
PROJET HFGL

Document d'architecture du logiciel

Version 1.0

HFGL	Version: 1.0
	Date: 16/05/2014

Table des matières

1.	Introduction	3
1.1	Objectifs	3
1.2	Portée	3
2.	Objectif du logiciel	3
2.1	Contexte	3
2.2	Besoins fonctionnels	3
2.3	Besoins non fonctionnels	3
3.	Structure	4
3.1	Vue des couches	4
3.1.1	La couche Interface	4
3.1.2	La couche Contrôle	4
3.1.3	La couche Model	5
4.	Comportement	6
4.1	Réalisation des cas d'utilisation	6
5.	Autres vues	7
5.1	Vue implémentation	7
5.2	Vue données	7
6.	Concepts du domaine	8
7.	Qualités de l'architecture	9

HFGL	Version: 1.0
	Date: 16/05/2014

Document d'architecture HFGL

1. Introduction

1.1 Objectifs

Ce document fournit une vue globale de l'architecture du système, de haut niveau, grâce à un certain nombre de vues qui décrivent les différents aspects du système. Ces vues montrent les décisions significatives prises concernant l'architecture du système qui influenceront la conception.

1.2 Portée

Le document d'architecture logiciel est destiné à l'architecte, aux concepteurs et aux superviseurs du projet. Toute modification sur le document d'architecture du logiciel sera répercutée sur les cas d'utilisation de l'application.

2. Objectif du logiciel

2.1 Contexte

Le projet HFGL sous le cadre du Génie Logiciel avancé mène à la réalisation d'une application avec une simulation graphique fonctionnelle d'une plateforme d'apprentissage. Autrement dit nous comptons programmer un site d'apprentissage permettant à chaque usager (étudiant) de rejoindre les cours proposé par un professeur aussi que de répondre aux questions / examens.

- modélisation des composants de l'application ;
- définition de la présentation des composants du site avec une interface graphique ergonomique dédié aux étudiants / professeurs ;
- Ajouter, Supprimer, Soumettre les cours et les devoirs

2.2 Besoins fonctionnels

Cette section présente les cas d'utilisation les plus importants pour l'architecture du logiciel (cf. Modèle des cas d'utilisation 1.0). Les cas d'utilisation que nous avons retenus sont les suivants :

Le système doit permettre de :

- Créer un nouveau compte utilisateur
- Connecter, Déconnecter, Supprimer un utilisateur
- Afficher une liste de cours correspondant aux critères définis par l'étudiant
- Inscrire, Désinscrire un étudiant à un cours

HFGL	Version: 1.0
	Date: 16/05/2014

- Télécharger ou mettre en ligne les notes de cours d'un cours et mettre le barème
- Télécharger l'énoncé d'un projet ou d'un mémoire.
- Charger, enregistrée ou finir une feuille d'exercice
- Accéder à tous les devoirs, exercices et examens que l'étudiant doit faire
- Indique le type de feuille d'exercice et de correction
- Faire les exercices des différents types et sa soumission

2.3 Besoins non fonctionnels

Cette section décrit les qualités et les contraintes importantes pour l'architecture du logiciel :

➤ Performance :

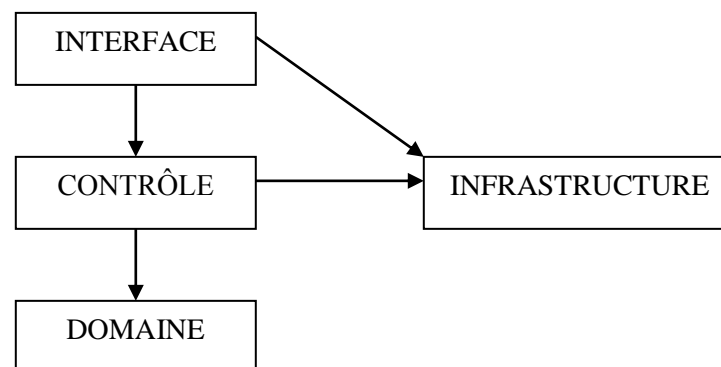
Temps de réponse : Le logiciel devra accomplir les fonctions demandées en un minimum de temps. Ainsi que d'avertir le client au cas où un problème de connexion sur le site ou lors de la soumission d'un projet ou examen.

3. Structure

Cette section décrit la structure statique de la vue logique de l'architecture, en montrant les composants, leurs interconnexions et les interfaces offertes par ces composants.

3.1 Vue des couches

Le diagramme suivant représente les couches du logiciel. Nous expliquerons ensuite le rôle de chaque couche.



HFGL	Version: 1.0
	Date: 16/05/2014

3.1.1 La couche Interface

Le rôle de cette couche est de fournir une interface à l'utilisateur pour qu'il puisse manipuler le logiciel. Elle va interagir avec la couche contrôle et va lui transmettre les données saisies et l'appeler en fonction des événements déclenchés par l'utilisateur.

La couche Interface ne gère que les aspects graphiques de l'application, c'est le niveau le plus haut dans l'architecture du logiciel et c'est la seule couche qui va interagir avec l'utilisateur :

Exemple : Interface « S'inscrire à un cours ».

3.1.2 La couche Contrôle

Le rôle de cette couche est de piloter les cas d'utilisation du logiciel. Chaque élément de la couche contrôle (contrôleur) est responsable d'un cas d'utilisation et c'est lui qui gère son fonctionnement. La couche contrôle agit sur 2 niveaux de l'architecture :

- Avec la couche Interface, afin de gérer l'Interface Homme-Machine
- Avec la couche Model, afin d'effectuer les traitements sur les objets métier de l'application, c'est-à-dire au niveau interne.

Exemple : Contrôleur « ControlleurProf ».

3.1.3 La couche Model

Le rôle de cette couche est de gérer les composants élémentaires de l'application, c'est-à-dire les objets Métier (ou entités), qui contient la logique métier.

4. Comportement

4.1 Réalisation des cas d'utilisation

.

HFGL	Version: 1.0
	Date: 16/05/2014

5. Autres vues

5.1 Vue implémentation

A Regarder sur Docs/Rapport.pdf.

5.2 Vue données

Cette section décrit les données persistantes du logiciel :

- La base de données en SQL.
- Génération du site Web projet (HTML/CSS/PHP MVC).

6. Concepts du domaine

Cette section décrit les concepts spécifiques du domaine et leurs relations.

Figure 1 : diagramme de classe des éléments du domaine

RATP	Version: 1.0
	Date: 02/04/2014

7. Qualités de l'architecture

➤ **Avantages :**

L'utilisation d'une architecture en couche de type Interface/Contrôle/Model permet de décomposer de façon claire la réalisation d'un cas d'utilisation. Ce système permet de déléguer à chaque couche une responsabilité lors de l'utilisation d'une fonctionnalité du logiciel. Chacune d'elles communiquera avec une ou plusieurs couches sous-jacentes.

➤ **Inconvénients :**

Les développeurs peuvent avoir des problèmes concernant la localisation de certaines structures de données, les accès entre composants de couches différentes...