

ENSIIE  
PROJET WEB  
RAPPORT

---

# Usagi Life

---

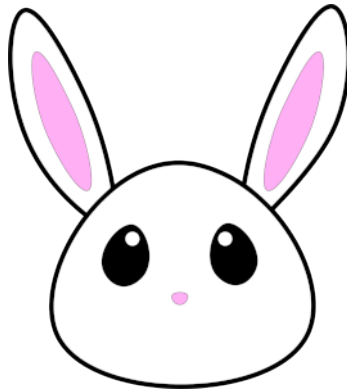
*Auteurs:*

Tania GONDY  
Théo MORNICO  
Kélian DEVILLIER  
Kelly MINOTTI

*Encadrants:*

Team MATTERS

14/05/2019



# Contents

<b>1</b>	<b>Problématique</b>	<b>2</b>
<b>2</b>	<b>Déroulement</b>	<b>2</b>
2.1	Approche . . . . .	2
2.2	Problèmes rencontrés et solutions apportées . . . . .	3
2.3	Respect du cahier des charges . . . . .	3
2.3.1	Authentification . . . . .	3
2.3.2	Le compte administrateur . . . . .	4
2.3.3	Profil utilisateur éditable . . . . .	4
2.3.4	Base de donnée relationnelle, au moins trois tables . . . . .	4
2.3.5	Au moins une jointure dans une requête . . . . .	5
2.3.6	INSERT, DELETE, UPDATE, SELECT . . . . .	5
2.3.7	CRUD . . . . .	5
<b>3</b>	<b>Répartition du travail</b>	<b>5</b>
3.1	Kelly Minotti . . . . .	5
3.2	Kélian Devillier . . . . .	5
3.3	Théo Mornico . . . . .	5
3.4	Tania Gondy . . . . .	5
<b>4</b>	<b>Conclusion - Bilan et limites de Usagi-Life</b>	<b>6</b>

# 1 Problématique

Notre projet vise à combler un manque crucial dans notre société : celui d'un réseau social pour les lapins.

En effet, comment se fait-il qu'au 21<sup>e</sup> siècle, à l'ère du tout-numérique, votre animal de compagnie favori ne puisse pas communiquer avec ses amis ? Lorsque cette aberration nous est apparue, nous avons décidé de prendre les choses en main et de lancer *Usagi Life*<sup>TM</sup>, notre réseau social spécialement destiné à nos amis bondissants.

Sur ce site, votre compagnon pourra, après une inscription rapide, créer son profil, poster ses plus beaux accomplissements, voir ceux de ces amis, les commenter et les aimer, ou non. Il pourra ainsi communiquer avec ses partenaires de bonds et leur apporter tout son soutien.

## 2 Déroulement

### 2.1 Approche

Pour répondre à la problématique soulevée, nous avons décidé de partir sur un site inspiré de Facebook<sup>TM</sup>. Pour voir comment les réseaux sociaux étaient codés au niveau du front end, nous sommes allés regarder le code source des pages de sites comme Twitter<sup>TM</sup> (car leur code source était le plus lisible de tous ceux étudiés).

Nous avons essayé d'appliquer une approche méthodique dès le début, c'est-à-dire en séparant les pages gérant les vues, celles travaillant sur la base de données et celles qui gèrent les informations rentrées par les utilisateurs. Nous avons également dès le début choisi le leader du projet et défini et réparti les premières tâches à réaliser entre tous les membres du groupe.

Petit à petit des tâches se sont naturellement rajoutées pour combler des soucis techniques ou d'esthétisme.

## 2.2 Problèmes rencontrés et solutions apportées

Le principal problème que nous avons dû rencontrer reste Docker, qui nous aura demandé tellement de temps et de configurations pour simplement commencer à travailler (en vain) que nous avons préféré commencer le projet sur Wamp/Windows avec MySQL. Bien que cela ne respecte pas les consignes par rapport aux outils de travail demandés, étant donné qu'aucun des ordinateurs de l'équipe (même ceux étant sous Linux) ne pouvaient faire marcher Docker, nous avons préféré au moins nous donner les moyens de faire ce projet qui nous tiens à coeur.

La connexion à la base de donnée aura aussi été difficile dans les premiers temps, que ce soit sur Docker ou sur Wamp, mais une solution a rapidement été trouvée grâce notamment à Google et aux aînés de l'école pour s'y connecter efficacement.

Un autre point quelque peu complexe a été de garder les sessions actives entre les différentes pages. Ce problème a été vite réglé de la même manière.

Quant au CSS, il a parfois été compliqué de gérer la disposition des éléments de la page. Mais à force de tests et de recherches Google là encore, on a finit par obtenir un affichage quasiment entièrement dynamique et esthétiquement correct.

## 2.3 Respect du cahier des charges

### 2.3.1 Authentification

L'authentification sur le site est fonctionnelle : soit on est déjà utilisateur et il suffit de se connecter sur le site, soit on ne l'est pas et on doit passer par l'étape inscription au site. Si les informations données ne sont pas conformes à ce qu'on attend de l'utilisateur, celui-ci est informé de ce qu'il a mal fait et est redirigé vers l'accueil où il pourra retenter sa connexion.

Toute la mécanique de traitement des données entrées par l'utilisateur s'il essaye de se connecter est dans le fichier `okforperso.php`, et le fichier `traitementInscription.php` vérifie que l'inscription d'un nouvel utilisateur sur le site est possible.

### 2.3.2 Le compte administrateur

Les administrateurs du sites se distinguent des autres par un simple booléen, `$is_admin`, qui vaut 0 si l'utilisateur n'est pas administrateur et 1 sinon.

Les administrateurs ont le droit et le devoir de gérer la publication des post sur le site Usagi-Life. en effet, les utilisateurs lambda vont vouloir poster de nombreuses informations à leur sujet pour les faire connaître à tout le monde, mais leurs messages ne pourront être publiés que s'ils sont d'abord vus et validés par l'un des administrateurs du site, afin d'éviter tout propos blessant et/ou inapproprié sur le fil d'actualité du site.

Ce travail se fait sur la page 'Gérer' du site qui est visible et accessible uniquement par les administrateurs de Usagi-Life.

### 2.3.3 Profil utilisateur éditable

Les utilisateurs ont la possibilité de visionner leur profil (dans l'onglet 'Profil' de la barre de menu) ainsi que de le modifier. Seuls les champs 'Adresse Mail' et 'Description personnelle' sont éligibles à une modification cependant, car nous avons estimé que la clientèle visée ne s'intéressait pas au nom qu'elle possède.

### 2.3.4 Base de donnée relationnelle, au moins trois tables

Nous avons une base de donnée relationnelle comportant trois tables :

- **user** : la table contenant toutes les informations concernant les utilisateurs, utilisée pour maintenir sa session, pour vérifier son identité, modifier ses données personnelles lorsqu'il le souhaite...
- **post** : la table contenant tous les post en attente ou validés du site Usagi-Life. Les messages non validés sont supprimés.
- **like** : la table contenant toutes les informations relatives aux like inter-publications. Elle permettra d'ordonner les messages à afficher sur la page 'Perso'.

### **2.3.5 Au moins une jointure dans une requête**

Quand l'utilisateur va consulter son profil il peut y voir toutes les publications dont il est l'auteur. Cela se fait avec une jointure entre les tables `user` et `post`.

### **2.3.6 INSERT, DELETE, UPDATE, SELECT**

Ces quatre commandes sont bien requises au moins une fois sur tout le code de Usagi-Life.

### **2.3.7 CRUD**

Le CRUD (Create Read Update Delete) est validé : en effet on utilise les commandes `INSERT INTO`, `SELECT`, `UPDATE` et `DELETE` au moins une fois dans tout le code de Usagi-Life.

## **3 Répartition du travail**

### **3.1 Kelly Minotti**

Kelly a rédigé en PHP/HTML la conception de la page d'accueil, l'inscription, le changement de mot de passe et de la page de profil personnelle de l'utilisateur.

### **3.2 Kélian Devillier**

Kélian a rédigé la base de donnée et s'est occupé des requêtes en SQL.

### **3.3 Théo Mornico**

Théo a rédigé toutes les pages PHP/HTML qui concernent le traitement des données entrées par l'utilisateur.

### **3.4 Tania Gondy**

Tania a conçu et rédigé en CSS tout le design du site, ainsi que son web-design et son approche UI/UX. C'était aussi le leader du groupe, c'est-à-dire qu'elle fixait ce qu'il y avait à faire et qu'elle le communiquait aux autres membres.

## 4 Conclusion - Bilan et limites de Usagi-Life

Au stade où il en est, Usagi-Life ne propose pas encore de système d'amis/followers entre ses membres, ni de possibilité de liker ou non une publication, ou encore de chatter avec ses amis. Cependant les lapins connectés sur le site peuvent déjà communiquer leurs sentiments et pensées en postant sur le mur principal du site, permettant ainsi à toute la communauté bondissante d'Usagi-Life de voir leurs messages et d'y réagir ou non.