
- Rapport de projet - Site Internet de réseau social

DE OLIVEIRA Anthony
SIMON Adélaïde
SOLAS Yoan

DUT Informatique - 2nde année

Tuteur : LEBAS Ludovic

Table des matières

1. INTRODUCTION	3
2. DESCRIPTION DU PROJET	4
2.1 Exigences fonctionnelles	4
2.2 Exigences non fonctionnelles.....	5
3. ANALYSE ET CONCEPTION.....	6
3.1 Spécification des exigences : les cas d'utilisations.....	6
3.2 Analyse du domaine : le diagramme de classe.....	7
3.3 La conception fonctionnelle.....	8
3.4 La conception technique.....	9
3.5 La conception graphique.....	9
4. REALISATION TECHNIQUE.....	12
4.1 Les langages utilisés.....	12
4.4 Les tests	14
5. ORGANISATION ET BILAN DU PROJET	18
5.1 Organisation du groupe : méthode de travail.....	18
5.2 Résultat	19
5.3 Améliorations envisagées.....	20
6. CONCLUSION.....	21

1. Introduction

La tendance actuelle du web, dans l'évolution technologique comme dans l'évolution sociale représente une avancée dans ses utilisations.

Illustré par le terme "Web 2.0", les sites de réseaux sociaux, aussi appelés "social networking", sont des exemples parmi ces évolutions sociales. Ils permettent de connecter des internautes selon leurs centres d'intérêt afin de partager des flux d'informations, tels que des vidéos (Youtube), des photos (Flickr), de la musique (Last.fm) ou des marque-pages (del.icio.us), et d'agrandir leurs réseaux d'amis ou de faire parti de communautés. Ces exemples montrent l'ampleur que prend ce phénomène, aussi bien pour les loisirs, que pour des fins professionnels (LinkedIn).

Nous avons donc choisi pour notre projet tutoré, de développer notre propre réseau social en s'appuyant sur des grands sites web comme MySpace, Mixi, Facebook et d'autres qui représentent le web 2.0 par leurs technologies utilisées.

Ainsi l'objectif est de développer un site Internet de réseau social pour les loisirs des internautes, en mettant à leurs dispositions des outils de base pour échanger des informations, des idées, et partager des centres d'intérêts. Nous ne visons pas l'exhaustivité des outils et/ou des informations partagées. Néanmoins nous suivons certaines caractéristiques des réseaux sociaux, comme l'interaction des utilisateurs à travers un groupe d'amis et de communautés.

Pour mener à bien ce projet, nous avons tout d'abord listé et analysé les exigences requises, afin d'obtenir une expression précise des besoins. Cette analyse a permis de développer plus efficacement les différentes fonctionnalités. Ensuite, nous avons effectué des tests dans le but de comparer les exigences attendues aux résultats obtenus et d'améliorer ces derniers. Et enfin, nous avons procédé à un bilan de notre projet.

2. Description du projet

Puisqu'il s'agit d'un site Internet de réseau social, formés d'internautes, nous avons choisi pour le nom du site : « Community ». Nous pensons qu'il résume bien le sujet du site et qu'il illustre le penchant des internautes Français pour les mots anglais.

Community devra donc regrouper toutes les fonctionnalités nécessaires au partage de l'information telles que : envoyer et recevoir des emails, gérer son journal, participer dans des communautés, rechercher des membres, rechercher des communautés, inviter des internautes..., et toutes les fonctionnalités techniques comme : gérer son compte, s'inscrire, s'authentifier, se déconnecter... Mais aussi, il devra répondre à des exigences non fonctionnelles par sa qualité et ses performances.

2.1 Exigences fonctionnelles

• Inscription

Un internaute peut s'inscrire uniquement s'il a reçu une invitation de la part d'un membre de Community. Lors de cette inscription, il saisit des informations personnelles, crée son profil afin de devenir membre.

• Authentification

Afin d'accéder au site et pouvoir accéder aux différentes fonctionnalités (recherche, messagerie...), tout membre doit s'authentifier.

• Journal

Tout membre du site possède un journal, auquel il peut ajouter des articles (communément appelé blog). Il peut ainsi poster des messages contenant tout types d'information (remarques, détails de sa journée, résumé de ses vacances...)... que ses amis ou autres membres pourront visualiser et commenter. De même, un membre a la possibilité de consulter le journal des autres et de le commenter.

• Messagerie

Par le biais de sa messagerie, un membre peut consulter ses messages reçus et envoyés. Il a également la possibilité de répondre à ses messages, mais aussi de les supprimer.

• Recherche de membres

A tout moment, un membre peut rechercher d'autres membres selon différents critères : pseudo, région, ville, âge..., de manière à l'ajouter dans sa liste d'amis.

• Liste d'amis

Un membre possède une liste d'amis, qui est constituée de membres. Il gère cette liste, puisque à tout moment il peut ajouter des membres dans celle-ci.

• Invitation

Un membre a la possibilité d'inviter des internautes afin de découvrir le site. Cette invitation est envoyée par email.

• Communautés

Un membre a la possibilité de faire parti d'une ou plusieurs communautés, qui porte sur un thème spécifique, et dans lesquelles il peut poster des messages. Pour faire parti d'une communauté, il doit s'inscrire dans celle-ci. Il a la possibilité de se désinscrire d'une communauté quand il le souhaite.

- **Recherche de communautés**

Une recherche relative aux communautés doit être présente afin de permettre à un membre de s'inscrire dans une communauté qui l'intéresse en ayant préalablement consulté son profil.

- **Compte**

Un membre a des droits sur ses informations personnelles. Il peut donc modifier à tout moment ses données telles que son pseudo, son mot de passe... mais aussi son image personnelle (avatar). Il doit également pouvoir choisir si ses informations (son profil) peut être consultable par d'autres membres ou non.

- **Visibilité**

Un membre peut souhaiter divulguer ses informations personnelles, les articles de son journal uniquement à ses amis, c'est-à-dire il désire que les autres membres exceptés ses amis, n'aient pas accès à ces données. Il faut donc lui donner cette possibilité.

2.2 Exigences non fonctionnelles

- **Exigences de qualité**

Afin que le site donne envie aux membres de le faire connaître à leurs proches et de lui être fidèle, il est important de répondre aux exigences de qualité suivantes :

- ^ Ergonomie efficace

Partager des informations avec des membres, des amis doit être un plaisir. La mise en page du site doit faciliter un maximum la démarche à l'aide d'une présentation claire et intuitive. Nous reviendrons sur ce facteur de qualité dans la partie conception.

- ^ Interface graphique

Les différentes couleurs et choix typographiques doivent permettre à un utilisateur de repérer les différentes fonctionnalités qui s'offrent à lui.

- ^ Charte d'utilisation

Puisque les utilisateurs du site sont tous différents (sexe, origine...), une charte d'utilisation doit être consultable par tous les utilisateurs, afin qu'ils connaissent les règles d'utilisation du site.

- ^ Aide en ligne

À tout moment, l'utilisateur doit pouvoir consulter des pages d'aide.

- ^ Contact

Pour toute questions ou informations, l'utilisateur doit pouvoir contacter le webmaster.

- **Exigences de performance**

Il faut aussi prendre en compte les exigences quantitatives :

- ^ le site Community doit pouvoir supporter plus de 100 connexions simultanées ;

3. Analyse et conception

Dans cette partie, nous utilisons la modélisation UML pour représenter les spécifications des exigences grâce au diagramme de cas d'utilisation, mais aussi pour analyser le domaine avec le diagramme de classe. Par la suite, nous abordons la conception, d'un point de vue fonctionnel, technique et graphique.

3.1 Spécification des exigences : les cas d'utilisations

Nous allons répondre aux questions suivantes : Quels sont les utilisateurs du système ? Quelles sont leurs interactions avec celui-ci ? Il faut donc identifier les différents acteurs ainsi que les cas d'utilisation c'est-à-dire les différentes fonctionnalités du système.

Les acteurs pour le site Community sont les suivants :

- **L'internaute / l'invité** : personne qui n'est pas membre du site et qui souhaite le devenir.
- **Le webmaster** : rôle des personnes qui ont en charge la maintenance du site Web.
- **Le membre** : personne qui possède un compte, qui peut rechercher des membres, fait parti de communautés, possède des amis...

Les principaux cas d'utilisation des acteurs précédemment identifiés, ont été bien mis en évidence dans la partie précédente. Voici donc le diagramme de cas d'utilisation.



Diagramme de cas d'utilisation.

Remarque : par la suite, le webmaster ne sera plus pris en compte.

L'annexe A décrit de façon détaillée les cas d'utilisation précédemment identifié, dans le but d'obtenir une expression précise des besoins. Pour chaque cas d'utilisation, on trouvera les différentes interactions entre les utilisateurs et le système, les scénarios des cas d'utilisation ainsi que les alternatives ou erreurs que peut rencontrer le système, dans le but de connaître les différentes étapes à développer pour chaque fonctionnalité.

3.2 Analyse du domaine : le diagramme de classe

Dans cette partie, nous étudierons les entités statiques du système. Ceci est illustré par le diagramme de classes suivant :

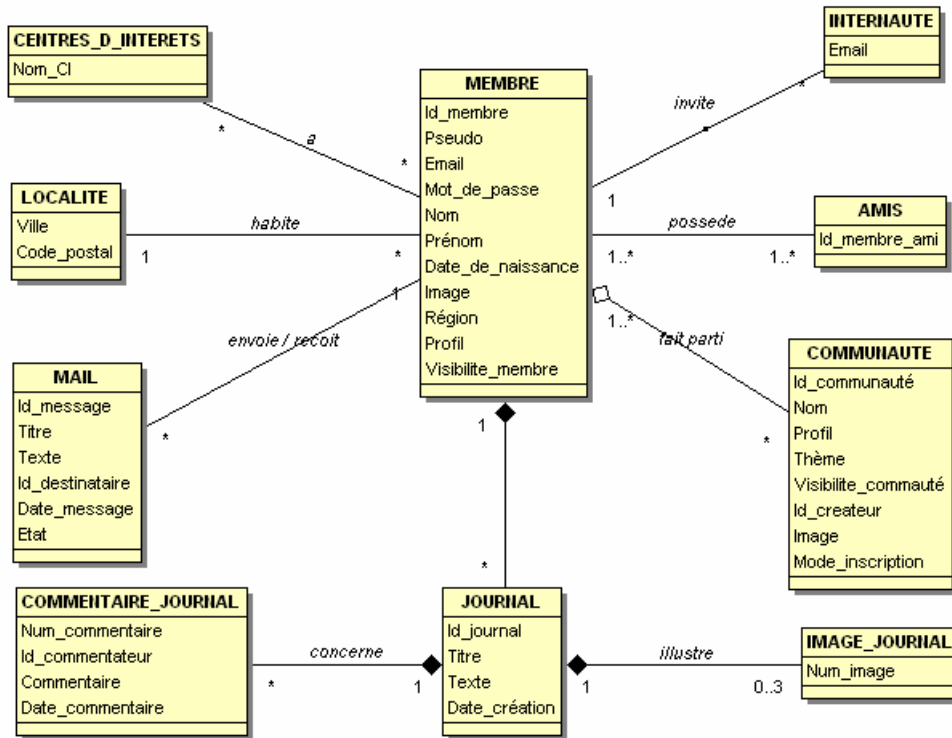


Diagramme de classes.

Remarques sur les différentes associations et cardinalités

Relation	Type d'association	Multiplicité
Invite	Association	<ul style="list-style-type: none"> Un membre peut inviter 0 ou plusieurs internautes. (*) Un internaute est invité par 1 seul membre. (1)
Possède	Association	<ul style="list-style-type: none"> Un membre possède 1 ou plusieurs amis (au moins celui qui l'a invité). (1..*) Une personne est amie avec 1 ou plusieurs membres. (1..*)
Fait parti	Agrégation	<ul style="list-style-type: none"> Un membre fait parti d'aucune ou plusieurs communautés. (*) Un communauté est composée de 1 ou plusieurs membres (au minimum son créateur). (1..*)
Poste	Composition	<ul style="list-style-type: none"> Un membre poste 0 ou plusieurs articles dans son journal. (*) Les articles d'un journal ne concernent qu'un seul membre. (1) La destruction du membre entraîne bien sur la destruction de tous les articles de son journal.
Concerne	Composition	<ul style="list-style-type: none"> Un article contient 0 ou plusieurs commentaires. (*) Un commentaire concerne un seul article. (1) La destruction du journal entraîne la destruction de tous ses commentaires.
Illustre	Composition	<ul style="list-style-type: none"> Une image illustre 1 seul journal. (1) Un journal peut être illustré par 0 ou 3 images. (0..3) La destruction du journal entraîne la destruction des images l'illustrant.
Envoie / Reçoit	Association	<ul style="list-style-type: none"> Un membre peut envoyer ou recevoir 0 ou plusieurs emails. (*) Un mail est envoyé ou reçu par un seul membre. (1)
Habite	Association	<ul style="list-style-type: none"> Un membre habite 1 seule localité. (1) Dans une localité, on peut trouver 0 ou plusieurs membres. (*)
A	Association	<ul style="list-style-type: none"> Un membre possède 0 ou plusieurs centres d'intérêts. (*) Un centre d'intérêt concerne 0 ou plusieurs membres. (*)

3.3 La conception fonctionnelle

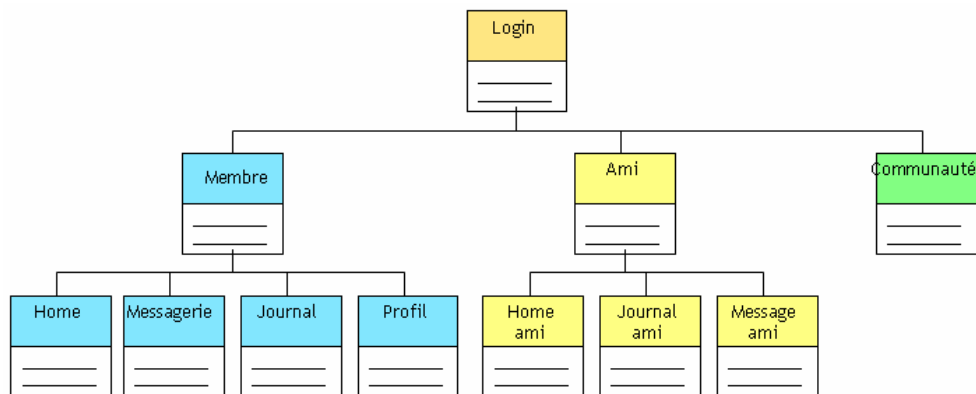
□ La structure du site

Le site web est basé sur une structure que l'on appelle structure hiérarchisée. Il s'agit d'une structure découpée en blocs d'informations, ou de fonctionnalités, reliés à un même noeud.

La page d'authentification fait office de noeud principal qui donne racine à diverses fonctionnalités une fois authentifié.

Ces fonctionnalités sont présentes dans 3 barres d'outils regroupées par thèmes:

- fonctionnalités d'un membre
- fonctionnalités d'un membre face à ses amis
- fonctionnalités d'un membre dans ses communautés

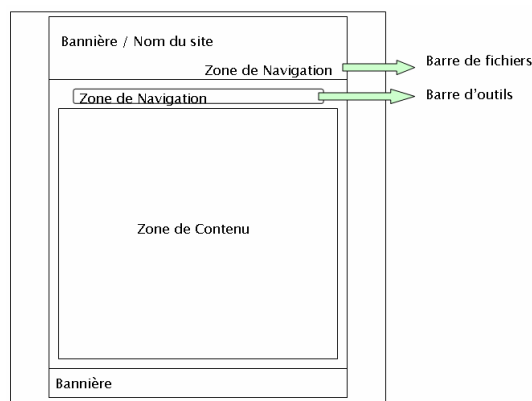


Structure hiérarchisée simplifiée.

Nous avons donc choisi ce type de schéma d'organisation puisqu'il s'adapte facilement à nos objectifs, qui sont d'une part, une facilité de navigation pour l'utilisateur en se familiarisant très rapidement à ce schéma, et d'autre part une organisation soignée de nos informations et fonctionnalités.

□ La structure des pages

Toutes les pages du site Community ont une même structure.



Trame simplifiée des pages.

Les éléments changeant entre les différentes pages sont le contenu, ainsi que la barre d'outils. On reviendra sur cela lors de la conception graphique.

Le site s'organise d'un conteneur principal, l'index.

L'index est présent continuellement pendant la navigation puisqu'il affiche toutes les autres pages du site. C'est lui qui se charge d'afficher les deux bannières, mais aussi de changer la zone de contenu.

Les pages a affiché sont envoyées à travers la méthode GET et sa variable « page » lorsqu'un utilisateur clique sur un lien pour naviguer.

Exemple :

```
index.php?page=home  
index.php?page=aide
```

La variable « page » est analysée par une page de traitement des pages (include_page.php), qui va afficher le contenu désiré ainsi que la barre d'outils correspondante si nécessaire.

Exemple:

```
$page = $_GET['page'];  
  
if ( $page == "home" )  
{  
    include("include_barredoutils_home.php");  
    include("include_home.php");  
    $test= 1;  
}  
  
if ( $page == "aide" )  
{  
    include("include_aide.php");  
    $test= 1;  
}
```

Nous avons donc choisi ce type de navigation, puisqu'il était très bien approprié à un site web fourni en page et bien qu'il soit considéré comme brutal par certains informaticiens, il a fait ses preuves en terme de sécurité.

De plus, il est d'une part très dynamique, puisqu'il nous permet d'ajouter des pages très facilement sans réorganiser tout le système de navigation en parcourant les nombreuses pages. D'autre part, il n'y a pas besoin de dupliquer l'index puisqu'il fait office de conteneur. Et enfin, il améliore la rapidité du site, car le conteneur (l'index) est chargé qu'une fois puis gardé en cache tout au long de la navigation.

3.4 La conception technique

✧ Les règles de nommage et de codage

Nous devons respecter certaines règles de nommage et de codage que nous nous sommes définis. Elles permettront une compréhension plus rapide du code par les différents membres du groupe. L'*annexe B* définit ces règles.

3.5 La conception graphique

L'ergonomie est un point important. Son objectif principal est de faciliter l'utilisation du site par les utilisateurs. Pour cela, nous nous imposerons certaines contraintes :

✧ Accessibilité des navigateurs

Le site Community doit être fonctionnel sous les navigateurs les plus utilisés, c'est-à-dire : Firefox et Internet Explorer. Les différentes fonctionnalités développées devront donc être testées sur les deux navigateurs.

✧ Le système de navigation

La navigation doit être simple, même pour les usagers débutants. Il est donc indispensable que l'apprentissage du site ne demande pas des efforts à l'utilisateur. L'utilisation de menu qui répertorie les principales fonctionnalités permet à l'utilisateur de minimiser son effort mental.

La barre de fichiers, qui est un menu horizontal statique est donc composé de :

- home : la page d'accueil
- inviter : inviter des personnes qui n'utilisent pas le site
- articles : derniers articles créés
- rechercher membre
- rechercher communauté
- aide
- déconnexion

A horizontal menu bar with a blue gradient background. It contains seven items: Home, Inviter, Rechercher Membre, Articles, Rechercher Communauté, Aide, and Déconnecter, each separated by a vertical line.

Barre de fichiers.

La barre d'outils quant à elle changera selon la navigation de l'utilisateur. C'est-à-dire selon s'il se trouve sur une page qui le concerne, ou une page qui concerne un ami ou un autre membre, ou une page relative à ses communautés.

Ainsi lorsqu'il navigue que des pages le concernant, la barre d'outils sera composée des onglets suivants :

- home : sa page d'accueil
- messagerie : sa boîte mail
- journal : son journal avec ses articles
- profil : sa description
- compte : ses données personnelles qu'il peut modifier.

A horizontal menu bar with a blue gradient background. It contains five items: Home, Messagerie, Journal, Profil, and Compte, each separated by a vertical line.

Barre d'outils : utilisateur.

Lorsque l'utilisateur navigue sur une page d'un autre membre ou ami, la barre d'outils sera composée des onglets :

- home : le profil de l'ami / du membre
- message : envoi d'un message à l'ami / membre
- journal : les différents articles de l'ami / membre
- ajouter ami : si l'utilisateur souhaite demander à ce membre d'être son ami

A horizontal menu bar with a yellow gradient background. It contains four items: Home, Envoyer message, Liste journal, and Ajouter ami, each separated by a vertical line.

Barre d'outils : ami / autre membre.

Lorsque l'utilisateur navigue sur une page relative aux communautés, la barre d'outil sera composée des onglets suivants :

- home : le profil de la communauté
- ajouter communauté : si l'utilisateur souhaite s'inscrire dans une communauté
- inviter communauté : inviter des membres dans une communauté
- quitter communauté

A horizontal menu bar with a green gradient background. It contains four items: home, ajouter communauté, invite communauté, and quitter communauté, each separated by a vertical line.

Barre d'outils : communauté.

✧ Apparence du site

➤ La charte graphique

Afin de donner à l'utilisateur des repères au sein du site et de faciliter sa visite, il doit y avoir une cohérence entre chaque page. La charte graphique détermine les différentes règles graphiques et ergonomiques.

Elle contient :

- les dimensions des pages
- les couleurs à employer (le texte, les liens, les liens actifs, les boutons...)
- les styles à employer (famille, taille...)
- les types d'images...

L'*annexe C* précise la charte graphique du site.

➤ Choix des couleurs

Pour réaliser l'image du nom « Community », nous avons cherché les tendances actuelles. Il en a découlé qu'une image avec des formes rondes avec des couleurs bleu et orange correspondait le mieux, mais le choix était encore assez vaste pour la création.

Anthony a des bases sur Photoshop, il a développé quelque chose correspondant aux tendances actuelles. Il a cherché une police correspondante : assez ronde et avec des lettres pas très définies.

Une fois l'image intégrée sur le fond, il fallait trouver quelque chose permettant de la faire ressortir de la bannière : une lueur blanche extérieure au logo correspondait parfaitement.

Les bannières et l'image du nom « Community » réalisées et approuvées par les autres membres du groupe, il a donc fallu s'adapter aux couleurs utilisées pour choisir une palette de couleur pour le site.

Nous avons décidé que la couleur dominante pour le site sera le bleu. Bien que cette couleur a pour symbolique principal le froid, c'est pourtant la couleur préférée des personnes (à hauteur de 33% d'après le site <http://www.1001-votes.com>). De plus le bleu donne une impression d'espace, et apporte un équilibre tout en accentuant les formes.

Une fois cette couleur définie (#6699FF) nous avons utilisé le logiciel Color Scheme afin de connaître les couleurs qui s'harmonisent le plus avec celle-ci. D'après un accord de quatre tons de couleur, nous avons choisi le orange et le vert. D'autant plus que le orange a pour caractère psychologique la sociabilité et le fait d'être ouvert, et le vert représente les loisirs, la détente.

➤ Contenu

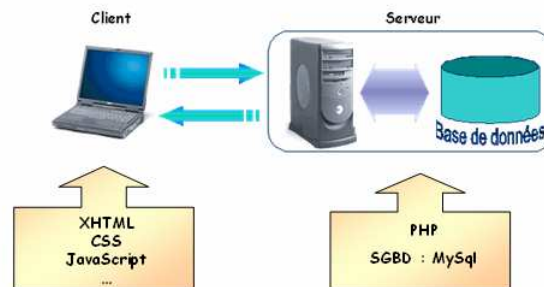
Chaque page doit avoir un titre propre permettant à l'utilisateur de savoir sans réfléchir dans quelle section du site il se trouve.

Exemple : Rechercher un membre

Les phrases simples et les textes courts sont privilégiés car cela favorise une lecture rapide pour l'utilisateur.

4. Réalisation technique

4.1 Les langages utilisés



□ XHTML 1.0

Afin d'être au standard du W3C sur la famille HTML, nous avons utilisé le XHTML 1.0, plutôt qu'une mise en page en HTML difficile à manipuler. Le contenu et la présentation seront développés de façon distincte.

□ CSS

Afin de manipuler la présentation, nous avons utilisé des feuilles de style CSS. Voici les raisons : D'une part, il permet d'alléger le code source écrit en XHTML, puisque tout ce qui est relatif à la présentation est géré dans un fichier séparé. Ce qui entraîne donc un chargement plus rapide des pages, qui est après manipulé par la feuille de style.

Et d'autre part, il permet de nous retrouver plus facilement dans notre code et ainsi facilite les modifications à effectuer, puisqu'au lieu d'avoir à modifier toutes les pages une à une, nous avons juste à modifier le fichier CSS.

Exemples d'utilisation :

- tous les choix graphiques
- pages différentes lors que l'utilisateur s'est authentifié ou non.

□ PHP

Pour le côté serveur, nous avons choisi le langage PHP, ceci pour plusieurs raisons :

Tout d'abord, le PHP gère très bien les requêtes SQL. Ce qui est important pour manipuler les résultats de requêtes SQL.

Mais aussi, le PHP possède beaucoup de fonctions utiles telle que la cryptologie...

Enfin, le PHP est un des langages côté serveur le plus utilisé, il existe une grande communauté d'utilisateurs de PHP.

Exemples d'utilisation :

- cryptage de mot de passe
- traitement des différents formulaires
- utilisation de variables de session

□ SQL

L'association PHP/MySQL est de plus en plus utilisée ces dernières années. C'est pourquoi nous avons choisi le serveur de base de données MySQL qui offre de grandes performances en flexibilité et montée en charge. Il intègre de nombreuses fonctionnalités et assure une bonne sécurité des données.

□ AJAX

Nous avons intégré à notre projet la méthode développement AJAX dans plusieurs de nos modules. En effet, nous l'utilisons en grande partie pour ses requêtes asynchrone qui nous permettent de faire et de manipuler les requêtes au serveur sans déranger la page de l'utilisateur.

Exemple d'utilisation :

- chargement des listes d'amis et communautés dans les pages home
- recherche des utilisateurs
- module de messagerie

□ XML

Nous exploitons le XML pour le traitement des résultats des requêtes envoyées au serveur. Le XML est généré grâce au PHP et aux requêtes SQL afin d'extraire les données de la BD et les traitées sous forme XML.

□ DOM et JAVASCRIPT

Une fois la feuille XML générée, nous pouvons afficher et interagir dynamiquement avec la page de l'utilisateur grâce au JavaScript et au DOM.

Exemple d'utilisation :

- affichage dynamique des villes selon un code postal saisi, sans recharger la page
- affichage dynamique des emails reçus
- affichage dynamique du formulaire de réponse à un mail en ayant cliquer préalablement sur un bouton

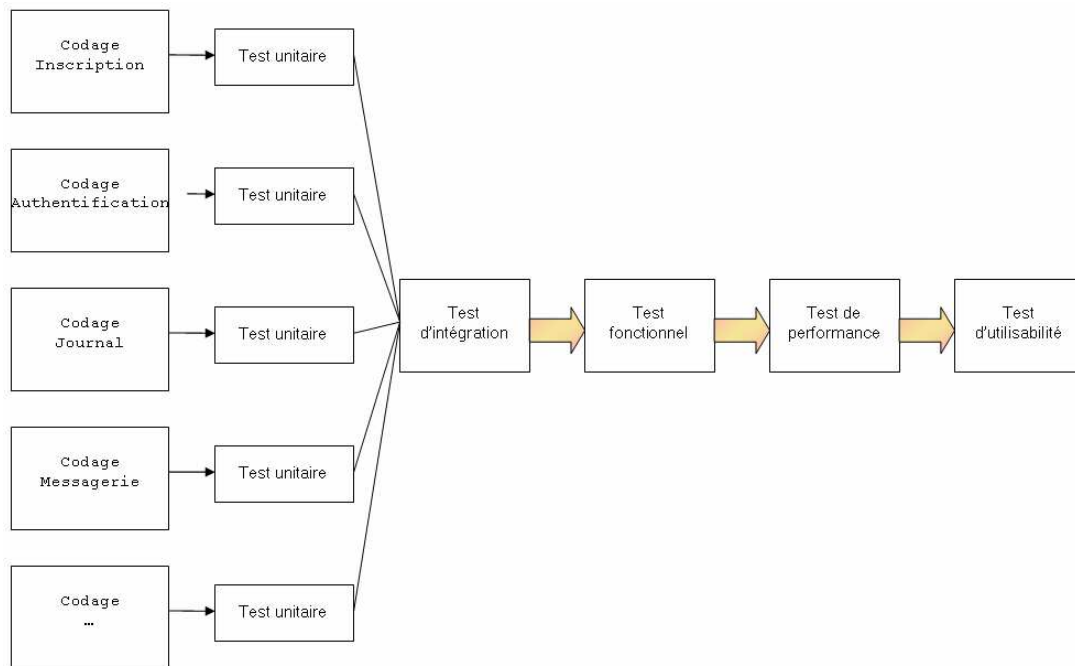
Nous mairions plusieurs langages pour plusieurs raisons : leurs performances, leurs standardisations et leurs portabilité face à une multitude de langage Web.

4.4 Les tests

Afin que le site soit de qualité, il est important de réaliser des tests. Ces tests ont pour but de détecter les erreurs éventuelles et permettent de les corriger pour qu'il n'y ait pas d'impact sur le fonctionnement du site.

Pour cela au cours du développement, nous avons effectué 5 types de tests :

- des tests unitaires
- des tests d'intégrations
- des tests fonctionnels
- des tests de performances
- des tests d'utilisabilité.



Etapes des tests.

□ Les tests unitaires

Ces tests permettent de valider chaque fonctionnalité développée indépendamment des autres. L'objectif de ce test est de vérifier que chaque fonctionnalité développée correspond aux attentes des spécifications. Une fois les tests validés, le programme pouvait être envoyé à Yoan pour l'intégration.

L'*annexe D* illustre les tests unitaires pour le module de l'inscription.

Résultats des tests

Toutes les fonctionnalités : inscription, création de message, affichage de la liste de messages, réponse à un message, création d'article... ont été testées avant d'être envoyées à Yoan. Le résultat était conforme à ce que nous avons défini dans le cahier des charges, et les cas d'erreurs ont été traités. Une fois les fichiers reçus, Yoan pouvait procéder à l'intégration des fonctionnalités, et réaliser des tests d'intégration.

□ Les tests d'intégrations

Les tests d'intégration consistent en l'assemblage des fonctionnalités testées unitairement. L'objectif de ces tests est donc de valider l'intégration des différents modules entre eux, et de vérifier que le résultat est conforme à la conception.

L'intégration d'un module dans le site est une phase délicate. Elle peut soulever des problèmes non prise en compte durant la phase d'analyse du module tout comme conduire à de simples bugs. C'est pourquoi les tests d'intégrations sont très importants puisqu'ils peuvent déceler ces bugs qui ne se sont pas apparus lors des tests unitaires, et qu'il faut corriger.

Mise en place des tests

On teste tout d'abord la compatibilité du module avec les autres. Ensuite on fait des tests plus poussés sur les fonctionnalités du module. Puis on teste les autres modules qui peuvent interagir avec celui-ci, en exécutant des tâches plus précises qui pourrait avoir des répercussions sur ce module. Bien entendu des corrections sont faites pour passer à des tests plus poussés.

L'*annexe E* illustre un test d'intégration simple, celle du formulaire.

Problèmes souvent rencontrés lors de l'intégration

Les problèmes les plus souvent rencontrés dans ces tests sont la redondance des importations de bibliothèques en PHP, redondance des variables de session, ainsi que les fonctions PHP incompatibles entre elles (exemple : header). Mais les problèmes peuvent être beaucoup plus importants. En effet, certains modules utilisent beaucoup de ressources. Ils doivent envoyer plusieurs requêtes lors de chaque interaction avec l'utilisateur. C'est pourquoi certaines fonctions AJAX intégrées dans ces modules doivent être correctement dissociées, et isolées pour éviter tout conflit.

Ces tests permettent de démarrer une série de tests dans leur élément final.

□ Les tests fonctionnels

Ces tests permettent de savoir si les différentes fonctionnalités sont opérationnelles, c'est-à-dire si elles sont conformes au cahier des charges, et se déroulent correctement ; mais aussi ils permettent d'observer le comportement du système lorsque des valeurs inattendues ou incorrectes sont saisies (tests de robustesse), c'est-à-dire tester les cas d'erreurs prévus via des messages d'erreur.

L'objectif de ce test est de vérifier que le site est conforme au cahier des charges et s'il résiste à des comportements anormaux. Si le système réagit bien à ces erreurs, cela veut dire que le site a une bonne aptitude à résister aux erreurs.

Mise en place des tests

Pour chaque fonctionnalité développer, nous avons regardé si elles étaient conformes au cahier des charges, et si les différentes erreurs que peut rencontrer le système sont gérées (par exemple, par des messages d'erreur).

L'*annexe F* rapporte quelques tests que nous avons effectués sur différentes fonctionnalités.

Résultats des tests

Selon les valeurs d'entrées, les sorties correspondaient bien à nos attentes. Elles sont traitées par des messages d'erreur qui permettent d'améliorer la sécurité, notamment lors de saisie d'URL incorrectes. Le cahier des charges est respecté pour toutes les fonctionnalités excepté lors des tests du module email où nous avons remarqué que la fonctionnalité « supprimer un email » ne correspondait pas à nos attentes.

□ Les tests de performance

Pour effectuer les tests de performance nous avons utilisé « Jmeter ». Cet outil permet de tester la montée en charge du site en effectuant un scénario défini. Ce scénario sera réalisé pour X utilisateurs virtuels, que nous choisirons. Il permet de voir comment réagit notre site en situation réelle. Nous pourrions ensuite étudier les résultats et en déduire les pages qui ont besoins d'être optimisées.

L'objectif de ces tests est donc de détecter les pages qui peuvent être améliorées, notamment en modifiant les requêtes. Ces tests sont importantes pour les pages les plus fréquentées, puisqu'elles se doivent d'être le plus fluides possible. Si à la suite de ces améliorations, les pages sont encore trop lentes à s'afficher pour un certains nombre de connexions d'utilisateur, il est envisageable de séparer celles-ci en plusieurs pages.

Une fois les pages optimisées, le second objectif est de connaître le nombre de connexions simultanées qui peut être gérées.

Mise en place des tests

Nous avons choisit pour illustrer ces tests la page d'accueil et le fichier PHP qui génère le fichier XML permettant l'affichage de la liste des amis et des communautés dans la page d'accueil. Nous avons effectué une première vague de tests avec des nombres de connexions différents. Par la suite, nous avons simplifié une requête et réaliser une seconde vague de tests, toujours avec le même nombre de connexions.

L'*annexe G* illustre ces tests.

Résultats des tests

Grâce aux données de « Jmeter », nous élaborons un graphique de performances d'après trois critères : le temps moyen d'affichage, le temps nécessaire à satisfaire 90% des requêtes, et le temps maximal d'affichage.

Après la simplification de la requête, on remarque clairement que le temps d'affichage est divisé par deux. Le fichier PHP optimisé est donc deux fois plus rapide lors de son chargement que son ancienne version.

Le code de la page d'accueil n'a pas été directement modifié, mais son temps de chargement a nettement diminué. Ce gain de rapidité est dû à l'amélioration du fichier PHP, puisque il est utilisé lors du chargement de la page d'accueil.

□ Les tests d'utilisabilité

Ces tests permettent d'observer le comportement de l'utilisateur lorsqu'il navigue sur le site : connaître ses impressions sur le graphisme utilisé, voir les problèmes qu'il rencontre, les questions qu'il peut se poser, les fonctionnalités qu'il apprécie.

Mise en place des tests

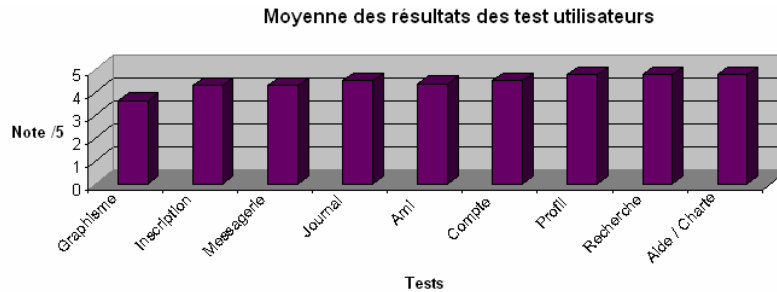
Nous avons réalisé ces tests grâce à quatre personnes de notre entourage. Ces personnes ont été choisies selon leur âge, entre 20 et 28 ans, qui reflètent les utilisateurs potentiels du site Community. Nous leur distribuons un questionnaire à remplir.

Il laisse ses critiques sur la feuille préalablement distribuée. Sur cette feuille nous nous intéressons à deux critères : graphique et fonctionnel.

L'*annexe H* comporte les tests d'utilisabilité de ces quatre personnes.

Résultats des tests

Pour synthétiser ces résultats, nous avons reporté sur un graphique les évaluations générales des utilisateurs.



Nous pouvons en conclure, que les utilisateurs sont satisfaits des fonctionnalités par leur facilité d'utilisation, et leurs fluidité, cependant, coté graphique la seule remarque récurrente est "trop de bleu".

Nous pouvons apporter quelques modifications immédiates à la suite de ces résultats :

- sur la page d'authentification, le terme "Login" doit être remplacé par "Email" de façon à ce que l'utilisateur ne pense pas qu'il s'agisse de son pseudo.
- dans la messagerie, les titres : "Emails reçus" et "Emails envoyés" doivent être rajoutés de manière à ce que l'utilisateur arrive à se situer dans la navigation.
- dans la barre d'outils, les différents onglets doivent être personnalisés pour que l'utilisateur soit plus impliqué (Ma messagerie, mon journal...)

Les modifications futures que nous devons effectués sont :

- par rapport au graphisme, notamment les couleurs utilisées

5. Organisation et bilan du projet

Comment s'est-on organisé pour mener à bien ce projet ? Quels sont les résultats du projet ? A-t-on réussi à atteindre les objectifs fixés ? Quelles sont les différences entre objectifs et résultats ? Quelles améliorations sont envisagées ? Dans cette partie nous répondrons à ces différentes questions.

5.1 Organisation du groupe : méthode de travail

Au début du projet nous avons décidé de se répartir les tâches en fonctions des langages utilisés. Nous avons remis en question ce découpage puisque nous ne pouvions pas développer ensemble, car nous n'habitons pas à proximité les uns des autres. Par conséquent, nous devons développer chacun de notre côté puis mettre en commun.

Le mode de travail envisagé au début s'est donc vu abandonné pour une structure de développement dite « outsourcing ».

Détaillons cette structure à travers l'exemple d'un module :

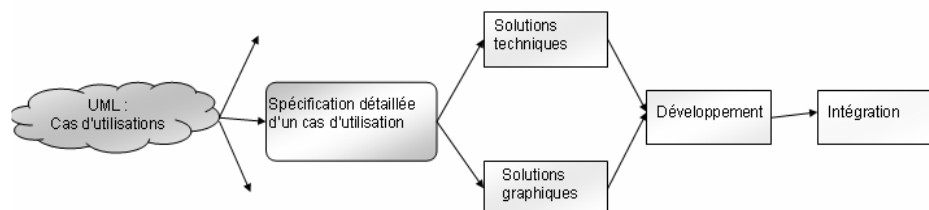


Schéma illustrant notre méthode de travail.

Tout d'abord, suite à l'élaboration du diagramme de cas d'utilisation, nous élaborons le diagramme de séquence système du module en question afin de cerner toutes les fonctionnalités. Nous avons réalisé cette tâche tous ensemble afin d'avoir toutes les opinions des membres du groupe dans le but d'avoir la meilleure analyse possible.

Ensuite une fois l'UML établi, nous définissons les différents éléments graphiques du module de sorte à ce qu'un membre puisse commencer à développer la mise en page relative au module, le CSS. Mais aussi nous définissons les différents éléments techniques en listant les différentes étapes (fonctionnalités) à réaliser relatifs au bon fonctionnement du module.

Selon les modules, leurs complexités, et les différentes compétences de chacun de nos membres, nous développons à un ou deux les différents éléments du module, indépendamment les uns des autres. Une fois l'élément développé et les différentes fonctionnalités réalisées, nous effectuons des tests unitaires. (Nous reviendrons sur ces tests un peu plus tard). Lorsqu'ils sont validés, l'élément est transmis pour l'intégrer au site lui-même.

L'élément est donc intégré dans l'ossature du site. Des modifications du code sont alors nécessaires, car les techniques utilisées par chacun d'entre nous ne sont pas communes (par exemple, innerHTML vs DOM). Une fois cette tâche réalisée, il faut la valider en effectuant des tests d'intégration.

Par la suite, nous vérifions que le produit répond aux attentes, c'est-à-dire que le module correspond au scénario prévu, sur le point de vue graphique (effet visible par l'utilisateur) et sur le point technique par les résultats de la base de données (ce que ne voit pas l'utilisateur).

Notre méthode de travail possède certains avantages :

- les programmes sont testés à plusieurs reprises ;
- le développement est séparé de l'intégration, ce qui permet lors du développement de ne pas se soucier de l'intégration, et donc d'être plus productif.

Mais aussi quelques défauts :

- possibilité pour un développeur d'être bloqué sur un élément du module ;
- la manière de coder est différente les uns des autres.

Les clés de notre méthode :

- la propreté du code et respect de la charte de codage ;
- la communication entre les membres du groupe quant à l'avancement des modules.

Mais les clés de cette méthode peuvent devenir des limites si elles ne sont pas respectées.

5.2 Résultat

Les exigences fonctionnelles fixées dès le début par notre cahier des charges ont presque été menées à bien. En effet, les principaux modules caractérisant un réseau social ont été intégrés.

Les exigences non fonctionnelles qui assurent la qualité et la performance du site ont toutes été développées. L'ergonomie du site est satisfaisante pour les utilisateurs d'après les tests d'utilisabilité, le site est compatible avec Internet Explorer et Firefox. Les actions telles que contacter le webmaster, accéder à l'aide en ligne, et à la charte d'utilisation sont possibles.

La performance du site a été prouvée d'après les tests de montée en charge permettant ainsi à plus de 100 utilisateurs de se connecter en même temps.

Toutefois, certains points n'ont pu être développés par manque de temps.

Le module mail, où la suppression d'un message n'a pas été réalisée correctement. La création d'une nouvelle table dans la base de données aurait été la solution, mais notre analyse préliminaire n'était pas correcte et nous nous en sommes rendu compte trop tard. Si notre analyse de départ avait été correcte, nous aurions pu valider complètement ce module.

Le module communauté, n'a pas été totalement finalisé : la création de communauté, la recherche de communautés ont été développés, mais par souci de temps, n'ont pu être intégrés. Nous avons donc laissé de côté ce module, mais en laissant les liens s'y rapportant qui pourrait donc être intégré sans grande difficulté.

Bien que nous aurions pu développer le module communauté et modifier le problème rencontré dans le module messagerie, nous avons préféré vérifier et améliorer la performance et la qualité du site, notamment grâce aux différents tests.

Cette solution a été envisagée lorsque nous nous sommes rendu compte d'un manque de temps, nous avons donc réfléchi afin de mener à bien notre projet de la meilleure façon possible. C'est pourquoi nous avons décidé de recarder notre sujet en ne gardant que les modules les plus importants et en contrepartie nous avons décidé d'améliorer la qualité et la performance de notre application.

L'application finale n'intègre pas tous les modules envisagés dans notre cahier des charges. Néanmoins elle est conforme à nos objectifs qui étaient d'intégrer les modules de bases caractérisant les réseaux sociaux. De plus l'application a été améliorée en termes de sécurité, performance et qualité qui rend son utilisation beaucoup plus rassurante, rapide et facile à défaut de l'exhaustivité des outils.

5.3 Améliorations envisagées

La finalité du projet était de le mettre en ligne, mais n'ayant pas pu mener à bien cette tâche, nous avons l'intention de continuer à développer ce site, après notre séjour universitaire. Nous envisageons de le mettre en ligne, afin de le voir évoluer en situation réelle.

Pour cela nous devons nous informer des différentes étapes de la mise en ligne, mais auparavant nous devons compléter le site actuel et améliorer quelques outils déjà présents.

La première amélioration sera de finaliser le module communauté, où les utilisateurs pourront partager leurs centres d'intérêt, non seulement entre amis mais aussi avec des gens qui ont les mêmes centres d'intérêt. Le cercle d'amis de chaque utilisateur se verra donc agrandi, par le biais des membres de la communauté, mais aussi par les amis de ces derniers.

La communauté fonctionnera sur un principe similaire à celui du journal, chaque membre écrira un article qui sera partagé et en rapport avec le thème de la communauté.

Exemple : une communauté football, où chaque utilisateur pourra débattre avec les autres sur un match, ou ils pourront se donner les dernières info relatives au monde du football, supporter leurs équipes favorites, etc...

La communauté sera gérée par son fondateur, il aura le droit de faire entrer ou non un membre dans la communauté. A lui aussi le pouvoir de bannir quelqu'un de la communauté, pour des raisons propres, discours déplacé, pas en rapport avec la communauté, propos injurieux, racistes, ou encore mauvaise ambiance créé dans la communauté par un utilisateur.

Ensuite nous améliorerons le module mails afin que la suppression de mails soit opérationnelle. Nous reviendrons donc sur cette fonctionnalité.

Nous envisageons aussi d'améliorer le profil de l'utilisateur, de manière à ce qu'il puisse partager des photos, de liens Internet, et des vidéos. Les vidéos seront « uploadé » sur le serveur, et comprimé en un format propre au lecteur universel que nous utiliserons.

Mais aussi, le développement d'un module « back office » sera indispensable pour réguler les informations circulant entre les utilisateurs.

Enfin les tests effectués avec Jmeter nous pousseront à optimiser certaines pages, donc probablement d'en revoir la structure entière.

Après ça le site devrait être opérationnel pour une mise en ligne. Nous ouvrirons une communauté libre à tout utilisateur, lié au développement du site où chacun pourra laisser ses idées pour un site lui ressemblant, de sorte à l'améliorer en fonctions des gens qui y passent du temps.

6. Conclusion

Le projet tutoré est beaucoup plus complexe que les projets que nous avons déjà réalisés (miniprojets de BP, site Web de vente en ligne Medelib, miniprojet IHM en GL...). Il permet de mettre en relation plusieurs enseignements, utilisés sous une dimension professionnelle.

Tout d'abord, ce projet nous a permis d'appliquer les connaissances que nous avons acquises durant notre DUT, telles que la modélisation, les différents langages de programmation, la gestion de projet mais aussi la communication.

Grâce à celui-ci chaque membre de l'équipe a pu renforcer ses connaissances mais aussi apporter aux autres membres son savoir et ses compétences afin d'harmoniser l'efficacité de l'équipe.

A travers des méthodes de travail et des outils, ce projet nous a permis de nous immerger dans un univers professionnel.

Il est vrai que de créer une application et respecter un cahier des charges rendent un projet tutoré intéressant et professionnel mais il y a aussi toutes les démarches qui ne sont pas visibles et qui rendent enrichissante une telle expérience : écouter l'opinion de chacun des membres de l'équipe, savoir communiquer et argumenter afin d'opter pour les meilleurs choix, s'organiser sur les plans personnels et collectifs, gérer les imprévus, respecter des délais pour ne pas gêner ses collègues et pour ne pas retarder tout le projet.

Le projet nous apporte donc à chacun une idée sur l'organisation dans le monde professionnel et qui permettra de nous adapter plus facilement lors de notre stage.

Les sources d'information et bibliographie

CHU Nicolas. Réussir un projet de site web, 3^{ème} éd. Paris : Eyrolles, 2005, 242p.

SALZMAN Claude, Développer un projet internet. Paris : Dunod, 2001, 255p.

HARDY Jean-Marc, Check-list pour réussir son site Web. Paris : Dunod, 2004, 152p.

ROQUES Pascal, UML Modéliser un site e-commerce. Paris : Eyrolles, 2002, 152p.

ULLMAN Lary, PHP & MySql : Développement de sites web. Paris : Campus Press, 2004, 408p.

CARFANTAN Jean, JavaScript. Paris : Micro Application, 2005, 192p.

PLASSE Michel, Développer en Ajax. Paris : Eyrolles, 2006, 314p.

ENGELS Jean, XHTML et CSS. Paris : Eyrolles, 2006, 508p.

<http://fr.wikipedia.org/wiki>

<http://www.manuelphp.com/>

<http://www.phpdebutant.org/>