

# LSINF1212 Pojet : IOU

Omar Kram 2708-16-00

Oliver Perdaens 2382-16-00

Thibault Vereerstraeten 7882-16-00

May 2018

# 1 Introduction

Dans le cadre du cours LSINF1212 d'approfondissement en informatique il nous a été demandé de réaliser un site sur un sujet libre.

Dès lors, nous avons pensé à créer un site possédant une réelle utilité, qui pourrait notamment aider à résoudre certains problèmes du quotidien.

Pour cela, nous avons donc pensé aux petites dettes que nous avons régulièrement entre amis et surtout au fait qu'il est difficile de se rappeler de chacune d'elle, ce qui est embêtant car nous perdons les comptes et au final nous ne savons plus qui doit de l'argent à qui.

C'est en vue de palier à ce petit problème que nous avons pensé "IOU" (I Owe You), une application web permettant de gérer simplement et efficacement les petites dettes que nous contractons envers nos amis.

## 2 Choix d'analyse

Tout d'abord, il nous faut rappeler que l'objectif cible de cette application est bien la gestion de faibles dettes que l'on a envers nos amis. Il n'est nullement question ici de régler des affaires juridiques. De plus, nous partons du postulat de base que les deux parties sont de bonne volonté.

Par ailleurs, nous allons souvent employer deux termes particuliers: "créancier" et "débiteur". Afin d'éviter tout problème de compréhension, nous vous rappelons qu'un débiteur est la personne qui a emprunté une somme d'argent tandis que le créancier est la personne qui a prêté de l'argent. Le débiteur doit de l'argent à son créancier.

Cette précision étant faite nous pouvons passer au coeur du sujet.

### 2.1 Points importants

Lors de la conception de notre application web il nous est apparu que certains points étaient très importants pour le bon fonctionnement du site web:

Premièrement, les informations des utilisateurs, comme les dettes, mots de passe etc.. , ne doivent être accessibles qu'aux utilisateurs concernés. C'est à dire qu'une personne n'étant ni créancier, ni débiteur d'une dette ne doit en aucun pouvoir accéder à cette dette.

Pour ce faire nous avons créé un système d'identifiant et de mot de passe classique. De cette manière, au moment où vous vous connectez à un compte, seules vos dettes sont affichées et en aucun cas celles d'un autre utilisateur.

De ce fait nous pouvons affirmer que ces informations sont bel et bien protégées.

Deuxièmement, un utilisateur ne doit pas pouvoir effectuer un remboursement sur l'application sans le faire en même temps dans la réalité. Autrement dit, la fraude ne doit pas être possible.

Pour ce faire, nous avons conçu un système de validation.

En effet, lorsqu'un débiteur veut effectuer le remboursement d'une de ses dettes, il doit effectuer le paiement à son créancier. Dans un autre temps, il va sur le site et sélectionne la dette qu'il vient de payer et envoie une demande de validation de remboursement, à ce moment son créancier peut valider ou non cette demande, ici le débiteur a effectué le paiement donc le créancier, de bonne volonté, va accepter la demande de validation. Une fois cela fait la dette disparaît.

Les étapes sont donc: la sélection d'une dette, la demande de validation et la validation. Il n'y a que trois étapes et cela est important car la simplicité d'utilisation permet une plus grande

accessibilité au près des utilisateurs.

## **2.2 Interface utilisateur**

Finalement, l'utilisation de l'application doit être la plus simple ,intuitive et agréable possible. Cela peut sembler anodin mais en réalité c'est très important. Car, l'interface est l'unique moyen de contact entre l'utilisateur et l'application. Dès lors, ce contact conditionnera l'utilisation du site et sera un élément clef de la réussite ou non de celle-ci. On peut prendre Apple comme exemple, qui a véritablement révolutionner l'utilisation des téléphones portable notamment grâce à son acharnement à concevoir une interface intuitive.

C'est avec cette idée en tête que nous avons conçu notre interface.

En effet, la page principale affiche directement 3 tableaux reprenants les dettes en cours et les dettes en demande d'acceptation et en demande de validation de remboursement.

Le plus important se situe à gauche reprenant l'entièreté des emprunt et des prêt de l'utilisateur. À la fin de ce tableau la balance totale est affichée.

Cette balance représente l'addition et soustraction des dettes afin de savoir si au final l'utilisateur gagnera de l'argent ou bien s'il en perdra. Cette balance sera aussi afficher en haut à droite de manière à lire cette information directement sans devoir scroller ou chercher dans la page.

Les deux autres tableaux sont plus petits et chacun d'eux affiche soit toutes les demandes de dettes, soit les demandes de validation de remboursement.

Afin de faciliter la lecture de ces informations nous avons décidé d'utiliser un code couleur connu internationalement. Ce code est très simple, le rouge est utilisé pour afficher ce que doit l'utilisateur et le vert ce qu'on doit à l'utilisateur. De ce fait, l'utilisateur a une vue globale de ses comptes au premier regard.

Page principale:

# IOU

Bonjour root root

You Owe Me

I Owe You

Balance générale:

-27.25

€

Ami	Balance
omar kraitm	-29.25 €
Olivier Perdaens	+ 2 €
Balance Totale: -27.25 €	

Demande de dettes reçues, en attente !

Ami	Description	Date	Montant	Action
Olivier Perdaens	Restaurant	a few seconds ago	- 11€	<div>Accepter</div> <div>Refuser</div>

Demande de dettes envoyées, en attente !

Ami	Description	Date	Montant
-----	-------------	------	---------

Détails des dettes:

### Détail des dettes avec omar kraitm

il y a une heure  
Place de concert

25€

il y a une heure  
Sandwich a midi


4.25€

Close

Totale: -29.25 €

Demande de dettes envoyées, en attente !

Ami	Description	Date
omar kraitm	123	an hour



### Connection requise

☐ Se souvenir de moi

[Se connecter](#)

[Créer un compte](#)

© 2017-2018

(a) Connexion au compte

### Nouvelle dette

*J'ai prêté y € à x ! x me doit maintenant y €*

Description :

Montant :

 €

Utilisateur concerné :

Voulez-vous anti-dater la dette ? ☐

[Annuler](#) [Enregistrer](#)

(b) Création de dettes

Gestion d'amis:

I Owe You
Dettes
Amis 1
Olivier Perdaens

☒

#### Mes amis:

Prénom	Nom	Action
Jean	Dubois	
oliper	oliper	

#### Demandes d'ajout:

Prenom	Nom	Action
root	root	<input checked="" type="checkbox"/> <input type="checkbox"/>

### 3 Choix de conception

Pour la conception du site nous avons été relativement libre ce qui nous a permis de faire des choix de conception étant précisément adaptés à notre projet. Néanmoins certains éléments nous ont été imposés:

L'utilisation de

1. MongoDB pour la gestion de la base de données.
2. Java script pour gérer les interactions principales entre le client et le serveur.
3. Node.js qui est environnement bas niveau permettant l'exécution de JavaScript côté serveur.

#### 3.1 HTML, CSS

Évidemment pour ce qui est de la mise en page, nous avons dû utiliser HTML qui est très largement utilisé pour l'organisation interne des pages hypertext.

En plus d'HTML nous avons employé CSS qui permet de gérer tout le côté esthétique des pages du site web.

Plus particulièrement, nous avons utilisé Bootstrap qui est un framework que l'on utilise pour le design de site web.

Un framework est une collection d'outil contenant ,dans ce cas-ci, du code CSS et HTML, et qui facilite le travail des développeurs en gérant une partie du code et de ce fait, rendant l'utilisation du langage plus intuitif.

C'est donc un outil très pratique pour obtenir un résultat fort satisfaisant assez rapidement.

#### 3.2 JS

L'utilisation du Java Script nous a été primordial dans l'implémentation du côté serveur car il nous a permis de récupérer les informations que notre site stock. Autrement dit, toutes les requêtes faites à la base de données ont été effectuées à l'aide de Java Script qui travaille de paire avec notre base de données MongoDB.

En outre, il nous a été tout aussi important dans l'élaboration du côté client. En effet, son utilisation a permis une plus grande fluidité au niveau graphique. Notamment lors de l'affichage de pop up ou autre animation.

De plus, nous l'avons mélangé avec des techniques dites ajax qui ont rendu possible une gestion asynchrone du site. Cette particularité asynchrone permet de charger des données uniquement lorsque l'utilisateur les demande et cela sans recharger la page.

Tout cela rend l'emploi de notre application web plus agréable ce qui contribue à remplir l'un des besoins de l'utilisateur défini précédemment.

#### 3.3 Machine learning

Il nous était aussi imposé d'utiliser du machine learning au sein de notre application, cependant le choix de son utilité nous était réservé.

Ainsi nous avons réalisé un système de suggestion d'amis.

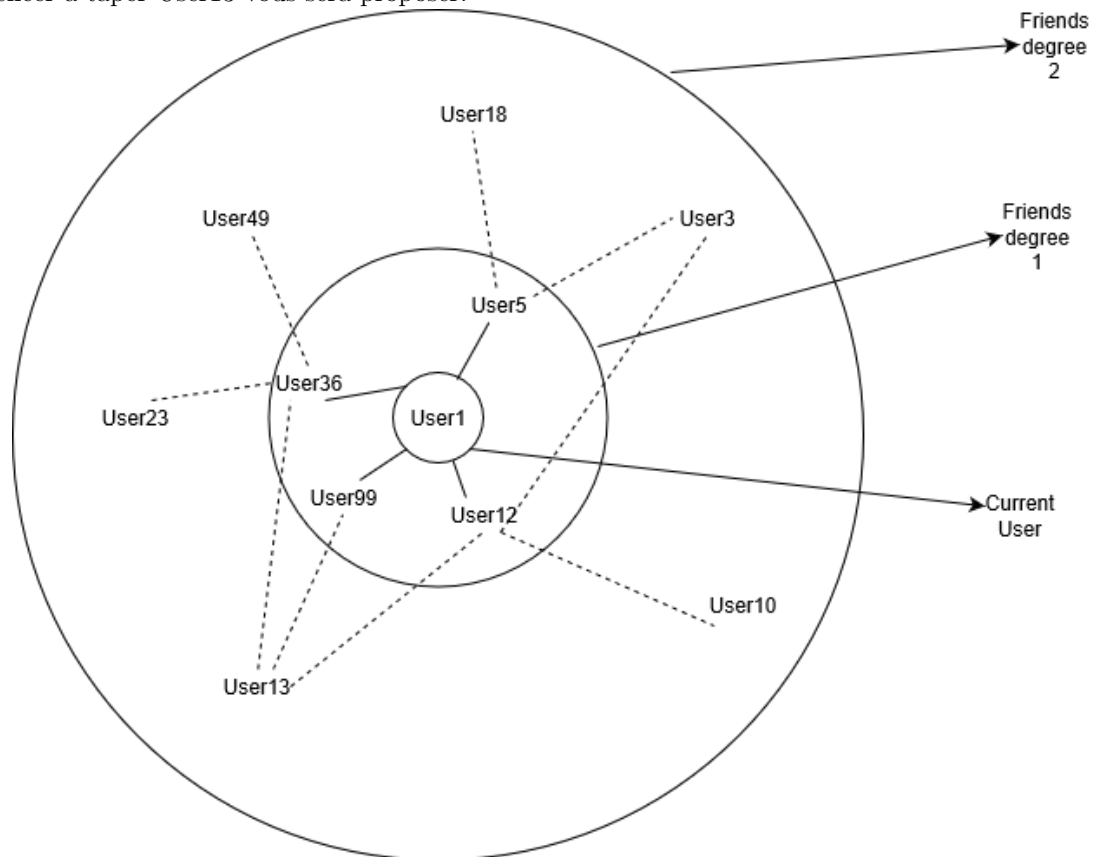
Or, il ne fallait pas que des utilisateurs vous soient proposés de manière totalement hasardeuse. C'est pourquoi nous avons utilisé un algorithme se basant sur les amis d'amis pour vous faire des suggestions.

L'application analyse la liste d'amis de chacun de vos amis. Ensuite elle compare toutes ces listes. Il en résulte que les utilisateurs ayant le plus de liens avec l'ensemble de vos amis vous seront suggérés en premiers.

Prenons un exemple avec le schéma ci-dessous. Le user au centre dans le premier cercle est votre nom de compte. Ensuite dans le deuxième cercle vous pouvez voir le nom de vos amis. Et enfin, dans le troisième cercle vous voyez les amis de vos amis.

Pour plus de facilité nous avons appelé nos amis directs les "friends degree 1" et les amis de vos amis les "friends degree 2".

On voit ici que le User13 est amis avec 3 de vos amis (user36, user99, user12). Donc lorsque vous voudrez rajouter un amis en allant dans la barre de recherche avant même que vous n'ayez commencer à taper User13 vous sera proposer.



## 4 Implémentation du système

Nous pouvons diviser l'implémentation en deux parties distinctes. D'abord, l'aspect technique de la manipulation de données à l'aide de MongoDB et ensuite la structure du code de l'application et de ses différentes entités.

### 4.1 MongoDB

Contrairement au langage SQL, MongoDB n'est pas basé sur une structure relationnelle. En réalité, MongoDB se structure d'après les objets élaborés dans le code, c'est à dire les entités principales de l'application web. MongoDB s'inscrit dans un courant dit "noSQL" afin de proposer une alternative à ce dernier.

Les tables qui forment la base de données sont donc appelées des "collections" composées d'objets. Pour notre site, nous avons établi 3 tables de données: En premier lieu, la table "users" contenant toutes les informations relatives à un utilisateur donné (email, nom, prénom, mot de passe). Puis, la table "friends" qui reprend les identifiants des objets users pour les "lier" ensemble. Et

pour finir, la table "debts" contenant les id's des deux utilisateurs , l'information sur le créateur de la dette, le montant exacte de la somme prêtée , la date à laquelle la transaction a été effectuée , l'état de confirmation de la dette (si oui ou non elle a été confirmée), ainsi qu'une description succincte de la-dite transaction.

## 4.2 Code

### 4.2.1 Client/Serveur

En ce qui concerne le code à proprement parlé, les scripts cotés serveurs sont utilisés pour gérer l'accès à la base de donnée, par contre le code coté client lui, est utilisé pour gérer les animations , la vérification des doubles entrées pour le mot de passe , ainsi que le traitement de données.

Nous avons également séparé le code en différentes classes pour gérer les méthodes concernant les divers entités les "friends" , "users" , "auth" (pour l'authentification), profil , etc... Pour illustrer cela prenons un exemple, dans la classe "friends" nous pouvons donc trouver toutes les fonctions qui concernent les amis c'est à dire la création d'une amitié , sa confirmation et d'autres.

### 4.2.2 Fonctionnement Machine Learning

Le fonctionnement l'algorithme friend predictor comme expliqué au point 3, reprend une liste d'amis et va donc chercher les amis de ces amis . Cette matrice comparera chaque ligne et verifera le nombre de répétition de chaque objet (utilisateur) et attribue donc un score.

Une fois la matrix parcourue nous appliquons un mergesort des users en se basant sur les points recus. La liste sera donc trié du moins probable au plus probable en fonction de l'utilisateur qui utilise cette méthode .

## 4.3 Difficultés rencontrée

Tout d'abord , utiliser une base de données noSQL obligé de réfléchir au données de maniere vraiment différentes de ce que nous avons pu rencontrer auparavant. Ensuite, le fait de coder en async a rendu la tâche encore plus compliquée du fait que certaines fonctions n'ont pas pu appliquer leurs opérations sur les attributs reçu en entrée correctement car les informations attendus de la base de donnée n'arrivaient qu'après avoir terminé les opérations.

Voila pourquoi l'élément "machine learning" appelé "friendPredictor" n'a pas pu être correctement implémenté a temps

## 5 Améliorations futures

Comme vous l'avez vu précédemment, notre application web supporte divers fonctionnalités. Cependant, on peut encore l'améliorer. C'est pourquoi nous proposons ces différentes améliorations possibles.

Première amélioration: Enregistrer toutes les informations afin de les stocker dans un historique, de ce fait vous pourrez voir qui sont les personnes qui vous empreinte le plus, pour quel motif et inversement.

Deuxième amélioration: L'interdiction pour un utilisateur de se dés-inscrire lorsque sa balance totale n'est égale ou supérieure à zéro. Car lorsqu'un utilisateur supprime son compte les dettes qui lui sont liées disparaissent. En rendant impossible cette suppression en vérifiant la balance totale, on améliorerait le défense anti-fraude du site web.

Troisième amélioration: Nous pourrions utiliser les informations contenues dans l'historique et calculer le temps moyen de remboursement pour un utilisateur donnée. Ainsi, ce temps moyen serait une sorte de cote d'utilisateur comme celle que l'on peut voir sur d'autres sites, mais à la



différence de ceux-ci, notre site aurait une cotation totalement objective et plus parlante pour les autres utilisateurs. Cette côte serait visible à tout moment sur le profil de l'utilisateur et serait rappelée au créancier s'appêtant à prêter de l'argent à un ami.

Évidement on peut toujours améliorer une application à l'infini mais ce sont les principaux éléments qui, selon nous, pourraient rendre le site web meilleur qu'il ne l'est actuellement.

## **6 Conclusion**

En conclusion, nous avons implémenté un site internet pleinement fonctionnel répondant à un problème du quotidien. Ce site a été conçu de manière à faciliter son utilisation au travers d'une interface simple et user-friendly afin que son accessibilité soit des plus inclusive. Tout ceci, respectant le cahier des charges imposé au commencement de ce projet.