

TD 1- Génie logiciel

Présentation du projet « Studway »

Notre projet consistait à réaliser une application multiplateforme de recherches d'emplois/stages pour les étudiants. Elle centralise les offres de plusieurs sites et les entreprises peuvent aussi poster des offres d'emploi/stage dans cette application afin d'établir un contact direct avec les étudiants.

Difficultés rencontrées

Difficultés organisationnelles

La principale difficulté rencontrée est l'organisation. En effet le projet a été réalisé par un groupe de 6 étudiants et la communication entre nous n'était pas efficace, surtout vers les deux dernières semaines. La cadence de développement s'est intensifiée et le semblant de bonnes pratiques que nous avions a disparu. Cela est dû à notre inexpérience dans ce domaine. Pour résoudre cette difficulté, nous avons utilisé l'outil d'organisation Trello qui permet d'organiser les tâches et les assigner à chacune des personnes travaillant dans ce projet.

Difficultés liées au code

Duplicate defects

Concernant les difficultés liées au code, nous avons remarqué qu'il y avait beaucoup de duplication dans les mécanismes de requêtes et authentification. Cela a eu pour conséquence directe un ralentissement de production car le code devenait de moins en moins compréhensible et maintenable. Pour résoudre ce problème nous avons fait appel au principe du **DRY** (Don't Repeat Yourself) et aux **design pattern** (*Template Method*). Cela a eu aussi pour conséquence indirecte un refactoring complet du système d'authentification et d'envoi de requêtes, puis par la suite fluidification de la production. Cela nous a coûté **3h par personne** : nous étions 3 pour régler ce problème et cela nous a pris 1h chacun.

Performance defects

Concernant les difficultés de performance, nous avons réalisé que les requêtes prenaient beaucoup de temps pour être effectuées et cela est dû à une mauvaise implémentation des méthodes. Cela a eu pour conséquence directe le ralentissement de l'application (page blanche pendant le chargement). Pour résoudre ce problème, nous avons **réduit les appels à la base de données** qu'au strict nécessaire (le ralentissement était dû à ça). Pour les conséquences indirectes, nous avons opté pour une différente manière pour dialoguer avec la base de données. Cela nous a coûté **8h par personne** : nous étions 2 pour régler ce problème et cela nous a pris 4h chacun.

Structural defects

Concernant les difficultés/défauts de structure, nous avons remarqué que la structure de la base de données n'était cohérente, notamment les relations entre les utilisateurs et entreprises pour qu'ils puissent s'envoyer des messages. Cela a eu pour conséquence le refactoring de la base de données ainsi que les relations, ce qui a eu pour conséquence indirecte la modification de code déjà existant qui fait des requêtes SQL avec ces tables. Cela nous a coûté **9h par personne** : nous étions 3 personnes pour régler ce problème et ça nous a pris 3 heures chacun.

Youcef
MEDILEH
22004750

Binôme : Jordan MALCLES

Problème	Evitable ? Comment ?
Organisation	Ce problème aurait pu clairement être évité grâce à l'utilisation d'outils d'organisation dès le début du projet et l'utiliser jusqu'au bout (c'est-à-dire respecter le planning etc...) et avec la mise place de réunions quotidiennes/hebdomadaires pour discuter de l'avancement du projet et réduire le risque de désorganisation.
Duplication de code	Ce problème aurait pu être évité s'il y avait eu une meilleure communication entre les membres de l'équipe et une préparation préalable.
Perte de performance de l'application	Ce problème aurait pu être évité si on avait réfléchi à la structure du code et des méthodes afin d'éviter les appels récurrents vers la base de données.
Mauvaise structure de base de données	Ce problème aurait pu être évité si on avait bien réfléchi à la structure de la base de données avant de commencer le projet.