

Projet Web

Romain LOIRS, Damien LU, Rémi DECOUTY, Florian LECOMTE

12 mai 2020

Table des matières

1	Cahier des charges	2
1.1	Connexion	2
1.2	Les tweets	2
1.3	Les abonnements	2
2	Organisation des tâches	3
2.1	Modélisation (fin Mars- début Avril)	3
2.2	Implémentation (mi-Avril jusqu'à début Mai)	3
2.3	Mise en forme du projet (fin Avril-mai)	4
3	Base de données	4
3.1	Présentation succincte	4
3.2	MCD de la base de données	4
3.3	Schéma relationnel de la base de données	5
4	Analyse technique	6
4.1	Organisation des pages	6
4.2	Accessibilité	6
4.2.1	Page de connexion	7
4.2.2	Page d'inscription	7
4.2.3	Page HOME	7
4.2.3.1	Mon Feed	7
4.2.3.2	Mes publications	7
4.2.3.3	Mes abonnés	8
4.2.3.4	Mes abonnements	8
4.2.3.5	Utilisateurs	8
4.2.3.6	Tendances	8
4.3	Méthode utilisée pour afficher les vues dynamiques	8
4.4	Organisation du code CSS	9
5	Conclusion	10

Avant- Propos

Nous avons décidé de travailler sur le projet [Le twitter de l'ENSIIE]. Une version de notre site est en ligne et est accessible à l'adresse suivante <http://dlu02.freeboxos.fr/src/View/connection.html> sous réserve de disponibilité. Vous êtes bien sur invitée à créer un nouvel utilisateur mais pour voir une page plus fourni, nous vous conseillons de vous connectez avec le login "unex" et le mot de passe "mdp".

1 Cahier des charges

Notre site Web contiendra les fonctionnalités présentées ci-dessous.

1.1 Connexion

Un utilisateur possède un login unique qui permet de l' identifier. Celui-ci est associé à un mot de passe afin que l'utilisateur puisse se connecter et ainsi avoir accès à toutes les autres fonctionnalités proposées par le site web.

1.2 Les tweets

Le but de l'application est d'envoyer des messages nommés "tweet" entre les utilisateurs. Ce sont des textes de longueur maximale fixée à 280 caractères. Ils sont écrits par un utilisateur et envoyés à d'autres utilisateurs. Des fonctionnalités sont associées à ces tweets parmi lesquelles :

retweet : permet à un utilisateur d'ajouter un message dans son feed d'actualité, même si il n'en est pas l'auteur.

like : appréciation d'un tweet qui peut être fait par tout les utilisateurs. au plus une fois par tweet.

hashtag : mot-clé présent dans un tweet qui permet de rassembler un ensemble de tweets qui contiennent le même hashtag. Le mot-clé est un mot ou groupe de mot (sans espace) précédé du symbole #.

message ciblé : un utilisateur peut envoyer un message à une personne (même si elle n'est pas abonnée avec elle) en utilisant @.

1.3 Les abonnements

Un utilisateur peut s'abonner à un autre utilisateur. Il a alors la possibilité de consulter les tweets que ce dernier publie, en plus des siens. Une liste des abonnés et des abonnements d'un utilisateur est accessible par ce dernier.

2 Organisation des tâches

Notre travail s'est déroulé en trois grandes étapes :

- la modélisation du site (maquettes fonctionnelles + MCD)
- l'implémentation
- la mise en forme du projet

2.1 Modélisation (fin Mars- début Avril)

Durant cette phase, le but a été de définir le cahier des charges précédemment exposé de notre site web afin de construire une base de données répondant à nos besoins. C'est également durant cette étape que nous avons défini l'organisation et le design général du site, à l'aide de maquettes fonctionnelles. Enfin, le projet devant respecter le modèle MVC, nous avons établis une liste de fichier nécessaire pour réaliser le site, que ce soit des vues, des modèles ou des contrôleurs.

	Rémi	Florian	Damien	Romain
Définir le cahier des charges fonctionnelles	x	x	x	x
Modéliser la base de données		x		
Réaliser des maquettes du site			x	x
Définir une charte graphique (couleur, image)			x	

2.2 Implémentation (mi-Avril jusqu'à début Mai)

Cette partie a été la plus chronophage du projet. Les fonctionnalités du site web ont été implémenté par tous les membres de l'équipe. Rémi a défini des tâches et chacun s'est vu attribué plusieurs d'entre elles.

	Rémi	Florian	Damien	Romain
Intégrer la BD dans le site		x		
Définir un design pour le site (CSS)			x	
Créer la page de connexion	x			x
Vérifier les informations de la page de connexion PHP		x		
Créer la page d'inscription	x			x
Vérifier que les champs du formulaire sont valides en JavaScript				x
Utiliser les informations du formulaire en PHP		x		
Afficher la page d'accueil après connexion	x			
Envoi d'un tweet à la BD	x	x		
Implémenter les fonctions d'abonnement et le système de like	x			
Implémenter système de gestion de liens cliquables pour accéder au feed d'un utilisateur	x			

2.3 Mise en forme du projet (fin Avril-mai)

En parallèle de l'implémentation, des tâches de mises en formes et de vérification du projet ont été défini vers la fin du projet.

	Rémi	Florian	Damien	Romain
Mise en place du projet dans un serveur			X	
Test des fonctionnalités (travail en continue tout au long du projet)			X	
Rapport final	X			X

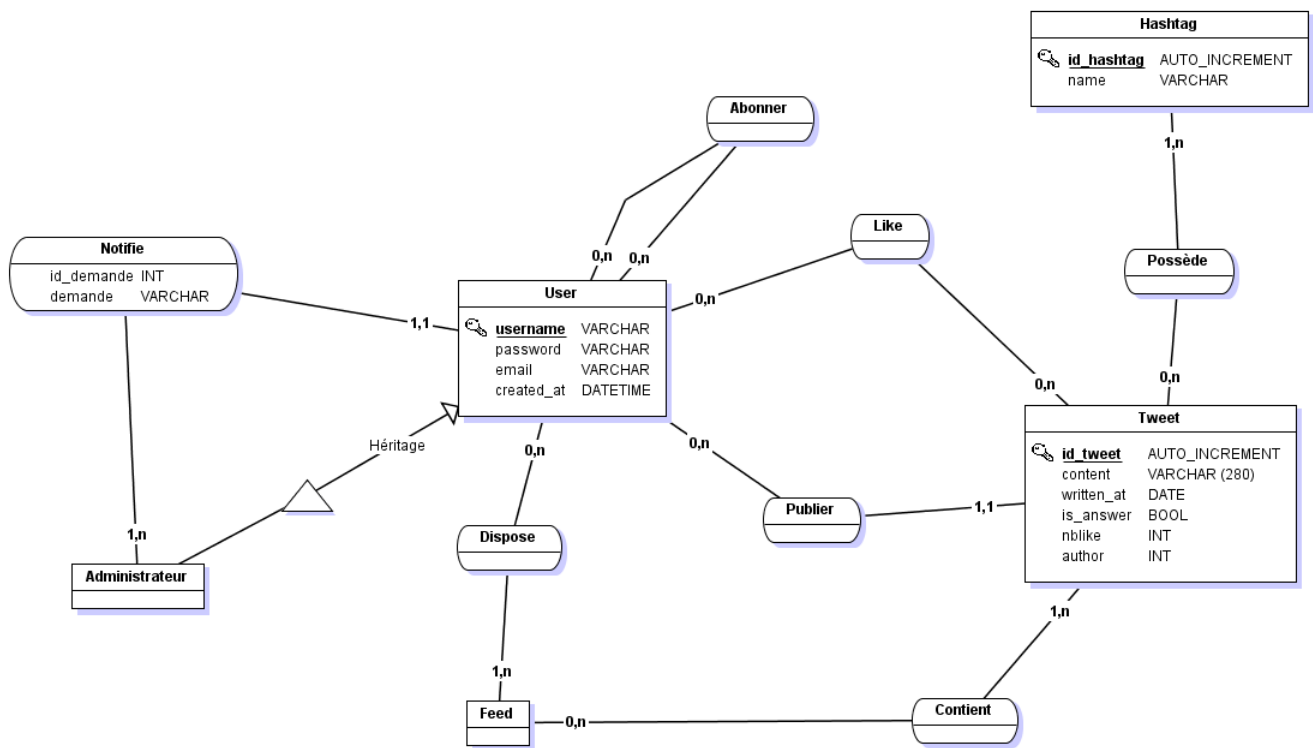
3 Base de données

3.1 Présentation succincte

La base de données de l'application est organisée autour de deux tables principales :

- la table user dans laquelle les informations des utilisateurs sont sauvegardées (nom d'utilisateur, mot de passe, email et date de création)
- la table tweet où sont consignés les tweets (contenue, date d'envoi, expéditeur et nombre de likes)

3.2 MCD de la base de données

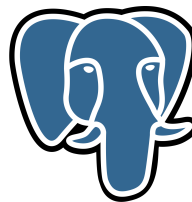


3.3 Schéma relationnel de la base de données

```
user(#username,password,email,created_at)
follow(followed => user,follower => user,#(followed,follower))
administrator(#username => user)
tweet(#id_tweet,content,written_at,is_answer,author,nblike,author=>user)
like(#id_tweet => tweet,#username => user)
hashtag (#id_hashtag,name)
own(id_hahtag => hashtag,id_tweet => tweet,#(id_hashtag,id_tweet))
feed(id_tweet => tweet,username => user,#(id_tweet,#username))
```

Nous avons réaliser la modélisation de la base de données à l'aide du logiciel JMerise. Celui-ci permet de générer un script SQL correspondant à la modélisation de notre MCD, c'est donc un outil très intéressant permettant d'avoir une bonne vision du back-end de notre site.

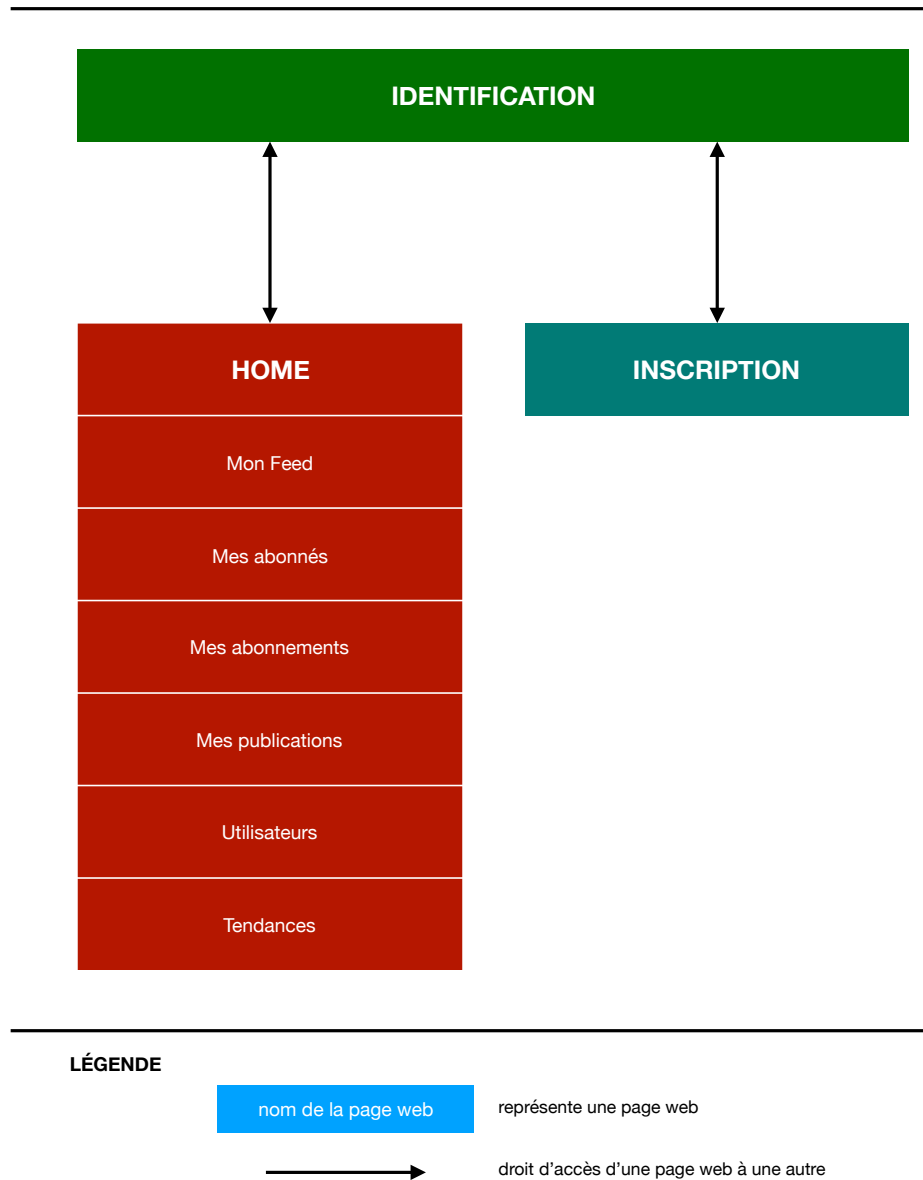
Après cette phase de modélisation, nous avons dû nous organiser afin d'installer la base de données dans un SGBD. Rémi a alors proposé l'utilisation de LAMP, un ensemble de logiciels permettant de construire une base de données ainsi que de faire fonctionner le site sur un serveur web Apache. Nous avons alors effectué une séance d'installation de cet outil, que nous avons complété par l'installation de pgAdmin 4 ainsi que de phpPgAdmin, qui permettent d'administrer une base de données PostGreSQL. La principale difficulté que nous avons rencontré concerne les différences d'installation entre les systèmes d'exploitation. En effet, certains membres étaient sous Linux tandis que d'autres disposaient d'un OS MacOS ou bien Windows. Il a donc fallu s'adapter, mais nous avons tout de même réussi à surmonter cette difficulté.



4 Analyse technique

4.1 Organisation des pages

organisation du projet



4.2 Accessibilité

Depuis toutes les pages du site, l'utilisateur peut retourner à la page de connexion. Dans la page Home, l'utilisateur peut utiliser librement toutes les pages (en rouge sur le schéma précédemment exposé)

4.2.1 Page de connexion

La page d'accueil est la page de connexion. Il s'agit de la première page auquel accède l'utilisateur. Elle est codée dans le fichier `connexion.html`. Lorsqu'un utilisateur se connecte, il renseigne son nom d'utilisateur et son mot de passe. Ces informations sont ensuite vérifiées par le script PHP `checkIfUserExists.php` qui renvoie soit sur la page de profil si elles sont bonnes, soit sur la page de connexion si elles sont mauvaises.

4.2.2 Page d'inscription

Le nouvel utilisateur remplit un formulaire. Les informations fournies par le formulaire sont envoyées vers la page `newuser.php` qui vérifie si l'utilisateur n'existe pas déjà. Elles sont ensuite écrites dans la table `user` de la base de données grâce à la fonction `insert`

4.2.3 Page HOME

Il s'agit de la page centrale du projet. C'est sur cette page que toutes les fonctionnalités relatives à l'envoi de tweets et aux abonnements sont disponibles.

4.2.3.1 Mon Feed

Cette page est personnalisée grâce à la fonction PHP `getCurrentUser` qui permet de connaître l'identité de l'utilisateur. C'est sur cette page que l'utilisateur est redirigé après sa connexion. C'est également sur cette page que l'utilisateur peut "tweeter". Il peut envoyer des tweets et lire tous les tweets qui lui sont adressés (ses propres tweets et ceux auquel il est abonné). Il peut également liker un tweet et retweeter.

Toutes les fonctionnalités relatives aux "tweets" sont codées à l'aide de fonctions PHP accessibles depuis le fichier `TweetRepository.php`. Ces fonctions sont :

- `getMyTweets` : permet d'accéder à la table 'tweet' de la base de données et de récupérer tous les tweets de l'utilisateur
- `sendTweet` : permet d'ajouter à la base de données un tweet.
- `showTweets` : permet d'afficher les tweets de l'utilisateur
- `liketweet` : permet de liker un tweet
- `retweet` : permet de retweeter.

4.2.3.2 Mes publications

Cette section reprend les mêmes fonctionnalités d'envoi de tweet (envoi, retweet, like) que pour "Mon Feed". Cependant, sur cette page, les seuls tweets affichés sont ceux que l'utilisateur a envoyé.

4.2.3.3 Mes abonnés

L'utilisateur a accès sur cette page à la liste de ces abonnés grâce à la fonction PHP `getMyFollowers` et `showFollows`

4.2.3.4 Mes abonnements

L'utilisateur peut sur cette page s'abonner à un autre utilisateur et voir la liste des utilisateurs auxquels il est abonné.

4.2.3.5 Utilisateurs

Ici, l'utilisateur peut voir la liste de tous les utilisateurs du site. Lorsqu'il clique sur le nom d'un utilisateur, il y a deux possibilités :

- si il s'agit d'une personne auquel l'utilisateur est abonné, celui-ci accède aux messages de cette personne
- si l'utilisateur n'est pas abonné, il peut s'abonner en cliquant sur le bouton **suivre nom**, la base de données est modifiée avec la fonction php `follow`
- si l'utilisateur clique sur son propre pseudo, il aura accès à ses propres tweets.

4.2.3.6 Tendances

L'utilisateur peut voir tous les # qui ont été utilisés par les utilisateurs du site. Le fichier `HashtagRepository.php` définit les fonctions qui permettent de :

- récupérer la liste des hashtags
- ajouter un hashtag à la base de données
- afficher la liste des hashtags

4.3 Méthode utilisée pour afficher les vues dynamiques

Le site disposant de nombreuses fonctionnalités, nous nous sommes organisés pour faciliter au maximum l'affichage des informations dynamiques, selon l'action de l'utilisateur. Ainsi, nous avons créé un fichier `loadView.php`, qui récupère l'action saisie par l'utilisateur. En effet, dès lors que l'on clique sur une vue du menu de navigation, alors on passe dans l'URL certaines informations, qui permettront d'effectuer l'action désirée. Par exemple, si un utilisateur désire afficher la liste des tweets de l'utilisateur "césar" alors la requête effectuée sera celle ci-dessous.

```
127.0.0.1/projet-ensiie-2020-master/src/View/home.php?action=viewOtherUsers&pseudo=césar
```

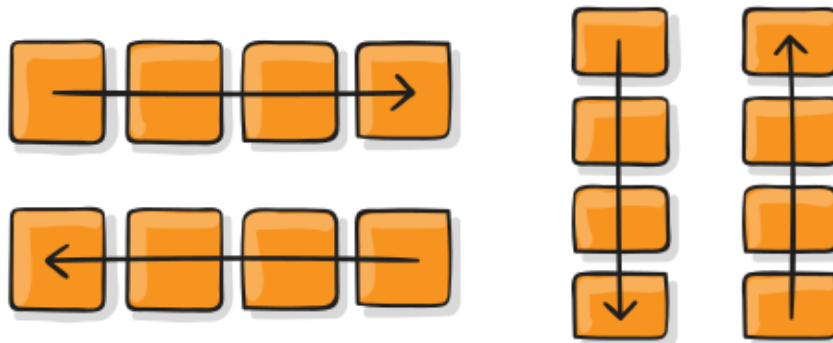
Par la suite, on va récupérer ces informations contenues dans l'URL via la méthode `$GET`. Vous pourrez trouver dans notre code source un fichier `loadView.php`, et bien c'est celui-ci qui va effectuer l'analyse de l'action saisie par l'utilisateur, afin d'afficher la vue correspondante. Ce fichier sert en quelque sorte de "routeur" des informations, et appellera les différentes fonctions d'affichage contenues dans le fichier `viewManager.php`. C'est ainsi un moyen d'organiser

plus méthodiquement l'organisation du code, ce qui permet également d'ajouter facilement de nouvelles fonctionnalités.

4.4 Organisation du code CSS

Concernant l'aspect visuel du site, nous avons essayé d'utiliser des couleurs neutres, pas trop voyantes, afin de ne pas altérer l'expérience de l'utilisateur. C'est pourquoi nous avons principalement eu recours à des nuances de bleu clair, ainsi qu'à des dégradés de cette même couleur.

Pour ce qui est du positionnement des éléments, nous avons très majoritairement utilisé des flexbox, qui est une méthode permettant de distribuer l'espace entre des objets d'une interface, ainsi que de les aligner. Grâce à cet outil, on évite ainsi d'utiliser des "position : absolute" ou encore le positionnement flottant, qui peuvent rapidement devenir difficile à manipuler. Cela permet d'ailleurs d'ajouter des éléments très facilement, sans avoir à faire sans cesse des ajustements dans le CSS. Voici un exemple ci-dessous du principe des flexbox, l'image de gauche représentant une flexbox ayant une direction horizontale, tandis que celle de droite positionne les éléments verticalement.



5 Conclusion

Concernant les fonctionnalités que nous n'avons malheureusement pas su implémenter, il y a tout d'abord celle de super-utilisateur, permettant à celui-ci de supprimer certains contenus signalés par d'autres utilisateurs. La fonctionnalité de réponse à un tweet n'a pas pu être mise en oeuvre, bien que nous l'avions modélisée dans la table `tweet` grâce à l'attribut `isanswer`. Nous avons tout de même cherché à coder ces fonctions, mais elles se sont avérées non fonctionnelles et provoquaient des bugs, c'est pourquoi elles n'apparaissent plus dans le code source du projet.

Pour terminer ce rapport, nous aimerions énoncer quelques améliorations qui pourraient être réalisées à notre site. Tout d'abord, on pourrait offrir à l'utilisateur la possibilité d'envoyer des émojis, mais aussi des hypertextes ou bien des images. Cette fonctionnalité demande néanmoins beaucoup de travail supplémentaire, c'est pour cela que nous n'avons pas eu le temps de l'implémenter. Enfin, on pourrait aussi ajouter des effets visuels, codés en JavaScript, lorsque l'utilisateur s'abonne par exemple. Cela améliorerait ainsi grandement l'expérience utilisateur, en lui apportant des informations supplémentaires.

Ce projet nous aura à tous permis de réaliser notre premier site Web, à l'aide d'outils dont on ignorait l'existence il y a encore quelques mois. Nous avons ainsi pu réaliser l'installation d'un environnement LAMP, dans lequel nous avons développé notre propre projet. Ce fut donc une expérience très enrichissante pour notre parcours d'élève ingénieur, qui nous a également appris à collaborer afin de mener à bien des projets complexes.