|  |
| --- |
| EISTI |
| Journal de Développement |
|  |
|  |
| **Laurent Hatier** |
| **19/11/2010** |

|  |
| --- |
| Journal de développement du projet OPTIM’EISTI – côté Client. Développement JAVA |

Voici le journal de développement du projet OPTIM’EISTI :

Semaine du 11 Octobre :

Semaine du 18 Octobre :

 Questionnement sur l’utilisation du pattern « Monteur » permettant de gérer les droits des différents rôles des utilisateurs : Choix d’utiliser une interface Action ayant des classes qui implémente de celle-ci et de redéfinir la méthode « execute » en fonction de l’action choisi.

Semaine du 25 Octobre :

 Test unitaire des boutons concernant les problèmes.

 Problème rencontré sur le panel gestion groupe qui est commun à l’administrateur et au responsable de groupe : création de 2 classes différentes permettant d’avoir des panels ayant des composants propres à chacun.

Semaine du 1er Novembre :

 Création du test Unitaire Administrateur : test de l’ensemble des boutons du panel administrateur concernant la partie administrative du logiciel : ensemble des actions possibles sur les utilisateurs et les groupes.

 Choix de la mise en forme du fichier source. Le fichier comportera le rôle, le nom, le prénom, l’adresse, etc ...

 Intégration du fichier externe dans le logiciel et création de l’interaction entre le fichier et le logiciel : Problème de lisibilité du code lors de la récupération des différentes informations et interaction entre logiciel et données :

* Amélioration de la lisibilité par :
	+ Création de la classe « Application » permettant de gérer les différentes listes créées avec des « Getter & add » et des méthodes agissant sur les listes.
	+ Création de la classe « ApplicationHelper » permettant de tester les appartenances ou autres méthodes concernant les tests à effectuer.

Semaine du 8 Novembre :

 Mise en commun du travail d’Emmanuel Declercq (responsable du côté mathématique du Client) : Emmanuel Declercq devient responsable Réseau

* Amélioration du diagramme de classe : Mise en application de différents « design pattern » appris en cours :
	+ « Factory »
	+ « Adapter
	+ « Mediator »
	+ Mise en évidence du pattern « Strategy » -> Interface Action et ses classes qui l’implémentent
	+ Questionnement sur l’utilisation du pattern « Proxy » pour la classe utilisateur -> créer une « light class » : Réponse positive à la demande.

Semaine du 15 Novembre :

 Mauvaise conception du pattern « Médiator » : Le pattern a été effectué pour gérer les panels ce qui n’est pas très pratique d’un point vue conception : Remodelage de ce dernier avec l’application de ce pattern sur l’ensemble des boutons du logiciel. Ceci permettra de gérer l’ensemble des évènements liés affichage ou non des panels (à la vue). D’un autre côté, les listeners «  de base » seront employés à l’interaction avec le model/contrôleur(Application)

 Amélioration de l’IHM : Le fait qu’un utilisateur peut rejoindre un autre groupe quand il est « loggé » n’a pas été pensé -> Création d’un nouveau panel permettant cette action.

 Soucis de lecture du code avec le passage en paramètre du « Médiator » : Utilisation du design pattern « Singleton » permettant de récupérer l’instance de ce dernier et donc d’effectuer le médiateur .