École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique

Cahier de spécification système & plan de développement			
Projet :	**Adaptation des images d'un site web pour la compensation du daltonisme**		
Emetteur :	Pierre Fourreau	Coordonnées : EPU-DI	
Date d'émission :	10 janvier 2013		
Validation			
Nom	Date	Valide (O/N)	Commentaires

Mohamed Slimane : 20/12/2012 ; O ; Commentaires faits sur le document par écrit
Sébastien Aupetit : 20/12/2012 ; O ; Commentaires faits sur le document par écrit

Historique des modifications		
Version	Date	Description de la modification

1.1 : 01/12/12 ; Version initiale
1.2 : 28/12/12 ; Modification de la version initiale (points à revoir)

Table des matières

Cahier de spécification système

1.1 Introduction

Dans le cadre de notre formation au sein du Département Informatique de l'école Polytechnique de l'Université de Tours, un projet de fin d'études (PFE) est effectué afin de mettre en pratique les connaissances acquises au cours de notre formation.

Ce document a pour but de spécifier le projet de fin d'études intitulé : « adaptation des images d'un site web pour la compensation du daltonisme ».

Ce projet est réalisé au sein de l'équipe Handicap et Nouvelles Technologies (HaNT) de François Rabelais.

Ce cahier de spécifications est rédigé par moi-même, Pierre Fourreau. Les professeurs chargés de l'encadrement et de la relecture de ce projet sont Mohamed Slimane et Sébatien Aupetit.

1.2 Contexte de la réalisation

1.2.1 Contexte

Au départ, le web avait été mit en place pour pouvoir être utilisé par tout le monde. Mais il s'avère qu'aujourd'hui, cet objectif n'est pas atteint. Actuellement, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). Ainsi, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web. Ceci peut être expliqué lorsque l'on sait que le développement d'un site web est soumis à un ensemble de contraintes (financières ou d'ergonomies). De plus, il n'est pas rare qu'un site web soit délaissé par celui qui a conçu ce site.[4]

Ici, le but est de rendre plus simple la navigation de l'utilisateur sur les sites web sans qu'il ait besoin de suivre un apprentissage spécifique. Les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

L'équipe HaNT de l'Université François Rabelais tente de mettre en place un outil facilitant l'accès au web pour les personnes handicapées. L'objectif de cet outil est de traiter les pages web afin de les rendre plus accessibles.

Mon projet de fin d'études est donc directement en relation avec l'outil mis en place par l'équipe HaNT puisque le sujet de mon projet est d'adapter des images d'un site web pour la compensation du daltonisme.

1.2.2 Objectifs

L'objectif est de permettre l'utilisation d'un web non accessible/non utilisable à des personnes ayant une déficience visuelle.

Dans ce projet, on cherche à compenser les pertes visuelles dues au daltonisme sur les images, à différencier les images du décors et les autres images, à prioriser les traitements dans le cas où le nombre d'images à traiter est trop importants pour le temps disponible. L'ensemble du projet sera réalisé en Java au sein de la plateforme SWAP (Smart Web Accessibility Proxy).

1.2.3 L'accessibilité sur le web

Dans le domaine du web, l'accessibilité signifie que les personnes handicapées peuvent utiliser le web. Cela permet à ces personnes de percevoir, comprendre, naviguer et interagir avec le Web ainsi que de pouvoir apporter leur contribution et créer du contenu. L'accessibilité du web vise toutes les déficiences, qu'elles soient visuelles, auditives, motrices, cognitives, neurologiques ou liées à la parole. Cependant, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). En effet, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web.[6]

L'accessibilité peut être exigée par la loi. En effet, afin de limiter les problèmes et permettre un accès au web à tous, les associations, lobbys et autres législateurs demandent à ce qu'un certain nombre de principes et normes (WCAG1, WCAG2) soient respectés par les concepteurs de site web. Des lois existent et tentent d'obliger les concepteurs à concevoir des sites web accessibles. Malheureusement, ces lois, normes et principes ne sont pas suffisants.

Rendre un site web accessible à 100% reste utopique pour de nombreuses raisons. Cela serait trop coûteux en temps et argent, trop complexe et trop contraignant. En effet, beaucoup de sites ne sont plus maintenus mais existent toujours, ou d'autres sont faits manuellement sans tenir compte de la notion d'accessibilité. Par ailleurs, de nombreux sites web ne sont pas d'origine française ; les lois et les contraintes peuvent donc être différentes voire ne pas exister. Un site web peut être accessible à une personne ayant le handicap X sans l'être pour autant l'être avec une autre ayant le handicap Y. De ce fait, pour répondre au problème d'accessibilité sur des sites web, une solution peut consister à viser une meilleure accessibilité sur ces derniers plutôt que de viser une accessibilité complète.

1.2.4 La notion d'utilisabilité

Il faut bien faire la différence entre l'accessibilité d'un site web et l'utilisabilité de ce dernier. En effet, un site web peut très bien être accessible à une personne ayant une déficience visuelle mais cette personne peut être dans l'incapacité d'utiliser ce site. L'objectif de l'utilisabilité du web est de rendre la navigation sur les sites web plus aisée pour l'utilisateur, sans qu'il ait besoin de suivre un apprentissage spécifique. En effet, les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

Dans un site web, l'ergonomie correspond à répondre efficacement aux attentes de l'utilisateur afin de leur fournir une navigation plus aisée. Le problème est que les visiteurs du site web n'ont pas tous le même profil. Il faut donc prendre en considération les attentes de l'utilisateur, ses habitudes, son âge, ses équipements ainsi que son niveau de connaissance.[2]

1.2.5 Le daltonisme

Le projet concerne les personnes ayant une déficience visuelle et plus particulièrement les personnes atteintes de daltonisme.

Le daltonisme est une maladie qui est répandue dans le monde. Son apparition est due à une anomalie de la rétine. C'est un dysfonctionnement de la vision des couleurs, entraînant le plus souvent la confusion entre le rouge et le vert.[1] Le daltonisme est une anomalie dans laquelle un ou plusieurs des trois types de cônes de la rétine oculaire, responsables de la perception des couleurs sont déficients. Habituellement, le daltonisme est classé comme une infirmité légère. Cependant, il y a des situations où les daltoniens peuvent avoir un avantage sur les individus ayant une vision normale (percer certains camouflages basés sur la couleur).

Il existe plusieurs formes de daltonisme (aussi appelé dyschromatopsie) partielle, la plus fréquente étant la confusion du vert et du rouge. Les autres formes de daltonisme sont nettement plus rares, comme la confusion du bleu et du jaune, la plus rare de toutes étant la déficience totale de la perception des couleurs (achromatopsie), où le sujet ne perçoit que des nuances de gris.

Les différentes formes de daltonisme sont les suivantes :
- Achromatopsie
- Deutéranopie
- Deutéranomalie
- Protanopie
- Protanomalie
- Tritanopie
- Tritanomalie

Chacune de ces formes de daltonisme possède des caractéristiques particulières.[7]

1.2.6 Bases méthodologiques

Chaque semaine, un compte-rendu sera à rédiger par email et à envoyer aux encadrants. Dans chaque compte rendu, il faudra préciser les activités de la semaine, répondre aux questions ouvertes par le précédent compte-rendu, poser de nouvelles questions en indiquant le destinataire de la question, et enfin faire un rétroplanning de ce qui est à faire dans la semaine suivante.

La rédaction d'un compte rendu hebdomadaire présente un réel intérêt. En effet, cela permet de se rendre compte rapidement si le projet n'avance pas suffisamment, et de pouvoir retrouver facilement tout ce qui a été fait précédemment. De plus, ces comptes-rendus permettent aux encadrants de mieux suivre l'avancement du projet, et peuvent aussi permettre de discuter et d'échanger avec les encadrants.

Tous les documents relatifs au projet sont déposés sur un serveur Subversion qui est un logiciel de gestion de versions. Ces documents peuvent être des rapports, des comptes rendus ou même des codes sources.

L'avantage principal d'utiliser un serveur Subversion pour stocker tous ces documents est que l'on possède toutes les différentes versions de développement. Il est alors très aisé de revenir a un état antérieur de la phase de développement grâce à un logiciel client de SVN installé sur ma propre machine comme TortoiseSVN.[5]

De plus, l'autre avantage d'utiliser un serveur Subversion pour stocker tous ces documents est qu'en cas de problème, comme une perte des données, suite à un plantage d'une machine par exemple, tous les documents relatifs au projet pourront alors être récupérés grâce au serveur.

Le langage utilisé pour le développement de ce projet sera le Java 1.7.

1.3 Description générale

1.3.1 Environnement du projet

Ce projet s'insère dans le projet SWAP (Smart Web Accessibility Proxy). SWAP est un ensemble de classes Java permettant le développement rapide d'un proxy. Le proxy SWAP a des applications spéciales dans l'amélioration de l'accessibilité du site Web. Il est capable de transformer les pages web non accessibles en page web plus accessibles.

1.3.2 Caractéristiques des utilisateurs

Il est assez facile de caractériser les utilisateurs. En effet, l'objectif de ce projet est de permettre l'utilisation d'un site web non accessible/non utilisable à des personnes ayant une déficience visuelle. Plus particulièrement, l'utilisateur sera daltonien.

Cet utilisateur n'a pas nécessairement besoin de connaître l'informatique mis à part de savoir utiliser un navigateur internet et naviguer sur des sites web.

1.3.3 Fonctionnalités et structure générale du système

L'équipe HaNT a développé et mis en place un système de proxy appelé SWAP. Cet acronyme signifie Smart Web Accessibility Proxy. Ce proxy est codé en Java et doit donc être présent sur la machine de l'utilisateur ayant une déficience visuelle. Ce système permet de récupérer une page web et d'effectuer des traitements sur cette même page pour ensuite pouvoir l'afficher à l'utilisateur.

L'intérêt de ce système est de concevoir et mettre en place un proxy HTTP intelligent capable de transformer les pages web non accessibles en page web plus accessibles. Grâce à l'utilisation d'un proxy HTTP, il est possible d'intervenir directement sur le document et sa forme sans pour autant changer les habitudes et les logiciels spécifiques de l'utilisateur comme le lecteur braille ou la synthèse vocale. De ce fait, le proxy est présent pour faire bénéficier l'utilisateur d'améliorations.

Dans le système, le proxy a une place importante. En effet, lorsqu'un utilisateur envoie une requête (demande une page web), le proxy va interagir avec le navigateur web sans que l'utilisateur s'en rende compte. Le proxy va alors se charger de récupérer et de traiter les données du navigateur pour finalement renvoyer une réponse à l'utilisateur (la page web demandée).

L'avantage d'un tel système est que le traitement réalisé par le proxy est invisible pour l'utilisateur.

De plus, des fonctionnalités sont déjà implémentées sur ce proxy. Il faudra donc en partie utiliser les fonctionnalités existantes afin de pouvoir permettre à une personne daltonienne de visualiser une page web comme un utilisateur n'ayant pas de problème de vision.

1.3.4 Contraintes de développement, d'exploitation et de maintenance

Contraintes de développement

L'ensemble du projet sera réalisé en Java au sein de la plateforme SWAP. Ce langage est évidemment choisi puisque le proxy a été développé avec ce langage.

Le projet doit être réalisé de préférence sur une machine LINUX.
Cette machine LINUX doit posséder :
- le kit de développement JDK 1.7
- le logiciel de gestion de versions Apache Subversion 1.7
- le logiciel pour la gestion et l'automatisation de production des projets Java Apache Maven 3

Le développement se fera avec l'IDE Eclipse version 4.2. Ce dernier doit posséder le plugin Subversive permettant l'utilisation de Subversion et le plugin Maven Integration (m2e).

Maintenance et évolution du système

La maintenance sera assurée par le logiciel de gestion de versions Apache Subversion (SVN).

1.4 Description des interfaces externes du logiciel

Comme expliqué précédemment, le proxy développé par le laboratoire HaNT permet de récupérer une page web et d'effectuer des traitements sur cette même page pour ensuite pouvoir l'afficher à l'utilisateur. Il y a donc une liaison entre le navigateur web de l'utilisateur et le proxy.

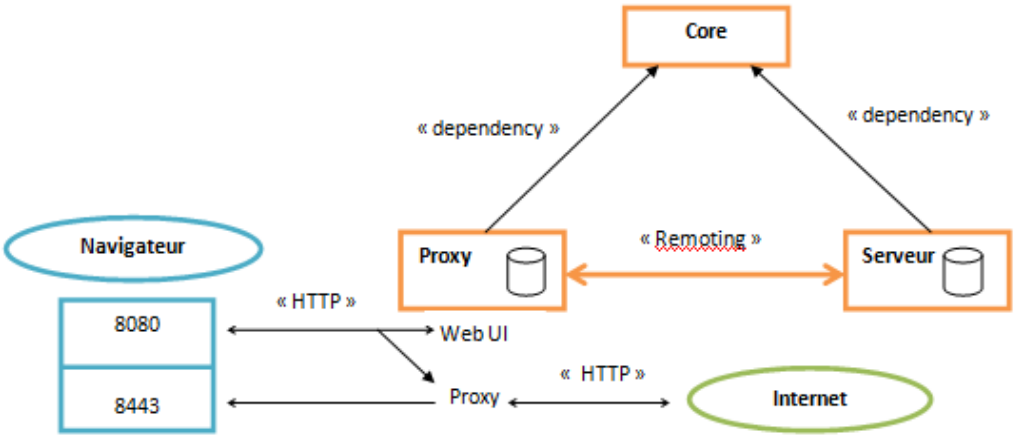

FIGURE 1.1 – Architecture du système

Dans ce schéma on voit que le composant Core est un composant commun du proxy et du serveur. Ces deux derniers peuvent communiquer grâce au remoting (appel de procédure distante RPC).

L'intérêt de ce schéma est de montrer que le navigateur peut accéder à une page web via le proxy. Dans ce cas ce sont les ports HTTP (8080) et HTTPS (8443) qui sont utilisés. Mais ce schéma montre aussi que le navigateur peut accéder à une page sans que le proxy ait a accéder à internet. En effet, le Web UI est un composant du proxy, et dans le cas où la page demandée par le navigateur existe déjà sur le proxy, le Web UI se charge alors de renvoyer la page demandée.

1.5 Architecture générale du système

La plateforme SWAP possède de multiples composants. Ces derniers peuvent être regroupés en trois grandes parties. Il y a les composants centraux (core components), les composants du proxy, et ceux du serveur. [3]

1.5.1 Composants centraux

Pour les composants centraux (core components), différentes classes (content helper classes) existent comme les classes HTTP, les classes d'encodage, les classes HTML et les classes CSS.

Les composants de SWAP sont centrés sur un environnement objet. Les objets managent le cycle de vie de l'application, l'accès à la configuration et le contexte de l'application. La plupart des composants de SWAP reçoivent une référence sur l'interface Environment. Il y a 2 environnements concrets : ProxyEnvironment and ServerEnvironment. Ces deux derniers représentent une application proxy ou une application serveur.

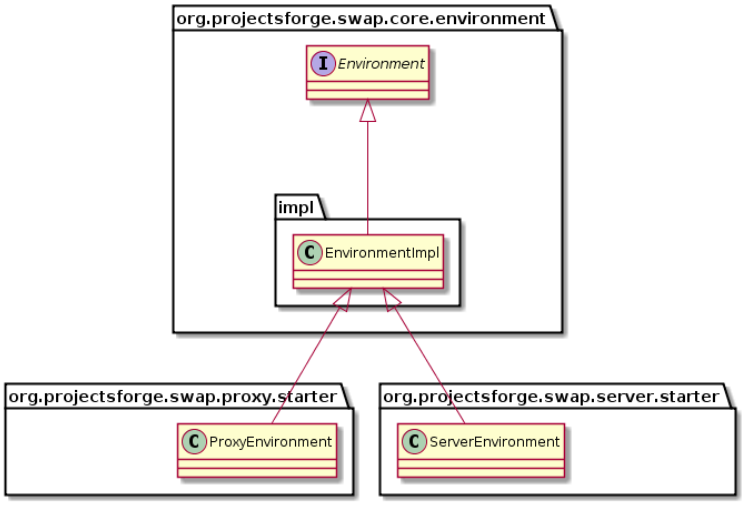

FIGURE 1.2 – Environnement SWAP

Lorsque l'on a un traitement à faire et que l'on a plusieurs tâches. Si on a plusieurs classes qui sont déclarées en tant qu'handler, on va pour gérer les exécutions parallèles de l'ensemble des tâches.

Les handlers sont aussi capables de se partager des ressources.

D'autre part, des classes permettent de manipuler le contenu des pages web (content helper classes). Il y a des classes permettant de construire une requête HTTP et de manipuler le cache (HTTP helper classes) ou alors de calculer l'encodage du document pour une manipulation correcte du contenu(encoding helper classes). Enfin, d'autres classes existent et permettent notamment de manipuler le HTML et le CSS.

1.5.2 Composants du proxy

Pour les composants du proxy, les gestionnaires de contenu (content handlers) sont composés des gestionnaires MIME, HTML, CSS et des gestionnaires d'images. Ensuite, il y a plusieurs servlets pour la partie proxy avec une servlet pour la commutation, pour le proxy et pour le WebUI.

1.5.3 Composants du serveur

Pour les composants du serveur, il existe aussi plusieurs servlets notamment pour le remoting et pour le WebUI.

1.6 Description des fonctionnalités

1.6.1 Définition de la fonction permettant de simuler une vue daltonienne

Identification de la fonction permettant de simuler une vue daltonienne

Le nom de cette fonction sera SimulationDaltonisme. Son rôle sera de simuler les trois types de daltonisme des images présentes sur le web. Il y a donc la vision deutéranope, protanope et tritanope à simuler. Avant de pour effectuer cette simulation, il faudra intercepter toutes les images présentes sur le page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP et se charge d'intercepter les images et de donner en entrée un flux de bytes correspondant à l'image interceptée.

La priorité de cette fonction est primordiale car elle constitue la base du traitement à effectuer.

Description de la fonction permettant de simuler une vue daltonienne

En entrée, nous auront une page HTML avec l'ensemble des balises possibles. Le handler qui intercepte les images fournit un flux de bytes correspondant à l'image interceptée.

1.6.2 Définition de la fonction permettant de marquer les images modifiées

Identification de la fonction permettant de marquer les images modifiées

Le nom de cette fonction sera MarquerImage. Son rôle sera d'ajouter une marque aux images qui ont été modifiées par le proxy avec la simulation de la vue daltonienne. Comme pour la fonction précédente, avant de pour effectuer cette simulation, il faudra intercepter toutes les images présentes sur le page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP et se charge d'intercepter les images et de donner en entrée un flux de bytes correspondant à l'image interceptée.

Cette fonction permettrait de savoir quelles images ont été simulées ou non sur la page web.

Description de la fonction permettant de marquer les images modifiées

En entrée, nous auront une page HTML avec l'ensemble des balises possibles. Le handler qui intercepte les images fournit un flux de bytes correspondant à l'image interceptée. Le handler fournit aussi diverses informations sur la page web et sur l'image.

1.6.3 Définition de la fonction permettant d'intercepter toutes les images d'une page web

Identification de la fonction permettant d'intercepter toutes les images d'une page web

Le nom de cette fonction sera IntercepterHTML. Son rôle sera d'intercepter toutes les images présentes sur une page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP, ce handler gère une image à la fois. De plus, un handler HTML existe pour gérer le HTML et un autre handler gère le CSS. Il faut donc détecter les images dans le HTML et dans le CSS et faire du prefetch (mise en cache) des images référencées.

Description de la fonction permettant d'intercepter toutes les images d'une page web

Il faut détecter les images dans le handler HTML puis les récupérer de manière asynchrone. Lorsque que le navigateur analysera la page HTML, il détectera les images et les demandera une par une au proxy. Si le proxy les a déjà en cache alors elles seront retournées tout de suite. Enfin, un marquage des requêtes pourra être effectué. Ceci sera effectué grâce à la marque qui aura été précédemment effectué sur les images.

1.6.4 Définition de la fonction permettant de mettre en cache les images modifiées

Identification de la fonction permettant de mettre en cache les images modifiées

Le nom de cette fonction sera MiseEnCacheImagesModifiees. Son rôle sera de mettre en cache les images qui auront été modifiées.

Description de la fonction permettant de mettre en cache les images modifiées

Cette fonctionnalité fait suite aux précédentes puisqu'elle prend en entrée les images modifiées. Une image modifiée est une image qui a été traitée par le proxy. Cette image peut être modifiée pour simuler une vision daltonienne.

1.6.5 Définition de la fonction permettant de mettre en cache les pages voisines

Identification de la fonction permettant de mettre en cache les pages voisines

Le nom de cette fonction sera MiseEnCachePagesVoisines. Son rôle sera de mettre en cache les pages voisines de la page courante dans le navigateur.

Description de la fonction permettant de mettre en cache les pages voisines

Cette fonctionnalité consiste à faire du calcul anticipé. Cette fonction devra télécharger les pages HTML (via un objet nommé Request).

1.7 Conditions de fonctionnement

1.7.1 Performances

Le proxy étant un intermédiaire entre un utilisateur et internet, il va falloir faire en sorte de retourner des résultats en un temps limité et raisonnable. Le but est de faire le mieux possible pendant ce temps limité.

1.7.2 Méthodologie

Chaque semaine, un compte-rendu indiquant les activités réalisées pendant la semaine est rédigé et envoyé aux encadrants de ce PFE. De plus, nous utilisons un SVN afin de garder une trace de l'avancement de notre projet. Enfin, différents comptes-rendus seront rédigés suite à des réunions effectuées avec les encadrants de ce PFE.

1.7.3 Contrôlabilité

Pour suivre l'éxécution des traitement, il faudra utiliser des fichiers de log. En effet, dans ce projet, des loggueurs seront accessibles.

Plan de développement

2.1 Découpage du projet en tâches

Cette partie du cahier de spécifications permet de structurer le projet en tâches. Une tâche correspond ici à un ensemble de réalisations ayant une cohésion d'ensemble.

2.1.1 Recherches et documentation

Description de la tâche

Tout d'abord, le projet commence par une longue phase de recherches et de documentation. Cela consiste à se documenter sur tout ce qui tourne au daltonisme, comprendre comment fonctionne cette déficience visuelle. Il faut d'abord étudier la notion d'accessibilité et d'utilisabilité sur le web, puis trouver comment modifier les couleurs d'une image afin de passer de la vue d'une « personne normale » à une personne daltonienne. Il faudra aussi comprendre comment une couleur est représentée au niveau du codage (RGB par exemple), mais aussi de découvrir les différents algorithmes existants permettant de simuler la vue d'une personne daltonienne.

Livrables

Une synthèse résumant toutes les recherches doit être faite. Cette synthèse sera utile pour la rédaction du rapport final de ce projet. De plus, l'intérêt de cette synthèse est de réaliser une bibliographie avec tous les liens et documents qui ont été utilisés pour ces recherches. L'intérêt de cette bibliographie est de pouvoir retrouver facilement les différents liens qui m'auront permis de comprendre mon projet de fin d'études.

Estimation de charge

Cette tâche est estimée à 8 jours/homme.

2.1.2 Installation de l'environnement de développement

Description de la tâche

Cette tâche consiste à installer tous les outils nécessaires au développement sur la plateforme SWAP. Différentes installations doivent être effectuées comme l'installation de l'IDE Eclipse ainsi qu'un logiciel de gestion de version comme SVN. Un kit de développement JDK 1.7. De plus, plusieurs plugins sont à installer avec Eclipse comme Subversive et Maven 3.

Par ailleurs, il faudra récupérer le projet SWAP sur le serveur mis à disposition par nos encadrants. Pour cela il faudra utiliser le logiciel de gestion de versions SVN.

Estimation de charge

Cette tâche est estimée à 2 jours/homme.

2.1.3 Analyse de l'existant de la plateforme SWAP

Description de la tâche

Cette tâche consiste à prendre en main le code source existant du projet SWAP. Ce dernier étant composé de multiples classes, il faut pouvoir comprendre comment est structuré ce projet ainsi que savoir comment fonctionnent toutes ces classes.

Livrables

Une synthèse expliquant le fonctionnement général de la plateforme SWAP pourra être rédigée. Cette synthèse indiquera comment les différents composants de la plateforme SWAP interagissent entre eux.

Estimation de charge

Cette tâche est estimée à 4 jours/homme.

2.1.4 Rédaction du cahier de spécifications

Description de la tâche

Dans ce projet, un cahier de spécifications doit être rédigé. Ce document permet de décrire le projet d'un point de vue général d'abord, puis de spécifier l'architecture du système, les différentes fonctionnalités et de découper le projet en tâches.

Livrables

Le livrable est un document de type PDF réalisé avec le système de composition de documents nommé Latex.

Estimation de charge

Cette tâche est estimée à 5 jours/homme.

2.1.5 Préparation de la présentation orale à mi-parcours

Description de la tâche

Cette tâche consiste à préparer une présentation orale qui aura lieu au mois de Janvier. Il s'agit aussi de préparer un poster sur ce projet.

Livrables

Le livrable correspond ici à un rapport PDF du cahier de spécifications ainsi que d'un poster.

Estimation de charge

Cette tâche est estimée à 3 jours/homme.

2.1.6 Développement permettant de simuler une vue daltonienne

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de simuler une vue daltonienne.
1.6.1

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués, comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.7 Développement permettant de marquer les images modifiées

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de marquer les images modifiées.
1.6.2

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués, comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.8 Développement permettant d'intercepter toutes les images d'une page web

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant d'intercepter toutes les images du contenu HTML et CSS ainsi que de mettre en cache ces images.
1.6.3

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 12 jours/homme.

2.1.9 Développement permettant de mettre en cache les images modifiées

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de mettre en cache les images modifiées.
1.6.4

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.10 Tests sur l'ensemble de l'application

Description de la tâche

Cette tâche consiste à tester l'ensemble de l'application qui comprend les différentes fonctions développées préalablement.
1.6.5

Cycle de vie

Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.11 Rédaction du rapport final du projet de fin d'études

Description de la tâche

Cette tâche consiste à rédiger le rapport final de ce projet de fin d'études.

Livrables

Le livrable correspond ici à un rapport PDF.

Estimation de charge

Cette tâche est estimée à 15 jours/homme.

2.1.12 Préparation de la présentation orale finale

Description de la tâche

Cette tâche consiste à préparer une présentation orale finale qui aura lieu le 3, le 6 et le 7 Mai 2013.

Livrables

Le livrable correspond ici à un rapport PDF final du projet de fin d'études.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.2 Planning

Le diagramme de Gantt correspondant aux tâches citées précédemment est le suivant :

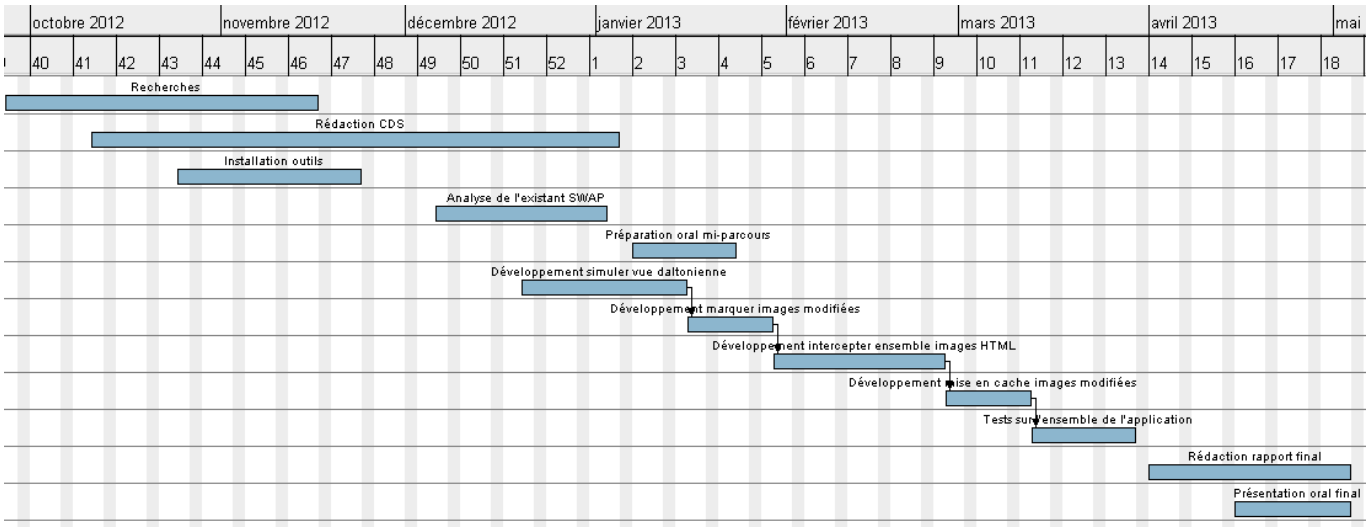

FIGURE 2.3 – Diagramme de Gantt

Il faut tenir compte du fait que le nombre de jours travaillés par semaine dans l'année peut être différent. En effet, de Septembre à fin Décembre, une seule journée par semaine est consacrée au projet de fin d'études, tandis qu'à partir du mois de Janvier et jusqu'à la fin du projet, trois journées y sont consacrées. Cela revient donc à un jour/homme par semaine sur le premier semestre et à 3 jours/homme sur le deuxième semestre.

Glossaire

Achromatopsie
Pathologie de la vision qui correspond à la non-perception des couleurs. Le patient achromate ne peut distinguer que le blanc, le noir, et les nuances de gris.

Dyschromatopsie
Défaut de la vision, caractérisé par une déficience partielle ou totale dans la distinction des couleurs (synonyme de daltonisme).

HTML
HyperText Markup Langage. Langage permettant de créer des pages Web, il utilise une structure formé avec des balises permettant la mise en forme du texte.

HTTP
Protocole de communication utilisé pour l'accès à un serveur web sécurisé.

HTTPS
HyperText Transfer Protocol Secure est la combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS.

IDE
Integrated Development Environment en anglais et Environnement de Développement Intégré (EDI) en français. C'est un programme regroupant un ensemble d'outils pour le développement de logiciels.

JDK
Java Development Kit. Logiciel édité par Sun pour le développement d'application en Java.

Proxy
Mandataire en informatique (proxy en anglais) est un composant logiciel qui se place entre deux autres pour faciliter ou surveiller leurs échanges.

RGB
Red Green Blue. Mode de description des couleurs comme mélange de trois composantes de bases, le rouge, le vert et le bleu.

Servlet
Une servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP.

WCAG
Web Content Accessibility Guidelines se développe à travers le processus du W3C en collaboration avec des personnes et des organisations du monde entier. Les documents WCAG expliquent comment rendre les contenus Web plus accessibles aux personnes handicapées.

Bibliographie

[1] Daltonisme. Le daltonisme. *http://www.daltonisme.com/*.

[2] Frédéric Dufour. Les dix règles de jacob nielsen. *http://fr.slideshare.net/fredericdufour/lutilisabilit-dun-site-web*.

[3] Projects Forge. Documentation swap. *https://projectsforge.org/projects/swap/wiki/doc/Index*.

[4] Sébastien Aupetit Mohamed Slimane. Smart web proxy for web accessiblity. *http://hant.li.univ-tours.fr/webhant/index5fbc.html?pageid=67*.

[5] Dev nozav. Subversion. *http://dev.nozav.org/intro_svn.html*.

[6] W3C. Introduction à l'accessibilité du web. *http://www.w3qc.org/ressources/traductions/introduction-accessibilite-web/?/docs/accessibilite/*.

[7] Wikipédia. Le daltonisme. *http://fr.wikipedia.org/wiki/Daltonisme*.