

**I**nstitut **S**upérieur

d'**I**nformatique, de **M**odélisation

et leurs **A**pplications

TP de Grails

Segmentation Fault

RABERIN Alexandre & ELYSEE Jonas

Année scolaire 2015/2016

Encadrant : PLANTIN Alexis

Sommaire

Introduction

I- Travail à faire

A – Consigne

Dans le contexte du cours de Développement Rapide, nous avions à développer une application qui reprenait le concept du site web [www.StackOverflow.com](http://www.StackOverflow.com).

L'application permet donc à un utilisateur de poser des questions sur un forum; ces questions seront consultables par tous les utilisateurs de l'application.

L'utilisateur aura besoin de s'authentifier (et donc de posséder un compte) afin de poser une question, de répondre à une question ou encore commenter la réponse d'un autre utilisateur.

Chaque utilisateur aura donc un profile qui sera consultable, et qui affichera diverses informations sur l'utilisateur telles que la date à laquelle il s'est inscrit et un résumé de l'ensemble de ses activités.

Il existera un ensemble de catégories qui permettra de regrouper les questions autour de certains sujets.

Les questions (et les réponses à ces questions) seront sujets à un système de vote qui indiquera si la question est intéressantes ou si une ou plusieurs réponses répondent bien à la question posée.

Un système de gratification devra exister qui récompense les utilisateurs selon certaines actions effectuées.

B – Spécificités

Certaines contraintes se rajoutent à la consigne énoncée précédemment:

* L'application devra être internationalisée, c'est-à-dire qu'il devra prendre en compte la langue utilisée par le système de l'utilisateur et s'afficher en "correspondance".
* L'application devra être codée en Anglais
* L'application devra être testée, c'est-à-dire qu'un système de tests unitaires devra être mis en place enfin de tester les différentes fonctionnalités mises en place.

C – Outils Utilisés

Afin de développement ce projet, certains outils sont mis à disposition afin d'alléger le travail à effectuer. En effet, dans le cadre du cours, l'outil de développement d’application web Grails a été utilisé. Grails s’appuie sur l’utilisation du langage Groovy pour effectuer le code de notre application.

Concernant l’environnement de développement, nous avons utilisé IntelliJ qui possède un support du framework Grails.

II – Travail de Conception Réalisé

A – UML

La première étape du développement de l'application revient à isoler les entités que nous allions manipuler tout au long du développement et du fonctionnement de notre application. Pour cela, nous avons construit le diagramme UML suivant qui donne une vision générale de notre projet.

* Insert Diagram

« Commentaire diagramme »

B - Maquettes

Sachant que notre application serait une application web, il a donc un aspect graphique à prendre en compte. La deuxième étape à réaliser a donc été de savoir à quoi nous voulions que notre application ressemble. Pour cela, nous avons utilisé le programme "Balsamiq" afin de créer différents écrans de notre application, ce qui permet d’avoir une meilleure appréhension lors de la réalisation des vues de l’application.

* Exemple d'écran, le reste en annexe

III – Produit Final

A – Architecture

La charge se répartie sur différents controlleurs qui se chargent de gérer différentes parties de l’application. Chaque controlleur, en fonction des besoins du site web, fait appel à des services qui sont en chargent de gérer des processus plus complets comme par exemple la logique d’ajout de réputation. Ceci explique pourquoi les différents services peuvent eux même avoir des relations avec d’autres services. Cette manière de procéder nous permet par la même occasion de factoriser des logique métiers et donc ne laisser que la logique du contrôle sur les différentes routes aux controlleurs.

L’application repose sur un mécanisme de permissions qui permet de créer une interface différente en fonction des privilèges accordés aux différents utilisateurs. On notera principalement la possibilité de poster une question ou non en fonction de l’état connecté ou non de l’utilisateur, etc…

Notre application fait aussi l’objet d’une série de tests qui ont été mis en place afin de tester les différents modèles (domain class) et controllers que nous avons créés.

1 – Détails des controlleurs

TODO

2 – Détails des services

TODO

B – Ce qu'il reste à faire

D’un point de vue fonctionnel, notre application possède la majorité des fonctionnalités attendues pour une imitation du site StackOverflow.

Il resterait cependant un travail assez conséquent à réaliser sur l’aspect visuel du site. Cela passerait par la mise en place d’une charte graphique spécifique au site. Notre production, s’appuie sur différents bootstrap de sources diverses afin de produire un visuel un peu plus agréable. Cependant, il est certain que ceci n’est pas envisageable pour une application finale, mais n’aura pas pu être mis en place dans le temps imparti.

Conclusion

Putin j'en ai marre

Pas assez de temps