UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

Rapport de projet
Licence 3

Editeur de sites web

Réalisé par :

Pierre Burc, Olivier Duplouy,
Hamza Erraji, Issame Amal,
Mickaël Berger, Joachim Divet,
Zaydane Sadiki et Abdelhamid Belarbi

Sous la direction de :

Michel Meynard

Année universitaire 2011-2012

Remerciements

Nous tenons à remercier tout particulièrement M. Michel Meynard, notre tuteur de projet qui nous a guidés et épaulés tout au long de ces quelques mois de travail, égrénant ça et là différents conseils utiles à souhait.

Bien entendu nous n'oublions pas de remercier chaleureusement toute l'équipe pédagogique de l'UM2 qui nous a apporté son soutien.

Et il va sans dire que tous les membres du groupe se remercient les uns les autres.

Table des matières

Table des figures

Introduction

C'est par une froide journée d'hiver que nous nous réunîmes pour la première fois à l'université de Montpellier. Huit, tous étudiants préposés au projet numéro vingt-trois nous attendons à notre table l'arrivée de notre tuteur M. Meynard. Ce dernier se présente, nous salue et prononce ce discours mémorable qui restera gravé dans nos mémoires.

> ``Dans le cadre de développement de sites Web, on souhaiterait utiliser un éditeur multi-fichiers permettant de réaliser différentes actions sur des fichiers relatifs à un site : édition de fichiers Html, Php, JavaScript et Css.
>
> Il serait également appréciable que cet éditeur proposât un mode de visualisation du site dans un navigateur et un mode arborescence.
>
> On pourrait aussi avoir de l'autocomplétion, de la coloration syntaxique, un accès aux manuels des langages cités précédemment et pourquoi pas une validation Html. ''

Après ce laïus prononcé d'une traite M. Meynard disparut soudain, nous laissant là, le regard vide.

Néanmoins, nous nous remîmes assez vite de notre ahurissement et un sage parmi nous s'écria soudain :

> ``Nous commencerons par étudier quelques éditeurs existants sur le marché pour nous faire une idée. Ensuite nous concevrons le programme avec le langage UML en nous organisant pour la réalisation. Enfin, nous implémenterons l'application insolents et sûrs de nous. Qu'en dites-vous ?''

Nous n'en dîmes que du bien. En effet, l'idée loin d'être incroyablement novatrice avait l'avantage d'être logique et cohérente.

C'est ainsi que démarra le projet *Éditeur web* décrit dans ce document, entrons donc sans plus attendre dans le vif du sujet.

Première partie

Analyse

Chapitre 1

Cahier des charges

Voici une liste exhaustive des fonctionnalités prescrites par M. Meynard :
- Édition des fichiers JavaScript, Php, Css et Html ;
- Un système multi-onglets permettant de naviguer rapidement entre plusieurs fichiers d'un même projet ;
- Possibilité de visualiser le site sous forme d'arborescence ;
- Mode autocomplétion et coloration syntaxique ;
- Mode auto-indentation ;
- Accès en un clic au manuel Php/Html/Css/JavaScript ;
- Validation Html ;
- Visualisation du site dans un navigateur ;
- Squelette de site préexistant.

La liste d'exigences ci dessus se résume en une besoins graphiques, il nous fallait donc ``deviner'' ce qui se déroule en interne dans le programme. Malgré les différents avis données par M. Meynard, il était difficile pour la majorité des membres de comprendre comment fonctionnait un tel programme.

Après plusieurs colloques, discussions, palabres et réunions, il fut décidé que le meilleur moyen de se rendre compte de ce que représentait un tel cahier des charges en terme de produit fini était d'étudier les différents éditeurs existants offrants ce genre de fonctionnalités.

Chapitre 2

Étude de projets existants

2.1 Espresso

Figure 2.1 – Espresso

Le premier programme d'édition de site web que nous avons étudié propose les mêmes fonctionnalités que le logiciel que nous nous proposons de développer, à savoir coloration syntaxique, indentation automatique, arborescence des codes, etc. Il sera à priori une bonne source d'inspiration pour nous d'autant plus que l'interface est très soignée et agréable d'utilisation.

2.2 Aptana

Figure 2.2 – Aptana

Cet éditeur propose un ensemble de fonctionnalités nombreuses et variées. Beaucoup plus fourni que Espresso, il propose des fonctionnalités complexes dans divers langages : déboguage, déploiement automatique, gestionnaire de version inclus, terminal intégré et moult autres outils. Pour nous c'est un bon modèle à suivre en évitant tout de même de tomber dans le piège du sur-nombre d'options qui importuneraient l'utilisateur.

2.3 Dreamweaver

Figure 2.3 – Dreamweaver

Dreamweaver est une référence en la matière. Il réunit les atouts des deux éditeurs sus-cités en joignant l'utile à l'agréable. Il possède en outre un mode dit WYSIWYG permettant de dessiner l'interface d'un site web.

2.4 Bilan comparatif

Le tableau 2.1, donne la distributions de quelques fonctionnalités parmi les logiciels cités précédemment. Il est inutile de mettre dans ce tableau des fonctionnalités telles que la coloration ou l'édition de fichier, car il est évident que des lesdits logiciels possèdent ce genre de spécificités.

Table 2.1 – Fonctionnalités spéciales dans les éditeurs étudiés

Éditeur	onglets	autocomplétion	auto-indentation[1]	validation Html	documentations
Espresso	oui	oui	non	non	non
Aptana	oui	oui	non	oui	oui
Dreamweaver	oui	oui	oui	oui	non

Ainsi, il ressort du tableau 2.1 qu'aucun des outils étudiés ne possède toutes les spécificités de notre programme. Il faudra donc, si nous voulons de l'inspiration, naviguer d'un éditeur à l'autre selon que nous voulons implémenter telle ou telle fonctionnalité.

Maintenant que l'objectif est discernable et que le groupe a une idée générale du logiciel à produire, il est temps de penser à choisir les outils à utiliser.

1. À noter que l'auto-indentation dans le tableau 2.1 concerne la capacité de formater tout un fichier et non pas d'ajouter une tabulation après saut de ligne.

Chapitre 3

Choix des outils

Il est des choix qui influent sur l'ensemble d'un projet et le choix des outils est de ceux là. Nous nous devions de faire un choix judicieux compte tenu de différents critères, à savoir la taille de l'équipe (huit personnes), l'ampleur du projet, le temps imparti et autres menus détails.

La modélisation formelle des spécifications du programme nécessite un langage adapté à ce genre de besoin. C'est donc presque sans discussion que nous prîmes l'évidente décision d'utiliser le langage UML pour ce faire. Les diagrammes seront dessinés grâce à une application web nommée yUML (cf. sitographie, p.24).

Concernant le langage de programmation principal, nous choisîmes C++. Ce choix est motivé par deux raisons, d'une part nous apprenons actuellement ce langage en cours, d'autre part, il s'agit d'un langage stable, documenté et qu'il fait bon d'avoir dans sa besace. C'est ainsi que M. Meynard nous proposé le framework Qt. Un peu austère de prime abord, presque effrayant, il s'avéra finalement très sympathique grâce notamment à une syntaxe lisible et à une documentation bien feuillue.

Pour travailler en équipe sur un projet de ce genre il est utile voire indispensable de disposer d'un outil de synchronisation et de partage. Notre choix s'est porté sur le gestionnaire de version Git. Ceci sans raison particulière car ses semblables offrent des fonctionnalités similaires.

Bien que les huit membres de notre groupe furent géographiquement proches à vol d'oiseau, aucun d'entre nous ne possédait l'étonnante capacité de se déplacer dans les airs. Par conséquent, hormis les outils sus-cités il fallut quelques outils auxiliaires de communication.
Nous utilisâmes donc des outils web tels que :

MicroMobs.com : Pour les discussions professionnelles ;

Facebook.com : Pour les annonces importantes, les messages, les dates de réunions, etc.

Citons aussi TeamGantt.com, une application qui permet de dessiner un (très joli) diagramme de Gantt en ligne.

Chapitre 4

Organisation

4.1 Diagramme de Gantt

Le projet se déroulant sur quatre mois, il a fallu faire un planning qui s'étale sur ledit laps de temps. Voici toutes les étapes du projet inscrites sur un diagramme de Gantt.

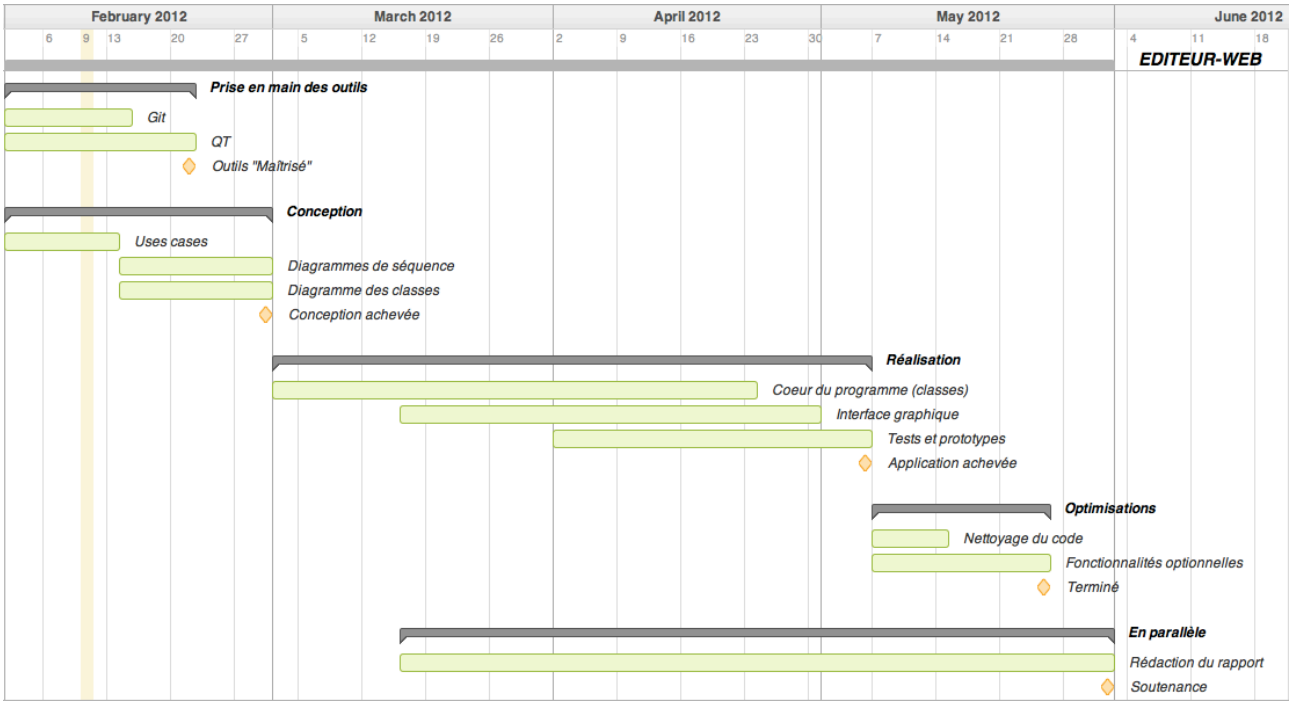

Figure 4.1 – Diagramme de Gantt

Deuxième partie

Conception

Chapitre 5

Décomposition en sous systèmes

Le programme se découpe grossièrement en trois sous-systèmes complémentaires comme sur le diagramme 5.1.

Figure 5.1 – Décomposition en sous systèmes

Le système de gestion des fichiers est un système très important de l'application et c'est pourquoi il est requis par le système de gestion des projets.
Le système de gestion du code est vue comme une extension aux fichiers. En effet, il serait possible d'éditer un site web sans gérer la coloration, l'indentation et toutes les fonctionnalités qui rendent un éditeur de code si agréable à utiliser.

5.1 Gestion des projets

Le sous-système de gestion des projets est un élément important de l'application, en ce sens qu'il permet à l'utilisateur de bien s'organiser de manière simple et efficace.
Un projet est concrètement représenté par un dossier sur la machine de l'utilisateur. Partant de cette idée, notre système de gestion de projet sera implémenté en utilisant les normes et références communément admises en termes de création, modification et destruction de dossiers.

L'unique caractéristique d'un projet est que l'on pourra trouver à la racine un fichier spécial résumant les paramètres utilisateur, les caractéristiques du projet, et autres informations diverses.
Relativement à un projet, l'application fournit les fonctionnalités décrites dans le diagramme 5.2

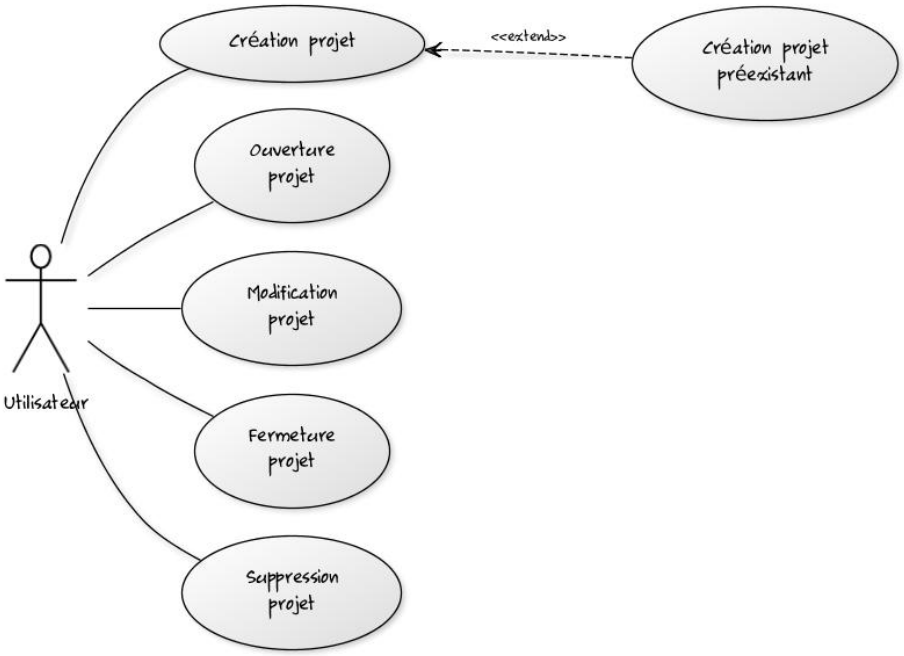

Figure 5.2 – Sous système de gestion des projets

5.2 Gestion des fichiers

Le diagramme 5.3 détaille le contenu du système de gestion des fichiers. Notez l'incroyable similitude entre ce dernier et le système de gestion des projets décrits précédemment.
Plusieurs raisons expliquent notre choix. Primo, les procédures d'entrée et sortie restent les mêmes (création, lecture, écriture, ...), secundo, la sémantique est similaire et facilite le travail autant au programmeur qu'à l'utilisateur final qui suivront les mêmes étapes pour travailler sur un fichier ou un projet.

La présence d'une

5.3 Gestion du code

Derrière ce nom quelque peu singulier se cache un concept fort simple, la gestion du code consiste à indenter, colorer, visualiser et sublimer le code source présent dans un fichier. Le diagramme 5.4 parle de lui même.

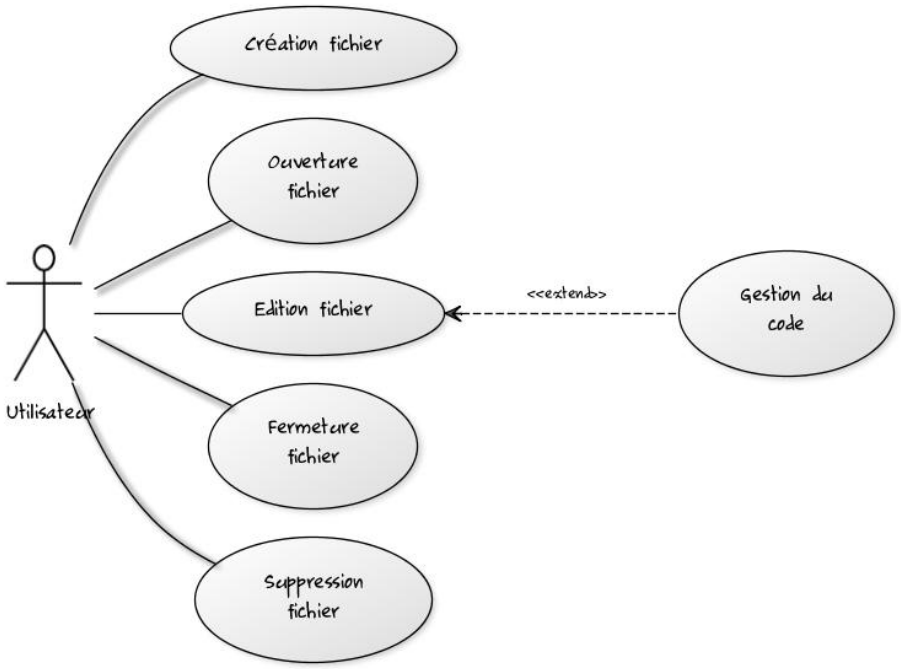

Figure 5.3 – Sous système de gestion des fichiers

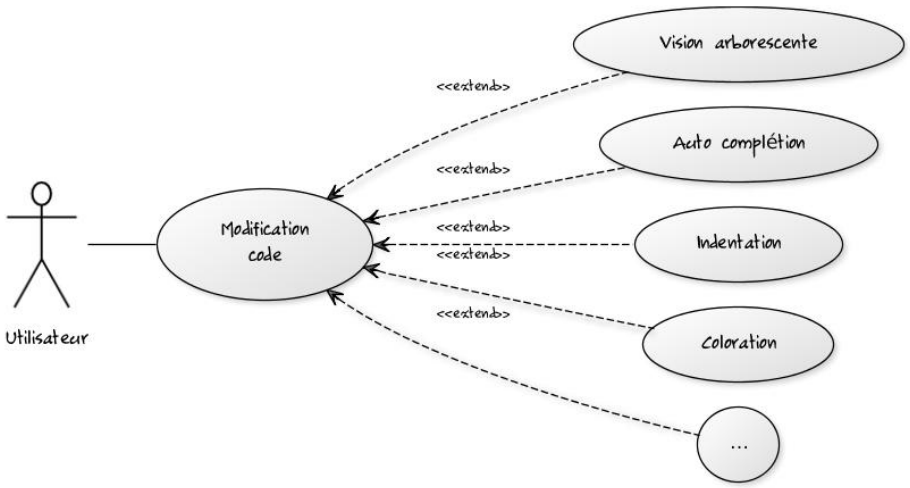

Figure 5.4 – Sous système de gestion du code

Chapitre 6

Diagrammes des classes

6.1 Coloration

La conception de la partie « coloration syntaxique » de notre application a requis l'introduction de plusieurs classes différentes. Premièrement des classes dites de données, suffixées du mot anglais *data*, qui contiennent les différents mots clés relatifs à chaque langage sous forme d'expressions régulières. Dans le diagramme 6.1, nous donnons en exemple une seule classe *HtmlData*, car les autres sont similaires, aux particularités du langage près.

Figure 6.1 – Classe de données pour le langage Html

De la même manière, on trouve les classes *CssData, PhpData et JavaScriptData*.

Une autre part de la coloration est l'utilisation concrète des classes décrites précédemment. Sur le diagramme 6.2 sont représentées toutes les classes qui permettent la coloration à l'écran. Toutes les classes héritent d'une superclasse *Highlighter* qui factorise la méthode *highlightBlock()*, celle-ci, comme son nom l'indique colore le texte bloc par bloc en fonction des règles données dans les classes *Data*.

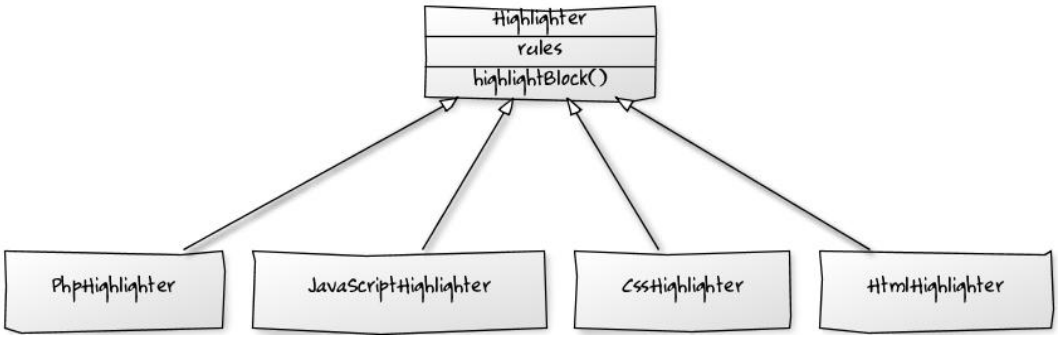

Figure 6.2 – Classes de coloration

Chapitre 7

Diagrammes d'états-transitions

Pas sûr on verra si on à le temps.

Troisième partie

L'oeuvre

Chapitre 8

Travail de groupe

Là on parle des réunions, pourquoi pas mettre un extrait de journal.

Chapitre 9

Implémentation

9.1 Noyau

9.2 Interface

Ici on aborde les problèmes de la documentation, de nouvelles bibliothèques, de mettre en place nos algorithmes ``en vrai'', ...

Chapitre 10

Résultat

On va essayer de caser des trucs ici.

Chapitre 11

Discussion

Critique (positive) du résultat, le cahier des charges est-il respecté ? améliorations possibles, erreurs qu'on à pu faire.

Conclusion

Oué on s'est bien marré et tout et tout.lol!!!!

Glossaire

Autocomplétion l'autocomplétion, est une fonctionnalité informatique permettant à l'utilisateur de limiter la quantité d'informations qu'il saisit avec son clavier, en se voyant proposer un complément qui pourrait convenir à la chaîne de caractères qu'il a commencé à taper. 4, 6

Diagramme de Gantt Le diagramme de Gantt est un outil utilisé en ordonnancement et gestion de projet et permettant de visualiser dans le temps les diverses tâches liées composant un projet. Il permet de représenter graphiquement l'avancement du projet. 10

Expression régulière Une expression rationnelle ou expression régulière est en informatique une chaîne de caractères que l'on appelle parfois un motif et qui décrit un ensemble de chaînes de caractères possibles selon une syntaxe précise. Les expressions rationnelles sont issues des théories mathématiques des langages formels. 15

Git Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, le créateur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. 9

Qt Qt est un framework orienté objet et développé en C++. Il offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. 9

UML Unified Modeling Language, langage de modélisation graphique à base de pictogrammes. 24

WYSIWYG Un WYSIWYG est une interface utilisateur qui permet de composer visuellement le résultat voulu, typiquement pour un logiciel de mise en page, un traitement de texte ou d'image. C'est une interface « intuitive » : l'utilisateur voit directement à l'écran à quoi ressemblera le résultat final. 8

Sitographie

[1] Wikipédia : www.wikipedia.org
L'encyclopédie en ligne de laquelle j'ai tiré certaines définitions présentes dans le glossaire.

[2] yUML : www.yuml.me
Ce site permet de générer à la volée des diagrammes UML, extrêmement utile lorsqu'il s'agit de travail de groupe.

[3] Micromobs : www.micromobs.com
Le site de discussion de groupe au principe intéressant : il vaut mieux une interface dédiée aux discussions plutôt qu'une boîte e-mail encombrée et mal organisée.

[4] TeamGantt : www.teamgantt.com
Création en ligne de diagrammes de Gantt beaux et lisibles.

[5] Facebook : www.facebook.com
Le réseau social que l'on ne présente plus, il nous a permis d'échanger des messages moins techniques que sur Micromobs et de nous organiser grâce à différents évènements.