

Génie logiciel avancé - 2017

TD n°2

Système de planification des vols

La présentation faite durant le TD précédent équivaut à un document contractuel.

Ce document est un livrable à part entière et doit être complété avec toutes les informations nécessaires à la bonne compréhension du projet par un intervenant externe.

L'objectif de ce TD est de compléter ce livrable pour le dimanche 5 février au plus tard.

Il doit être :

- mis sur la plateforme en ligne du cours ;
- rédigé en anglais ;
- soigné, la qualité visuelle de la présentation est importante.

Il est important de s'assurer au fil des étapes que toutes les fonctionnalités indiquées dans le sujet du projet sont bien incluses dans votre livrable.

Décomposition en composants

A minima, le système doit contenir les composants suivants :

- Un système de persistance. Le système à utiliser est Elasticsearch.
- Un serveur qui s'appuie sur la couche de persistance pour proposer des web-services REST. Le serveur d'application utilisé doit être Jetty (version 9) et les web-services seront réalisés avec le framework Jersey ;
- Une interface graphique pour le CCO ;
- Une interface graphique pour les équipages ;
- Un système d'alerte. Ce système d'alerte utilisera la librairie « Quartz » (<http://www.quartz-scheduler.org/>).

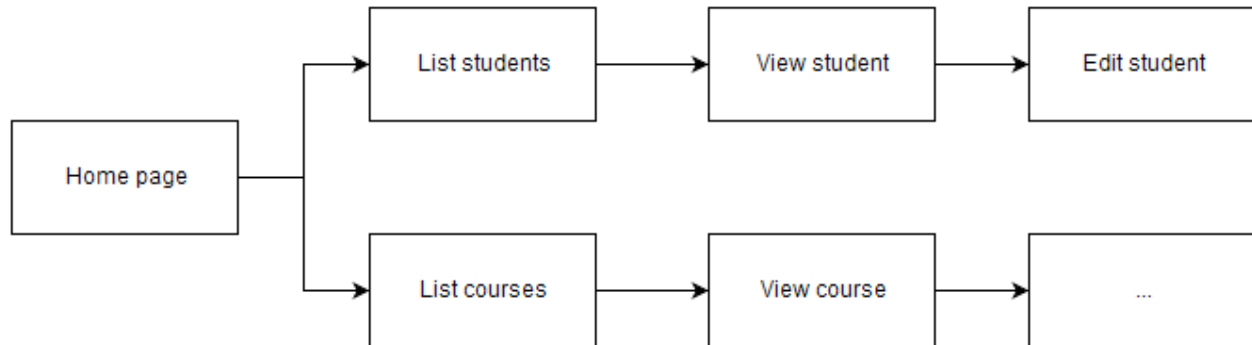
Chaque interface graphique doit-être réalisée en HTML et en Javascript. Aucun framework Javascript n'est imposé. Ces interfaces ne doivent pas reposer sur des technologies PHP, JSP ou ASP. Le code Javascript présent dans les pages doit appeler les web-services afin de charger les données et les afficher.

L'IDE utilisée pour le projet sera Eclipse, une structure de départ vous sera fournie au prochain TD.

Question 1 – Schéma de navigation

Pour chaque interface graphique, réaliser un schéma de navigation incluant tous les écrans ainsi que les liens importants permettant de passer d'un écran à un autre.

Voici un exemple d'un tel schéma réalisé à partir de l'outil « Draw.io » :



Question 2 – Mock-ups

Pour chaque écran des deux schémas de navigation, réaliser son mockup. Le mock-up peut contenir des notes afin de commenter les fonctionnalités.

Voici un exemple d'un tel mock-up réalisé à partir de l'outil « Pencil » :

List of students

This list contains all the students registered to the "Génie logiciel avancé" course

| <input type="checkbox"/> | Firstname | Lastname | Phone number |
|-------------------------------------|-----------|----------|----------------|
| <input checked="" type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |
| <input type="checkbox"/> | John | Doe | +33 1 23 45 67 |

1 | 2 | 3 »

Question 3 – Web-services

A partir des mock-ups, il faut définir les web-services nécessaires pour charger les données.

Quatre types de web-services REST peuvent être utilisés :

- GET : pour récupérer des données depuis le serveur
- PUT : pour ajouter un nouvel élément
- POST : pour modifier un élément existant
- DELETE : pour supprimer un élément

Chaque web-service doit être défini par :

- Son type (GET, PUT, POST ou DELETE)
- Son url
- Son comportement

Définir l'ensemble des web-services nécessaire au bon fonctionnement des interfaces graphiques ainsi qu'au système d'alerte.

Quelques exemples de description sont donnés ci-dessous :

| Type | URL | Behavior |
|---------------|---------------|--|
| GET | /student | Returns the list of students |
| GET | /student/{id} | Return details of student for the corresponding "id" |
| POST | /student/{id} | Modify the details of student for the corresponding "id" |
| PUT | /student | Add a student |
| DELETE | /student/{id} | Delete a student |