

CSI 3540. Structures, techniques et normes du Web

Hiver 2010

Projet : partie 1 (10 % de la note du projet)

Échéance : vendredi 5 février 2010, 16 h

(Version provisoire)

[[PDF](#)]

Objectifs

- Choisir un sujet porteur pour votre projet
- Concevoir un plan détaillé du projet
- Maîtriser les concepts du langage de balisage XHTML 1.1
- Introduction aux feuilles de styles en cascade (CSS 2.1)

Introduction

Vous devez soumettre une proposition de projet pour le cours. Ce travail pratique devra permettre la mise en application des concepts vus en classe. Entre autres, il doit comporter les éléments suivants :

- Application Web utilisant au maximum les techniques et normes vues en classe (voir plan de cours pour plus de détails) ;
- Possède une interface usager non triviale comportant des éléments de programmation côté client (validation de champs, personnaliser l'interface, etc.) ;
- L'interface sera ultérieurement développée afin de la rendre dynamique à l'aide de technologies telles qu'Ajax, ou de façon plus générale, par l'échange de données en format XML entre le client et le serveur ;
- Comporte une application côté serveur non trivial, possiblement sous forme de service Web ;
- Finalement, le projet doit être suffisamment important pour être intéressant, mais réalisable par une équipe de deux personnes au cours du semestre.

Idéalement l'application devrait être utile et intéressante. Son développement pourrait se poursuivre après le cours, de sorte que l'application pourrait éventuellement être distribuée sur [«source forge»](#) ou commercialisée. Cette application sera votre carte de visite!

Consignes

Vous devez préférablement faire le travail en équipe de deux, mais vous pouvez aussi faire le travail individuellement. Vous devez soumettre, via WebCT, les éléments suivants :

1. Faites-moi parvenir une ébauche de votre proposition pour **vendredi 15 janvier, 16 h**.
2. **Proposition.** La proposition doit être suffisamment détaillée pour que l'on puisse déterminer si le projet satisfait tous les objectifs du cours. Assurez-vous que le document répond aux questions suivantes.
 - Quelle est la proposition ? Quelle est la problématique ?

- Quels seront les principaux éléments de l'interface graphique ?
- Quelles sont les fonctionnalités implémentées du côté client ? Et celles implémentées du côté serveur ?
- Comment comptez-vous rendre l'interface usager dynamique («rich client»)?
- En quoi est-ce que l'application est utile ou intéressante ?

3. **Prototype statique.** Afin de mieux présenter cette proposition, vous devez soumettre au moins deux documents HTML en format XHTML 1.1 strict.

- Les éléments qui suivent devraient se retrouver dans au moins un document à remettre : un tableau (élément **table**), un formulaire (élément **form**), une image et un hyperlien ;
- Dans le cours, nous allons mettre l'emphase sur la séparation du contenu et de la préparation. Ainsi, vous devez utiliser les feuilles de style en cascade (CSS 2.1) pour décrire la présentation des documents HTML. Vous devez inclure au minimum 5 éléments de styles.

Cette partie compte pour 10% de la note totale du projet ; la proposition vaut 5%, le prototype HTML vaut 5%.

Quelques idées pour les projets

1. **Migration d'une application conventionnelle sur le Web.** Au cours des dernières années, nous avons assisté à la migration de plusieurs applications vers le Web. Par exemple, [Google Docs](#) permet l'édition de textes et la création de chiffriers sur le Web. Ainsi, les documents sont accessibles en tout temps, à partir de plusieurs plateformes. Certaines applications ont plus de potentiel que d'autres pour ce projet. Par exemple, l'application [SubEthaEdit](#) est un environnement de programmation collaboratif ; plusieurs usagers travaillent simultanément au développement d'un même fichier source. Pour l'instant cette application ne tourne que sous «Mac OS», ce qui limite le bassin d'utilisateurs. De plus, afin de participer au développement, les usagers doivent forcément être branchés au réseau Internet, par conséquent l'application semble être un bon candidat pour sa migration vers le Web. L'interface usager devra être dynamique ; on imagine mal un éditeur de programme dont la page principale devrait être rechargée à chaque fois qu'un usager fait une modification au code source !
2. **Planificateur d'événements.** La planification d'une réunion impliquant plusieurs participants ayant des horaires complexes n'est pas chose facile. Il y a bien quelques outils, mais en général, ils fonctionnent mieux si tous les participants appartiennent à la même entreprise et utilisent un environnement commun (Microsoft Exchange). Pour ce projet, vous devez développer une application Web pour la planification d'un événement. Les tâches principales sont : la création d'un nouvel événement (description, durée de la réunion, une liste de dates potentielles), l'envoi d'invitations, chaque invité, pour chaque date, doit indiquer sa disponibilité. Le système permet l'ajout de commentaires et la possibilité d'importer son horaire, en format iCal, par exemple. Suggestion : vous pourriez faire une ré-implémentation de [Doodle](#) à l'aide de [Google Calendar API](#).
3. **Serveur Web pour Profile-Dynalign.** Nous avons développé un algorithme, nommé profile Dynalign, pour résoudre un certain problème en bioinformatique. Le projet consiste à développer une interface Web pour ce programme. Puisque l'exécution du programme prend beaucoup de temps (jusqu'à deux semaines), il faut une interface qui permette aux utilisateurs de consulter le statut des tâches.
4. **Géoportail agrégateur de nouvelles.** Les agrégateurs de toutes sortes sont très populaires. Par exemple, [bloglines.com](#) rassemble plusieurs sources de nouvelles en format RSS. Pour ce projet, vous devez utiliser un service, tel que [Google Maps API](#), afin d'afficher sur une carte géographique des nouvelles tirées de plusieurs sources. L'utilisateur consulte une carte du monde sur laquelle il y a des icônes, chaque icône comprend le nom du lieu et le nombre de nouvelles qui s'y rattache.
5. **Géoportail pour le suivi de colis.** Toujours à l'aide d'un service, tel que [Google Maps API](#), vous devez concevoir un système permettant le suivi de colis. L'interface affiche sur une carte géographique la position courante du colis. FedEx et Postes Canada peuvent servir de sources de données.
6. **Ottawa Food.** Lorsque je suis arrivé à Ottawa, il y 8 ans, j'ai mis beaucoup de temps à découvrir les marchés d'alimentation spécialisés. Où trouve-t-on des spécialités du Moyen-Orient, de la Chine, de l'Inde, de l'Afrique ? Quel est le meilleur pâtissier, saucissier, etc. Encore aujourd'hui, je ne sais

pas où trouver des violettes cristallisées à Ottawa ; les violettes cristallisées sont utilisées en pâtisserie. Ce projet consiste à concevoir un portail où les usagers pourront publier de telles informations. Les concepteurs du site devraient exploiter les idées des réseaux sociaux Web 2.0. Ainsi, chaque article posté aurait une étiquette sémantique associée. Lors d'une recherche sur le site, l'interface dynamiquement proposait plusieurs façons de compléter l'expression ; ces informations seraient déterminées à l'aide des balises sémantiques.

7. **Gofer Ottawa.** Concevoir un portail pour troquer des services. Un usager cherche quelqu'un pour promener son chien, pour faire une emplette, pour déneiger son stationnement, etc. Les usagers ont un profil et ils acquièrent des cotes de fiabilité.
8. **Jeu Internet.** Luis von Ahn (Carnegie Mellon) a trouvé plusieurs façons de créer des applications Web qui soient à la fois intéressantes et utiles. Il a créé plusieurs jeux sur Internet où les joueurs s'amuse tout en faisant un travail. L'annotation automatique d'images numériques est un problème important, mais difficile à résoudre ; l'efficacité d'un engin de recherche tel que **Google Images** dépend de la qualité des annotations. L'application Web **ESP Game** est un programme informatique qui se sert d'humains pour faire l'annotation d'images ! **Peekaboom** pousse l'idée un peu plus loin. Inspirez-vous de la présentation vidéo **Human Computation** de Luis von Ahn afin de concevoir le prochain jeu populaire sur Internet.
9. **Sondage électronique.** Concevoir une application où les usagers doivent répondre à une série de questions. Par exemple afin d'évaluer la qualité d'un service, connaître leurs opinions politiques, faire des études de marché, etc. Le système doit comporter un nombre non trivial de questions. À tout moment, l'usager devrait savoir à étape du questionnaire il ou elle est. Vous pourriez transformer le projet en jeu didactique. Plusieurs élèves sont branchés à un même serveur, ils commencent le questionnaire au même moment. Afin de passer à la prochaine question, l'usager doit répondre correctement. À tout moment, l'usager connaît son rang dans le groupe (i/n , i ème de n usagers). Le système peut par la suite analyser la performance des élèves, et identifier les difficultés.
10. **maps.uottawa.ca.** Ce projet a pour but la conception d'un site semblable à maps.google.ca, cependant à l'échelle du campus. 1) Tout comme maps.google.ca, on pourrait visualiser la carte du campus à plusieurs niveaux de résolution. Pour ce faire, la carte www.uottawa.ca/maps/CampusMap.pdf serait découpée en une série d'images (format PNG), et ce, pour plusieurs résolutions. 2) L'usager serait en mesure de sélectionner deux immeubles et le serveur lui proposerait le meilleur chemin. Il faudrait créer une structure de graphe associée à la carte du campus et utiliser un algorithme permettant de trouver le plus court chemin. 3) Finalement, le site pourrait aussi permettre des requêtes telles "où est située la cantine la plus près de ...".
11. Soyez créatifs. Vous pouvez explorer des technologies telles que Google Wave, Google App Java Engine, Amazon Elastic Cloud Computing.
12. Finalement, vous pouvez aussi consulter la page de projets de **Guy-Vincent Jourdan**.

En principe, vous n'avez pas une connaissance approfondie des concepts tels que la programmation d'applications Web côté client ou serveur, le développement d'applications dynamiques (Ajax) ou des services Web. Voici, quelques idées pour vous aider à décrire votre application.

- On peut utiliser des techniques de programmation côté client afin de faire la validation de champs (est-ce que le numéro de téléphone est valide ?), pour personnaliser l'interface (changer la taille du texte, par exemple), pour afficher ou cacher des détails d'un document (présentation abrégée des informations), etc.
- L'exemple classique d'interface usager dynamique est **Google Maps**. L'interface permet à l'usager de changer la résolution de la carte, de se déplacer sur la terre, etc. sans précharger toutes ces informations et sans recharger la page principale de l'UI.

Modifié le : 8 janvier 2010