
Concrétisation disciplinaire

Modélisation et conception d'un schéma pour des données issues d'études sur les risques au travail

M2 :

Hugues DUMONT
Guillaume HUET
Zineb LOUKILI

ESTER :

Natacha FOUQUET
Anna LLOYD

M1 :

Nidal BEDYOUCH
Imane BELHOUARI
Théo DÉZÉ
Charles MALLET

Référent :

Vincent BARICHARD
Laurent GARCIA

Table des matières

1	Introduction	1
2	Gestion du projet	2
2.1	Présentation du projet	2
2.2	Choix technologies	2
2.2.1	Technologie côté serveur	2
2.2.1.1	JEE	2
2.2.2	Technologie côté client	2
2.2.2.1	Highcharts	2
2.2.3	Base de données	2
2.2.3.1	Besoin	2
2.2.3.2	MongoDB	2
2.3	Planification et répartition des tâches	3
2.3.1	Outils utilisés	3
2.3.2	Diagramme de Gantt	3
2.3.3	Répartitions des rôles	3
3	Back-end	4
3.1	Base de Données	4
3.1.1	Choix	4
3.1.1.1	Besoin	4
3.1.1.2	NoSQL : MongoDB	4
3.1.2	Implémentation	4
3.1.2.1	Bilan	4
3.2	Connexion	5
3.3	Administration e-mail	5
4	Front-end	6
4.1	Structure du site et style	6
4.2	Questionnaire	6
4.3	Résultat	6
5	Conclusion	7

Chapitre 1

Introduction

Chapitre 2

Gestion du projet

2.1 Présentation du projet

2.2 Choix technologies

2.2.1 Technologie côté serveur

2.2.1.1 JEE

Nous avons proposer d'utilisé NodeJS côté serveur, car il s'intégrais particulièrement bien a MongoDB et propose une maniere simple d'intégrer des modules. De plus pour la parti formation nous aurions du simplement apprendre le javascript et ainsi éviter d'apprendre une autre technologie.

Mes nos chefs de projet on décider de choisir le JEE, car c'était une technologie qu'ils connaissait et cela nous permetais de découvrir la technologie qui nous sera utile pour le second semestre.

2.2.2 Technologie côté client

2.2.2.1 Highcharts

2.2.3 Base de données

2.2.3.1 Besoin

Pour le projet, nous avons eu besoin d'une base de données modulaire car un des besoins était que les questionnaires pouvait évoluer création, modification ou ajout de question. Et un autre des besoins était que l'utilisateur puisse faire des sauvegarde partiel pour reprendre le questionnaires en case de problème ou si l'utilisateur veut faire une pause.

En plus des besoins spécifique pour la sauvegarde des questionnaires et des réponses. Il y a des besoins plus génériques comme la gestion des comptes que nous verrons plus en détails dans un autre partie.

2.2.3.2 MongoDB

Nous somme partie sur du MongoDB qui fais partie de la mouvance NoSQL qui s'écarte du paradigme classique des bases relationnelles. Cela nous permet de nous affranchir d'une des contraintes des base de données SQL qui est de devoir définir un schéma prédéfini. Nous somme quand même partie d'un schéma de base pour avoir des données en partie structurée.

En offrant une plus grande flexibilité en permettant de gérer des données hétérogènes. Dans ce cas cela est particulièrement utile pour les questionnaires et les réponses car si un questionnaire est modifié, il faut que les anciennes réponses restent en partie utilisables.



FIGURE 2.1 – Logo de MongoDB

Ce choix du type de la base de données a été proposé par nos chefs de projet. La raison du choix de MongoDB est car il est le membre le plus populaire de la famille NoSQL.

2.3 Planification et répartition des tâches

2.3.1 Outils utilisés

2.3.2 Diagramme de Gantt

2.3.3 Répartitions des rôles

Chapitre 3

Back-end

3.1 Base de Données

3.1.1 Choix

3.1.1.1 Besoin

Pour le projet, nous avons eu besoin d'une base de données modulaire car un des besoins était que les questionnaires pouvait évoluer création, modification ou ajout de question. Et un autre des besoins était que l'utilisateur puisse faire des sauvegarde partiel pour reprendre le questionnaires en case de problème ou si l'utilisateur veut faire une pause.

En plus des besoins spécifique pour la sauvegarde des questionnaires et des réponses. Il y a des besoins plus génériques comme la gestion des comptes que nous verrons plus en détails dans un autre partie.

3.1.1.2 NoSQL : MongoDB

3.1.2 Implémentation

Pour l'implémentation nous avons utilisé le driver officiel proposer par MongoDB pour le Java. Nous avons crée un classe en Java qui nous héritons et que permet des interfaces pour simplifier l'utilisation dans le reste du code.

Nous avons deviser en trois partie, une première "Utilisateur" qui contient les information sur les comptes (Entreprise, Salarie et Utilisateur Ester).

Un autre "Questionnaires" (ID du questionnaire, ID de celui qui la soumis, Date de création, HTML du questionnaire).

Et enfin la partie "Réponses" (ID du question, ID de celui qui a répondu, réponses (ID question, réponse)). La modélisation de la parti "Utilisateur" correspond à celle fournis par les M2.

3.1.2.1 Bilan

La base de données est fonctionnel, mais la parti enregistrement questionnaires n'est pas optimal, faute de temps nous enregistrons le code HTML du questionnaire en dur sans le transformer en structure qui nous permettrai de récupérer la liste des questions et des réponses possible. Actuellement, il n'y a pas de calcul de score mais seulement celui du pourcentage de personne qui on répondu.

Pour avoir un version

Priorité	Nom	Raison
1	Sauvegarde des données en cours de saisie	Doit être vérifié en premier car sinon [...]
2	Tache 2	On doit pouvoir [...]
3	Enregistrement auto de la date	Comme les principales fonctionnalités permettant de tester sont opérationnelles, nous pouvons passer à cette tâche.
4	Création et gestion de comptes pour différents types d'utilisateurs	Parce que [...]
5	Sauvegarde des questionnaires	La tâche 5 fait partie des principales [...].
6	Sauvegarde de toutes les données traitées	Dernière fonctionnalité essentielle à mettre en place.
7	Tache 7	Non-essentiel, mais apporterait un plus au projet.
8	Tache 8	Non-essentiel, mais apporterait un plus au projet.

FIGURE 3.1 – Tableau récapitulatif des tâches

3.2 Connexion

3.3 Administration e-mail

Chapitre 4

Front-end

4.1 Structure du site et style

4.2 Questionnaire

4.3 Résultat

Chapitre 5

Conclusion