



IUT 2015-2016 PROJET JAVA

Planification des soutenances de stages

Table des matières

Objet du projet : Planification des soutenances de stages	2
Modélisation du problème	3
Eléments obligatoires.....	3
Les besoins	3
Analyse & conception architecturale.....	3
Développements - Java	4
Gestion des erreurs	4
Internationalisation	4
Interface homme-machine.....	4
Proposition de planning.....	4
Sauvegardes	4
Communication	4
Export WEB	4

Sujet du projet : Planification des soutenances de stages

L'objet du projet est de développer un logiciel pour la planification des soutenances de stage et de gestion de ressources. Cet outil fera l'objet d'une modélisation UML complète suivi d'un développement de l'application en Java.

L'outil devra se présenter comme un agenda, les événements de l'agenda seront des soutenances. Chaque soutenance disposera

- d'une salle de soutenance
- d'une date et d'une heure
- d'un nombre non défini de personnes représentant le Jury (le jury étant lui-même constitué de membres institutionnels – généralement vos professeurs – et de membres professionnels – les personnes représentant l'entreprise dans laquelle a été effectué le stage).
- d'un étudiant (qui effectue sa soutenance)
- Contraintes (non exhaustif)
 - Contraintes concernant les membres institutionnels :
 - pas plus de deux demi-journées de soutenances
 - pas plus de N soutenances par enseignant
 - pas plus de N soutenances
 - une pause de X minutes le matin
 - une pause déjeuner d'au moins de minimum H heure
 - pas plus de NS soutenances d'affilée
 - le membre du Jury est disponible sur un ensemble de créneaux horaires
 - le membre du Jury est disponible sur un ensemble de jours
 - ...
 - Contraintes concernant le membre professionnel du Jury
 - le membre du Jury est disponible sur un ensemble de créneaux horaires
 - le membre du Jury est disponible sur un ensemble de jours
 - ...
 - Contraintes concernant l'étudiant
 - un étudiant est disponible sur un ensemble de créneaux
 - un étudiant doit passer le jour J et pas J' -> facile
 - Il est possible d'utiliser une bibliothèque d'optimisations pour proposer, en fonction des contraintes, une répartition des soutenances
- Pour chaque soutenance, nous devons avoir la possibilité de visualiser le rapport, la présentation
- Pour chaque soutenance, il devra y avoir un ensemble de documents partagés qu'il est possible de modifier durant la soutenance (pour prendre des notes par exemple) : il est possible de s'inspirer de Google document.
- Pour chaque soutenance, il devra y avoir un document d'évaluation
- Un système de recherche devra être mis en place
 - Pour un professeur donné recherche des salles, des jurys dont il fait partie
 - Pour un étudiant donnée, recherche la salle, des membres du jury ...

L'outil devra disposer d'une fonctionnalité de publication WEB (sous forme de Servlet/JSP), d'un système de *Chat* (ou de *Visio conf*) afin que les enseignants travaillant sur logiciel puissent communiquer en temps réel.

L'outil devra pouvoir sauvegarder/charger les données liées à la planification un fichier XML (et/ou dans une base de données).

Modélisation du problème

Éléments obligatoires

- Gestion de sessions de l'application (*Singleton*)
- Gestion avancées des erreurs : *Exceptions* / *Logger*
- Mise en place de *packages* (en Java le *package* racine devra être *edu.iut*)
- Une MVC
- Un plan de tests

Les besoins

- Définir le contexte général dans lequel le projet va être réalisé.
- Définir les acteurs humains ou non qui vont interagir avec le système.
- Définir les fonctionnalités attendues du système.
- Définir le contexte technique du projet (serveurs, réseau, etc...)
- Définir le fonctionnement dynamique de chaque fonctionnalité.
- Définir les besoins en interface homme machine (IHM).
- Rédiger un cahier des charges fonctionnel et technique qui permettra à la fois de réaliser le projet et de le valider au fur et à mesure
- Diagrammes
 - **Diagramme de cas d'utilisation (Obligatoire)**
 - Diagramme de séquence
 - Diagramme de déploiement
 - Diagramme d'état
 - Diagramme d'activité

Analyse & conception architecturale

- Identifier les classes candidates.
- Définir les packages et l'architecture générale.
- Regrouper les classes par package.
- Définir les interfaces des packages.
- Affiner la définition statique et dynamique des classes.
- Générer l'architecture de départ
- Diagrammes
 - **Diagramme de classe (Obligatoire)**
 - **Diagramme de package (Obligatoire)**
 - Diagramme de séquence
 - Diagramme d'état
 - Diagramme d'activité
 - Diagramme d'objets

Développements - Java

L'ensemble des développements demandés seront abordés en TD/TP

Le projet sera développé dans le langage Java, vous devrez répondre au sujet et respecter les contraintes suivantes :

Gestion des erreurs

L'application devra disposer d'un système avancé de gestion d'erreur en intégrant vos propres Exceptions et un *Logger*.

(Cours 2)

Internationalisation

L'application devra être multilingue.

(Cours 2)

Interface homme-machine

L'interface devra être développée avec Java Swing, il est possible de s'inspirer de Google Agenda par exemple.

(Cours 1)

Proposition de planning

Il sera possible d'intégrer une bibliothèque pour les problèmes de résolution de contraintes (v. Choco).

(Recherches personnels)

Sauvegardes

Il devra être possible de sauvegarder et charger le planning au format XML (et éventuellement dans une base de données).

(Cours 5)

Communication

Il faudra intégrer un système de *Chat* conçu avec un protocole réseau de votre choix.

(Cours 7)

Export WEB

Il faudra intégrer un système d'export Web pour une visualisation rapide du planning à l'aide de Servlets.

(Cours 7)