Documentation technique

Pango-Scrum

Présentation de l'application

Pango-Scrum est une application de gestion de projet SCRUM, c'est un client lourd qui utilise une bibliothèque graphique et un framework (SWT et Jface).

Les données saisie dans l'application son enregistrées dans une base de données afin de les rendre persistantes.

I Caractéristiques technique

- Langage de programmation: Java.
- Type de programmation : Programmation orientée objet.
- Utilisation de l'ORM (Object Relational Mapper) Hibernate, pour le mappage relationnel/objet.
- Application de type client lourd : utilisation de la bibliothèque graphique SWT (Standard Widget Toolkit) et du Framework Jface. Jface permet l'implémentation du modèle MVC.
- Plateforme de développement : WAMP.
- Base de données : MySql.
- Environnement de développement : Eclipse (version Kepler).

II MVC

Le développement de cette application s'est effectué avec le patron de conception MVC (Modèle-Vue-Controlleur) afin de séparer les traitements, les vues, et les données. Cela permet de mieux structurer l'application.

III Arborescence du projet

Packages de l'application :

Nom	*Description*
net.controllers	Contient les contrôleurs de l'application
net.db	Contient le script de création de la base de données.
Net.images	Contient les différentes images utilisées dans l'application.
Net.technics	Contient toutes les classes techniques comme les classes utilitaire, les providers et les classes passerelle entre le programme et la base de données.
Net.vues	Contient les vues de l'application

1.Classes metier

Nom Classe	*Description*
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 d'instancier un événement
Eventtype	Classe permettant d'instancier un type d'événement
Participate	Classe permettant l'instanciation d'un objet Participate
ParticipateId	C'est une classe générée par hibernate et qui permet d'affecter un id à une instance de Participate
PlayRole	Classe qui permet d'instancier un objet PlayRole (rôle joué)
PlayRoleId	PlayRole
Product	Classe permettant d'instancier un Produit
Réalize	Classe permettant l'instanciation d'un objet Réalize, elle permet de préciser quel est la tâche d'un collaborateur.
RealizeId	C'est une classe générée par hibernate et qui permet d'affecter un id à une instance de Realize
Role	Classe permettant d'instancier un rôle
Sprint	Classe qui permet d'instancier un sprint
Statut	Classe permettant l'instanciation d'un statut
Userstory	Classe permettant l'instanciation d'une user story

2. Classes techniques

Classes utilitaires :

Nom classe	*Description*
HibernateUtil	Méthodes utilitaires liées à la session
Utils	Méthodes utilitaires (chargement d'images...)

Classes d'accès à la base de données (DAO) :

Nom classe	*Description*
DAOCollaborator	Classe passerelle : accès aux données qui concernent les collaborateurs.
DAOProduct	Classe passerelle : accès aux données qui concernent les produits.
DAOUserStory	Classe passerelle : accès aux données qui concernent les user stories.
DAOSprint	Classe passerelle : accès aux données qui concernent les sprints.

Providers :

Nom Classe	*Description*
CollaboratorTvProvider	Définit comment les collaborateurs doivent s'afficher dans un tableau ou une liste
ProductTvProvider	Définit comment les produts doivent s'afficher dans un tableau ou une liste
TvSprintProvider	Définit comment les sprints doivent s'afficher dans un tableau ou une liste
TvProviderOverview	Définit comment les sprints et les userStory doivent s'afficher dans un tableau ou une liste de la page overview

3.Contrôleurs

Nom	*Description*
AccueilController	Contrôleur lié à l'accueil
AffectationScrumMasterProjetController	Contrôleur lié à l'affectation de scrum-master à un projet
AffectationController	Contrôleur lié à l'affectation de collaborateur sur des user stories
AppController	Classe utilisée lors du lancement de l'application
CollaboratorController	Contrôleur lié à la gestion des collaborateurs
LoginController	Contrôleur lié à la gestion de l'authentification d'un utilisateur
MdpModificationController	Contrôleur lié à la modification du mot de passe
MyProfileController	Contrôleur lié à la gestion du profil de l'utilisateur
OverviewController	Contrôleur lié aux informations d'un produit (sprint, user stories : en cours, a faire, fini)
ProductController	Contrôleur lié à la gestion des produits
Prog	Contrôleur lié au démarrage de l'application
SprintController	Contrôleur lié à la gestion des sprints
UserStoriesController	Contrôleur lié à la gestion des user stories

IV Bonne pratiques et normes à respecter

Architecture 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 se terminera toujours par "Controller", pour montrer qu'il s'agit bien d'un contrôleur

Vues :

- leur nom commence toujours par un "V" majuscule, afin de démontrer qu'il s'agit d'une vue.

Il essentiel que les vues aient un nom qui correspond à leur fonction : cela permet de savoir ce qu'elle affiche sans allez dans le code.

Constantes, variables et paramètres :

De manière générale :

- le nommage doit être cohérent.
- respect de 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 :

Chaque développeurs qui participe au projet réalisera la documentation des fonctions de l'application qu'il aura fait.