



IUT INFORMATIQUE DE BELFORT-MONTBÉLIARD

PROJET DE BASE DE DONNÉES

Site internet administratif de l'association Vélocampus

Auteur :

Morgane CABROL

Pierre LIMBALLE

Geoffrey GLANGINE

Superviseur :

Alexandru DOBRILA



28 mai 2014

Table des matières

1	Le cahier des charges	3
1.1	La partie administration	3
1.2	La partie client	3
2	Base de données	5
2.1	Le modèle conceptuel de données	5
2.2	La réalisation du MCD	5
3	Le fonctionnement du site internet	6
3.1	La Structure	6
3.2	Les fonctions importantes	6
3.2.1	La partie location	6
3.2.2	L'affichage des demandes	7
3.2.3	L'ajout d'image	8
3.2.4	Le Postit	9
3.3	Inscription sur le site client	9
3.4	Gestion du site client depuis la partie administration	10
4	Les problèmes rencontrés	11
4.1	Adblock	11
4.2	Les dates	11
4.3	La sécurité	11
4.4	Répartition du travail	11
5	Les améliorations possibles	12

Introduction

Vélocampus est une association proposant une location de vélos aux étudiants à très petit tarifs. De plus, elle s'occupe de la réparation et de l'entretien de vélos loués à l'association ou non.

Pour ce projet, nous avons voulu réaliser un site web administratif permettant à Vélocampus de gérer son administration plus facilement. Pour cela nous vous présenterons d'abord le cahier des charges que nous avons conçu avec les membres de l'association, puis nous expliquerons la base de données que nous avons décidés de mettre en place, ainsi que les fonctionnalités que nous avons implémenté sur le site. Enfin, nous aborderons les problèmes que nous avons rencontrés puis, les améliorations qui pourraient être apportées au site par la suite.

Chapitre 1

Le cahier des charges

Après une réunion avec les membres de l'association, nous en avons conclu que le site contiendrait une partie client, rendant ainsi l'association présente sur le web et une partie administration, permettant toute la gestion de l'association. Nous avons finalement construit le cahier des charges suivant.

1.1 La partie administration

Elle sera accessible à tous les membres de l'association.

- Une mailing-list des adhérents (*Permettant à l'administrateur d'importer la liste sur une boîte mail afin de pouvoir envoyer des messages groupés*)
- Un suivi des vélos (*Afin de savoir quel vélo est en location, quel vélo a subi quelle réparation...*)
- Une gestion des demandes d'adhésion faites en ligne
- Une messagerie interne (*Permettant aux différents administrateurs de communiquer entre eux*)
- Une gestion des vélos (*Pour ajouter des vélos, supprimer des vélos ou encore modifier des vélos*)
- Une gestion de réparation (*Permettant d'ajouter des réparations à certains vélos et de connaître le prix de ces réparations*)
- Une gestion du site côté client (*Ajouter des photos, modifier les partenaires, modifier le message d'accueil ...*)
- L'importation d'un calendrier Gmail (*Permettant d'importer un calendrier événement sur la partie client du site et d'afficher le calendrier interne de l'association sur la partie administration du site*)

1.2 La partie client

- Une formulaire d'adhésion
(*Permettant aux visiteurs de s'inscrire en ligne*)
- Un formulaire de demande de réparation*
(*Permettant aux adhérents de demander la réparations de leurs vélos*)
- Calendrier en ligne avec les événements de l'association
- Une page d'informations
(*Contenant la présentation de l'association, ses membres et ses partenaires*)

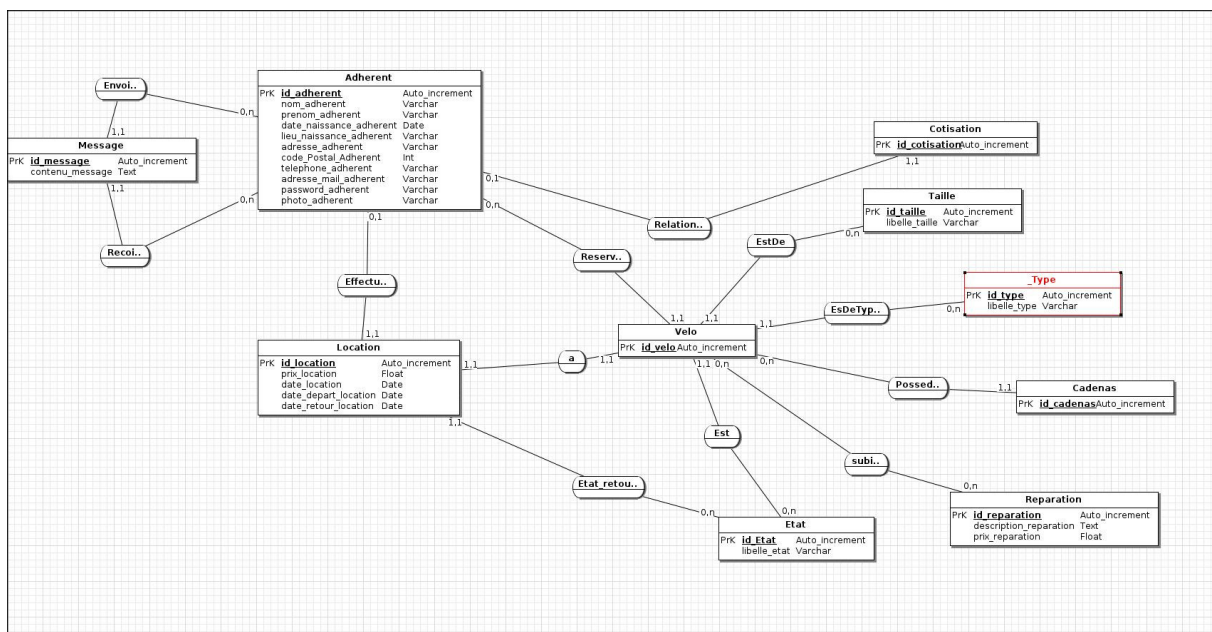
*. Cette partie n'est accessible que par les adhérents connectés

— Une page de photos et vidéos

Chapitre 2

Base de données

2.1 Le modèle conceptuel de données



2.2 La réalisation du MCD

Tout d'abord, nous avons mis en place un MCD pour avoir une représentation graphique de notre base de données avant de l'implémenter. Le MCD devait répondre à plusieurs contraintes :

- Les adhérents doivent avoir payé une cotisation
- Les adhérents ont plusieurs attributs (nom, prénoms, etc..)
- Les adhérents peuvent louer des vélos
- Une location possède un prix, une date de départ, une date d'arrivée et un état de retour
- Si l'état de retour est différent de l'état de base du vélo alors on modifie l'état du vélo et on impose une taxe au locataire, où on garde la caution.
- Un vélo a :
 - Un état
 - Une taille
 - 3 cadenas
 - Un type (modèle)
- Un vélo peut subir des réparations

Chapitre 3

Le fonctionnement du site internet

Le site est organisé en deux parties, une partie administration assez conséquente et une partie client.

3.1 La Structure

Tous d'abord nous avons utilisé Github qui est un logiciel de gestion de version. Vous pouvez donc trouver notre projet via ce lien : <https://github.com/kwidz/VeloCampus> De plus nous avons utilisé un framework CSS nommé Bootstrap, pour avoir un site ergonomique et complet sans avoir à utiliser des procédés complexes de CSS et de HTML5. Bootstrap gère aussi le "mobile-responsive" pour adapter le site web sur un smartphone. Enfin, afin de ne pas se perdre dans tous les fichiers, nous avons mis en place une arborescence (possibilité de la visionner entièrement sur notre Github) :

- index.php
(Ce fichier inclue les différentes pages html)
- footer.html
- header.html
- menu.html
(Ce fichier contient le menu que l'utilisateur voit lorsqu'il n'est pas connecté)
- menulog.html
(Ce fichier contient le menu que l'utilisateur voit lorsqu'il est connecté)

puis des dossiers portant le nom de chaque sous partie du menu, contenant un dossier de traitement ainsi qu'un index.

3.2 Les fonctions importantes

Ce projet comporte beaucoup de fonctionnalités donc par la suite nous expliquerons le fonctionnement de seulement quelques parties.

3.2.1 La partie location

La partie Location se trouve dans la partie administration, elle permet de :

- Créer une location
- Voir les vélos non loués
- Voir les personnes qui n'ont pas encore rendu leurs vélos
- Voir toutes les locations de l'année
- Ajouter un retour de location

Cette partie a été délicate à gérer car lorsque l'on loue un vélo il faut connaître son état, la date de location, les noms des cadenas qui sont loués avec le vélo ainsi que les informations de

l'adhérent qui loue le vélo. Il nous a donc fallu faire plusieurs jointure.

Par exemple : Cette requête sert à afficher les vélos non-loués.

```
SELECT v.id_velo, c.id_cadenas, ty.libelle_type, t.libelle_taille
FROM Velo v, Cadenas c, Taille t, _Type ty
WHERE v.id_velo not in
(
    SELECT id_velo
    FROM Location
    WHERE date_retour_location is null
)
AND c.id_velo=v.id_velo
AND v.id_type=ty.id_type
AND v.id_taille=t.id_taille
ORDER BY(id_velo);
```

De plus lorsque l'adhérent retourne son vélo, l'administrateur doit pouvoir modifier l'état du vélo et ajouter une date de retour.

3.2.2 L'affichage des demandes

Lorsque l'on fait une demande d'adhésion sur le site côté client, celle-ci est affichée sur la page d'administration en attente d'être validé ou refusé.

La demande d'inscription est validée si les informations données sont correctes et si la personne qui a fait cette demande est venu payer la cotisation à l'association. C'est seulement à ce moment là que la personne pourra se connecter sur le site public.

Voici un exemple :

Demandes d'inscription (à valider)				
Limballe	Pierre	Plus d'infos	✓	✗
test	t	Plus d'infos	✓	✗
cabrol	Morgane	Plus d'infos	✓	✗

Demandes de réparation (à valider)	
velo en feu	Limballe Pierre a@a.fr
cadre plié	Limballe Pierre a@a.fr
pneu creve	Glangine Geoffrey geoffrey.glangine@gmail.com

Si l'on clique sur le bouton valider à coté d'un prénom, la personne sera ajoutée à la table cotisation qui contient toutes les personnes cotisantes.

```
INSERT INTO Cotisation ('id_cotisation', 'id_adherent')
VALUES (NULL, ".$_GET['id'].");
```

Par contre si l'on clique sur le bouton supprimer, la personne sera retirée de la table adhérent.

```
DELETE from Adherent where id_adherent='".$_GET['id'].''
```

3.2.3 L'ajout d'image

Nous voulions créer une fonctionnalité simple qui permettrait aux membres de Vélocampus d'ajouter ou supprimer des photos sur leur site. De plus nous ne voulions pas ajouter seulement un lien vers l'image dans la base de données car nous voulions que lors de la suppression d'image, l'image soit réellement supprimée et pas seulement le lien qui mène à elle dans la base de données.



Nous avons mis en place un formulaire d'upload de fichier "classique" avec un bouton parcourir ainsi qu'un drag and drop pour uploader plusieurs images d'un seul coup. Les images sont mises dans un dossier portant le nom de l'album et nous affichons les images de manières récursives :

```
Fonction parcoursDossier(UnDossier){
    pour chaque fichier dans UnDossier:
        si fichier.Type = DOSSIER:
            parcoursDossier(fichier)
        sinon :
            on affiche l'image
        fin si
    fin pour
}
```

3.2.4 Le Postit

Les membres avaient besoin de pouvoir communiquer entre eux sur la page d'administration. Nous avons d'abord pensé à créer un compte administrateur pour chaque membre afin qu'ils puissent communiquer. Finalement, puisque tous les membres avaient tous les mêmes droits sur la page administration, il semblait plus logique de ne faire qu'une seule session administrateur. C'est ainsi que nous avons créé "le postit" :



Postit est une table à part dans la base de données, reliée à aucune autres. C'est pour cela qu'elle n'est pas présentée sur le MCD plus haut. Ce postit fonctionne simplement avec un formulaire qui envoie les valeurs des champs "nom" et "message" à une page php qui elle même les ajoute dans la base de données.

3.3 Inscription sur le site client

Dans ce projet nous avons du créer un formulaire d'inscription sur le site client afin que les visiteurs puisse demander à devenir adhérents.

A registration form titled 'Inscription' is shown. It includes a warning message at the top. The form has fields for 'Nom', 'Prénom', 'Date de naissance', 'Adresse', 'Code Postal', 'N° Tel', 'Adresse Mail', 'Mot de passe', and 'Confirmer'. There is a 'Valider' button at the bottom right and a 'Fermer' button at the bottom left. A red circle highlights the 'Valider' button.

Sur cette partie, nous avons du faire face à plusieurs situations :

- La personne non inscrite
- La personne qui est inscrite mais qui n'a pas cotisé
- La personne qui essaie de se réinscrire en étant déjà inscrite
- La personne qui a cotisé

Lorsqu'une personne tente de s'inscrire, une fonction Ajax vérifie que l'adresse mail entrée n'a jamais été utilisée si elle l'a déjà été, la fonction vérifie que l'adhérent n'est pas déjà cotisant donc déjà inscrit. Si c'est le cas elle retourne un message d'erreur :

Algorithme de la fonction :

```
// adresse_mail est la variable adresse_mail récupéré par la fonction.
SI (adresse_mail == VRAI) :
    RESULTAT = SELECT Count(*) from Adherent
                WHERE adresse_mail_adherent=adresse_mail
SI (RESULTAT == 0) :
    Message = OK !
SINON :
    RESULTAT = SELECT count(c.id_adherent)
    FROM Adherent a, Cotisation c
    WHERE adresse_mail_adherent= adresse_mail
    AND c.id_adherent=a.id_adherent
SI (RESULTAT == 0):
    Message ="attention cette adresse mail est utilisé pour un compte
    en attente de cotisation."
SINON :
    Message ="adresse mail déjà utilisée pour un autre compte"
```

3.4 Gestion du site client depuis la partie administration

L'intérêt de ce site était aussi de permettre aux administrateurs de gérer facilement la partie client de leur site. Nous avons donc créé quelques fonctions de gestion tels que "modifier le texte de l'accueil" ou "modifier les partenaires".

La fonction "modifier le texte de l'accueil" est géré par des fichiers *.txt* dans lesquels nous enregistrons les modifications.

La fonction "modifier les partenaires" est elle par contre gérée par une table partenaires. Elle permet aussi l'ajout de photo.

Voir le calendrier des réunions
Créer la mailling list
Modifier le texte de l'accueil
Ajouter des photos d'évènements
Modifier les partenaires
Modifier les membres
Changer le mot de passe admin
Voir le site public

Chapitre 4

Les problèmes rencontrés

4.1 Adblock

Adblock est un plug-in pour navigateur web permettant de bloquer les publicités sur les navigateurs. Elle est donc installée sur beaucoup d'ordinateurs. Malheureusement, elle ne bloque pas que les publicités, lors de la réalisation du projet nous nous sommes aperçu que celui-ci bloquait aussi toutes les requêtes Ajax du site (qui nous permettaient le plus souvent d'empêcher l'utilisateur de rentrer des données erronées dans un formulaire sur le site). Nous avons donc du modifier notre façon de fonctionner. Cela nous a fait perdre un peu de temps.

4.2 Les dates

En fonction des navigateurs, la gestion des dates est différente. Par exemple, sur le navigateur Google Chrome, un champ de type date est géré avec un calendrier. Sous Firefox, ce même champ est géré de la même manière qu'un champ de type text. Il a donc fallu traiter le format de date manuellement en fonction du navigateur que la personne utilise.

4.3 La sécurité

Ce projet étant utilisés par la suite sur internet, il était primordiale de le sécuriser. Or, nous n'avons aucun cours sur la sécurité informatique et internet à l'IUT. Nous avons donc dû étudier cet aspect de l'informatique par nous même. Cela nous a donc apporté une charge de travail supplémentaire.

4.4 Répartition du travail

Finalement, le plus gros problème que nous avons rencontrés est la répartition du travail. Le chef de projet a d'abord partagé les premières tâches en pensant vérifier à la fin de la semaine leurs réalisations pour, ensuite, pouvoir en fournir d'autres. Malheureusement, seul trois membres ont travaillés, le quatrième n'avait rien réalisé à la fin de la semaine. La tâche de celui-ci étant indispensable pour la suite, le chef de projet a préféré la coder lui même car il s'est rendu compte qu'elle ne sera pas faite. Ce petit manège a recommencé à chaque nouvelle distribution de tâches. C'est seulement à la fin du projet que l'élève en question a fini par avouer qu'il avait décidé de ne rien faire pour ce projet car la matière n'était pas assez importante à ses yeux. Cette mésaventure nous a fait perdre beaucoup de temps car nous avons souvent dû attendre sur le travail de l'élève en question pour continuer à avancer. De plus, nous avons du travailler plus pour rattraper le travail manquant de ce membre. Ce projet étant difficilement réalisable à quatre, la difficulté a évolué dès lors que nous sommes passés à seulement trois membres actifs.

Chapitre 5

Les améliorations possibles

Ce projet n'étant pas entièrement réalisable en un mois, il reste beaucoup de fonctionnalités qui auraient pu être implémentées plus tard, tels que :

- La réparation des vélos non loués
- La demande de location en ligne (*Nous avons décidé de ne pas la faire car il faut de toute façon se rendre à l'association pour récupérer le vélo loué.*)
- Gestion des comptes-rendus de réunions (*Il faudrait créer un éditeur sur la partie administration du site afin que le secrétaire de l'association puisse noter directement en ligne le compte-rendu des réunions et les sauvegarder en ligne.*)
- Ajout d'un forum (*L'association Vélocampus aurait aimé que nous mettions en place un forum pour tous les adhérents, nous n'avons malheureusement pas eu le temps*)
- Ajout d'une fonctionnalité de paiement en ligne
(*Nous aurions pu ajouter un lien Paypal mais cela compliquait les choses, de plus l'association préfère le contact humain.*)
- Créer une application Android

Conclusion

Finalement, ce projet était enrichissant car il devait à la fois répondre à la demande du professeur et à la demande du client. De plus, il nous a permis de découvrir le travail en collaboration avec un client qui compte sur nous. Enfin, le fait de développer un site qui sera utilisé par une association permet d'avoir une motivation autre qu'un projet banal, et de travailler dans une autre dynamique. L'envie de faire mieux est constamment présente.