

Projet : un gestionnaire de services communaux

UCBL - Département Informatique de Lyon 1 – BDW - automne 2023

- Ce projet est évalué par plusieurs notes : diagramme E/A, QCM SQL, QCM programmation web, soutenance de projet et qualité du code.
- Cette UE est avant tout une introduction aux **bases de données**. La non-utilisation d'une BD relationnelle dans le projet peut se voir attribuer la note 0.
- Le rendu du projet et la soutenance ont lieu lors de la dernière séance de TP. Tout rendu ne respectant pas les consignes peut se voir attribuer la note 0.
- Le projet est réalisé en monôme ou en binôme (pas de trinôme). Un binôme signifie deux étudiant·e-s **du même groupe de TD**.
- Attention à la gestion de votre temps : concentrez-vous sur l'implémentation des fonctionnalités demandées, et pas plus. Cela ne sert à rien de gagner un point bonus pour quelques améliorations mineures mais de perdre plusieurs points pour des fonctionnalités non développées.

L'objectif du projet est de développer un site web permettant de gérer les services proposés par les communes (e.g., certains services proposés au niveau national sur [service-public](#)).

1 Des spécifications au script SQL

Les spécifications sont les suivantes. Une commune est définie par un identifiant, un code postal, un nom, des coordonnées, un code INSEE et l'adresse de la mairie. Elle est rattachée à un département (code INSEE et nom), lui-même situé dans une région (code INSEE et nom). Une commune propose différents services à ses citoyen·ne·s. Un service possède un libellé, une éventuelle description et une période d'ouverture propre à la commune (e.g., inscriptions scolaires du printemps jusque la rentrée pour Villeurbanne). Les services peuvent être gratuits ou payants, selon la législation nationale ou sur décision de la commune. Les citoyen·ne·s peuvent effectuer des demandes, chacune concernant un seul service. Chaque demande se caractérise par un identifiant, une date, un message optionnel, la personne à l'origine de la demande et une liste de justificatifs. Chaque justificatif est numéroté à partir de 1 pour chaque demande, et on stocke également son type, une description et un chemin vers le fichier. Des informations supplémentaires sont enregistrées selon le type de service demandé. Le service d'état civil permet de préciser le type de document souhaité (e.g., passeport) et la date de mise à disposition de ce document. Pour le service élections (e.g., pour une inscription sur les listes électorales), seul le bureau de vote est stocké. Pour un signalement (e.g., voiture gênante, panneau cassé), on enregistre son type et un lieu de localisation. Les lieux sont issus d'une source de données existante et comportent un identifiant, un nom, un type et des coordonnées géographiques. Le service d'union civile est dédié à la reconnaissance légale d'un couple. Pour chaque union, on stocke donc son type (e.g., mariage, PACS), la date prévue ainsi que les deux citoyen·ne·s concernées (email, nom, prénom, adresse, date de naissance, téléphone). Deux mêmes citoyen·ne·s peuvent s'unir sous différents contrats (e.g., concubinage puis PACS). Le service scolaire concerne l'inscription d'enfant à l'école. La demande nécessite de préciser un nom et numéro de téléphone d'une personne contact. Chaque demande concerne un seul enfant, pour lequel on connaît son nom et son prénom. Un enfant est inscrit dans une classe donnée dans une école, qui est un lieu avec une adresse et un nombre de classes. On indique également si c'est la première inscription de l'enfant dans cette école. Notez qu'en plus des renouvellements annuels, des changements d'école peuvent avoir lieu en cours d'année, y compris dans une même commune (e.g., déménagement). Le service restauration (scolaire) est utilisé pour la cantine. En général, des tarifs réduits sont accordés sur le prix des repas en fonction du quotient familial renseigné. Une demande de restauration précise aussi la cantine souhaitée. Cette dernière possède un nom, une adresse, un nombre de places, un nombre de services, et des coordonnées géographiques. Les enfants mangent dans une cantine pendant une période donnée,

et le système stocke le nombre de jours d'absence (repas non comptabilisés).

Dans un premier temps, nous allons concevoir la base de données à partir des spécifications. Votre modélisation doit respecter au mieux ces spécifications, mais votre site web n'utilisera pas tous les concepts décrits.

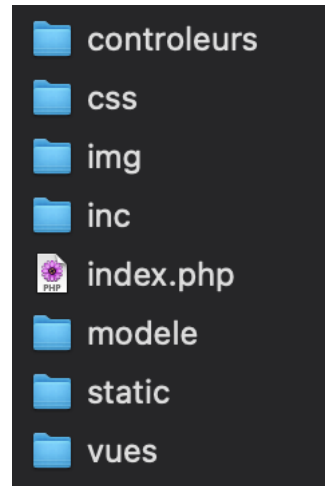
1. Produisez un diagramme entité / association pour ces spécifications (par exemple avec [MoCoDo](#), [Looping](#), [AnalyseSI](#) ou [JMerise](#) - attention, chaque outil a ses contraintes et limitations).
2. Produisez le schéma relationnel dérivé de votre diagramme E/A. Si vous générez ce schéma avec un outil de modélisation, il est recommandé de le vérifier et éventuellement de le corriger / compléter.
3. Produisez le script SQL permettant la création de la base de données. Si vous générez ce script avec un outil de modélisation, il sera nécessaire de le vérifier et de le corriger / compléter (ces modifications devraient être stockées dans un autre fichier que celui généré par l'outil de modélisation). Enfin, utilisez PHPMYAdmin pour insérer des instances fictives dans les tables les plus importantes.

2 Design du site et pages statiques

Votre site est codé selon l'architecture MVC, et vos fichiers doivent impérativement respecter l'arborescence suivante : un fichier `index.php`, des répertoires `controleurs`, `css`, `modele`, `static` et `vues`. L'évaluation du code sera automatisée et le non-respect de cette architecture vous pénalisera.

Vous êtes libres de concevoir vos pages comme bon vous semble. Mais chacune de vos pages doit avoir les zones suivantes :

- Un entête (`<header>`), avec un logo et un nom de site ;
- Un menu (`<nav>`), dont les libellés seront explicites ;
- Le contenu de la page, qui correspond aux fonctionnalités développées dans les sections suivantes. Prévoyez également de créer une page d'accueil qui décrit les objectifs de votre site ;
- Un pied de page (`<footer>`), avec un lien vers la page de BDW et des remerciements (e.g., pour les images).



Il ne faut **pas** implémenter un système d'authentification¹ ! La mise en page et mise en forme se feront évidemment avec des styles CSS. L'esthétique est prise en compte lors de la notation, aussi, soignez votre site. Le projet sera évalué avec le navigateur **Mozilla Firefox**, donc vérifiez le rendu de votre site avec ce navigateur !

3 Fonctionnalité 1 : afficher des statistiques

Cette fonctionnalité consiste simplement à créer une page qui affiche le résultat des statistiques suivantes :

- Nombre d'instances pour 3 tables de votre choix ;
- Liste de chaque enfant et de son école actuelle ;
- Liste des enfants avec le nom de la cantine où ils mangeront le 01/01/2024 ;
- Paires d'enfants ayant les mêmes nom et prénom, mais inscrits dans des écoles différentes ;
- Top-3 des départements ayant le plus de communes ;
- Top-3 des services les plus demandés (par les citoyen-ne-s) ;
- Top-3 des services les plus proposés (par les communes) ;
- Top-3 des communes qui réalisent le plus d'unions.

4 Fonctionnalité 2 : intégrer les communes de la région

Cette fonctionnalité consiste à migrer des données fournies dans votre base de données. Ces données ne sont malheureusement pas correctement modélisées, et il est donc nécessaire de les transformer avant de les stocker dans votre base. C'est un processus fréquent appelé *intégration de données*. L'utilisation de ces données est **obligatoire**, et votre script d'intégration doit être **exécutable sur de nouveaux jeux de données** lors de la maintenance.

Vous allez donc écrire un script PHP ou SQL qui interroge le jeu de données fourni pour extraire les communes de la région Auvergne Rhône-Alpes (uniquement ces communes, pas celles des autres régions), les transformer si nécessaire, puis les insérer dans votre base. En pratique, vous allez tester votre script à plusieurs reprises, donc prévoyez de vérifier si les communes, départements ou région n'existent pas déjà avant d'insérer !

On peut décomposer l'étape d'intégration en trois étapes :

- Réfléchir aux correspondances entre les données existantes et votre schéma (e.g., à quel attribut de votre schéma correspond tel attribut fourni). Vérifier si les types de données sont cohérents entre attributs correspondants, et quelles sont les transformations nécessaires au niveau des valeurs. N'oubliez pas que le jeu de données fourni n'est pas bien modélisé : si vous modifiez votre schéma de BD pour intégrer plus facilement les données fournies, votre schéma ne respectera plus les spécifications demandées et vous serez pénalisé.e ;

¹Un système d'authentification se base sur des protocoles standardisés et sécurisés comme SSL, Kerberos ou CAS.

- Pour accéder aux données fournies et récupérer les informations pertinentes, il faut indiquer le nom de la BD (`dataset`) avant la table (`communes`) au niveau du `from` de vos requêtes : `SELECT * FROM dataset.Communes;`
- Pour chaque tuple résultat, appliquer éventuellement des transformations sur les valeurs (soit directement en SQL, soit en PHP), puis écrivez une requête `INSERT` pour peupler vos tables. Si certaines données sont manquantes et ne permettent pas de respecter les contraintes de votre schéma, utilisez un `NULL` ou choisissez une valeur par défaut. Enfin, n'oubliez pas de peupler les tables liées !

5 Fonctionnalité 3 : ajouter un service

Pour cette première fonctionnalité, nous allons créer une page pour ajouter un nouveau service. Celle-ci inclut un formulaire avec les informations à saisir pour le service. Le formulaire permet également de choisir la ou les communes qui vont proposer ce nouveau service. Pensez à proposer des valeurs par défaut pour chaque champ de saisie, afin de ne pas avoir à remplir tous les champs du formulaire lors des tests. Quand c'est nécessaire, on utilisera des listes pour sélectionner une valeur (pour les communes). Le système effectuera des vérifications avant d'insérer en base (e.g., pas de service existant avec le même intitulé) et affiche des messages pertinents pour faciliter la correction des erreurs.

Développez ensuite une fonctionnalité permettant d'afficher tous les services avec leurs informations.

Lors de la soutenance, il est fort probable que les enseignant-e-s testent votre application en saisissant des valeurs absurdes susceptibles de déclencher des erreurs dans votre application. Alors essayez de penser au pire !

6 Fonctionnalité 4 : générer une liste de périodes d'essai

Afin de faire connaître votre application aux différentes communes, vous proposez des périodes d'essai, i.e., d'offrir gratuitement un certain nombre de services à certaines communes. Par exemple, l'application pourrait proposer la période d'essai suivante : les services *Signalement*, *Restauration* et *Élection* à la commune de Villeurbanne pour une durée de 4 mois. La page doit donc générer une liste de périodes d'essai selon différents paramètres.

Certains paramètres sont fixes :

- Le nombre de communes est sélectionné aléatoirement entre 5 et 20 ;
- La durée de chaque période d'essai est sélectionnée aléatoirement parmi 3, 4 ou 6 mois ;
- Le nombre de services de chaque période d'essai est sélectionné aléatoirement entre 3 et 5 (inclus).

La page comporte un formulaire pour renseigner les paramètres configurables suivants :

- Département. Si un département est choisi, les communes sont sélectionnées aléatoirement dans ce département ;
- Nombre de mois maximum. Si une valeur est saisie, alors le total des durées de périodes d'essai générées ne peut excéder cette valeur ;
- Nombre de kilomètres. Si une valeur est saisie, alors les communes sont sélectionnées dans un périmètre (circulaire) dont le diamètre est égal à la valeur saisie. Cela permet d'obtenir des communes géographiquement proches et pas dispersées dans la région. MariaDB inclut des [fonctions géographiques](#) comme la distance sphérique entre 2 points.

Quand la liste est générée, les informations sur les périodes d'essai générées sont affichées sur la page. Selon les données présentes et l'algorithme développé, il peut arriver que toutes les contraintes ne soient pas satisfaites. Dans ce cas, il serait bien d'en informer l'utilisateur·ice. Enfin, ces données générées seront stockées en base.

7 Préparation des livrables

Deux livrables sont à rendre **le jour de soutenance de votre projet, avant 23h59, sur Tomuss** :

- Une archive de votre site web, en **zip** ou **rar** (colonne `archive_projet`), qui contient a-minima :
 - le répertoire de votre site (code commenté et indenté, respectant l'arborescence de la section 2) ;

- un fichier `.txt` ou `.sql` avec le script SQL **”exécutable”** de création de votre base de données.
- L’URL de votre site, sur le serveur `bdw.univ-lyon1.fr` (colonne `url_projet`) :
 - Elle doit être complète (login et répertoire(s), voire fichier index précisé si nécessaire) et elle doit pointer vers la page d’accueil de votre site (e.g., `http://bdw.univ-lyon1.fr/p1234567/mon_site/`). Prenez le temps de la vérifier une fois qu’elle est saisie dans Tomuss ;
 - Votre site doit déjà avoir des instances stockées en base **en nombre suffisant** pour simplifier les tests.

8 Soutenance

La soutenance dure 10 minutes par binôme et se décompose en deux parties : une démonstration du site (5 minutes) et une séance de questions (5 minutes). Les conditions suivantes s’appliquent :

- Pour la partie démonstration, respectez le temps ! Vous serez interrompus au bout des 5 minutes, et donc pénalisés ;
- Les deux membres du binôme doivent parler lors de la démonstration. Entraînez-vous ;
- Vous présentez soit sur votre machine, soit sur une machine de la fac. Prévoyez d’arriver quelques minutes en avance devant la salle, et quand on vous fait entrer, préparez-vous pour la présentation et les questions (i.e., lancement du site dans le navigateur, ouverture des fichiers source) ;
- Vous devez répondre à plusieurs questions pendant les 5 minutes. Donc donnez des **réponses claires et concises**. Si vous ne savez pas répondre à une question, dites-le pour ne pas perdre de temps ;
- Si vous arrivez en retard à votre soutenance, vous passez en fin de séance... ou pas du tout selon les disponibilités de votre jury.

Concours (optionnel) : Pour vous détendre (en cas de blocage sur un bug, d’engueulade avec votre binôme, etc.), deux concours sont organisés, celui du **meilleur nom de site** et celui du **plus beau logo**. Les deux binômes gagnants remporteront un paquet de chamallows. Indiquez si vous participez au concours lors de la soutenance. Soyez créatifs/ves !

