Rapport technique
21/03/12

Anthony Morales, Chen-Yang Gao, Irvin genieys, Simon Maby

Table des matières

I. Introduction

Nous présenterons dans ce rapport le compte rendu technique de notre projet Slab. Nous décrirons, dans un premier temps, les spécifications fonctionnelles des différentes classes puis nous présenterons les outils utilisés pour réaliser ce projet.

II. Spécifications Fonctionnelles

Cette partie donne une vue d'ensemble de la déclaration des différentes méthodes correspondant aux cas d'utilisation.

Int connexion (String login, String mdp)
Paramètres : le login et le mot de passe de l'enseignant
Retour : 0 en cas d'échec, 1 si l'enseignant est aussi un responsable et 2 sinon

Cette méthode permet de connecter un enseignant. Elle entraine le chargement en mémoire des données nécessaires à l'enseignant notamment ses enseignements, les créneaux, les groupes et les caractéristiques.

Int demanderSeance(Creneau creneau, Date date, Enseignement enseignement, ArrayList<caractéristique> caractéristiques)
Paramètres : le créneau, la date, l'enseignement, les caractéristiques de la salle
Retour : renvoie 1 en cas de succès et 0 sinon

Cette méthode permet de demander une réservation de séance. Elle doit lancer une exception si les caractéristiques données ne sont pas présentes dans la base de donnée ou si le créneau ou l'enseignement donné n'existe pas dans les données persistantes.

ArrayList<Caractéristiques> getCaractéristiques()
Paramètres : aucun
Retour : les caractéristiques des salles stockées en mémoire persistante

ArrayList<Creneau> getCreneaux()
Paramètres : aucun
Retour : les créneaux stockés dans la mémoire persistante

ArrayList<Salle> getSalleDisponibles(Seance reservation)
Paramètres : une réservation
Retour : une liste des salles disponibles

ArrayList<Enseignement> getEnseignements()
Paramètres : aucun
Retour : les enseignements correspondant à l'enseignant enregistré dans la session

ArrayList<Reservation> getDemandes()
Paramètres : aucun
Retour : les demandes en cours de l'enseignant logué

ArrayList<Reservation> getAllDemandes()
Paramètres : aucun
Retour : toutes les demandes stockées en mémoire persistante

ArrayList<Reservation> getPlanning()
Paramètres : aucun
Retour : le planning de l'enseignant logué

Reservation getPlanning (Creneau c, Date d)
Paramètres : un créneau et une date
Retour : les réservations pour un créneau et une date donnée

La méthode doit générer une exception si le créneau est introuvable dans mémoire persistante.

ArrayList<Reservation> getPlannning(Groupe groupe)
Paramètres : un groupe
Retour : la liste des réservations associées au groupe

ArrayList<Reservation> chargerReservations(Enseignement enseignement, Creneau creaneau, Date date)
Paramètres : un enseignement, un créneau et une date
Retour : la liste des réservations correspondantes

Cette méthode permet de charger les réservations correspondantes à un enseignement, un créneau, et une date.

Int attribuerSalle(reservation reservation, Salle salle)
Paramètres : une réservation et une salle
Retour : 1 en cas de succès, 2 si la salle n'est pas disponible, 3 si la reservation n'existe pas, 0 si l'écriture ne fonctionne pas.

Cette méthode attribue une salle à la réservation donnée dans les données persistantes. Elle lance une exception si la salle donnée n'existe pas dans les données persistantes.

Heure getNbHeureProgramme (Enseignement e)
Paramètres : un enseignement
Retour : le nombre d'heures programmées par l'enseignant pour l'enseignement sous format horaire

Cette méthode lance une exception si l'enseignement est introuvable.

Heure getNbHeuresFaites(Enseignement e)
Paramètres : un enseignement
Retour : le nombre d'heures effectuées par l'enseignant pour l'enseignement sous format horaire

Cette méthode lance une exception si l'enseignement est introuvable.

ArrayList<Groupe> getGroupes()
Paramètres : aucun
Retour : la liste des groupes

Void supprimerReservation(Reservation reservation)
Paramètres : une réservation
Action : supprime la réservation donnée en paramètre des données persistantes

ArrayList<Notification> getNotifications()
Paramètres : aucun
Retour : la liste des notifications de l'enseignant

Void supprimerNotifications()
Paramètres : aucun
Action : supprime toutes les notifications de l'enseignant

III. Outils utilisés

a. ArgoUml

Pour réaliser la conception objet de notre projet nous avons choisi d'utiliser le programme argoUml qui est un outil graphique de conception objet. Ce logiciel nous a permis par la suite de générer automatiquement du code à partir de notre conception ce qui a permis un aide de temps considérable.

b. Eclipse

Pour développer notre projet nous avons utilisé l'environnement de développement intégré eclipse. Nous avons choisi cet outils car c'est l'IDE le plus utilisé et le plus fiable pour développer des projets en Java et de plus il fournit de nombreux plug-in qui nous ont été d'une grande aide.

c. Subversion/Subclipse

Afin de gérer au mieux le travail collaboratif en équipe nous avons décidé d'utiliser le gestionnaire de version subversion. Pour stocker nos données nous avons utilisé l'hébergeur de dépôt svn assembla et notre projet est accessible à l'url http://subversion.assembla.com/svn/gestionedt/.
Nous avons de plus choisi d'utiliser le plug-in subclipse d'eclipse qui fournit un client svn intégré à l'éditeur et nous a permis de gérer plus simplement le gestionnaire de version.

d. Le framework JDBC

Pour interagir avec les données stockées en mémoire persistante nous avons utilisé le framework java JDBC qui permet une utilisation simple d'une base de données dans un programme java.

e. Le framework JDOM

Pour parcourir des fichiers xml nous avons utilisé le framework JDOM qui fournit une API pour la gestion des fichiers xml.

f. Swing4eclipse

Afin de faciliter le développement de l'interface utilisateur graphique nous avons utilisé le framework swing4eclipse qui fournit un outil visuel pour développer des interfaces graphiques en java.

g. Javadoc

Pour documenter notre code nous avons utilisé l'outil javadoc, de plus eclipse fournit un outil de génération de javadoc.