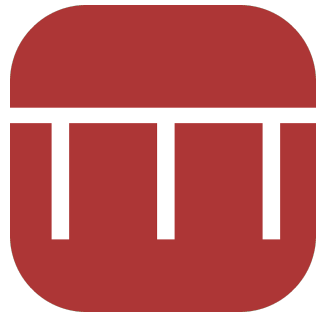


# Rapport de Projet Web



## “Trouve Ton Truc”

SELLAMBIN ANDIAPIN Hugo

GOSSET Matthieu

GONZALEZ Pablo

MAKAROFF Nicolas

# Sommaire

Sommaire	1
I - Introduction	2
II - Approche et mise en place	2
III - Répartition des rôles au sein de l'équipe	5
IV - Les limites du projet	6

# I - Introduction

Parmi les principaux problèmes que l'on rencontre lorsque l'on est étudiant sont : une rentrée d'argent faible; difficulté à se déplacer en dehors de son campus si on a pas de voiture; on peut venir de n'importe où en France et/ou du monde entier, et on ne connaît pas forcément le nouvel environnement dans lequel on se trouve et si les gens sont dignes de confiance sur les sites de revente de biens et de services. C'est pour cela que nous avons eu l'idée de créer "Trouve Ton Truc" (TTT). Nous voulions rassurer l'étudiant et lui permettre d'avoir une alternative à "Le bon coin". En effet, notre site est intra-campus (intra-université/école). Il permet de mettre en relation les étudiants d'un même campus pour qu'ils puissent revendre, prêter, et proposer des services entre-eux. Cela crée donc une interaction marchande, à des prix raisonnables pour des étudiants, entre des personnes d'une même localité, habitant proche les uns des autres, et surtout que l'on connaît ! Ainsi, "Trouve ton truc" favorisera l'échange entre ces jeunes adultes.

## II - Approche et mise en place

Nous avons d'abord dû trouver l'idée sur laquelle nous étions tous d'accord. Après plusieurs idées proposées, nous avons vite retenu l'idée d'un site pour étudiant sur lequel on pourrait poster des annonces et partager des bons plans intéressants pour les étudiants.

Le projet s'est d'abord appelé FindYourThing mais durant le processus de développement le nom est devenu Trouve Ton Truc TTT qui sonnait beaucoup mieux et qui résonnait comme nom d'une start-up florissante française.

Tout d'abord, nous avons réfléchi aux différentes pages qui nous seront utiles et la mise en page qu'elles devraient avoir mais aussi aux tables et aux jointures.

L'idée de base incluait:

- Une page d'accueil avec affichage des 5 derniers produits postés
- Une page par catégorie qui affiche tous les produits disponibles dans les catégories
- Possibilité de se créer un profil
- Possibilité de poster une annonce
- Une page de recherche

- Une page par produit

On a alors défini plusieurs tables :

- Une table « utilisateur » : id, nom, prénom, date de naissance, ville, mot de passe, id\_photo
- Une table « produit » : id\_produit, titre, description, prix, photo
- Une table « catégorie » : id\_cat, nom
- Une table « assoc\_prd\_cat » : id\_cat, id\_prod : **cette table permet notre jointure n x n (un produit peut avoir plusieurs catégories et une catégorie peut avoir plusieurs produits)**
- Une table photo\_profil qui contient les photos de profils
- Une table photo\_produit qui contient les photos des produits

On a d'abord travaillé sur le html de toutes les pages des formulaires : s'inscrire, ajouter un produit et recherche. Comme ses pages étaient formées nous avons pu directement attaquer la partie SQL qui était derrière. À l'instar de la page User et UserRepository, nous avons créé des pages similaires pour Catégories et Produits.

Pour s'inscrire, plusieurs vérifications étaient nécessaires et sont effectives : pseudo déjà existant, adresse mail valide, adresse mail déjà existante, mot de passe de minimum 8 caractères, mot de passe de vérification identique, avoir plus de 18 ans et avoir un âge raisonnable (moins de 130 ans)

Pour déposer une annonce, il y a aussi des vérifications comme la taille de la description qui doit avoir au minimum 30 caractères.

Les deux dernières pages permettent de récupérer avec la méthode post les données, de faire les vérifications et exécuter les requêtes pour les ajouter à la base de données.

Au contraire, la page search fonctionne avec la method get car, après recherche, il semble que c'est l'usage commun. De manière générale, la méthode get a été utilisée sur le site à chaque fois qu'il y avait besoin de récupérer des données mais pas d'ajout ou d'update dans les tables de données.

Il s'est ensuite posé la question de la sécurité. En effet, nous voulions essayer de minimiser au maximum le risque d'intrusion de code malveillant. Nous ne faisons confiance à aucune donnée rentrée par un utilisateur d'où l'utilisation systématique de la fonction htmlspecialchars et des fonctions de vérifications de fichiers pour les photos.

Pour les photos justement, nous avons vite vu qu'il n'était pas judicieux de séparer photo de profil et photo de produit. En effet, nous avons décidé de créer une seule table photo. Elle contient un id et l'extension de la photo. Quand une photo est ajoutée, elle est renommée en fonction de son id et stockée dans un dossier uploads. Par exemple, une photo moi.jpg sera renommée 5.jpg si il y a déjà 4 photos dans la base de données et on ajoutera dans la base de données la ligne : id\_photo : 5 et extension jpg. Pour récupérer une photo, on pourra alors recréer le chemin de la photo en faisant uploads/id\_photo.extension.

Nous avons aussi une page catégorie qui recense les annonces d'une catégorie en particulier. Mais également, des pages produits avec des photos du produit avec un affichage en JavaScript, une description du produit, un résumé des informations du vendeur et les autres annonces du vendeur.

Nous avons aussi mis en place une page profil qui permet de consulter le profil d'une personne donc d'avoir accès à son nom, prénom, ville, adresse mail et les annonces qu'elle a posté. De plus, cette page profil est différente s'il s'agit de celle de la session active. Il y a moyen de modifier ses informations : mot de passe, ville, mail ou photo de profil grâce à du JavaScript ou encore de supprimer ses annonces. À noter qu'il est impossible d'accéder à un profil sans être connecté, seul un aperçu du profil est disponible sur les annonces de l'utilisateur en question.

Pour la partie utilisateur, il ne reste plus que la page Contact qui permet de contacter l'équipe du site. Si l'on est connecté, il sera fourni aux administrateurs l'adresse email de son compte sinon l'utilisateur doit rentrer la sienne si il veut être contacté en retour. Nous avons alors créé une table message qui contient le titre, le contenu et la provenance d'un message.

Arrivé à ce stade, notre site fonctionnait correctement mais n'importe qui pouvait poster n'importe quoi. Nous avons alors défini le rôle d'administrateur qui a plusieurs rôles. Il doit valider les annonces avant qu'elles soient publiées, valider les profils et traiter les messages reçus. Un petit récapitulatif des choses à faire est disponible sur la page d'accueil pour les administrateurs. Nous avons dû alors modifier la base de données pour y mettre une colonne valide qui permet alors de connaître le statut de chaque type : pour les messages savoir s'ils sont à traiter, traités ou supprimés. Idem pour les produits et les profils. De plus, nous avons implémenté une interface utile pour les administrateurs afin d'avoir une vue globale sur les activités du site.

Pour conclure, notre site ne soulève aucune erreur lors des tests W3C.

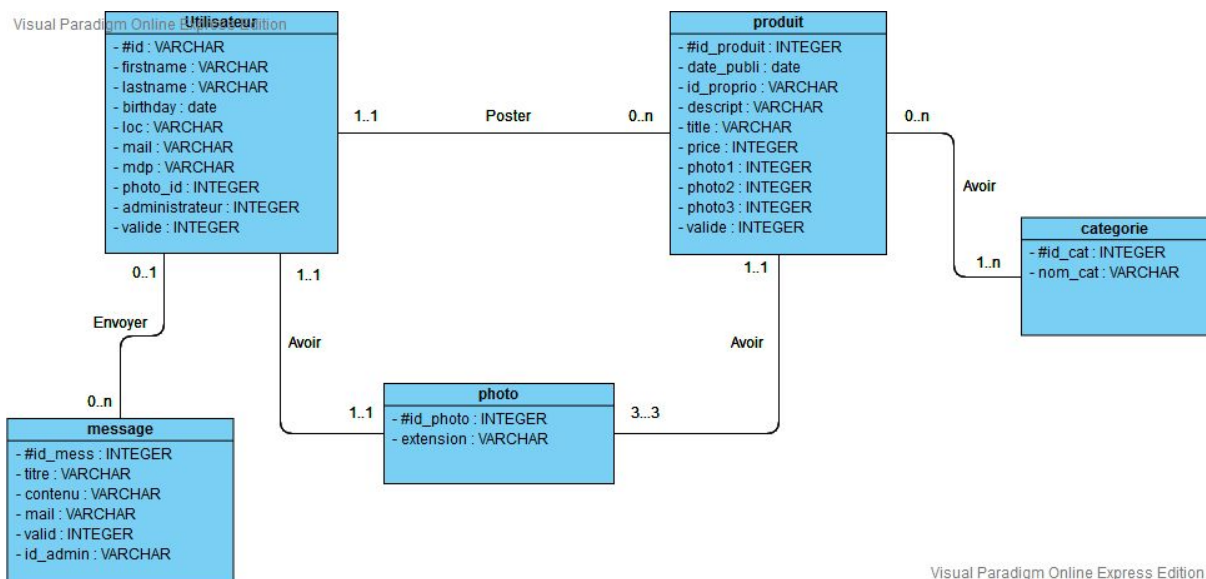


Diagramme UML de notre base de données.

## III - Répartition des rôles au sein de l'équipe

La répartition des rôles s'est faite plutôt naturellement. Comme dans une réelle entreprise ou start-up, nous avons décidé de séparer l'élaboration du back-end (PostgreSQL et PHP) et du front-end (HTML et CSS). Dans notre équipe, Matthieu Gosset s'est occupé de la création de la base de données et des classes PHP permettant d'exploiter les informations de cette dernière. Hugo Sellambin Andiapin s'est consacré à l'HTML/CSS et donc à la confection de la charte graphique et à la mise en page des informations sur le site. Cependant, cela n'exclut pas qu'à certains moments Matthieu fasse du front, et inversement, qu'Hugo fasse du end. De plus, Pablo Gonzalez et Nicolas Makaroff étaient plus mobiles, selon les besoins, il travaillaient soit en front soit en back afin d'aider et d'accélérer certaines tâches. Cette répartition nous permettait d'avancer assez rapidement tout en se complétant. En effet, à aucun moment nous avons été bloqué; lorsque l'un réalisait une fonction PHP, l'autre l'inclut directement dans le script HTML. C'est donc, de cette manière que nous avons réussi à construire notre site : "Trouve Ton Truc".

## IV - Les limites du projet

Ce projet présente néanmoins quelques limites. En effet, dans l'optique d'une utilisation à grande échelle, il serait plus judicieux de crypter les mots de passe et de ne conserver qu'un hash dans la base de données (fonctions `password_hash` et `password_verify` en PHP). Ceci garantirait plus de sécurité pour les différents comptes.

Lors d'une recherche l'application web charge tous les résultats correspondants. Un chargement limité du nombre de résultats et la possibilité de se déplacer entre différentes pages permettrait de réduire les temps de chargement des pages.

Enfin, en ce qui concerne l'expérience utilisateur, la navigation de l'utilisateur sur notre site web serait facilitée s'il y avait la possibilité de faire des recherches ne prenant pas en compte la casse.