pertinent aux vues, pour éviter de devoir lire le code pour déterminer leur utilité

Constantes, variables et paramètres :

De manière générale :

- on s'efforcera de donner un sens au nommage
- on respectera la différence singulier/pluriel : on utilisera un nom au pluriel pour désigner une collection d'objets, et un nom au singulier pour désigner une instance de classe
- on évitera l'utilisation d'accents, de caractères spéciaux, de chiffres dans les noms de variables

Documentation du code :

Pour chaque méthode, fonctions et procédures, développée :

- on s'efforcera de documenter le code, et d'indiquer de manière claire le rôle de chacune des méthodes : on doit pouvoir connaître le fonctionnement sans avoir à lire le code