

Esiea Paris 2020.

Démarches générales utilisées pour le projet.

Projet Chat-Room

Par

**Taleb Amine, Manon Fournier, Anaïs Dubois, Meegan Guillon et
Sacha Benhamou**

I. Introduction.

I.1 Objet :

Ce document décrit notre démarche générale que nous avons utilisé pour réaliser notre projet :

- Choix de la méthode de développement de notre projet.
- Démarches utilisées pour l'élaboration des exigences fonctionnelles et non fonctionnelles.

I.2 Généralités :

Dans le cadre des restrictions dû au corona virus, nous avons utilisé **Zoom** (fig 1) en mode « gratuit » session de 40 minutes afin de pouvoir échanger pour la durée de notre projet. Des sessions régulières une à deux fois par semaine en fonction des disponibilités de chacun.

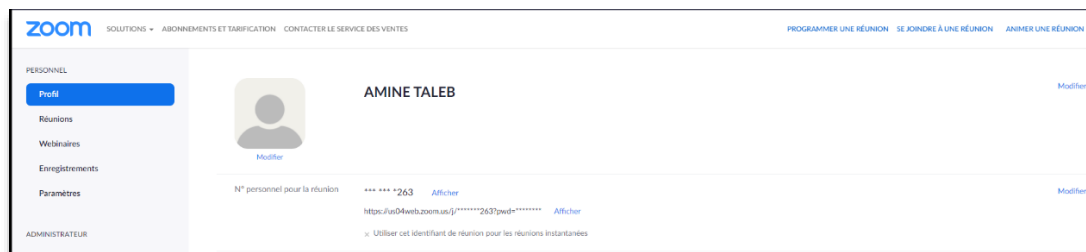


Fig 1

Nous avons par ailleurs, utiliser un groupe Facebook afin de pour échanger entre nous.

I.3 Démarches liées au choix du modèle de développement :

Nous avons opté pour une approche de développement en mode KANBAN (Fig 2), qui se prête bien pour ce type de projet à petite échelle.

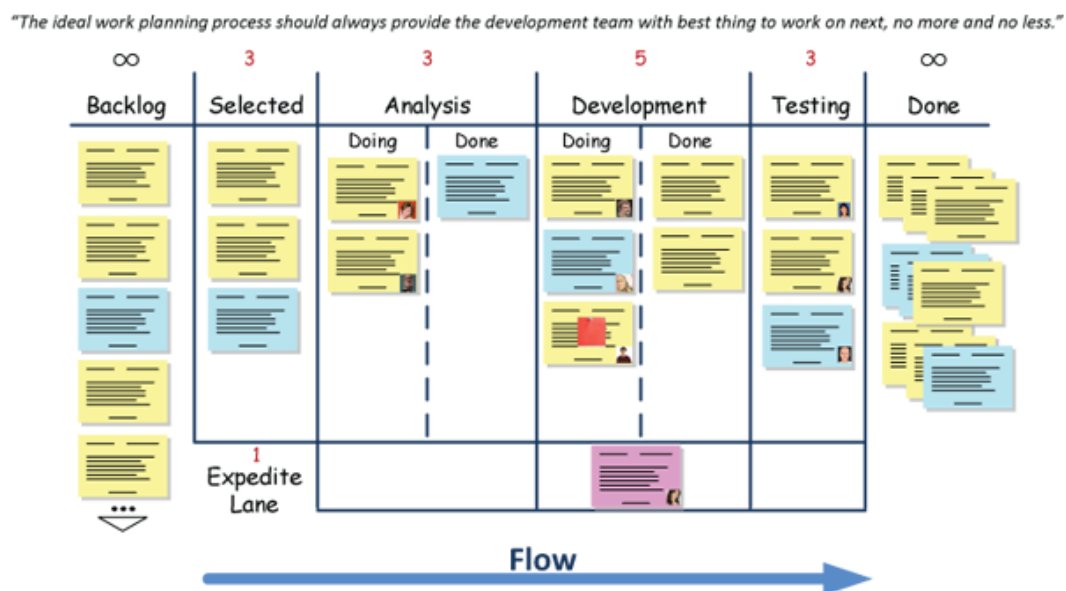


Fig 2

Dans ce projet, il était nécessaire d'avoir une vue globale du projet avec les différents grands blocs tels que (Le serveur et le client)

- Il a été nécessaire de spécifier les besoins fonctionnelles et non fonctionnelles de chaque grand bloc ainsi que leur interaction. (Définition des requirements fonctionnelles).
- Ex : « Le serveur doit-être démarré en premier »
- Définir une spécification fonctionnelle et logicielle, En partant de l'analyse des besoins, nous avons regrouper chaque besoin en catégorie (voir 1.3)
- La conception générale est la représentation macro des divers blocs avec l'emploi d'une architecture de type MVC par exemple.
- La conception détaillée est la phase qui a détaillée plus précisément les packages, classes, méthodes et diagrammes UML.

Les documentations ont été réalisés sous Word, l'ensemble des développements ont été réalisés sous IntelliJ Idea (Fig 3) qui offre une flexibilité et les divers plugins pour la gestion des builds, des commits (GIT) et tests (Maven).

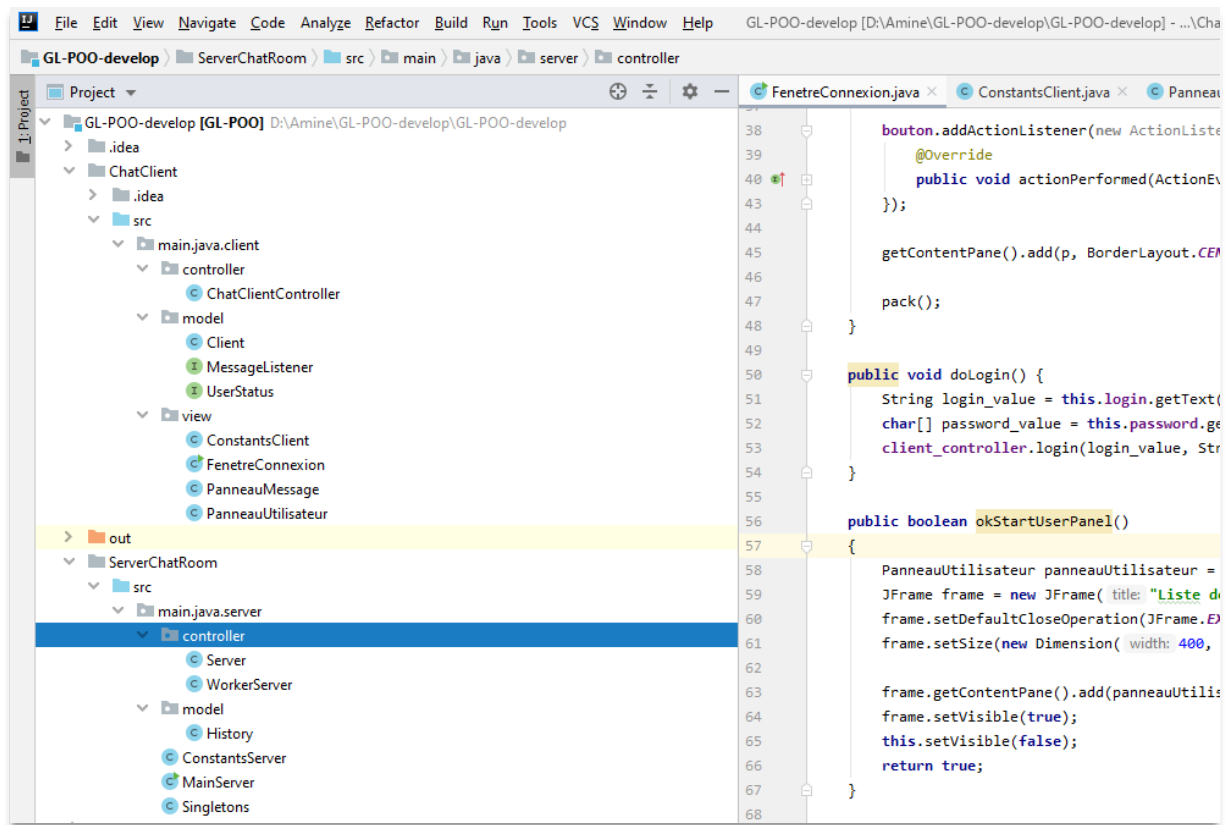


Fig 3.

1.3 Démarches liées à l'élaboration de la liste des exigences :

Initialement ce projet paraissait simple, car d'un point de vue « macro », nous avons tous utilisés des applications de chat, tel que « WhatsApp, Messenger, Zoom, etc... » de façon intensive. Ce qui peut être utile pour établir quelques exigences fonctionnelles et non fonctionnelles.

Cependant, une utilisation même intensive de certaines de ces applications ne reflète pas la difficulté de l'élaboration de ces listes d'exigences : **« C'est l'arbre qui cache la forêt ».**

Nous avons au préalable, chacun de notre côté établi des exigences en vrac, sans se soucier du caractère de celles-ci.

Les premiers échanges étaient de confirmer ou non les exigences de chacun et de recouper celles-ci en fonction de critères de catégories.

- Contraintes pesant sur le système
- Conformité du système à un environnement : normes réglementaires, documentaires, conformité aux licences acquises, ...
- Maintenabilité du système : traçage des erreurs, possibilité des mises à jour, extensibilité / « modifiabilité » du système initial, « supportabilité » en fonction de l'implantation géographique du futur système, testabilité...
- Performance du système : charge utilisateurs / transactions
- Portabilité du système : compatibilité avec diverses plateformes, facilité de remplacement d'autres systèmes en place, facilité d'installation et de désinstallation de l'application ...
- Fiabilité du système : capacité à gérer les erreurs du système, densité des défauts de qualité, capacité à être remis en état rapidement, capacité à résister aux attaques...
- Sécurité du système : traçage des mises à jour des données dans le système, gestion de la confidentialité, gestion de l'intégrité des données, protection des données personnelles ...
- Utilisation du système : facilité

Après ces quelques réunions nous avons élaborer les 2 documents de spécifications fonctionnelles et non-fonctionnelles. Néanmoins, certaines exigences fonctionnelles sont apparues un peu plus tard, ce qui est assez aisé de réadopter lorsqu'on adopte un modèle en KANBAN.