

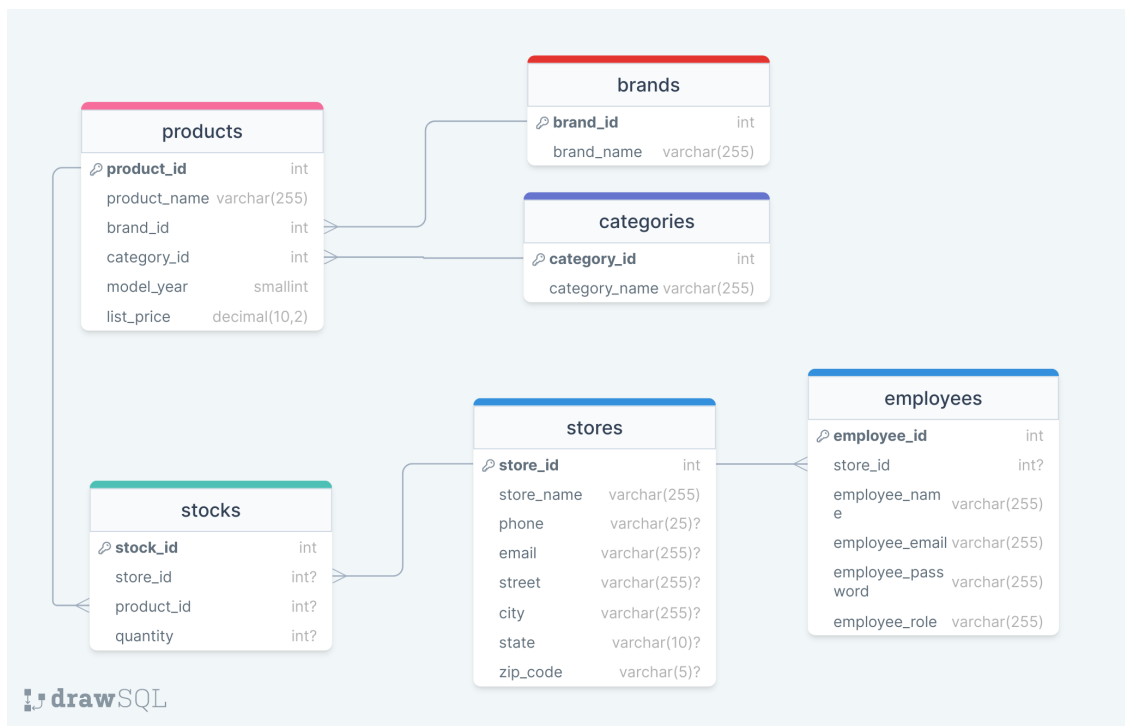
1 Projet Tuteuré de la SAÉ (Situation d'Apprentissage et d'Évaluation) 401D « Développer pour le Web »

L'objectif de cette SAÉ est de mobiliser les connaissances que vous avez acquises dans les ressources R401(Anglais), R403 (Design d'expérience), R404 (Expression, communication), R406D (Développement front), R407D (Développement back) et R408D (Déploiement de services) afin de développer un service Web.

Le service web que vous créerez sera à réaliser durant les séances dédiées à la SAÉ. Les heures de TD et TP sont encadrées par un enseignant. Celles de PT (Projet Tuteuré) sont en autonomie, un enseignant pourra néanmoins passer régulièrement vous voir pour faire le point et vous aider. Le cahier des charges du site décrit ce que vous devez respecter pour réaliser le site (qui est à réaliser individuellement).

2 Cahier des charges

Nous allons utiliser une base de données dont le schéma relationnel est présenté ci-dessous. Les clés primaires sont en gras et les clés étrangères sont mentionnées via les relations entre tables.



Cette base de données représente les stocks de trois magasins de vente de vélos situés aux États-Unis. Elle est constituée de 4 tables. Le fichier SQL « SAE401.sql » présent sur ecampus contient un script de création et de peuplement des tables. La base est structurée comme suit :

- La table *categories* contient les différentes catégories de vélos,
- La table *brands* contient les différentes marques de vélos,
- La table *stores* contient les différents modèles de vélos,
- La table *products* contient une description des vélos vendus dans les magasins. Ils sont relatifs à une marque (issue de la table *brands*), et une catégorie (issue de la table *brands*).
- La table *stocks* contient une description des stocks de vélos vendus dans les magasins. Ils sont relatifs à un modèle de vélo (issu de la table *products*), et une boutique (issue de la table *stores*),
- La table *employees* contient une description des employés des magasins. Ils sont relatifs à une boutique (issue de la table *stores*).

L'objectif du projet est de concevoir l'API d'un service web REST permettant de consulter et modifier cette base de données. Ce service sera utilisé par un site web qui aura une partie publique (le front office pour les clients) et une partie privée (le back office pour les employés des boutiques). Cette dernière sera soumise à authentification (tout employé devra se connecter pour accéder au back office).

2.1 Exigences fonctionnelles

2.1.1 Le web service

Votre webservice sera disponible sur votre site de dev, de manière publique (c'est-à-dire directement via son URL sans besoin de spécifier son etupass). Il devra respecter les normes d'une API REST. Il permettra d'interroger la base de données pour des actions de CRUD. Par exemple, pour une consultation de la base :

<https://dev-login.users.info.unicaen.fr/bikestores/stores> en GET doit retourner du JSON contenant la liste des magasins.

<https://dev-login.users.info.unicaen.fr/bikestores/products/10> en GET doit retourner du JSON contenant la description du produit numéro 10.

Les requêtes en POST permettront des insertions dans la base de données, celles en PUT des mises à jour et celles en DELETE des effacements. Pour ces requêtes de modification de la base de données, l'accès à votre API sera soumis à l'utilisation d'une clé d'accès unique qui devra être fournie pour chaque interrogation autre qu'en GET.

2.1.2 Le site web

Le site web sera un outil à destination des clients et des employés. La page d'accueil du site contiendra un formulaire de connexion qui permettra aux employés de se connecter avec leur email comme identifiant et leur mot de passe.

Si c'est un client qui consulte le site, il pourra parcourir la liste des produits. Des filtres lui permettront de consulter uniquement les produits d'une marque, d'une catégorie, d'une année donnée, et dans une fourchette de prix.

Les employés sont dans trois catégories : *employee*, *chief*, ou *it*. Ils sont affectés à un store.

Si c'est un employé de la catégorie *employee* qui consulte le site, il aura les mêmes fonctionnalités qu'un client avec en plus la possibilité

- d'ajouter/de modifier/d'effacer des marques, catégories, magasins, produits et stocks dans le magasin dans lequel il travaille
- de modifier ses informations de connexions et de se déconnecter

Un employé de la catégorie *chief* aura des droits supplémentaires. Il pourra aussi

- Consulter les employés de son magasin,
- Ajouter de nouveaux employés dans le magasin dans lequel il travaille (uniquement).

Un employé de la catégorie *it* pourra aussi

- Consulter les employés dans tous les magasins,
- Ajouter de nouveaux employés (de types *employee* ou *chief*) dans tous les magasins.

Les différentes actions qu'un utilisateur peut effectuer seront proposées dans un menu situé en haut de la page. Ce menu peut éventuellement contenir des sous-menus.

2.2 Exigences Techniques

Vous n'utiliserez pas de CMS, mais uniquement les langages de développement Web que vous avez appris en MMI : HTML, CSS, JS, SQL et PHP. L'usage de frameworks spécifiques de développement coté client et serveur est imposé, ainsi que de génération de documentation, ils sont précisés dans la suite.

Concernant le web service,

- Il sera conforme aux normes d'une API REST,
- Il sera hébergé sur votre site de dev à l'adresse suivante <https://dev-login.users.info.unicaen.fr/bikestores/>
- La page d'accueil du webservice affichera sa documentation.
- Votre site de dev étant soumis à une authentification, vous devrez la désactiver via le fichier « .htaccess ». Ce dernier contiendra également les règles de réécriture d'URL à appliquer.
- Il sera conçu en MVC avec le langage PHP, en exploitant le framework Doctrine.
- Il renverra du JSON pour toute interrogation, y compris pour préciser une erreur.
- La documentation du web service sera conçue avec l'outil swagger (<https://swagger.io/>) et son éditeur en ligne accessible à <https://editor.swagger.io>. Elle sera écrite en Anglais.
- Votre code PHP sera documenté en Anglais et sa documentation sera générée avec PHPDoc (<https://phpdoc.org>).
- Vous aurez une arborescence précise contenant vos codes selon le modèle MVC,
- La clé d'accès à votre API (pour les opérations autres que de consultation) sera « e8f1997c763 ».

Concernant le site web :

- Vous respecterez les normes du W3C,
- Vous utiliserez le framework bootstrap (<https://getbootstrap.com>) pour le style CSS de votre site (l'aspect responsive sera donc géré par bootstrap, vous n'avez pas à vous en préoccuper).
- L'interrogation de la base de données se fera obligatoirement via le web service pour l'affichage et l'édition des enregistrements des tables,
- La page d'accueil du site contiendra une carte géographique (une OpenStreet Map, voir <https://www.openstreetmap.fr>) permettant de visualiser la localisation des magasins. Lors de la consultation du site, l'utilisateur sera également géolocalisé sur la carte à l'aide d'un marqueur. Pour géolocaliser un utilisateur, vous pouvez récupérer son IP (avec par exemple <https://www.bigdatacloud.com/free-api/public-ip-address-api>) puis ses coordonnées GPS à partir de celle-ci (avec par exemple <https://apibundle.io>).
- L'authentification utilisera un système de cookies coté client.
- Pour l'interrogation du webservice, vous pouvez le faire en Vanilla JS ou en JQuery (<https://jquery.com>). Votre code JS sera commenté en Anglais à l'aide de JSDoc (<https://jsdoc.app>),
- Votre site sera intégralement en Anglais et contiendra des mentions légales spécifiques pour une utilisation du site web aux états-unis,
- Votre site web sera hébergé chez un hébergeur gratuit : <https://www.infinityfree.com> ou <https://www.alwaysdata.com/fr>.
- Vous aurez une arborescence précise pour stocker vos fichiers avec des répertoires dédiés (images, scripts JS/CSS/HTML).

Les codes de votre webservice et de votre site web seront disponibles sur un dépôt github.

3 Rendu final

Sur ecampus (uniquement, rien par email) vous rendrez un fichier texte contenant les URLs de vos webservices, site web, et dépôts github (un pour l'API et un autre pour le webservice). Votre dépôt github de l'API contiendra votre fichier « .htaccess ».

Vous remettrez également un manuel d'utilisation à destination d'une part du grand public et d'autre part des employés. Il n'est donc pas à destination de développeurs. Ce manuel aura donc deux parties. La première partie décrira comment le site web peut être utilisé pour sa partie publique. La seconde partie décrira comment le site web peut être utilisé pour sa partie privée. Ce manuel d'utilisation sera rédigé en Français.

4 Notation

Votre note sera une moyenne pondérée de notes relativement aux ressources en Anglais et Expression et communication (coefficient 0.15), Développement front (coefficient 0.35), Développement back et Déploiement de services (coefficient 0.5).