
Programmation Web Serveur : JEE

*Steven Kalwant
Sofia Latni
Javid Mougamadou*

LSI2 - 2014/2015

Responsables de projet :

- *Mr Barreaud*
- *Mr Chennebault*

Table des matières

Introduction

I.	Présentation du projet et de la base fournie	4
II.	La conception et choix techniques	Error! Bookmark not defined.
III.	Répartition du travail et intégration	11
IV.	Frameworks : Spring et bootstrap.....	12
V.	Manuel d'installation.....	14
1.	Le serveur à installer	15
2.	Pour ouvrir la page d'identification.....	15
3.	J'ai fait toutes les installations demandées, j'ai lancé l'application et il ne marche pas ?	15
VI.	Cahier d'utilisateur	16
1.	Comment s'identifier ?	16
2.	Quand vous vous connectez en tant que membre de la direction des études « DDE » :	17
a)	Comment faire le « peuplement » ?	17
b)	Comment vider la base ?	17
3.	Quand vous vous connectez en tant que responsable d'année « RA » :	18
a)	Comment désigner un enseignant comme responsable de module ?	19
b)	Comment gérer le déroulement parallèle et/ou en séquence des CM/TD/TP ?	20
VII.	Tests.....	21

Conclusion

Introduction

Ce projet traite le même thème que celui posé dans le cadre du projet PHP du module « Programmation WEB ». En effet, l'application réalisée est destinée au responsable de pôle des enseignements à l'Enssat, le but étant de lui faciliter le travail qui consiste à établir le planning général de la spécialité qui le concerne en début de chaque année. Le responsable doit faire ce planning à partir d'une maquette contenant les différentes Unités d'Enseignement « UE ». Cette application doit lui permettre de réaliser les actions suivantes :

- Définir le planning prévisionnel des enseignants du pôle dont il est le responsable.
- Attribuer à des enseignants la qualité de responsable d'un module de sa filière. Le responsable choisi doit être dans le groupe d'enseignant du pôle.
- Découper les interventions (chargé de cours ou encadrant d'un groupe de TP) entre enseignants intervenant dans le module.

Dans ce rapport, nous présenterons d'abord, avec plus de détails, notre projet ainsi que la modélisation adoptée et nos différentes phases de conception. Nous parlerons ensuite des techniques utilisées, notamment des frameworks que nous avons utilisés. Nous présenterons pour finir, un manuel d'installation et un cahier d'utilisateur pour expliquer les démarches à suivre pour installer et utiliser cette application web.

I. Présentation du projet et de la base fournie

Comme nous l'avons évoqué plus haut dans l'introduction, il s'agit d'une application qui sera utilisée par un responsable de pôle (Electronique, Informatique, Optronique, IMR, MSH) à l'Enssat pour établir un planning en début de chaque année.

Notre base de données est composée des classes suivantes :

ModuleDAO.java : c'est une interface qui nous permet d'accéder à un objet de type « Module »

ModuleDAOImpl.java : c'est la classe qui assure l'interaction avec le service web

Modules.java : classe des listes des modules

Module.java : classe des modules

Course.java : classe des cours

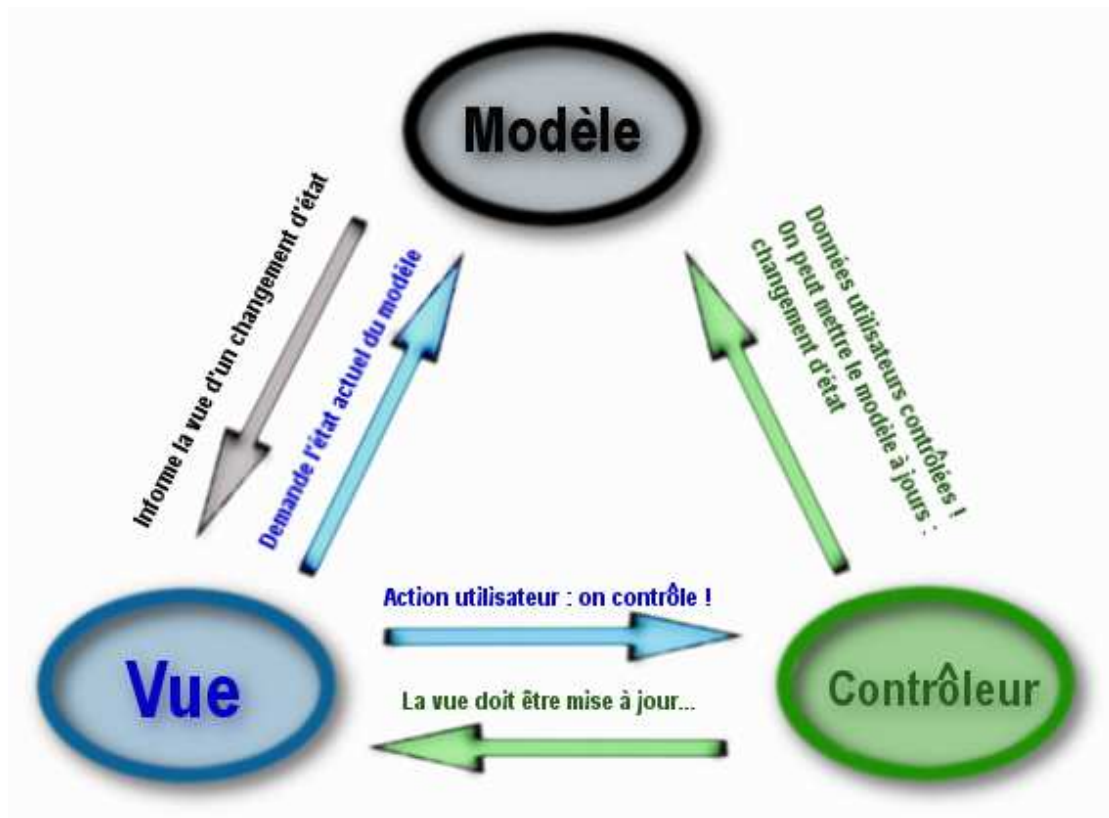
Main.java : la classe principale de notre application qui emploie les classes listées ci-dessus

Un responsable de pôle ou un utilisateur faisant partie de la Direction Des Études (DDE) peut peupler, en début d'année, la base des plannings par semaine. Il se connecte donc à l'application et déclenche le peuplement par un bouton. Une vue apparait et lui affiche quels modules ont été chargés et l'informe du bon déroulement de l'opération sinon lui affiche un message d'erreur. En effet, après cette opération la table « Module » dans la base de données change et contient les nouveaux modules introduits. Par la suite, suite à cette manipulation, le responsable de filière peut attribuer à chaque module lié à sa filière un enseignant responsable qu'il sélectionne parmi les enseignants présents dans la liste pour chaque module puis, d'un clic valide toute sa sélection. Les listes sont initialisées avec le nom du responsable si celui-ci est déjà renseigné en base.

Le responsable de filière peut aussi passer en revue les modules et peut décider de répartir les CMs, TDs et TPes entre plusieurs intervenants. Il peut aussi décider le nombre de groupes de TPes et TDs et attribue des intervenants différents s'il le souhaite à chaque groupe de TP/TD. Ceci se traduit au niveau de la base de données par un peuplement de la table « ContenuModule ». Nous pouvons distinguer deux cas de figure, deux intervenants différents qui se chargent d'un même groupe d'une manière séquentielle, ou bien deux groupes ou plus qui peuvent avoir des cours en même temps ou pas et ces cours sont donnés par un même intervenant ou des intervenants différents. Pour réaliser cette manipulation, le responsable d'année dispose d'une interface permettant de faire du séquençement et de la parallélisation. Une fois ce découpage effectué, le responsable de filière valide par un bouton. Cette validation entraine une modification directe de la table « ContenuModule » et l'envoi d'un mail au responsable de module décrivant l'opération, pour validation.

II. Cahier de conception

Pour répondre au cahier des charges, nous avons utilisé le modèle MVC.



L'architecture MVC

1. Cas 0 : Peuplement de la table Enseignant et Semaine

Déroulement :

1. L'utilisateur tombe sur la page principale.
2. Si la base de données est vide alors nous peuplons la table Enseignant et la table Semaine (1^{ère} utilisation).
3. Ensuite il est redirigé vers la page de connexion.

Pour cela, nous avons utilisé les fichiers suivants :

Modèle :

- responsable.xml : fichier contenant les informations des RA et DDE

Contrôleur :

- LoginServlet : Contrôleur de notre page de connexion

Vue :

- login.jsp : Vue HTML de notre page de connexion

Analyse et Implémentation :

Il s'agit du cas d'initialisation car la base de données est vide lors de la première utilisation. Il faut donc la remplir. Pour cela, nous utilisons notre fichier DAO responsable.xml contenant tous le nom, prénom des enseignants et leur statut (RA, DDE) et nous peuplons la table Enseignant par le biais des requêtes SQL (JDBC). Nous peuplons également la table Semaine. Ensuite l'utilisateur est redirigé vers la page d'accueil qui est notre page de connexion login.jsp par le biais de notre servlet LoginServlet.

L'utilisateur entre ensuite son identifiant et son mot de passe. Si les informations saisies sont correctes alors il est redirigé vers le menu principal RA s'il est un Responsable d'année ou le menu principal DDE s'il est membre de la direction des études. Dans le cas contraire, un message d'erreur est affiché.

2. Cas 1 : Peuplement de la table Module

Déroulement :

1. L'utilisateur connecté en tant que DDE tombe sur le menu principal DDE.
2. Il clique sur le bouton « Peupler la Base ».
3. Il tombe sur la page de peuplement de base avec un tableau comportant tous les modules insérés dans la base avec les messages d'erreurs le cas échéant.

Pour cela, nous avons utilisé les fichiers suivants :

Modèle :

- index.js : fichier JS nécessaire pour démarrer le Web Service

Contrôleur :

- ConnexionJDBC : Contrôleur de Connexion vers la BDD
- ModelFillingMain : Utilise le Web Service et peuple la base
- MenuDDEServlet : Contrôleur du menu principal DDE

- ModelFillingServlet : Contrôleur de la page de peuplement de la table module

Vue :

- menuDDE.jsp : Vue HTML du menu principal DDE
- modulefilling.jsp : Vue HTML de la page de peuplement de la table module

Analyse et Implémentation :

Lorsque le bouton « Peupler la Base » est cliqué, nous récupérons tous les modules parsés sous la forme d'un objet Java grâce au Web Service fournie par notre encadrant. Ainsi, nous pouvons manipuler plus facilement ces objets en Java. Par la suite, nous effectuons des requêtes INSERT vers la BDD et nous récupérons tous les messages d'erreurs ou de succès dans un ArrayList. Pour finir, nous affichons tous les messages dans un tableau. S'il s'agit d'une erreur alors il est marqué en rouge sinon en vert.

Problèmes / Bilan :

Le temps d'attente entre le moment où nous avons cliqué sur le bouton et le changement de page peut être long car celui-ci affiche tous les messages en même temps. Il aurait été préférable d'afficher les messages d'erreurs en continue par le biais de requête asynchrone avec une dynamicbar.

Nous avons rajouté un bouton « Vider la base » qui vide toutes les valeurs de la table Module. Cette fonctionnalité a été utile pour nos tests et permet au DDE de vider la base avant de la peupler.

3. Cas 2 : Affectation d'un enseignant à un module

Déroulement :

1. L'utilisateur connecté en tant que RA tombe sur le menu principal RA.
2. Il clique sur le bouton « Liste des modules».
3. Il tombe sur la page d'affichage des listes de modules
4. Il sélectionne un module et affecte un enseignant par le biais de la liste déroulante
5. Il valide sa sélection en cliquant sur le bouton « Valider »

Pour cela, nous avons utilisé les fichiers suivants :

Contrôleur :

- JsonServletModuleList : Servlet traitant la requête AJAX pour l'affectation d'un enseignant
- ModuleListServlet : Contrôleur de la page d'affichage des listes des modules

Vue :

- modulelist.jsp : Vue HTML de la page d’affichage des listes des modules

Analyse et Implémentation :

Lorsque le bouton « Liste des modules » est cliqué, nous affichons tous les modules de la promo concernée dans un tableau. Pour cela, nous effectuons une requête SQL SELECT pour les modules et une autre requête pour la liste des enseignants qui sera utile pour la liste déroulante. Cette liste est initialisée avec l’enseignant affecté au module.

A chaque changement d’enseignant effectué, nous ajoutons ces informations dans un fichier JSON sans doublons. Une fois que le bouton « Valider » est appuyé, une requête AJAX est envoyée vers `JsonServletModuleList` qui va effectuer plusieurs requêtes UPDATE dans la table Module.

Problèmes / Bilan :

Il aurait été préférable d’afficher dans la liste déroulante le nom et prénom de l’enseignant que de mettre son login. Pour cela, il aurait fallu modifier la requête en y insérant une jointure avec la table Enseignant. Nous avons rajouté une colonne Enseignant Actuelle qui est une information redondante compte tenu de la présence de la liste déroulante initialisé à l’enseignant actuel. Ceci est dû au fait que nous avons repris la même interface que dans notre projet précédant `PlanningParSemaine PHP` auquel nous n’avons pas eu de retour.

Pour effectuer les modifications dans la BDD, nous passons par une requête AJAX qui envoie un objet JSON avec des informations. Toutefois, il aurait été également possible de faire une simple requête POST pour cette opération.

4. Cas 3 : Découpage des modules

Déroulement :

1. L’utilisateur connecté en tant que RA tombe sur le menu principal RA.
2. Il clique sur le bouton « Découpage des modules ».
3. Il tombe sur la page des listes de modules.
4. Il clique sur le module à découper
5. Il tombe sur la page de découpage du module choisie avec une liste dynamique d’ajout et de suppression de cours (partie).
6. Il sélectionne le type du cours, son déroulement (séquence ou parallèle), le nombre d’heure et l’enseignant qui assurera le cours. Ensuite il valide sa sélection en cliquant sur le bouton « Ajouter »
7. Une ligne est ajoutée dans la liste dynamique (tableau dynamique).
8. Il peut également supprimer une ligne de la liste dynamique qui correspond à un cours.

Pour cela, nous avons utilisé les fichiers suivants :

Contrôleur :

- JsonServletModuleAdd : Servlet traitant la requête AJAX lors de l'ajout d'un cours dans la table ContenuModule
- JsonServletModuleDelete : Servlet traitant la requête AJAX lors de la suppression d'un cours dans la table ContenuModule
- ModuleDistributionServlet: Contrôleur de la page de découpage des modules

Vue :

- moduledistribution1.jsp : Vue HTML de la page du menu de découpage des modules
- moduledistribution2.jsp : Vue HTML de la page du découpage des modules

Analyse et Implémentation :

Lorsque le bouton « Découpage des modules » est cliqué, nous affichons tous les modules de la promo concernée dans un tableau. L'utilisateur sélectionne ensuite le module qu'il souhaite découper et tombe sur la page de découpage du module choisie. La vue de cette page se définit par 2 tableaux :

- tableau pour l'ajout des cours
- tableau pour la liste des cours ajoutés ou préalablement ajouté dans la table ContenuModule (liste dynamique)

Le premier tableau comporte plusieurs champs comme : le type du cours, son déroulement, le nombre de groupe (si le cours est en parallèle) et les enseignants à affecter. Le second tableau possède presque les mêmes informations : Partie, Type, nbHeures et Enseignant avec un bouton supprimer en fin de ligne.

Pour réaliser les opérations d'ajout et de suppressions de cours tout en ayant des opérations dynamiques sur la page, nous avons mis en place plusieurs ActionListener sur plusieurs composant ou champs de nos tableaux dont :

- liste déroulante du nombre d'heure : pour récupérer à la fois le nombre d'heures sélectionnées et à la fois pour mettre à jour le nombre max en fonction de du volumeHoraire max (donnée par le Webservice) et des cours déjà ajouté dans la BDD
- bouton Ajouter : actif si le nombre d'heure max est supérieur à 0 et à chaque clic effectue un ajout dans la BDD et dans le second tableau par le biais d'une requête AJAX.
- Liste déroulante du type : change le nombre d'heure max du volume Horaire en fonction du type choisie (CM, TD, TP) et récupère également le type.
- Checkbox Parallèle : si ce box est coché alors le cours s'effectue en parallèle. Nous ajoutons une colonne pour la sélection du nombre de groupe en parallèle (en réalité il s'agit d'un masquage/démasquage).
- Liste déroulante du nombre de groupe : change le nombre d'enseignant en fonction du nombre de groupe (max : 5).

- Liste déroulante des enseignants : permet de récupérer l'enseignant choisie mais également de faire un test, dans le cas du déroulement parallèle, si les profs de chaque groupes ne sont pas différent alors nous affichons un Alert().
- Bouton Supprimer : supprime la ligne et dans la BDD par le biais d'une requête AJAX

Lorsque le bouton « Ajouter » est cliqué, nous récupérons toutes les informations sélectionnées par l'utilisateur et nous l'ajoutons dans un JSON pour l'envoi d'une requête asynchrone AJAX vers sa servlet correspondante. Celle-ci va effectuer des requêtes INSERT dans la table ContenuModule. Ensuite, côté Client, nous ajoutons le cours dans le second tableau puis nous décrétons le nombre d'heure max du volume Horaire du type choisie.

Lorsque le bouton « Supprimer » est cliqué, nous récupérons la partie (au sens de la table ContenuModule) du cours et nous l'ajoutons dans un JSON pour l'envoi d'une requête asynchrone AJAX vers sa servlet correspondante. Celle-ci va effectuer des requêtes DELETE dans la table ContenuModule. Ensuite, côté Client, nous supprimons le cours dans le second tableau puis nous incrémentons le nombre d'heure max du volume Horaire du type choisie.

Pour assurer la cohérence des informations entre la BDD et notre liste dynamique ainsi que pour nos opérations d'ajouts et de suppressions de cours, nous avons implémenté beaucoup de fonctions JS permettant d'effectuer les manipulations correctement et dynamiquement.

Problèmes / Bilan :

Il aurait été possible de regrouper/fusionner l'interface du cas 2 et du cas 3 ensemble qui nous permettait d'avoir une vue multifonctionnalités. Cela a été séparé pour des raisons de modularité et de tests successifs. De plus l'interface d'ajout/suppression de cours aurait pu être encore plus dynamique en ajoutant des mini-fenêtres permettant de guider l'utilisateur.

Nous avons ajouté une contrainte concernant les cours en parallèle sur le fait que chaque enseignant de chaque groupe doit être différent. Par exemple, nous ne pouvons pas découper 2 cours en parallèle comme ceci : TP 2 G1 et TP 2 G2 assuré par Mr Barreaud. Pour effectuer les vérifications des enseignants de chaque groupe, nous avons ajouté les enseignants dans une liste puis supprimer les doublons. Ensuite nous effectuons une simple comparaison entre la taille de la liste sans doublons et le nombre de groupe choisi.

La valeur partie pour chaque cours est créée par une fonction JS qui s'auto-incrémente en fonction du dernier cours de la liste. Il aurait été possible de donner la liberté à l'utilisateur de mettre la partie voulue.

5. Cas 4 : Modifications des descriptions des modules (non implémenté)

Analyse et Implémentation Possible:

Ce cas optionnel n'a pas pu être implémenté par manque de temps. Toutefois, nous avons pu analyser les différentes étapes nécessaires pour réaliser ce cas.

Tout d'abord, nous récupérons les descriptions des modules grâce au Webservice. Pour cela nous modifions les classes java fournies par l'enseignant et nous ajoutons/modifions les champs `courseDescription` et `learningObjectives`. Ainsi nous aurons récupéré toutes les descriptions sous la forme d'objets JAVA.

Ensuite nous affichons ces descriptions dans une vue JSP. Elle aura une interface interactive similaire au cas 3 (Ajout/Suppression dans la liste dynamique). Des fonctions JS seront sûrement nécessaires pour la dynamique de l'affichage dans un tableau. Enfin, pour chaque modification effectuée nous les rajoutons dans les fichiers XML par le biais du Webservice avec une méthode POST (méthode prédéfinies dans les classes fournies).

III. Répartition du travail et intégration

Pour organiser le déroulement du projet nous avons mis en place un diagramme de Gantt afin de rendre le projet réalisable dans les temps et évoluer d'une manière organisée.

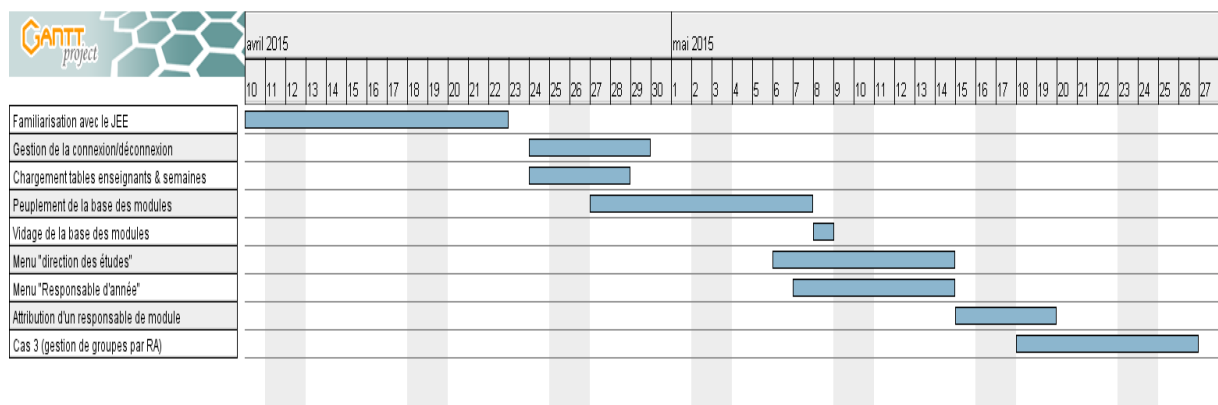


Diagramme de Gantt du projet

Répartition des tâches :

La première idée qui nous est venue quant à l'organisation de la suite du travail était de nous le partager par couche (Modèle Vue Contrôleur). Mais, finalement nous avons tous travaillé sur les différents cas, car cela est plus utile pédagogiquement et par rapport à notre apprentissage, il est meilleur que chacun de nous manipule toutes les couches. Ainsi, nous avons plutôt travaillé sur les différents cas ensemble en se répartissant les sous fonctionnalités de chaque cas.

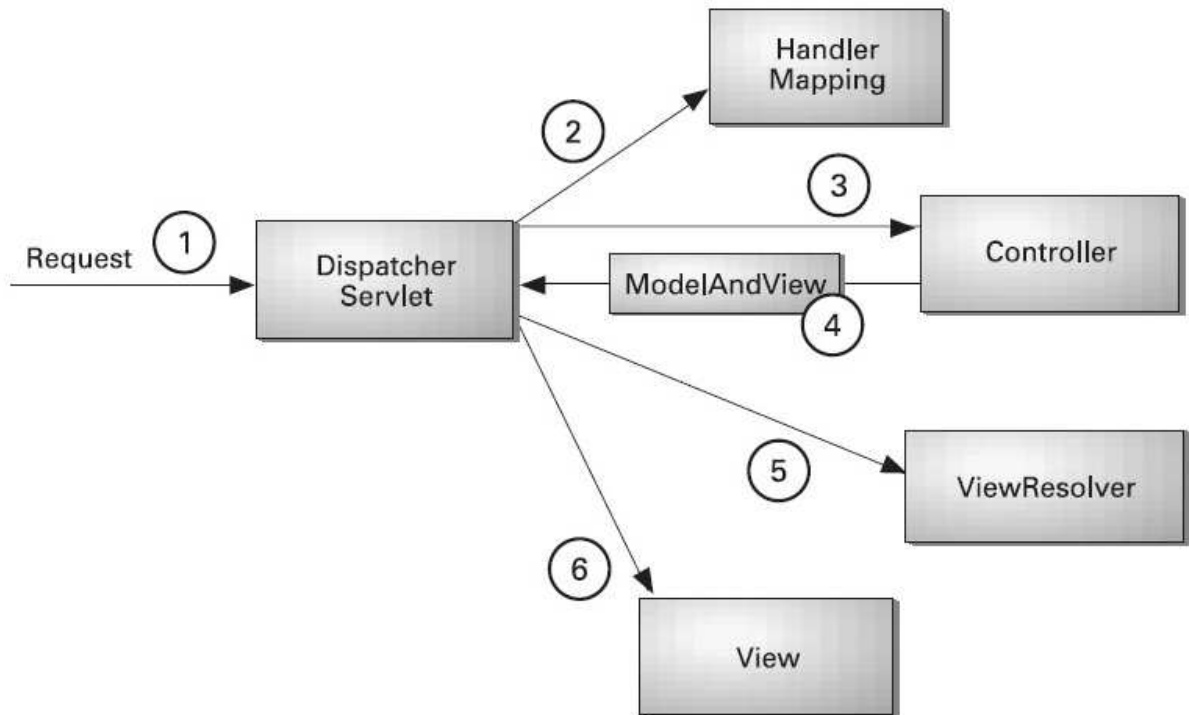
En ce qui concerne la phase d'intégration, nous n'avons pas eu beaucoup de mal pour cela car bien que chacun travaillait sur son PC, nous étions ensemble durant toutes les heures de travail sur le projet et nous testions chaque fonctionnalité à fur et à mesure de notre avancement. Cela nous a permis un gain énorme de temps de tester chaque partie avant de passer à la suivante et nous avons réussi à finir le projet dans les temps.

Dans l'ensemble le projet s'est bien déroulé malgré quelques manques techniques qui nous ont parfois ralenti mais nous trouvions toujours rapidement une solution. Comme nous avons travaillé ensemble et que nous discutons sur tous les cas nous n'avions pas particulièrement besoin de faire des réunions, le point sur l'avancement dans notre projet se faisait automatiquement au début de chaque séance de travail.

IV. Frameworks : Spring et bootstrap

Cycle de vie de Spring

Le Framework Spring MVC permet de construire et définir l'infrastructure de notre application. En effet, nous avons découpé notre application en 3 couches principales, notamment : Le contrôleur, le service et le DAO. Dans cette partie, nous n'allons pas détailler comment fonctionne exactement Spring MVC, mais nous allons plutôt mettre l'accent sur le cycle de vie de Spring, c'est-à-dire, comment Spring gère la requête d'un client du début de la demande jusqu'à l'envoi de la réponse. La figure ci-dessous met en lumière le fonctionnement du Framework Spring par rapport à la réception d'une requête du client.



1. Le cycle de vie du Framework Spring

Ainsi, nous pouvons décrire les étapes par leurs ordres d'exécution :

Le DispatcherServlet reçoit une requête dont l'URI-pattern

Le DispatcherServlet consulte son Handler Mapping pour connaître le contrôleur. En effet avec Spring MVC, nous créons nos contrôleurs qui remplacent les "vraies Servlets" dans une application développée sans ce Framework.

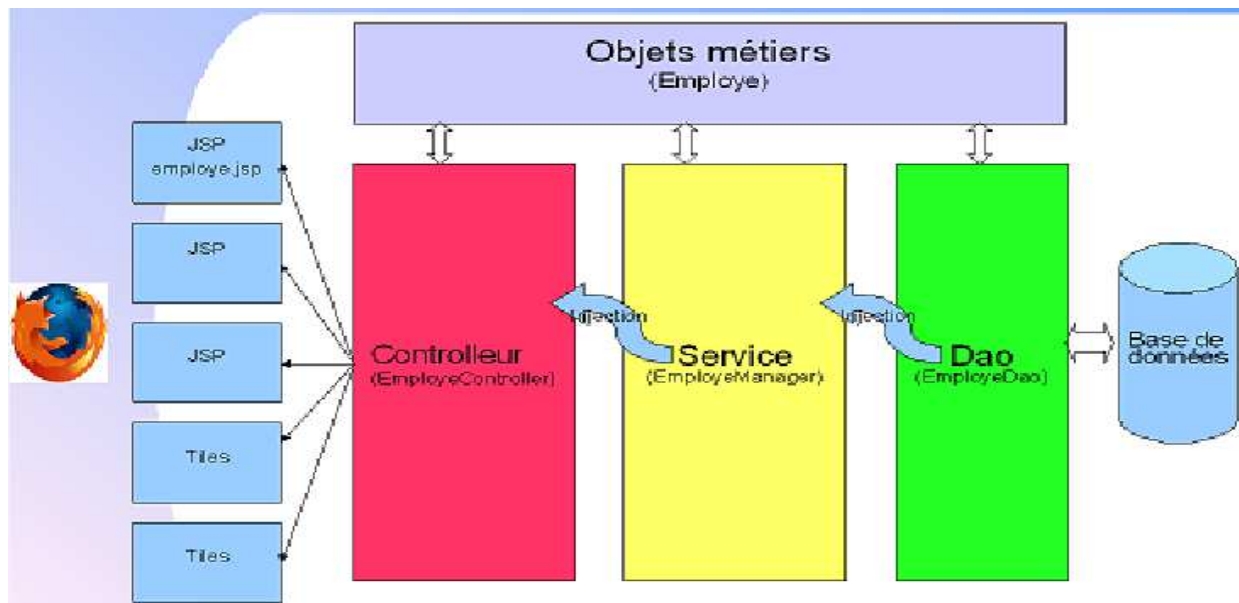
Le DispatcherServlet dispatche la requête au contrôleur identifié grâce à l'annotation @RequestParam qui identifie quelle action appliquer pour une telle requête.

Le contrôleur retourne au DispatcherServlet un objet de type ModelAndView possédant comme paramètre au minimum le nom logique de la vue à renvoyer.

Le DispatcherServlet consulte son View Resolver lui permettant de trouver la vue. Ici le type de View Resolver choisi est InternalResourceViewResolver et permet de concaténer le préfixe qui est 'xxx/' avec le suffixe '.aspx' afin de retrouver le chemin relatif dans notre application.

Le DispatcherServlet redirige la requête à la vue associée.

2. Architecture du Framework Spring MVC



Architecture d'une application basée sur Spring MVC

La partie précédente décrit le cycle de vie du Framework Spring MVC. Dans cette partie, nous allons mettre en avant la composition ou l'architecture de notre application basée sur ce Framework. Tout d'abord, un client envoie sa requête via une URI. Ensuite, le DispatcherServlet qui est le contrôleur général de Spring intercepte cette requête pour la renvoyer au bon contrôleur. Cette opération de "routage" est exécutée grâce à la composante de Spring "HandlerMapping" susmentionnée ci-dessus. HandlerMapping évalue le contenu de l'annotation @RequestParam pour nous rediriger vers le bon contrôleur. En conséquence, le contrôleur choisi renvoie la demande au métier qui exécutera la requête. Pour ce faire, le métier sollicite la couche DAO (DATA ACCESS OBJECT) qui permet d'interagir avec la base de données afin d'effectuer une opération particulière sur la base de données et nous renvoyer la réponse via de DAO puis le métier jusqu'au renvoi de la page.jsp.

Il importe de souligner le mécanisme d'injection des dépendances entre les différentes couches de notre framework Spring MVC-core. En effet, dans le fichier spring-beans.xml, nous créons les beans dont on aura besoin. Nous "injectons" un bean de type DAO dans notre classe métier et nous injectons un bean de type métier ou service dans notre classe contrôleur afin d'établir une communication entre nos différentes couches.

V. Manuel d'installation

1. Le serveur à installer

Notre application prend en compte le serveur Tomcat version 7. Il faut le télécharger, l'installer et configurer votre IDE.

Sous Eclipse:

- 1) Clic droit sur le projet créé et ouvert
- 2) Build path
- 3) Server
- 4) Choisir Tomcat v.7
- 5) Apply et OK

Il faut aussi télécharger et installer Node.js, puis lancer le terminal de commande de node.js à partir du menu « Démarrer » en choisissant « Node.js command prompt ». Il faut ensuite se positionner dans le répertoire du projet et saisir la commande *node nomfichier (index)*.

Il faut lancer le serveur « WAMPSERVER » et taper l'url suivante : ***http://localhost:8080/PlanningParSemaineJ2EE/*** dans un navigateur web, Google Chrome ou Mozilla Firefox par exemple.

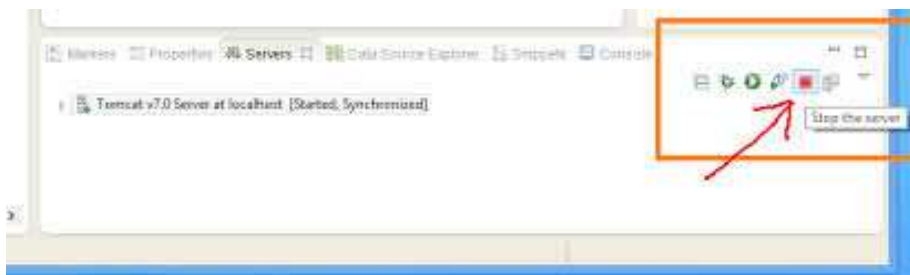
2. Pour ouvrir la page d'identification

Il faut saisir l'url suivante dans un navigateur web :

http://localhost:8080/PlanningParSemaineJ2EE/

3. J'ai fait toutes les installations demandées, j'ai lancé l'application et il ne marche pas ?

Il faut toujours s'assurer que votre serveur Tomcat tourne. Il faut le redémarrer souvent.



Dans cette image on appuie sur le bouton rouge pour l'arrêter. Pour le redémarrer c'est le bouton vert qu'il faut utiliser.

VI. Cahier d'utilisateur

En arrivant sur le site, vous devriez normalement être sur la page d'accueil du site qui est une page d'identification. Sur cette page se trouvent deux champs qu'il faudra remplir pour accéder aux fonctionnalités de notre application selon le service auquel appartient l'utilisateur (direction des études ou responsable d'année).

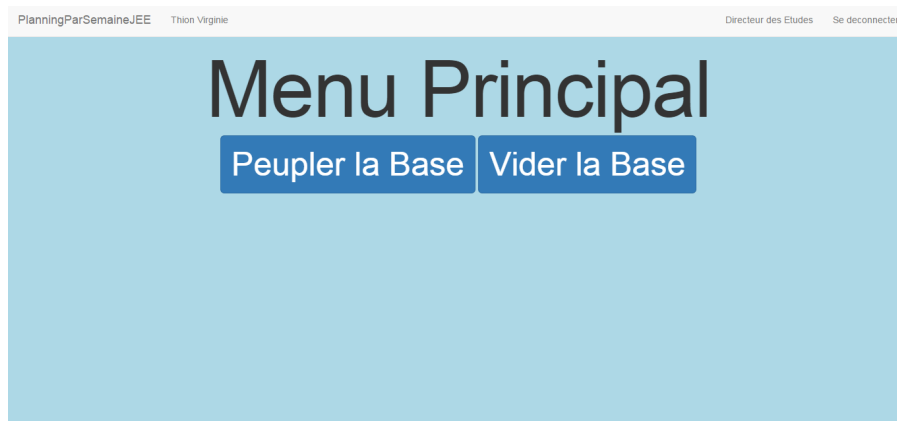
- Lorsque vous n'êtes pas connecté, aucune page n'est accessible
- Lorsque vous vous connectez, vous accédez à l'une des deux pages suivantes :
 - Menu RA (responsable d'année)
 - Menu DDE (direction des études)

1. Comment s'identifier ?

Pour s'identifier, il faut vous rendre sur la page **<http://localhost:8080/PlanningParSemaineJEE/>**. Une fois sur la page, vous devez remplir le formulaire «identification » avec votre login et votre mot de passe.

Si le login et le mot de passe que vous avez choisis sont corrects et que vous avez le droit d'accès, vous serez dirigé à l'une des pages MenuRA ou MenuDDE selon votre appartenance. En revanche, si le login ou le mot de passe sont incorrects ou que vous n'avez pas le droit d'accès, ou que vous n'avez rien saisi, un message d'erreur vous l'indique et vous devrez remplir à nouveau le formulaire en renseignant un login et un mot de passe corrects.

2. Quand vous vous connectez en tant que membre de la direction des études « DDE » :



Menu principal « Direction Des Etudes »

a) Comment faire le « peuplement » ?

Pour peupler la table des modules, il suffit de cliquer sur le bouton « Peuplement de la base », une nouvelle page s'affichera après quelques secondes. Cette page contient un listing de tous les modules qui ont été chargés. En effet, sur cette page, nous pouvons distinguer 2 couleurs. Toutes les lignes en verts, désignent les modules qui ont été chargés dans la table des modules avec succès. En rouge, les modules dont l'ajout a échoué, soit parce qu'ils existent déjà dans la table des modules ou que leur nom n'est pas compatible avec le code donné aux noms des modules ou bien pour une autres raisons.

En bas de la page, à la fin du listing des différents modules et du message qui les accompagne, un message est affiché pour dire que le peuplement est terminé et que la requête a été satisfaite.

b) Comment vider la base ?

Pour vider la table des modules, il suffit de cliquer sur le bouton « Vider la table ». Un message s'affichera en haut de la page pour demander une confirmation. Si la confirmation est faite, un autre message s'affiche en vert pour informer l'utilisateur que la table a bien été vidée.

Peuplement de la BDD

Connexion à la base de données
Connexion à la base de données réussie
Insertion du module:CO_TL5PIM libelle:Projet d'intégration dans la base de données effectuées avec succès
Insertion du module:CO_TL3PMM libelle:Projet Systèmes Numériques 2 dans la base de données effectuées avec succès
Insertion du module:CO_TL3AC2M libelle:Algorithmique et langage C 2 dans la base de données effectuées avec succès
Insertion du module:CO_TL3SA2M libelle:Structures algébriques dans la base de données effectuées avec succès
Insertion du module:CO_TL4ICAM libelle:Gestion de circuit d'interface dans la base de données effectuées avec succès
Insertion du module:CO_TL4TNM libelle:Communication Numérique dans la base de données effectuées avec succès
Insertion du module:CO_TL5MIMM libelle:Multimédia et interactions multimodales dans la base de données effectuées avec succès
Insertion du module:CO_TZ5ETRAR libelle:Contrat d'études dans université à l'étranger dans la base de données effectuées avec succès
Insertion du module:CO_TL5PIIM libelle:Projet d'intégration dans la base de données effectuées avec succès
Insertion du module:CO_TL5RGM TM libelle:Réseaux pair à pair dans la base de données effectuées avec succès

Résultat d'un peuplement

Insertion du module:CO_TZ5MASTR libelle:Partie théorique Master SISEA dans la base de données effectuées avec succès
Erreur lors de l'insertion de : CO_TZ5ETRAR Cause:Duplicate entry 'CO_TZ5ETRAR' for key 'PRIMARY'
Insertion du module:CO_TZ3COM libelle:Composants et Systèmes Optroniques dans la base de données effectuées avec succès
Insertion du module:CO_TZ3CMM libelle:Conférences sur les métiers dans la base de données effectuées avec succès
Insertion du module:CO_TE4MBDM libelle:Modèles et bases de données dans la base de données effectuées avec succès
Insertion du module:CO_TE4FSEM libelle:Fondements des systèmes d'exploitation dans la base de données effectuées avec succès
Insertion du module:CO_TE4UPM libelle:Unix : programmation dans la base de données effectuées avec succès
Insertion du module:CO_TZ3AC1M libelle:Algorithmique et langage C 1 dans la base de données effectuées avec succès
Insertion du module:CO_TZ3ETM libelle:Etudes thématiques dans la base de données effectuées avec succès
Insertion du module:CO_TZ3CE1TP libelle:Composants et Fonctions électroniques 1 dans la base de données effectuées avec succès

En rouge, une erreur d'insertion à cause de la présence de plusieurs occurrences du même module

3. Quand vous vous connectez en tant que responsable d'année « RA » :



Menu principal d'un « Responsable d'Année RA »

a) Comment désigner un enseignant comme responsable de module ?

Pour désigner un enseignant comme responsable de module, cliquer sur le bouton « Liste des modules ». Vous arriverez alors sur une nouvelle page contenant la liste de tous les modules de votre filière. Les modules sont affichés sous forme de tableau avec un module par ligne. Vous pouvez sélectionner l'enseignant dans la liste déroulante en fin de ligne. Cliquez ensuite sur le bouton « Valider » en fin de page pour valider votre choix.

Liste des Modules				
Identifiant	Libelle	Public	Enseignant Actuel	Enseignant à Affecter
CO_TEL3MNM	Mise à niveau en maths	LSI1	bvozel	bvozel ▼
CO_TL3AC2M	Algorithmique et langage C 2	LSI1	glecorve	glecorve ▼
CO_TL3BDDM	Bases de données	LSI1	fgoasdoue	fgoasdoue ▼
CO_TL3DOM	Développement objet	LSI1	null	null ▼
CO_TL3LMM	Logique mathématique	LSI1	null	null ▼
CO_TL3MDDM	Modèles de données	LSI1	null	null ▼
CO_TL3PMM	Projet Systèmes Numériques 2	LSI1	null	null ▼
CO_TL3PWM	Programmation WEB	LSI1	null	null ▼
CO_TL3SA2M	Structures algébriques	LSI1	null	null ▼
CO_TL3SDDM	Structures de données	LSI1	null	null ▼
CO_TL3SN2M	Intro Systèmes Numériques 2	LSI1	null	null ▼
CO_TL3UUM	Unix : utilisation	LSI1	null	null ▼

La liste des modules : pour désigner l'enseignant responsable de chaque module

b) Comment gérer le déroulement parallèle et/ou en séquence des CM/TD/TP ?

Pour répartir les cours d'un module, cliquer sur le bouton « Découpage des modules ». Vous arriverez sur une nouvelle fenêtre contenant les modules de votre filière. Cliquer sur un module. Vous arriverez sur une nouvelle page où vous pourrez répartir le cours en CM/TD/TP. Pour cela, sur la première ligne de la page, sélectionné le type de cours voulue grâce à la liste déroulante. Cochez ensuite la case si vous voulez que les cours se fassent en parallèle ou non. Choisissez ensuite le nombre d'heures grâce à la liste déroulante et l'enseignant avec la liste déroulante. Cliquez ensuite sur valider pour ajouter ce découpage. Si vous choisissez de mettre les cours en parallèle, cliquez sur la liste déroulante pour choisir le nombre de groupes à constituer et assigner un enseignant à chaque groupe grâce à la liste déroulante.

PlanningParSemaineJEE

Jaudoin Hélène

Responsable Année LSI1 (30 élèves)

Se deconnecter

Découpage des Modules

Module	Libelle
CO_TL3AC2M	Algorithmique et langage C 2
CO_TL3BDDM	Bases de données
CO_TL3DOM	Développement objet
CO_TL3SN2M	Intro Systèmes Numériques 2
CO_TL3LMM	Logique mathématique
CO_TEL3MNM	Mise à niveau en maths
CO_TL3MDDM	Modèles de données
CO_TL3PWM	Programmation WEB
CO_TL3PMM	Projet Systèmes Numériques 2
CO_TL3SA2M	Structures algébriques
CO_TL3SDDM	Structures de données

La liste des modules pour effectuer un découpage

PlanningParSemaineJEE
Jaudoin Hélène
Responsable Année LSI1 (30 élèves)
Se deconnecter

Découpage du Module : CO_TL3MDDM

Type	Déroulement	Groupe	Nb Heures	Enseignant	
TD ▼	<input checked="" type="checkbox"/> Parallèle	3 ▼	1 ▼	hjaudoin ▼ vbarreud ▼ opivert ▼	Ajouter

Partie	Type	Nb Heures	Enseignant	
TD 1 G1	TD	1	hjaudoin	Supprimer
TD 1 G2	TD	1	vbarreud	Supprimer
TD 1 G3	TD	1	opivert	Supprimer
CM 2	CM	8	hjaudoin	Supprimer

< Precedent
Retour au Menu

Exemple de découpage du module « Modèle de données »

VII. Tests

Pour tester les fonctionnalités vous pouvez utiliser les login suivants selon votre souhait.

Une fois sur la page d'accueil, pour vous connectez vous pouvez selon la fonctionnalité que vous souhaitez tester utiliser :

- Pour se connecter en tant que direction des études, utiliser le login *vthion* et utiliser le mot de passe *servicesENSSAT*
- Pour se connecter en tant que responsable d'année, utiliser le login *hjaudoin* et utiliser le mot de passe *servicesENSSAT*

Conclusion

Ce projet nous a permis de cerner les réelles difficultés qu'un développeur d'applications pourrait rencontrer au quotidien. En termes de difficultés, nous avons consacré un temps très important pour se familiariser avec le JEE. En termes de valeur ajoutée, ce projet nous a beaucoup apporté sur le plan personnel, car il nous a permis de nous confronter aux difficultés que nous pourrions rencontrer lors de développement d'application dans le monde professionnel. Par essence, l'idée du choix des frameworks venait du fait qu'ils sont abondamment utilisés dans le monde professionnel. Ainsi, nous avons pu mettre en œuvre des nouvelles technologies d'une importance non négligeable qui nous serviront sans nul doute dans les projets à venir.

