

# Rapport d'activité

## Table des matières

Objectifs et mission.....	2
Support matériel, installations.....	2
Modèle de développement.....	3
Modélisation.....	3
Réalisations logicielles.....	5
Tests.....	6
Production de documentation.....	6
Temps de travail.....	6
Problèmes rencontrés.....	7
Améliorations souhaitées.....	7
Conclusion.....	8

## **Objectifs et mission**

Ce projet a été pour moi l'occasion d'effectuer un certain nombre de tâches variées. Certaines d'entre elles sont communes à l'ensemble du groupe, d'autres sont plus spécifiques et liées à mon statut de responsable développement.

En tant que membre « général » du groupe LAPI.NET :

- Proposition du site et du nom LAPI.NET.
- Participation à la définition des besoins, des spécifications.
- Participation à la modélisation.
- Tests en local et sur le site en production.
- Participation au design.

En tant que responsable développement du projet, différentes tâches sont à ma charge :

- Validation de l'architecture, définie de manière collégiale.
- Veille à l'avancement du développement du projet.
- Participation importante au codage.
- Expertise php et javascript auprès des collègues.
- Débogage général du site, intégration des modules.
- Rédaction du rapport technique.

## **Support matériel, installations**

Le projet LAPI.NET, en terme de réalisation, a les spécificités suivantes :

- petit projet nécessitant malgré tout une étape de modélisation.
- projet de site web dynamique php et MySQL.
- utilisation d'un serveur apache pour la production (hébergé par free.fr)
- réalisation au sein d'un petit groupe de développeurs.
- réalisation dans différents lieux (locaux de l'université, domicile) sur différentes machines.
- très petit budget (environ 0 €).

### Machines :

Je possède 2 ordinateurs personnels :

- un ordinateur hp pavilion p6 (1 an) tournant sur ubuntu 10.04.
- un ordinateur portable acer 1350 (10 ans) tournant sur xubuntu 12.04.

### Logiciels :

Compte-tenu des éléments précédents et de certaines limitations dues à l'ancienneté du matériel ou de certains logiciel (ex : difficulté d'installer des plugin eclipse sous ubuntu 10.04) j'ai opté pour l'installation d'un ensemble facilement reproductible, léger, robuste et entièrement open-source pour mes deux ordinateurs afin de travailler collaborativement de manière satisfaisante.

*Concernant la modélisation :*

- pour les MCD : OpenModelSphere 3.2
- pour les diagrammes UML : Argouml 0.34
- pour certains diagrammes : Calligra flow 2.4.0

*Concernant le serveur local en mode développement :*

- serveur apache 2.2.22 (LAMP)
- php 5 : 5.3.10 (chez free : 4.4.3)
- Mysql:5.5.29 (chez free : 5.077)
- phpMyAdmin

*Concernant le codage :*

- geany : 0.21

*Concernant la gestion de version :*

- rapidSVN : 0.12.0

Tests et débogage :

Pendant toute la phase de développement, le code était vérifié en local sur le serveur apache via Firefox ou Google Chrome en contrôlant le fichier /var/log/apache2/error.log pour les erreurs php et la console d'erreur pour les erreurs javascript. Le suivi de la bases de données a été permis par phpMyAdmin, pour le test des requêtes MySQL.

## **Modèle de développement**

A titre personnel, différents modèles de développement ont été abordés selon les caractéristiques des parties à coder (taille, difficultés de modélisation) :

- modèle en V pour les parties faciles à structurer (ex : profil utilisateur).
- modèle évolutif pour les parties plus difficiles (ex : tchat).
- modèle code-and-fix pour les parties très simples et très petites (ex : gestion des amis).

Le gain de l'expérience a permis également de changer de modèle dans le temps. La différence se fait sentir dans l'architecture du site entre les parties codées en début et en fin de projet. Cela contribue au rendu assez peu homogène de l'architecture. Nous avons malheureusement manqué de temps pour effectué un refactoring général de l'architecture (cf. améliorations souhaitées).

## **Modélisation**

Du point de vue de la modélisation, j'ai réalisé un brouillon du MCD général du site, en mettant de côté la messagerie qui est réalisée par quelqu'un d'autre. Je me suis ensuite attaqué à la modélisation du service de tchat de manière plus fine.

Résumé :

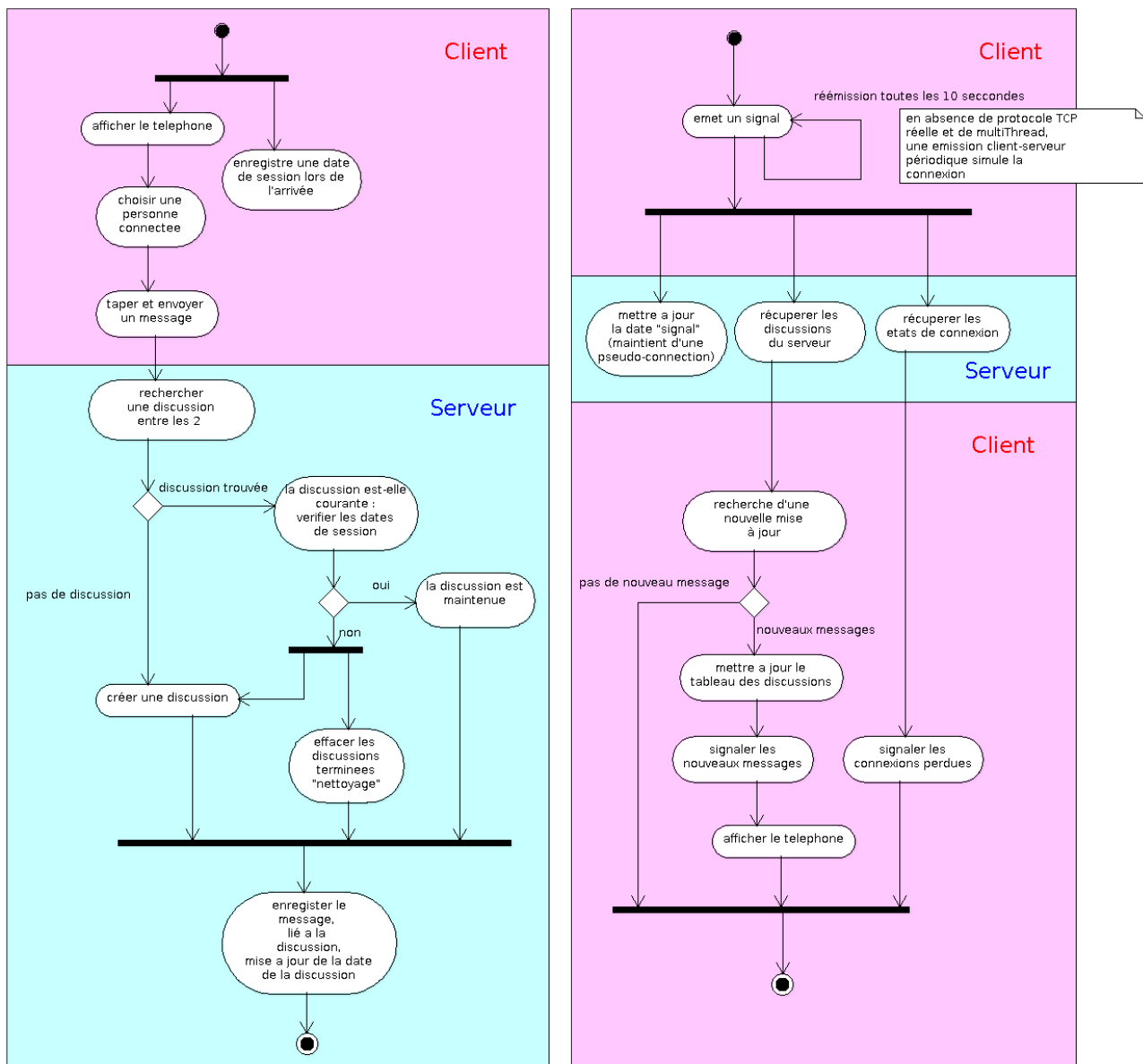
Compte-tenu des limitations imposées par l'hébergement sur les pages perso free.fr (pas de

socket et de multi thread) il était impossible de procéder à la réalisation d'un service de tchat classique en mode connecté, utilisant un patron de conception de type Ecouteur / Observateur. C'est au poste du client qu'il convient de s'enquérir des mises-à-jour en récupérant à intervalles régulier, par une requêtes ajax, les discussions le concernant. Se faisant, il met à jour une date de signal permettant de contrôler pour les machines tierce sa présence (pseudo-connexion) à posteriori sur le site.

Le diagramme d'activité qui en résulte est le suivant (Fig 1) :

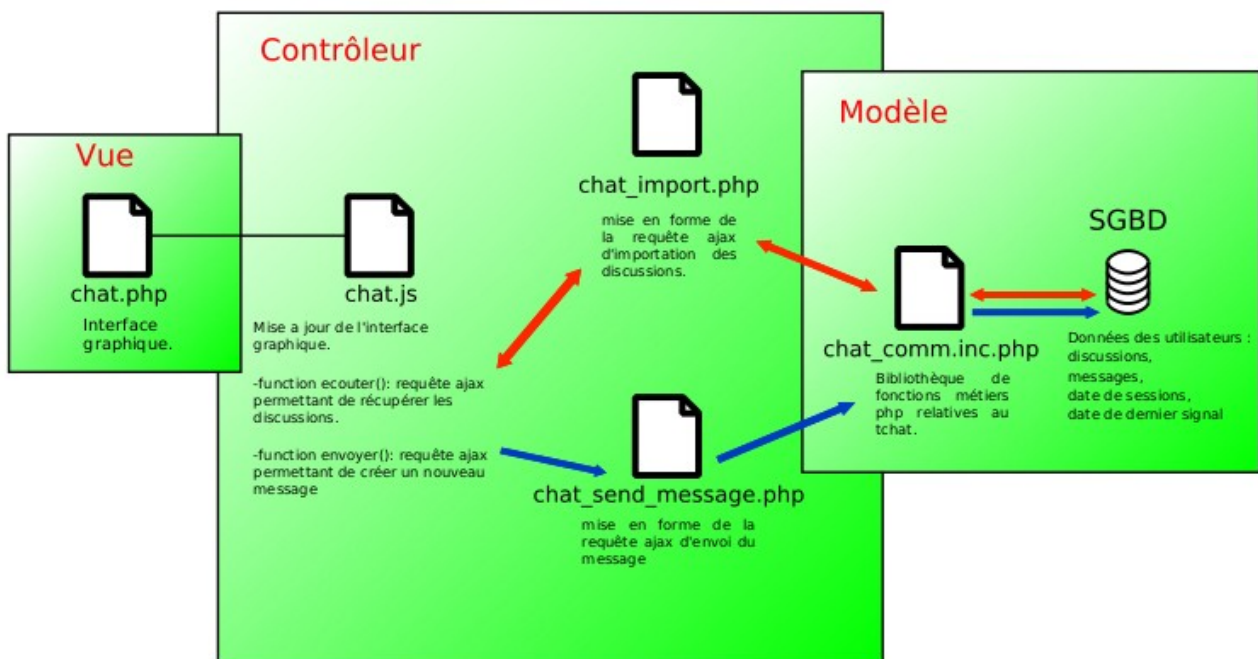
à gauche : l'émission d'un message

à droite : importation des discussions.



**Fig 1 :** diagrammes d'activité pour le tchat

Le déploiement se fait selon l'architecture suivante (patron modèle-vue-contrôleur) Fig 2 :



**Fig 2** : déploiement en modèle-vue-contrôleur pour le tchat

Concernant le site, de manière générale, le choix du déploiement avec inclusion de contenu variable dans la page index et la définition des différents répertoires et composants ont été fait de manière collégiale.

## Réalisations logicielles

Nous avons essayé de partager les tâches de développement au sein de l'équipe en suivant le déploiement logique de l'application et de l'inter-dépendances des différents modules. La seule exception, concerne le tchat qui est une partie de l'application quasiment indépendante des autres. Malgré cela, cette partie a été faite dans les premiers temps d'abord sous forme de prototype pendant la période de modélisation. Du fait de sa difficulté propre (gestion de requêtes ajax) elle a en effet nécessité de tester sa faisabilité rapidement.

L'autre (très grosse) partie a été la réalisation des modules de gestion des profils utilisateurs. Les profils utilisateurs sont en effets l'élément central du site, puisqu'ils sont nécessaires à tous les différents modules : services de messagerie, à la recherche *etc.*

Les différentes parties codées sont :

- Le tchat.
- La création d'un formulaire d'inscription.
- La validation de ce formulaire et la création du compte dans la base.<sup>1</sup>
- La modification du compte.
- La création de pages de profil publiques et privés.
- La gestion d'images apparaissant sur le profil.
- L'inscription de lapins dans les profils.
- La suppression et la modification des lapins.
- L'affichage d'un profil lapin.
- La possibilité de mémoriser ses « amis ».<sup>2</sup>

<sup>1</sup>Le cryptage du mot de passe a été réalisé par un autre membre.

<sup>2</sup>En binôme.

## **Tests**

Les composants logiciel ont été réalisés en séparant au mieux le code en fichiers, puis en fonctions et parfois en classes (pour le tchat). Cela a permis de faire des tests unitaires assez simples en vérifiant les résultats par les outils disponibles comme phpMyAdmin. Les modules (ou composants) ont toujours été testé minutieusement en local dans des conditions les plus proche possibles des conditions d'utilisation normale. Par exemple passer par l'inscription pour obtenir un profil avant de tester d'autres fonctionnalités. Les inscriptions ont été testées avec et sans javascript pour vérifier les validations côté client et côté serveur. Dans la mesure du possible, la mise-à-jour de version via SVN a toujours été conduite lorsque le débogage semblait satisfaisant. Le test du système réel a été plus délicat. En effet, différents bugs observés initialement était du à des limitations propres aux pages perso free.fr.

L'ensemble des composants a été testé. Le site répond aux besoins fonctionnels.

## **Production de documentation**

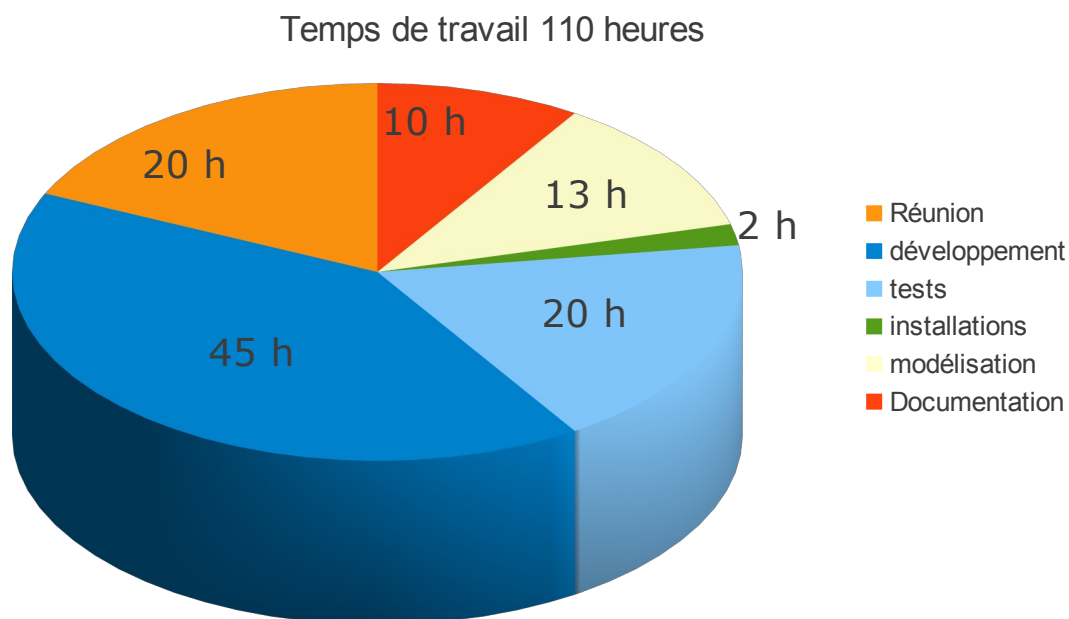
En plus d'une participation à l'architecture du site, j'ai été en charge du rapport technique de développement du site.

## **Temps de travail**

Le temps de travail total réalisé est de : **110 heures** cf Figure 3.

partagé en :

- Réunions et discussions : 20h
- Génie logiciel : installations et modélisation : 15h
  - installation : 2h
  - modélisation : 13h
- Développement : 45h
- Tests : 20H
- Documentation : 10h



**Fig 3 :** Décomposition du temps de travail.

## **Problèmes rencontrés**

Les principales difficultés rencontrées sont inhérentes au travail en équipe. Des difficultés récurrentes ont été rencontrées quant au suivi de l'avancement des différentes parties et de la bonne distribution des tâches. Les problèmes récurrents ont été la non-réalisation des tâches proposées dans l'équipe, le non respect des consignes et des délais, la dispersion, la réalisation de parties attribuées à d'autres personnes par certains membres.

D'un point de vue plus technique, la centralisation des fichiers via SVN ne s'appliquant pas aux tables MySQL utilisées en local, quelques difficultés ont été rencontrées, souvent en raison de mauvaises mise-à-jour.

## **Améliorations souhaitées**

Sont souhaitables :

- le partage des nombreux fichiers « métier » php du dossier include dans des sous-dossiers correspondant à des modules ou la création d'un répertoire pour les scripts métiers appelés et exécutés hors inclusion qui conduisant à une redirection (ex : vérification de formulaire).
- le renommage et l'homogénéisation des fichiers aux noms trop ambigus.
- la factorisation de certaines fonctions similaires.

## ***Conclusion***

La réalisation du site LAPI.NET a été une belle expérience collective qui a été pour la plupart d'entre-nous la première expérience de gestion de projet de développement. Cela a été également l'occasion de rentrer plus en profondeur dans les différents langages abordés en cours.