

inQuire - Une application web de gestion de sondages

Groupe 5 - Thiry Jérôme & Vermeiren Rémy

May 16, 2015

Contents

1	Phase d'analyse	1
2	Choix de conception	2
2.1	Couche utilisateur : HTML, CSS, Javascript, Hogan	3
2.2	Noyau de l'application : Node.js	3
2.3	Couche base de données	4
3	Choix d'implémentation du système	4

Introduction

Ce rapport concerne le projet d'approfondissement en sciences informatiques, mis en route dans le contexte du cours LSINF1212 dispensé à l'Université Catholique de Louvain. Il nous a été demandé de concevoir une application web dont le but était laissé au choix. Nous avons donc décidé de créer une application de gestion de sondages qui permettrait, entre autres, de créer, remplir et gérer des sondages en ligne, ainsi que d'en voir les résultats. Ce rapport a pour but d'expliquer les différentes étapes de la création de cette application, de la phase d'analyse, en passant par sa conception jusqu'à son implémentation en elle-même.

1 Phase d'analyse

Comme dans tout projet, celui-ci a commencé par une phase d'analyse. Celle-ci a été entre autres réalisée en utilisant le développement par comportements, et, en particulier, en accord avec la convention *cucumber* dont l'algorithme 1 donne un extrait d'un exemple décrit en début de projet, et qui concerne la création de sondage. Ces différents travaux nous ont permis de cerner plusieurs besoins.

Algorithm 1 BDD - Création de sondage

Feature: Survey creation
In order to collect data
As a user
I want to create a survey
Scenario: User can create a survey
Given the user is registered
When the user requests to create a survey
Then the application proposes different settings
And the user fills in the settings
And the user confirms his choices
And the application creates the survey

On peut par exemple citer le fait que l'utilisateur doive être connecté afin de pouvoir créer un sondage. En effet, les résultats d'un sondage n'étant accessibles que par son créateur, il était nécessaire de faire ce choix.

Citons encore le fait de pouvoir avoir des sondages publics mais aussi des sondages privés. Cela nous semblait important car les besoins sont multiples. Une enquête entre amis n'a pas la même ampleur qu'une enquête pour une étude, par exemple. Cependant, nous avons été amené, après réflexion, à autoriser les utilisateurs non connectés à pouvoir remplir un sondage. Cela est moins contraignant pour l'utilisateur et permet donc plus de flexibilité. Un bon compromis entre ces deux choix a été de ne pas reprendre les sondages privés sur la page principale. Ils ne sont donc accessibles qu'à partir du permalink leur correspondant. Il en va de même si la date de clôture du sondage est dépassée.

Il nous a aussi semblé important de permettre aux utilisateurs de modifier leurs données personnelles, de pouvoir avoir une vue d'ensemble sur leurs sondages, ainsi que de proposer à tous les visiteurs du site une liste des sondages publics les plus récents.

2 Choix de conception

Cette section décrit les choix de conception, et entre autres l'architecture du système. Le système se base sur plusieurs technologies qui s'occupent de différentes parties du système.

L'application se divise en trois couches importantes. En premier lieu, nous trouvons la couche utilisateur, qui gère l'affichage et envoie des requêtes au serveur. Ensuite vient le noyau de l'application, qui gère les requêtes et, enfin, la couche base de données, implémentée en MongoDB, un type de bases de données non-relationnelles, qui stocke les données sous forme d'objet en Javascript.

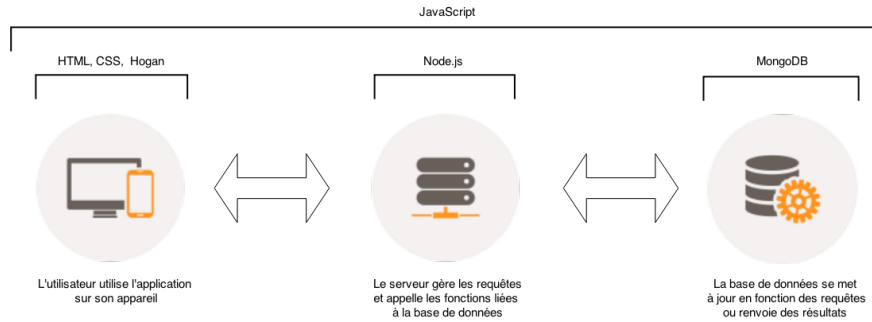


Figure 1: Couches de l'application

2.1 Couche utilisateur : HTML, CSS, Javascript, Hogan

Le dossier *views* contient des fichiers qui gèrent l'affichage de l'application dans le navigateur, entre autres les différentes pages HTML affichées sur le site. Ces pages utilisent le CSS de Bootstrap. Certaines pages HTML ne sont pas des pages complètes puisqu'elles ne servent qu'à venir s'insérer dans une autre page HTML. On trouve aussi les images (logos, background, ...) dans le dossier *images*, ainsi que des scripts Javascript, utiles pour changer de contenu dynamiquement en fonction des choix de l'utilisateur, notamment lors du choix du type de question à créer lors de l'initialisation d'un questionnaire.

Les pages HTML reçoivent des arguments lorsque le serveur indique qu'il faut les interpréter, qui sont alors affichés dans la page grâce à un outil de *templating*, Hogan.js.

2.2 Noyau de l'application : Node.js

Le noyau de l'application a été implémentée grâce à Javascript et Node.js. Celui-ci gère plusieurs opérations, dont entre autres :

- Dans *app.js* :
 - Le lancement du serveur ;
 - La liaison avec la base de données.
- Dans le dossier *persistance* : les différents fichiers gèrent les fonctions en rapport avec la base de données. Il y a un fichier par type d'objets que l'application utilise (utilisateurs, sessions, questions et réponses). On y trouve aussi bien des fonctions d'insertions que de récupération ou de mise à jour.
- Dans le dossier *routes* :

- Le routage des différentes pages de l’application, implémenté avec l’aide du module *Express* ;
- La gestion des requêtes GET et POST sur le serveur, c’est-à-dire les fonctions appelées lorsque l’on affiche une page (on peut alors l’interpréter et notamment lui passer des variables grâce à Hogan), où que l’on clique sur tel ou tel bouton par exemple. C’est ici aussi que l’on trouvera les appels aux fonctions du dossier persistence.

2.3 Couche base de données

3 Choix d’implémentation du système