



Description du Cours et des Projets

1 Introduction

Le *projet d'approfondissement en sciences informatiques* (LINFO1212) va vous permettre d'explorer les technologies de développement web.

Le projet sera réalisé par **groupe de 3 étudiants**. Le choix des coéquipiers est libre. Le cours se compose de deux phases :

- une phase préparatoire, dans laquelle vous travaillez sur un projet bien défini pour vous familiariser avec les technologies pour le développement sur le web ;
- une phase de réalisation, dans laquelle vous travaillez sur une application web de votre choix, en réutilisant la technologie utilisée dans la phase préparatoire.

Le projet se fera sur base du langage JavaScript, tant côté serveur que côté client. L'interface utilisateur se fera classiquement sur base du duo HTML5/CSS3, gérée soit côté serveur (web 1.0), soit côté client (web 2.0). La persistance sera implémentée sur une base de données MongoDB. Le projet donnera une brève introduction à la recherche d'information et ses applications.

Outre les aspects technologiques, ce projet sera l'occasion de mettre en oeuvre des techniques de génie logiciel, comme l'analyse des besoins du client et la création de tests.

2 Enoncé du Projet Préparatoire

Dans le projet préparatoire, on va créer un site web pour la commune d'Ottignies-Louvain-la-Neuve. Ce site web permettra aux citoyens de la commune de signaler des incidents dans l'espace public. Un exemple d'un tel site web est le suivant :

<https://fixmystreet.brussels>

Votre site web doit confirmer aux spécifications suivantes.

La page principale est illustré à la Figure 1. Cette page se compose de :

- le titre du site web ;
- un champs de recherche pour trouver des incidents ;
- le nom d'utilisateur, si l'utilisateur s'est identifié ;

Titre avec logo		Champs de recherche	Nom de l'utilisateur
Lien pour ajouter un incident		Jour actuel: <input style="width: 100px;" type="text"/>	
Description	Adresse	Rapporté par	Date
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>

FIGURE 1 – Page principale

- un lien pour ajouter un incident ;
- le jour actuel, dans le format “1 janvier 2019” ;
- un tableau qui montre tous les incidents enregistrés. Pour chaque incident, on montre :
 - une description de l’incident ;
 - l’adresse de l’incident ;
 - le nom de l’utilisateur qui a enregistré l’incident ;
 - la date sur laquelle l’incident a été enregistré.

Si l'utilisateur veut enregistrer un incident, mais ne s'est pas encore identifié, la page web à la Figure 2 est montrée.

Sur cette page web, l'utilisateur peut s'identifier (si elle a déjà un compte), ou peut créer un compte. Pour s'identifier, l'utilisateur doit spécifier un nom d'utilisateur et un mot de clé.

Pour créer un compte, l'utilisateur doit spécifier un nouveau nom d'utilisateur, un nouveau mot de clé, une adresse e-mail et un nom complet.

Au moment que l'utilisateur s'est identifié, le site web se souviendra du nom de l'utilisateur, et le montre sur chaque page web.

Titre avec logo		Champs de recherche	Nom de l'utilisateur
Compte existant			
Nom de l'utilisateur	<input type="text"/>		
Mot de passe	<input type="password"/>		
<input type="button" value="s'identifier"/>			
Nouveau compte			
Nom de l'utilisateur	<input type="text"/>		
Mot de passe	<input type="password"/>		
Nom complet	<input type="text"/>		
Adresse e-mail	<input type="text"/>		
<input type="button" value="s'enregistrer"/>			

FIGURE 2 – Page d'identification

Titre avec logo		Champs de recherche	Nom de l'utilisateur
Signaler nouveau incident			
Description			
<input type="text"/>			
Adresse			
<input type="text"/>			
<input type="submit" value="soumettre"/>			

FIGURE 3 – Page de registration d'un incident

Après s'être identifié, pour enregistrer un nouveau accident, la page web à la Figure 3 est montrée. Sur cette page web, l'utilisateur doit spécifier :

- une description de l'incident ;
- l'adresse de l'incident.

Après qu'un incident a été soumis, l'incident est ajouté à la page principale, où la date de création de l'incident et le nom du créateur sont montrés.

Si l'utilisateur spécifie un mot de recherche dans le champs de recherche, le site web montre une liste d'incidents les plus pertinents pour les termes de recherche spécifiés.

Il y a beaucoup de fonctionnalité que l'on pourrait ajouter au site web :

- de la fonctionnalité pour enlever des incidents ;
- des images pour les incidents ;
- des cartes montrant les incidents.

Il n'est pas nécessaire d'ajouter cette fonctionnalité (mais il est permis, si au moins la fonctionnalités spécifiée est présente).

Pour créer le site web, on a besoin de plusieurs technologies différentes. Ces technologies seront introduits pendant les premiers semaines du cours. Cela veut dire qu'on va aussi créer le site web en plusieurs étapes :

Étape 1 : Maquette On va commencer avec la création d'une maquette du site web. En utilisant HTML et CSS, on va créer des pages web statiques; cela veut dire que la page principale montre la même liste d'incidents, et que les pages de création de compte et de création d'incidents ne sont pas encore capable de stocker les résultats soumis par l'utilisateur.

Étape 2 : Création d'un serveur On va créer un serveur qui peut montrer les pages statiques, en utilisant un serveur NodeJS écrit en JavaScript; néanmoins, dans ces pages statiques, le jour actuel est déjà affiché de façon dynamique;

Étape 3 : Création d'une base de données On va créer une base de données pour stocker les incidents, et on va ajouter des incidents à cette base de données, pour un utilisateur fixe. (Alors, dans cette étape on suppose encore que c'est toujours le même utilisateur qui utilise le site web.)

Étape 4 : Utilisateurs On va ajouter la fonctionnalité pour créer des comptes, et pour s'identifier comme utilisateur.

Étape 5 : Fonctionnalité de Recherche On va ajouter la fonctionnalité pour rechercher des incidents.

Il faut seulement soumettre les résultats d'étape 1 et 5. Néanmoins, il est attendu que vous travaillez sur le projet chaque semaine!!! Les instructions spécifiques pour la soumission sont données sur Moodle. Les délais sont spécifiés plus en détail ci-dessous.

3 Enoncé du Projet Final

Vous êtes libre du sujet de l'application web final que vous développerez, mais l'application doit utiliser la technologie étudiée dans le projet préparatoire :

- l'application doit utiliser une base de données;
- l'application a des utilisateurs;
- l'application nécessite la recherche d'informations;
- l'application est mis en œuvre en utilisant NodeJS, HTML, CSS et MongoDB.

Afin d'être sûr que votre application rentre dans le cadre du projet, il vous sera demandé de soumettre les spécifications avant de commencer avec la mise en œuvre.

Pour vous orienter dans vos recherches de sujet, pensez à plusieurs domaines proche de votre vie quotidienne, dans lesquels vous pourriez apporter un outil facilitant la vie des utilisateurs :

- Vie étudiante, kot à projets, ...
- Vie familiale
- Cadre professionnel de vos proches
- Sports, scoutisme, ...
- Citoyenneté participative,
- Open data (<http://opendata.digitalwallonia.be/dataset>),
- ...

Le projet final sera divisé en deux étapes :

1. La première étape visera à mener à bien l'analyse et conception du projet (spécifications et architecture) ; les résultats de cette étape doivent être soumis indépendamment. Dans votre description, vous devez inclure :
 - un croquis de l'interface (comme dans ce document-ci pour le projet préparatoire) ;
 - une description de la base de données requise par le projet ;
 - une description de la fonctionnalité de recherches d'informations.

Le document a au plus 3 pages.

2. La seconde étape sera consacrée au gros du travail du projet, et sera menée en autonomie par chaque groupe. Aucune séance encadrée n'est prévue, mais **des contacts par mail et d'éventuels rendez-vous sont envisageables en cas de besoin**. Cette seconde étape couvrira la seconde moitié du quadrimestre, et se terminera par la remise du projet avant le blocus.

Délivrables finaux La remise finale du projet compte

- Le code source
 - Le code doit inclure un fichier texte expliquant comment préparer, lancer et tester l'application. Ce fichier est typiquement appelé **README**, et il doit se trouver dans la racine du projet.¹ Vous êtes libres d'ajouter toute autre information utile pour les utilisateurs de votre application.
- Un rapport écrit
 - Le code doit être accompagné d'un rapport de maximum 10 pages en format PDF décrivant les choix d'analyse (besoins de l'utilisateur), de conception (architecture du système) et d'implémentation du système.
- Données d'exemple (facultatives)
 - Facultativement, vous pouvez inclure dans votre soumission un fichier de données d'exemple généré via la commande **mongoexport**. Voir slides « Persistance des données en MongoDB » disponibles sur Moodle.
Même si vous fournissez des données d'exemple, votre application sera testée aussi à partir d'une base de données vide. Veuillez à vérifier que l'application marche dans les deux cas.

Ces résultats finals du développement sont à remettre sur le site Moodle sous forme d'un seul fichier compressé.

Fichiers de grande taille Si jamais une de vos soumissions dépasse la taille limite acceptée par Moodle, soumettez à sa place un fichier texte comprenant deux lignes :

1. Une adresse à partir de laquelle on peut télécharger le fichier compressé.²

1. Il s'agit d'une pratique standard adoptée par la grande majorité des projets open source. Il est conseillé d'utiliser le format Markdown, et dans ce cas d'appeler le fichier **README.md**.

2. Vous pouvez utiliser à cette fin un service tel que [Transvol](#), Dropbox, Google Drive, etc.

2. Le code SHA-1 du fichier compressé. Pour calculer ce code
 - sous Linux, utilisez la commande `shasum`.
 - sous Mac OS, regardez <https://support.apple.com/en-us/HT201259>.
 - sous Windows, regardez <https://support.microsoft.com/en-us/kb/889768>.

4 Modalités pédagogiques

Neuf séances encadrées de deux heures sont prévues, permettant de présenter les principes théoriques et technologies utilisés dans la mise en oeuvre du projet. Des petits exercices de mise en pratique seront proposés après les séances. Des forums seront mis en place sur le site Moodle du cours pour permettre les questions et discussions à propos de ces exercices.

5 Aperçu du Semestre et Échéances

Les échéances sont toujours à **10:45** le matin du jour indiqué.

Date	Sujet du cours	Echéance
24 septembre	Introduction, HTML	Étape 1 du projet préparatoire
25 septembre	HTML, CSS	
2 octobre	HTML, CSS	
9 octobre	JavaScript	
15 octobre	Serveur NodeJS	
16 octobre	MongoDB Base de données	
30 octobre	Comptes d'utilisateurs	Spécifications projet final
6 novembre	Recherche d'informations	Étape du 5 projet préparatoire
13 novembre	Discussion spécifications	Remise du projet
20 décembre	Assistance ; discussions d'avancement	

6 Évaluation

La note du cours sera scindée en trois parties :

- 20 % : le projet préparatoire ;
- 60 % : les livrables du projet final ;
- 20 % : la défense du projet final.


Évaluation des livrables pour le projet final Trois aspects seront pris en compte dans l'évaluation des livrables :

1. Tout d'abord, **l'analyse et la conception (20%)** évalueront la production des spécifications des besoins clients, la conception de l'interface utilisateur, l'architecture de l'application (diagramme objet), de la base de données et de l'utilisation de recherche d'information ; cette évaluation est basé en partie sur le document soumis après la première étape du projet final.
2. La deuxième partie de la note évalue la **réalisation du projet (60%)** : conformité par rapport aux spécifications, bon fonctionnement du code, et qualité de code source (propreté, bonne structuration, efficacité).
3. Enfin, la troisième partie évalue **le rapport écrit (20%)**. Ce rapport reprendra notamment les éléments de l'analyse initiales mises à jour suite aux feedbacks reçus, et tout autre élément que les étudiants jugeront nécessaires de présenter.

Défense finale Le projet sera clôturé par une défense orale organisée durant la session d'examen. La défense comprend trois parties :

1. **Démonstration du système (33.3%)**. Une présentation convaincante orientée utilisateur. Cette présentation doit montrer au vif le bon fonctionnement de l'application ainsi que ses avantages.
2. **Débriefing technique (33.3%)**. Bilan de la mise en œuvre du système. Cette présentation des aspects technique vous permet de démontrer que vous avez acquis les compétences techniques ciblées par ce projet.
3. **Discussion finale (33.3%)**. Un échange de questions-réponses portant aussi bien sur les spécifications que sur les aspects techniques. Le groupe doit laisser au moins 10 minutes pour cette partie.

Chaque groupe dispose de 50 minutes pour l'ensemble de la défense, y compris la préparation (branchements, etc.). Un projecteur vidéo sera à disposition dans la salle.

 Les étudiants se présentant en deuxième session devront ajouter au système des extensions de fonctionnalité proportionnellement au temps qu'ils auront par rapport à la première session. Les critères d'évaluation resteront inchangés.