

IUT BELFORT MONTBÉLIARD
Département informatique

PROJET DE BASE DE DONNÉES

SEMESTRE 3

Freétu

Rapport technique

Jean MERCADIER
Sofiane BELLAABOUSS
Nicolas FLEUROT
Gaëtan DUVAUX
Bastien DOUBLET
Jeremy RIGAULT
Alexis PLUMET

7 Janvier 2014

Table des matières

Introduction	1
1 Installation	3
1.1 Pré-requis	3
1.2 Installation	3
1.2.1 Installation automatique	3
1.2.2 Installation manuelle	4
2 Modélisation des données	5
2.1 Model conceptuel de données	5
2.2 Arborescence	6
2.3 Étude des fichiers	6
2.4 Problème algorithmiques et techniques	6
Conclusion	6

Introduction

Dans ce rapport, nous allons expliquer la partie technique du projet, en commençant par les logiciels nécessaire à l'application puis en détaillant la procédure d'instalation. Ensuite, nous étudierons le modèle conceptuel de donnés du projet puis son arborescence et nous terminerons en examinant l'utilité de chaque fichiers.

Chapitre 1

Installation

1.1 Pré-requis

Voici la liste des logiciels nécessaire au fonctionnement de l'application

- PHP 5.3.x minimum
- mysql 5.1.x minimum

de plus, l'application utilise les outils suivant :

- jquery 1.9.1
- API google MAP 3

en conséquence, une connexion internet est nécessaire pour utiliser l'application.

1.2 Installation

1.2.1 Installation automatique

Attention :

- L'installation automatique n'est disponible que sous les systèmes UNIX.
- Le port 443(https) doit être ouvert et disponible pour bénéficier de l'installation automatique, ce qui n'est pas le cas sur les ordinateurs de l'IUT

L'installation automatique se déroule de la manière suivante :

1. Ouvrir un terminal
2. exécuté la commande :
“*curl https ://raw.githubusercontent.com/Ricain/CoEtu/master/setup.sh > setup.sh*”
3. exécuté la commande :
“*bash ./setup.sh*”
Dans le cas où les ports ssh ne seraient pas disponibles, exécuté :
“*bash ./setup.sh -a*”
4. Suivez les instructions qui s'affichent à l'écran

L'application est désormais installée.

1.2.2 Installation manuelle

1. Récupérer le projet sur le site “<https://github.com/Ricain/CoEtu/>” ou avec la commande “`git clone https://github.com/Ricain/CoEtu.git`”
2. Placer le projet à la l’emplacement souhaité
3. crée un fichier `login.inc` à la racine du projet contenant le code suivant :

```
1 <?php
2     define( "LOGIN" ,/*"identifiant"*/ );
3     define( "PASSWORD" ,/*"mot_de_passe"*/ );
4     define( "BASE" , "coetu" );
5     define( "SERVER" ,/*"serveur"*/ );
6 ?>
```

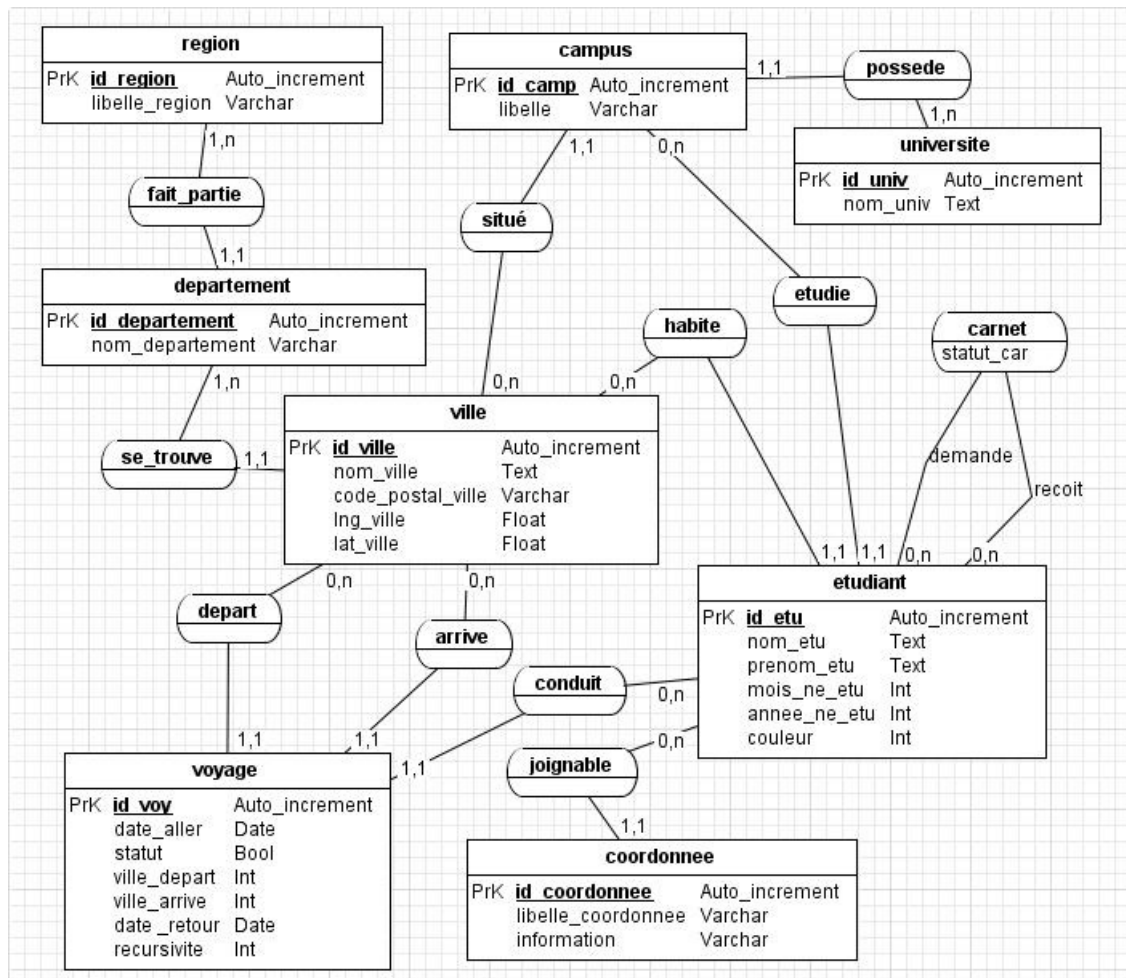
4. Exécuter la commande “`cd dev`”
5. Exécuter la commande “`mysql -u <identifiant> -h <serveur> -p <mot_de_passe>`”
6. Exécuter la commande “`source projetbdd.sql`”

L’application est désormais installée.

Chapitre 2

Modélisation des données

2.1 Model conceptuel de données



Ce MCD représente la base de données de notre application, il est implémenter sous forme d'un fichier SQL dans dev/projetbdd.sql, qui contient entre autre toutes les communes de France, permettant ainsi un fonctionnement optimal de l'application tout en permettant une comptabilité avec l'API Google MAP.

Le MCD est fait de telle sorte que chaque étudiant puisse fournir autant d'information qu'il le souhaite, dans le cas où il aurait plusieurs adresses email ou même Skype.

La relation réflexive **Carnet** représente la relation que chaque étudiant a avec les autres, cette relation est identifiée par *statut_car* qui définit si deux étudiants se connaissent ou non ou si il y a une demande de contact en cours.

2.2 Arborescence

2.3 Étude des fichiers

2.4 Problème algorithmiques et techniques

- Le moteur de recherche a posé et pose encore de nombreux problèmes. Dans son fonctionnement, il recherche tout les mots entrés en tant que personne puis en tant que voyage. Ainsi, si on écrit "Belfort", il renverra 1 résultat, si on écrit "Jean Belfort", sachant que Jean habite à Belfort, il renverra 2 résultats. Une fois la liste des résultats obtenue, le moteur de recherche compte et supprime les doublons. Plus le nombre de doublons est important, plus le résultat est pertinent. Le problème est que malgré tout, il arrive que certains résultats pertinents se retrouvent dans des positions peu élevées par rapport à d'autres résultats moins pertinents et vice versa.
- La liste des universités nous a posé un sérieux problème au niveau de la base de données. Nous pensions à l'origine utiliser des universités et des campus. Hélas, cela nécessitait énormément de travail au niveau de la récupération des données sur les différents campus de chaque université, nous avons donc dû modifier notre base de données pour n'utiliser que les académies et leur université associée. Or, même ainsi, avec 33 académies et entre 1 et 4 universités par académie, cela représentait un sérieux investissement en temps. Nous avons donc écrit un script en python se chargeant de récupérer toutes les académies et leur université et de créer les requêtes SQL associées.
- L'ajax nous pose un gros problème au niveau de l'optimisation. Nous avons commencé le projet en utilisant l'ajax pour renvoyer de l'HTML, ce qui, nous l'avons découvert plus tard, n'est ni beau, ni optimisé. Nous devrions donc modifier chaque requête ajax pour qu'elle renvoie de l'xml et créer les fonctions javascript associées pour les traiter.

Conclusion