

Rapport de projet
Licence 3

Editeur de sites web



Réalisé par :

Pierre Burc, Olivier Duploux,
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 la réalisation de ce projet.

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

Table des matières

Introduction	3
I Analyse	4
1 Cahier des charges	5
2 Étude de projets existants	6
2.1 Espresso	6
2.2 Aptana	6
2.3 Dreamweaver	7
3 Choix des outils	8
4 Organisation	9
4.1 Diagramme de Gantt	9
II Conception	10
5 Décomposition en sous systèmes	11
6 Cas d'utilisations	12
7 Diagrammes des classes	13
8 Diagrammes d'états-transitions	14
III L'oeuvre	15
9 Travail de groupe	16
10 Implémentation	17
11 Résultat	18
12 Discussion	19

Introduction

Au début du mois de Février, il nous a été demandé de réaliser un projet ayant pour but de mettre en place une application permettant d'utiliser un éditeur multi-fichiers effectuant différentes actions sur des fichiers relatif à un site web.

Ce dernier consiste à créer, modifier, sauvegarder... des fichiers contenant les langages JavaScript, HTML, CSS, PHP via une interface se rapprochant des IDEs présents sur le marché. Celle-ci aura un rôle primordial, car elle devra apporter à l'utilisateur une aide précieuse grâce à des fonctionnalités intuitives, comme par exemple l'auto-complétion.

A l'occasion de ce projet, un groupe de huit personnes a été formé pour répondre à la capacité de travail demandée. Nous devons donc nous répartir les tâches entre chaque membre voir créer des sous-groupes permettant de cloisonner chaque brique essentielle de l'application (interface, système, coloration/complétion). De plus, une période de trois mois nous a été accordée pour réaliser cet IDE.

Afin de ne pas gaspiller du temps à déterminer par où commencer et de quelle façon aborder le sujet, nous avons des réunions régulières avec notre tuteur M.Meynard, pour nous guider dans notre réflexion.

A travers ce compte-rendu nous essaierons de rendre compte des étapes qui nous ont conduits à élaborer cette application en essayant de répondre le plus fidèlement possible au problème posé.

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 23 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 à jamais dans les mémoires, du moins cette année.

“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'auto-complétion, 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 dûmes que du bien. En effet, l'idée loin d'être incroyablement novatrice avait l'avantage d'être logique et cohérente.

Première partie

Analyse

Chapitre 1

Cahier des charges

Pour nous donner une idée du rôle que devra jouer notre éditeur web, une liste des principales fonctionnalités nous a été fournie, présentée ci-dessous :

- 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 ;
- Auto-complétion et coloration syntaxique ;
- Auto-indentation ;
- Accès en un clic au manuel Php/Html/Css/JavaScript ;
- Validation Html ;
- Visualisation du site dans un navigateur ;
- Squelette de site préexistant.

Après avoir pris connaissance du cahier des charges initial, nous avons décidé de rajouter quelques spécifications sur certaines fonctionnalités. Concernant la visualisation du site web, 2 possibilités s'offraient à nous. La première était de réserver une fenêtre à l'intérieur de l'application destinée à afficher en temps réel le site web, après chaque modification. Nous avons opté pour la seconde solution, qui met à disposition de l'utilisateur un bouton permettant de visualiser le site à travers l'ouverture d'un navigateur.

Une contrainte a également été ajoutée pour la coloration syntaxique, car nous avons décidé d'obliger l'utilisateur à enregistrer le fichier dès sa création, de manière à récupérer l'extension du fichier pour déterminer le langage utilisé.

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 : débogage, 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. Derrière ce sigle barbare se cache la possibilité intéressante de dessiner l'interface d'un site web. Nous concernant, nous pourrions proposer à l'utilisateur de visualiser le site sur lequel il travaille.

Chapitre 3

Choix des outils

Avant de nous lancer dans la réalisation de l'application, nous avons du choisir les outils les plus adaptés avec lesquels nous avons conçus et programmer. Dans un premier temps, le langage de programmation devait être absolument choisis. Nous avons pensé au C++ pour plusieurs raisons. D'une part, nous l'avons utilisé durant cette année et est portable, donc un langage maîtrisé par les huit membres du groupe et utilisable sur n'importe quel système d'exploitation sans conflits. D'autre part, nous pouvons trouver très facilement beaucoup de documentation à son sujet, et il est également adapté à la réalisation de logiciels.

Par la suite, nous avons du adopter un IDE permettant de programmer avec le C++. Notre tuteur nous a orienté vers le framework QT Creator, intégrant de multiples bibliothèques utiles pour le projet. De plus, QT offre la possibilité de réaliser des interfaces graphiques grâce à QT Designer, simplifiant grandement la programmation par la présence de fonctionnalités déjà toutes faites.

Travaillant en groupe de huit, un gestionnaire de versions était indispensable pour partager les différents travaux effectués par chacun des membres. Nous avons donc choisis le gestionnaire de versions GIT, avec une prise en main accessible à tous.

A côté de la programmation pure, il fallait bien sur pouvoir communiquer entre nous de manière rapide à distance. N'habitant pas tous à Montpellier, il a fallu opter pour une solution collaborative. Le web nous offrant cette possibilité, nous avons sélectionné deux sites :

MicroMobs.com : Pour les discussions professionnelles ;

Facebook : Pour les annonces, les dates de réunions,etc... ;

Enfin, pour la conception et la modélisation des différentes structures à implémenter, nous avons utilisé le site YUML.me permettant de concevoir et générer des diagrammes UML. De plus, Team-Gantt.com nous a été utile pour établir le diagramme de Gantt, et ainsi planifier les différentes étapes du projet et séparer le groupe en plusieurs sous-groupes et savoir approximativement combien de temps nous prendrait chaque tâche.

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.

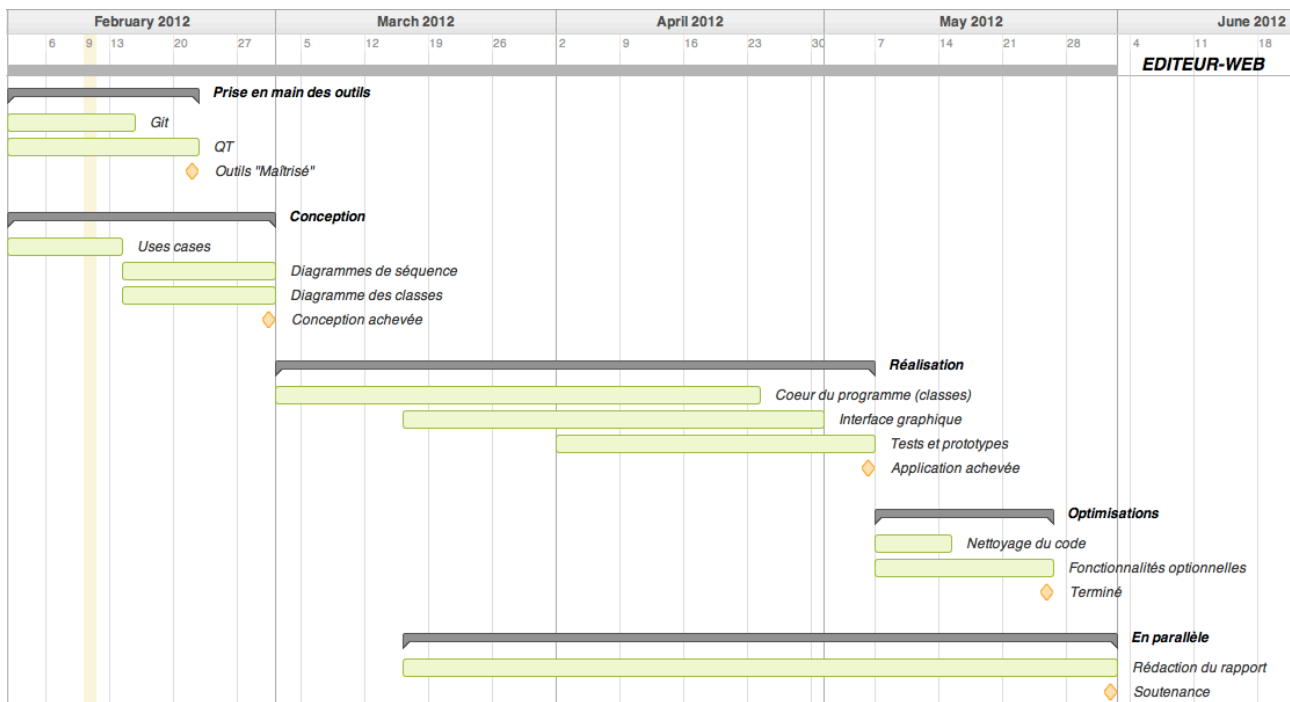


FIGURE 4.1 – Diagramme de Gantt

Deuxième partie

Conception

Chapitre 5

Décomposition en sous systèmes

Chapitre 6

Cas d'utilisations

Chapitre 7

Diagrammes des classes

Chapitre 8

Diagrammes d'états-transitions

Pas sûr on verra si on à le temps.

Troisième partie

L'oeuvre

Chapitre 9

Travail de groupe

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

Chapitre 10

Implémentation

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

Chapitre 11

Résultat

On va essayer de caser des trucs ici.

Chapitre 12

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

auto-complétion L'auto-complétion est une fonctionnalité consistant à compléter automatiquement le mot que l'utilisateur est en train decrire. 3

UML Unified Modeling Language, langage de modélisation graphique à base de pictogrammes. 22

Table des figures

2.1	Espresso	6
2.2	Aptana	6
2.3	Dreamweaver	7
4.1	Diagramme de Gantt	9

Sitographie

[1] Wikipédia : <http://fr.wikipedia.org/wiki>

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.