

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	8

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 l’emplacement souhaité
3. créer 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 ?>
```

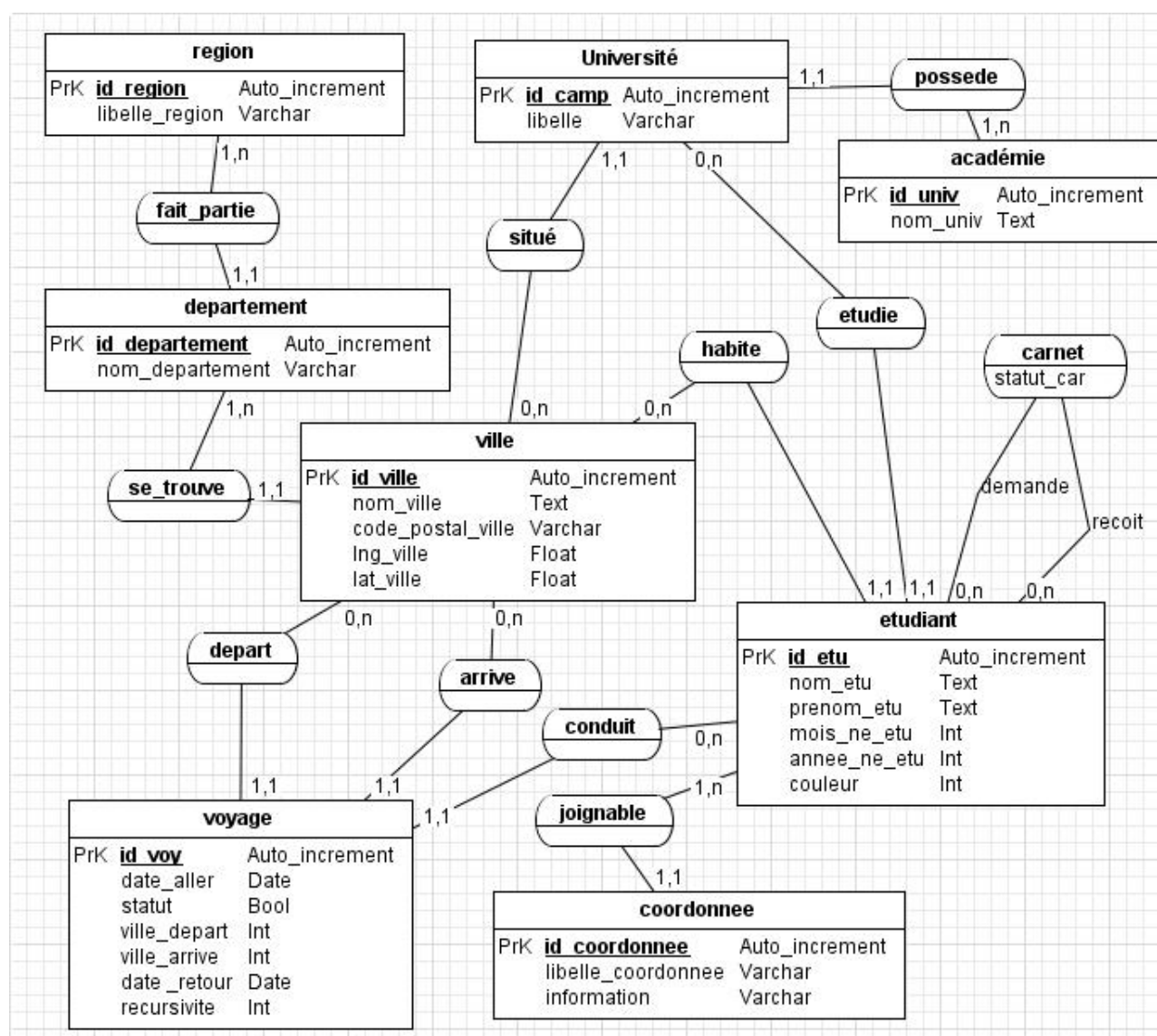
4. Exécuter la commande “`cd dev`”
5. Exécuter la commande “`mysql -u <identifiant> -h <serveur> -p <mot_de_passe>`”
6. Exécuter la commande “`source projetbdd.sql`”

L’application est désormais installée.

Chapitre 2

Modélisation des données

2.1 Model conceptuel de données



Ce MCD représente la base de donnée 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

- Ajax : Ce dossier contient toutes les pages PHP servant aux requêtes AJAX.
- getAllContact.php : Ce fichier permet de récupérer tous les contacts avec lesquels l'utilisateur courant est ami(e)s.
- getConversationCourante.php : Ce fichier permet de récupérer si il y a lieu, la conversation actuelle avec un contact.
- getCoordItineraire.php : Ce fichier permet de récupérer les latitudes et longitudes des points de départ et d'arrivée.
- getCoordVille.php : Ce fichier permet de récupérer les latitudes et longitudes d'une ville en fonction de son identifiant.
- getInfoContact.php : Ce fichier permet d'afficher les informations sur un contact ou propose d'ajouter la personne à ses contacts si elle n'en fait pas partie.

- getListeCampus.php : Ce fichier permet de récupérer la liste des universités pour l'autocomplé-
tion.
- getListeVille.php : Ce fichier permet de récupérer la liste des villes pour l'autocomplé-
tion.
- getNbNotification.php : Ce fichier permet de récupérer le nombre de notification non lu d'un
utilisateur.
- getNotification.php : Ce fichier permet d'afficher toutes les notification non lu ou d'afficher un
message disant qu'il n'y a aucune nouvelle notification.
- getNouveauMsg.php : Ce fichier permet de récupérer et de mettre en forme les message
entrant d'une conversation.
- getOpenConversations.php : Ce fichier permet de récupérer la liste des conversation en cours
et affiche un style particulier au conversation possédant des message non lus.
- getPersonneInfo.php : Ce fichier permet de récupérer les information d'un contact pour les
mettre dans un popup.
- getRecherche.php : Ce fichier gere le moteur de recherche, comme expliquer en 2.4 premier
tirez.
- getVille.php : Ce fichier créer la balise div permettant d'insérer correctement la carte google
map.
- getVoyageForm.php : Ce fichier permet d'afficher le formulaire de création de voyage.
- getVoyage.php : Ce fichier permet d'afficher les information sur un voyage.
- getVoyages.php : Ce fichier permet de récupérer une liste de tout les voyage de l'étudiant
courant.
- sendAccepteContact.php : Ce fichier permet de mettre à jour le statut d'une relation entre
deux étudiant.
- sendDemandeContact.php : Ce fichier permet de valider une demande de contact.
- sendFormVoyage.php : Ce fichier permet de faire la vérification des entrée dans le formulaire
de création de voyage.
- sendNouveauMsg.php : Ce fichier permet d'ajouter les nouveau message à la base de donnée.
- sendRefuseContact.php : Ce fichier permet de refuser une demande de contact.
- sendSupprContact.php : Ce fichier permet de supprimer un contact.
- sendSupprVoyage.php : Ce fichier permet de supprimer un voyage en vérifiant si l'on supprime
bien on voyage nous appartenent.
- carnet : Ce dossier contient la page de contact.
 - index.php : Ce fichier permet d'afficher la liste des contact avec leur information. Ils permet
aussi de naviger entre les contact avec le pavée directionnel ou avec les touches du clavier, en
appuyant sur la première lettre du nom du contact rechercher.
- cont : Ce dossier contient le fichier d'entete du site.
 - head.html : Ce fichier represente l'interieur de la balise <head> du site.
- css : Ce dossier contient les fichier de style du site.
 - color.php : Ce fichier permet de généré du css a partir de php, permettant ainsi au utilisateur
de choisir eu même le theme de leur comptes.
 - connec.css : Ce fichier représente la feuille de style de la page de connexion du site.
 - inside.css : Ce fichier représente la feuille de style du site après la connexion.
- dev : ce dossier contient divers fichier utiles au développement de l'application.
 - 1.mcd : Le mcd du projet
 - importBDD.php : Ce fichier permet d'importé la base de donnée depuis une page internet.
 - login.inc.sample : Un exemple de fichier de connexion.
 - mdpgen.php : Ce fichier permet de récupérer le sha256 d'une chaine de caractère.
 - page_template.php : fichier que nous utilisons en début de projet pour crée de nouvelle page.
 - projetbdd.sql : Le fichier SQL representant notre base de donnée.
 - recupUNIV.py : Un script python permettant de récupérer chaque academie et chacune de leur
universités.
- home : Ce dossier contient la page d'accueil du site une fois connecter.
 - index.php : Ce fichier represente la page d'accueil du site une fois connecter, il permet de
visualiser rapidement ses information personnel ainsi qu'un acces plus naturel au différentes
autre partie du site.
- img : Ce dossier contient toutes les image du site.
- js : Ce dossier contient tout les script javascript du site.
 - jscolor : ce dossier contient tous les élément permettant au selectionneur de couleur de fonctionné
 - jscolor.js : Ce fichier contient toute les fonction javascript nécessaire au bon fonctionnement
du selectionneur de couleur.
 - ajax.js : Ce fichier contient toutes fonction utilisant ou traitant de l'ajax

- `conne.js` : Ce fichier contient les fonction jquery nécessaire a l'autocompletion des ville et des universités dans la page d'accueil.
- `html.js` : Ce fichier contient des fonction dont le seul but est de générer ou modifier de l'html.
- `inside.js` : Ce fichier contient les fonction utile au fonctionnement de l'application après la connexion.
- `map.js` : Ce fichier contient différentes fonction nécessaire a l'utilisation de l'API google map.
- `popup.js` : Ce fichier contient les fonction nécessaire a la création et l'utilisation des différents popup du site.
- `suggestion.js` : Ce fichier contient les fonction jquery d'autocompletion lors de la création de voyage.
- `lib` : Ce dossier contient toutes les librairie php utiliser dans le projet.
- `sql` : Ce dossier contient toutes les librairie utilisant du sql.
 - `delete.php` : Ce fichier contient toutes les fonction permettant de supprimer des éléments de la base de donnée.
 - `insert.php` : Ce fichier contient toutes les fonction permettant d'ajouter des éléments de la base de donnée.
 - `select.php` : Ce fichier contient toutes les fonction permettant de récupérer des éléments de la base de donnée.
 - `update.php` : Ce fichier contient toutes les fonction permettant de modifier des éléments de la base de donnée.
- `bibli.php` : Ce fichier contient toutes les fonction un peu à part du programme.
- `deco.php` : Ce fichier permet de renvoyer à la page de connexion tout utilisateur n'étant pas connecté.
- `html.php` : Ce fichier contient toutes les fonction aillant pour but de générer de l'html.
- `securiter.php` : Ce fichier contient des sécurité vis à vis des attaques par injection SQL.
- `sql.php` : Ce fichier permet d'inclure en une fois toutes les bibliothèque SQL et fournit une fonction permettant de récupérer un objet PDO de manière propre.
- `message` : Ce dossier contient la page de tchat du site.
 - `index.php` : Ce fichier représente le tchat du site, permettant ainsi de discuter entre les étudiants. Il contient aussi les fonction permettant de naviguer entre les étudiants avec qui une conversation a été démarrée.
- `profil` : Ce dossier contient la page de profil utilisateur du site.
 - `index.php` : Ce fichier représente la page de profil utilisateur, permettant de modifier le thème du site ainsi que ses informations personnelles.
- `rechercher` : Ce dossier contient la page de recherche.
 - `index.php` : Ce fichier représente la page de recherche du site, permettant de rechercher voyages et étudiants.
- `voyage` : Ce dossier contient la page des voyages de l'utilisateur.
 - `index.php` : Ce fichier représente la page des voyages de l'utilisateur courant. Elle permet de créer et de supprimer des voyages.
- `README.md` : Rapide résumé du site en markdown.
- `index.php` : Page d'accueil du site, elle permet de s'inscrire et de se connecter.
- `login.inc` : Fichier contenant les informations de connexion à la base de donnée.
- `setup.sh` : Script d'installation automatique de l'application.

2.4 Problème algorithmiques et techniques

- Le premier problème auquel nous avons été confronté fut la connexion des utilisateurs. Nous voulions qu'elle soit sécurisée tout en étant relativement simple. Nous avons finalement opté pour la solution suivante :
 1. l'utilisateur rentre son mot de passe.
 2. le mot de passe est hashé en sha256.
 3. le hash est comparé avec celui contenu dans la base de donnée.
 4. si le hash correspond, l'utilisateur est identifié.
- 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

doublon. Plus le nombre de doublon 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.
- La création du script d'installation automatique nous a posé de nombreux problèmes, en premier lieu car il nécessite certains outils (wget, curl, git, unzip) qui ne sont pas forcément disponibles sur toutes les machines, mais au niveau de la sécurité des dites machines, qui, dans certains cas, bloque l'accès à certains ports, en particulier le port 443(https) rendant ainsi le script inutile et forçant une installation manuelle.