

Année universitaire 2020/2021

Rapport de projet

Matière: Réseaux & Base de données

Groupe A13

Étudiants :

* + Racim RIGHI
  + Marlyse Hangamalongo MAPAGA
  + Lydia TRABELSI

Fait le 04/12/2020

**Table des matières**

[1 INTRODUCTION 2](#__RefHeading___Toc1634_3580788858)

[1.1 Contexte du projet 3](#__RefHeading___Toc1636_3580788858)

[1.2 Objectifs et caractéristiques 3](#__RefHeading___Toc1638_3580788858)

[1.2.1 Partie base de données 3](#__RefHeading___Toc3632_3098746779)

[1.2.2 Partie réseau 3](#__RefHeading___Toc2189_3098746779)

[1.2.2.1 Serveur réseau 4](#__RefHeading___Toc1640_3580788858)

[1.2.2.2 Client réseau 4](#__RefHeading___Toc1642_3580788858)

[1.2.3 Partie web 4](#__RefHeading___Toc2191_3098746779)

[1.2.3.1 Serveur web 4](#__RefHeading___Toc1644_3580788858)

[1.2.3.2 Interface web 4](#__RefHeading___Toc1646_3580788858)

[1.3 Choix techniques 5](#__RefHeading___Toc1648_3580788858)

[2 BASE DE DONNÉES 5](#__RefHeading___Toc2193_3098746779)

[2.1 Dictionnaire de données 5](#__RefHeading___Toc1911_3580788858)

[2.2 MCD 7](#__RefHeading___Toc1913_3580788858)

[2.3 MLD 8](#__RefHeading___Toc1915_3580788858)

[2.4 Création de la base de données 8](#__RefHeading___Toc827_2390633755)

[2.5 Jeu de données 8](#__RefHeading___Toc829_2390633755)

[3 Partie réseau 9](#__RefHeading___Toc831_2390633755)

[3.1 Schéma d’architecture contextualisé 9](#__RefHeading___Toc833_2390633755)

[3.2 Diagramme applicatif 10](#__RefHeading___Toc835_2390633755)

# INTRODUCTION

Dans cette section, nous allons présenter les principaux aspects et objectifs de notre projet, et ce en décrivant le contexte, les fonctionnalités prévues de chaque partie ainsi que les choix techniques.

## Contexte du projet

À chaque rentrée universitaire, on remarque toujours des étudiants perdus, désorientés et à la recherche d’une aide si précieuse, en particulier pour les étudiants étrangers. Bien qu’il existe des sites web d’informations, des salons sur différentes applications dédiés aux sections (Discord, Teams...etc.) , il reste parfois compliqué de contacter des étudiants expérimentés pour avoir des informations sur certaines démarches, ou juste de réussir à s’adapter rapidement.

## Objectifs et caractéristiques

L’objectif de notre projet est d’offrir aux nouveaux étudiants un moyen de trouver des contacts rapidement et efficacement selon leurs besoins, et ce pour faciliter leur intégration que ce soit dans une nouvelle section, spécialité, université ou pays. Et en même temps de permettre aux anciens étudiants d’être utile à l’établissement.

Les fonctionnalités attendues du système sont les suivantes :

* Effectuer une demande d’information sur la borne ou sur internet
* Établir un contact soit en distanciel (message personnalisé) ou en présentiel entre étudiant en quête de renseignement et étudiant expérimenté
* Consulter la foire aux questions
* Faire le suivi de sa demande
* Gérer la foire aux questions

Le système doit être accessible de la façon la plus simple.

### Partie base de données

Dans la base de données, on devra prendre en compte les informations sur les bornes disponibles à travers la région, les demandes issues des utilisateurs ainsi que leurs informations personnelles, et toutes les interactions qui sont effectuées entre les demandeurs et les assistants, y compris les messages avec les métadonnées nécessaires. Nous aurons aussi un système d’authentification ce qui nécessite l’enregistrement des identifiants des utilisateurs

### Partie réseau

La partie réseau consiste en une borne d’information qui peut être posée aux différents endroits d’une université, elle permet aux étudiants à la recherche d’aide ou d’orientation de lancer une demande via le serveur réseau qui sera diffusée aux comptes des utilisateurs connectés.

#### Serveur réseau

Le serveur réseau va s’assurer de transmettre de manière sécurisée la demande d’un étudiant en détresse vers le serveur web. Il affichera ensuite à l’étudiant les éléments correspondants à sa demande. Par ailleurs, le serveur réseau restera en attente d’une réponse une fois la demande envoyée, afin de notifier l’étudiant d’un accusé de réception.

#### Client réseau

Le client réseau va proposer aux utilisateurs un service d’authentification rapide pour s’assurer qu’il est question d’un étudiant. Une fois connecté, grâce aux options prédéfinies, l’étudiant pourra soit effectuer une demande d’assistance physique s’il n’en a pas déjà initié une, soit consulter la foire aux questions s’il s’agit d’un renseignement courant, sinon poser une question.

### Partie web

Le site web va servir comme une plateforme d’assistance en ligne, où les utilisateurs avec un compte personnel, peuvent consulter les demandes, répondre à des questions pour enrichir les réponses des bornes et modifier leurs informations personnelles.

#### Serveur web

Ça sera un serveur web classique, qui va gérer les requêtes envoyées par l’interface pour satisfaire les demandes des utilisateurs, en plus du fait qu’il va gérer les nouvelles demandes depuis les bornes et les envoyer aux bons destinataires.

#### Interface web

L’interface web va nécessiter un service d’authentification et de vérification de l’ID étudiant, pour ensuite afficher à l’utilisateur les demandes d’aide récemment envoyées selon ses propres informations et sa région/université. Il pourra ensuite soit contacter la personne concernée via une messagerie ou envoyer une réponse rapide.

## Choix techniques

Pour la réalisation de notre projet, on a choisi le langage **C** pour toute la partie réseau (client et serveur), pour faciliter la connexion et les échanges entre eux. Comme protocole de connexion on a fais des connexion avec le protocole **TCP.**

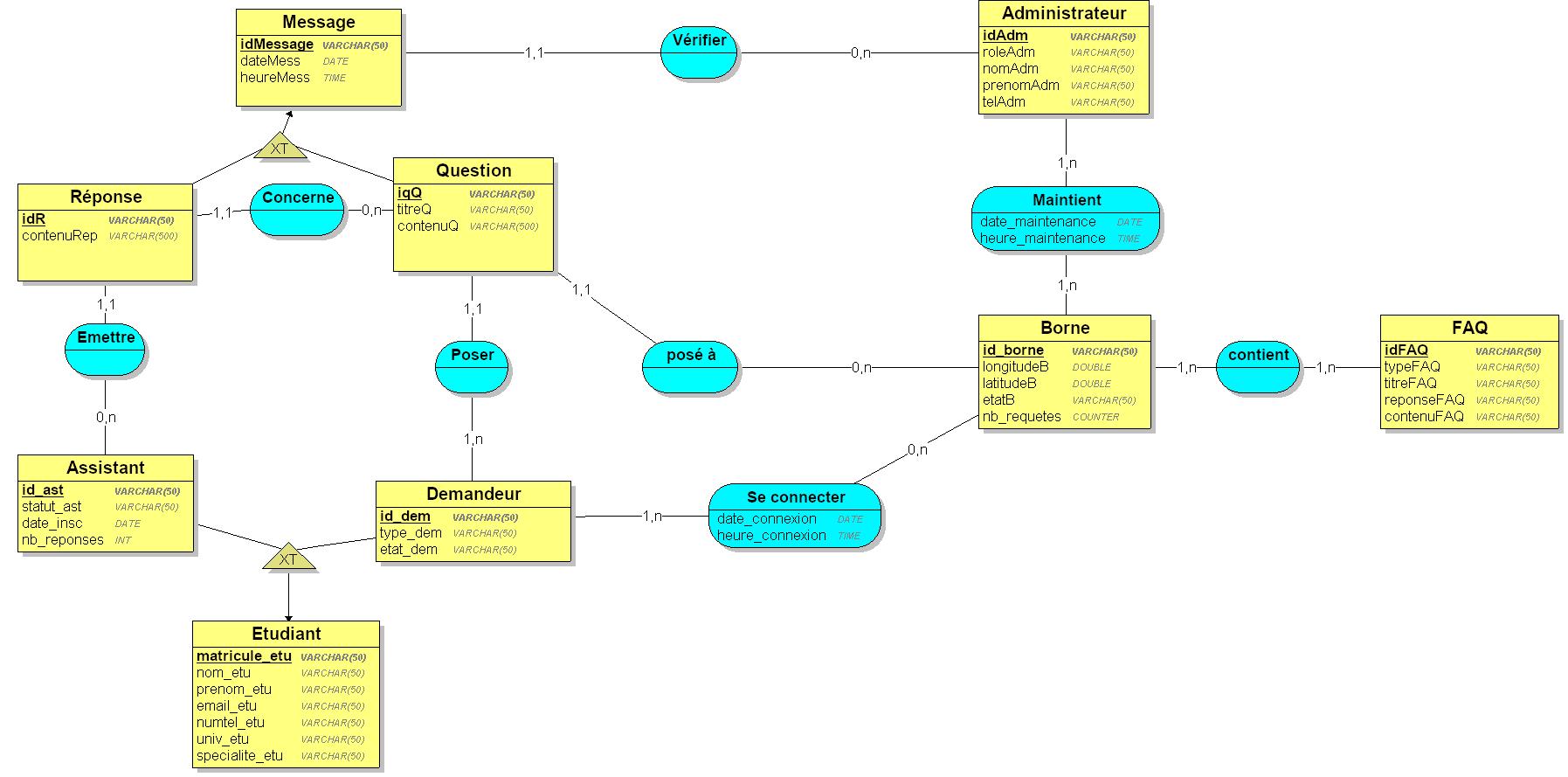
Pour le reste, on a choisi **PHP** pour le serveur web, et un site web en **HTML, CSS et Javascript.**

# BASE DE DONNÉES

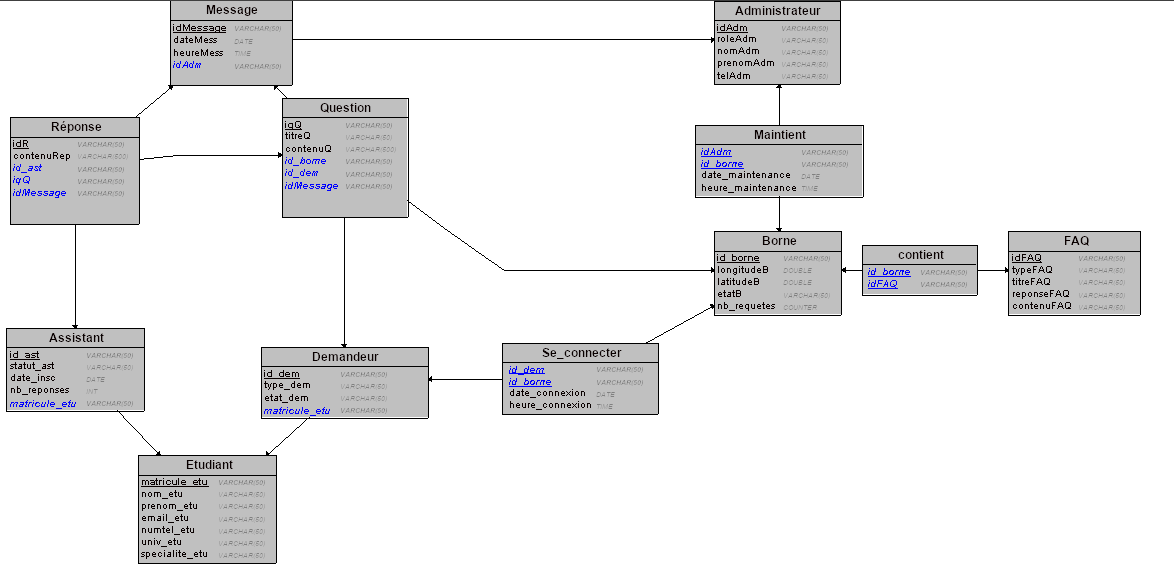
## Dictionnaire de données

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Description** | **Type** | **Remarque** |
| **Borne** |  | | |
| id\_borne | Identifiant unique de la borne | AN | Cet identifiant sera généré au préalable à la création des bornes |
| LongitudeB | Longitude de la localisation de la borne | N |  |
| LatitudeB | Latitude de la localisation de la borne | N |  |
| etat\_B | État de la borne (en marche, en panne ..etc.) | AN |  |
| nb\_requetes | Nombre de requêtes effectué sur une borne | N |  |
| **Administrateur** |  | | |
| IdAdm | Identifiant de l’administrateur | N |  |
| nomAdm | Nom a | AN |  |
| prenomAdm | Prénom de l’administrateur | AN |  |
| telAdm | Téléphone de l’administrateur | AN |  |
| roleAdm | Rôle de l’administrateur | AN |  |
| **FAQ** |  | | |
| idFAQ | ID généré par incrémentation | N |  |
| typeFAQ | Type de la question (Peut varier de Astuces, Spécialité, …etc) | AN |  |
| titreFAQ | Une question | AN |  |
| contenuFAQ | Détails sur la question | AN |  |
| reponseFAQ | Réponse à la question | AN |  |
| **Étudiant** |  | | |
| matricule\_etu | Matricule de l’étudiant | AN | C’est le matricule donné par l’université |
| nom\_E | Nom de l’étudiant | AN |  |
| prenom\_E | Prénom de l’étudiant | AN |  |
| tel\_E | Numéro de téléphone | AN |  |
| email\_E | E-mail de l’étudiant | AN |  |
| université\_E | Université de l'étudiant | AN |  |
| specialite\_E | Spécialité de l’étudiant | AN | Informatique, biologie, physique … etc |
| **Demande** |  | | |
| id\_dem | Id unique d’une demande faite par un étudiant | AN | Elle est générée par le serveur réseau en utilisant le matricule et la date et heure de la demande |
| type\_dem | Type de la demande si disponible | AN |  |
| etat\_dem | En cours de traitement, traité, annulé...etc |  |  |
| **Assistant** |  | | |
| id\_ast | Id unique de l’assistant | AN |  |
| statut\_ast | L’assistant est actif ou non | AN |  |
| date\_inscr | Date d’inscription de l’étudiant | AN |  |
| nb\_reponses | Nombre de réponses effectués par l’étudiant | N |  |
| **Connexion** |  | | |
| date\_connexion | Date de connexion d’un étudiant demandeur à une borne | DATE |  |
| heure\_connexion | Heure de connexion d’un étudiant demandeur à une borne | TIME |  |
| **Message** |  | | |
| IdMessage | Id unique d’un message, qui va être hérité aux questions et réponses | AN |  |
| dateMess | Date d’émission du message | DATE |  |
| heureMess | Heure d’émission du message | TIME |  |
| **Question** |  | | |
| idQ | Id unique d’une question | AN |  |
| titreQ | Titre de la question | AN |  |
| contenuQ | Description détaillée de la question | AN |  |
| **Réponse** |  | | |
| IdR | Id unique d’une réponse | AN |  |
| contenuR | Contenu de la réponse | AN |  |
| **Maintenance** |  | | |
| date\_maintenance | Date d’une maintenance d’une borne | DATE |  |
| heure\_maintenance | Heure d’une maintenance d’une borne | TIME |  |

## MCD

Figure 1 – Modèle conceptuel de données

## MLD

Figure 2 – Modèle logique de données

## Création de la base de données

Après avoir créé le fichier **creation.sql** (joint dans les livrables), on a plus qu’a exécuter le script en passant au compte postgres sous Linux avec la commande

**$ sudo su – postgres**

Ensuite on exécute le script avec la commande

**$ psql -d <nom\_BD> -U <utilistateur> -f <lien\_vers\_le\_fichier> -W**

Le fichier commende par des **drop cascade,** et ce pour pouvoir tester la base de données avec différents jeux de données indépendants, en lançant le fichier **insertions.sql**

## Requêtes

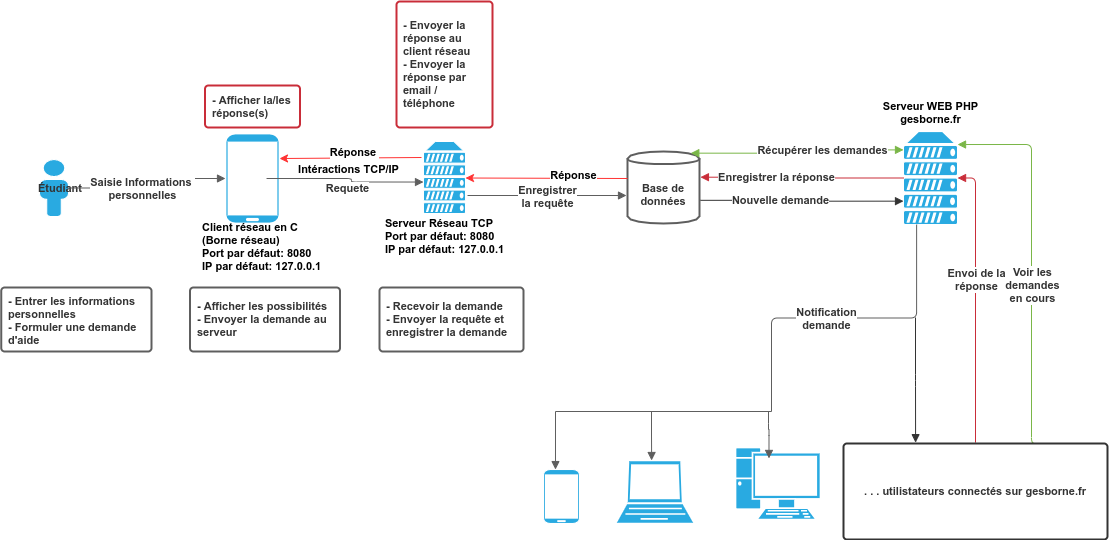
## Jeu de données

Pour tester les différentes partie du projet, on a créé un jeu de données polyvalent, en commençant par les données des bornes, qui sont injectés dès la création de la base car elles sont définitives et ne contiennent qu’un champ qui peut être changé par l’administrateur.

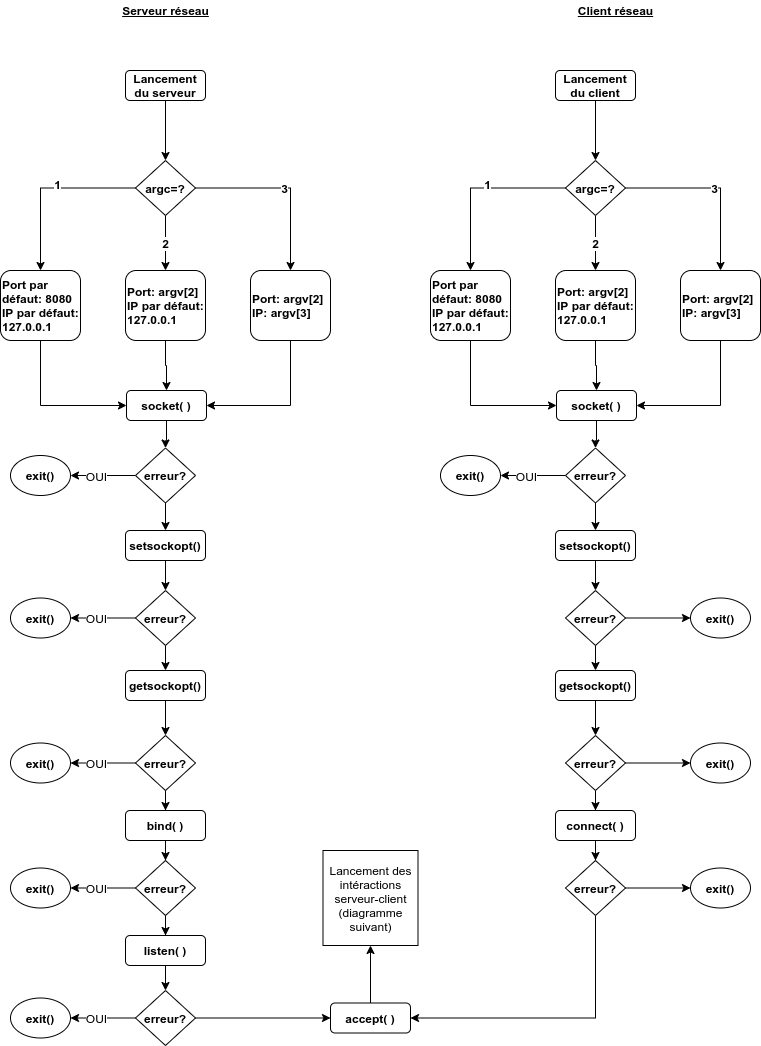
L’ensemble du code du jeu de données est dans les fichiers **insertions.sql** et **questions.sql**

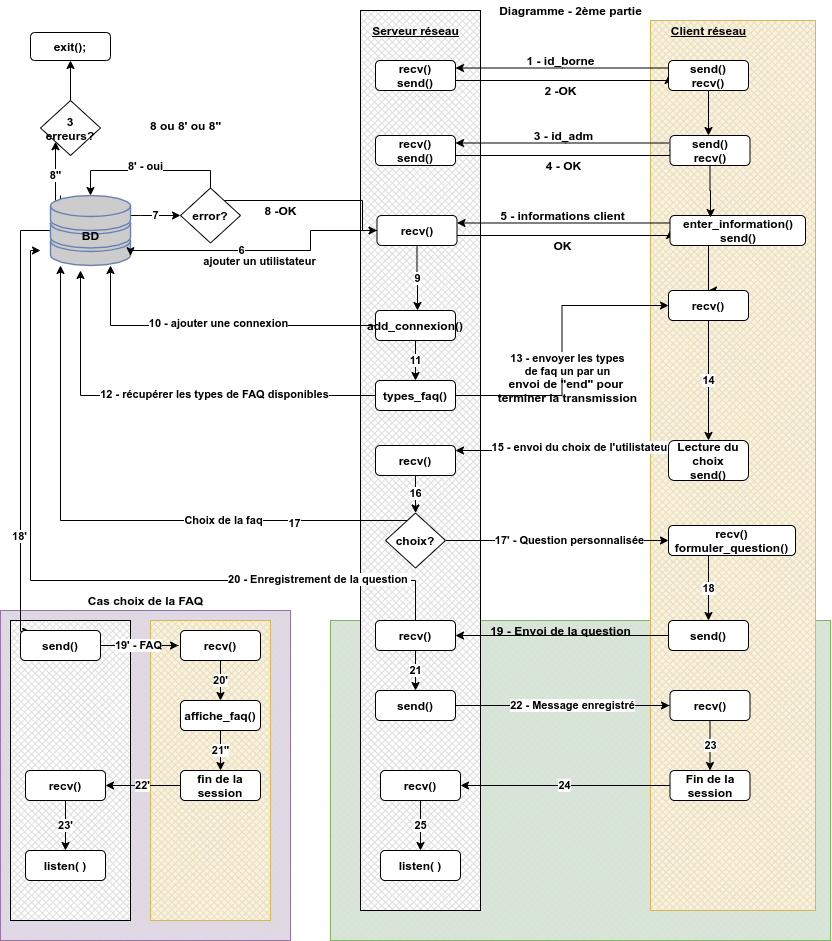
# Partie réseau

## Schéma d’architecture contextualisé

Figure 3 – Schéma d’architecture contextualisé

## Diagramme applicatif

Figure 4 – Diagramme applicatif (Partie 1 – Connexion client serveur)

Figure 5 – Diagramme applicatif (Partie 2 – Connexion et intéractions)