

# Dossier de définition des besoins pour l'hébergement de l'application SAE 345

<b>1. Description de l'application développée.....</b>	<b>3</b>
1.1. Description non technique de l'application.....	3
1.2. Description technique de l'application.....	3
<b>2. Description des besoins.....</b>	<b>4</b>
2.1. Besoins contraints.....	4
2.2. Besoins avec choix ouvert.....	5
Nom de domaine.....	5
2.3. Sécurisation - Utilisation de TLS.....	5

# 1. Description de l'application développée

## 1.1. Description non technique de l'application

L'application H2eau est un site web de e-commerce qui permet aux utilisateurs de rechercher et d'acheter des articles (plus précisément de l'eau). Le site comprend également des fonctionnalités telles que la possibilité de commenter et de noter des articles, de définir une ou plusieurs adresses pour l'expédition, de sélectionner des articles dans une liste d'envies, d'avoir un historique des derniers articles consultés et de filtrer les articles par type.

Les utilisateurs peuvent ajouter des articles à leur panier, supprimer des articles, modifier la quantité d'un article ajouté au panier, valider le panier, afficher le prix total de chaque commande et afficher le stock restant d'un article. Le back-office permet au vendeur de gérer les commandes en affichant toutes les commandes et en validant leur traitement, en affichant le détail d'une commande, en modifiant le stock d'un article et en visualisant les données sur les stocks pour l'administrateur.

Le site prend également en charge la gestion des commentaires et des notes, la gestion des adresses d'expédition et de livraison ainsi que la gestion des stocks pour les articles ou services de catégories différentes. Les utilisateurs peuvent ajouter des commentaires sur un article qu'ils ont acheté et ne peuvent saisir que trois commentaires par article. Ils ne peuvent saisir qu'une note par article ou la modifier. Les administrateurs peuvent voir s'il y a de nouveaux commentaires sur un article, supprimer un commentaire ou y répondre.

Les utilisateurs peuvent créer un maximum de quatre adresses et peuvent les modifier. Lors de la création d'une commande, l'utilisateur doit créer une adresse si elle n'existe pas, sinon la dernière adresse utilisée pour la facturation est proposée ou la dernière adresse utilisée pour la livraison est proposée dans la liste déroulante des adresses. Enfin, le site est capable d'afficher le nombre de commandes et la région de destination pour les ventes ainsi que le nombre de notes et la note moyenne par catégorie pour les commentaires.

Le stock dépend d'au moins une propriété d'un article (couleur et/ou taille), et le stock n'est pas affiché dans la page principale si l'article existe en plusieurs propriétés différentes. Le stock se décrémente lorsque l'on ajoute un ou plusieurs articles dans le panier et se réincrémente lorsque l'on supprime des articles du panier. Lors de l'ajout d'un article dans le panier, le client passe par une interface supplémentaire pour saisir la propriété, et l'administrateur peut ajouter des interfaces pour compléter le stock des articles en fonction de la propriété.

## 1.2. Description technique de l'application

L'architecture logicielle de l'application est un modèle MVC (Model-View-Controller) de manière légèrement différente de l'approche traditionnelle, car la partie View et Model ne sont pas vraiment séparées. Ce modèle permet de séparer les différentes parties fonctionnelles de l'application. Le modèle peut être représenté par la base de données MariaDB qui stocke les données, tandis que les vues sont représentées par les fichiers

HTML du dossier templates qui permettent d'afficher les informations. Le contrôleur est représenté par les fichiers Python du dossier controllers qui gèrent la logique applicative, effectuent des opérations sur la base de données et interagissent avec les vues. Flask est utilisé pour gérer les requêtes HTTP et les réponses de l'application, tandis que Jinja est utilisé comme moteur de templates pour générer les pages HTML dynamiquement en injectant des variables dans les fichiers HTML.

Les langages, bibliothèques et outils utilisés pour développer l'application incluent Python, Flask, Jinja, HTML, CSS, MariaDB et Bootstrap. Python est utilisé pour écrire la logique applicative, Flask est utilisé pour gérer les requêtes HTTP et les réponses de l'application, Jinja est utilisé comme moteur de templates pour générer les pages HTML dynamiquement en injectant des variables dans les fichiers HTML, HTML et CSS sont utilisés pour définir la structure et le style des pages web, MariaDB est utilisé pour stocker les données de l'application, tandis que Bootstrap est utilisé pour faciliter la mise en page et la création de composants web réutilisables.

L'arborescence des fichiers de l'application est structurée de manière claire et organisée. Le dossier controllers contient des fichiers Python qui gèrent la logique applicative de l'application, le dossier static contient les fichiers statiques tels que les images, les styles et les scripts, tandis que le dossier templates contient les fichiers HTML pour les différentes vues du site. Les fichiers HTML sont organisés en sous-dossiers tels que admin, auth et client, qui contiennent des sous-dossiers supplémentaires pour des fonctionnalités spécifiques telles que les articles et les commandes. Cela permet une organisation logique des fichiers pour faciliter la maintenance et l'extension de l'application à l'avenir.

## 2. Description des besoins

### 2.1. Besoins constraints

Pour que l'application H2Eau puisse fonctionner correctement en production, les éléments suivants doivent être installés sur l'environnement de production :

- Python 3.7 ou supérieur
- Flask 2.0.1 ou supérieur
- Jinja2 3.0.1 ou supérieur
- HTML5
- CSS3
- MariaDB 10.5.10 ou supérieur
- Bootstrap 5.1.0 ou supérieur

Il est recommandé d'avoir une version de MariaDB supérieure ou égale à 10.5.10 pour bénéficier de toutes les fonctionnalités de l'application.

L'espace disque nécessaire pour l'application dépendra de la quantité de données stockées sur la base de données et du nombre de fichiers statiques stockés sur le serveur.

Cependant, une estimation approximative est de 100 Mo pour l'application elle-même, en plus de l'espace nécessaire pour stocker les données et les fichiers statiques.

Il est également recommandé de créer un compte utilisateur distinct pour l'application, avec les permissions nécessaires pour administrer l'environnement de production à distance.

Pour permettre au client de déployer facilement une nouvelle version de l'application en production, il est recommandé d'utiliser un outil de gestion de versions tel que Git, et d'utiliser un processus de déploiement continu (CI/CD) pour automatiser le processus de déploiement.

## 2.2. Besoins avec choix ouvert

Les choix avec un certain degré de flexibilité pour l'environnement de production sont :

- Serveur web : Apache 2.4, Nginx 1.21.1 ou supérieur
- Serveur de messagerie : Postfix 3.5.10 ou supérieur, Exim 4.94 ou supérieur

Le choix du serveur web dépendra de la préférence et de l'expertise du client. Apache et Nginx sont tous deux de bonnes options, largement utilisées et ayant une forte communauté de support.

Quant au serveur de messagerie, Postfix et Exim sont des options courantes et fiables, avec des fonctionnalités similaires.

### Nom de domaine

Le nom de domaine pour le site web H2Eau n'a pas été spécifié. Voici cinq propositions de noms de domaine avec l'extension .fr :

1. [www.h2eau.fr](http://www.h2eau.fr)
2. [www.h2eau-ecommerce.fr](http://www.h2eau-ecommerce.fr)
3. [www.h2eau-boutique.fr](http://www.h2eau-boutique.fr)
4. [www.h2eau-shop.fr](http://www.h2eau-shop.fr)
5. [www.h2eau-store.fr](http://www.h2eau-store.fr)

Le coût d'achat d'un nom de domaine peut varier en fonction de plusieurs facteurs, tels que l'extension, la longueur du nom de domaine, la popularité de l'extension et la disponibilité du nom de domaine. Le coût moyen d'un nom de domaine en France est d'environ 10€ à 20€ par an.

## 2.3. Sécurisation - Utilisation de TLS

Il est primordial de sécuriser un site web pour protéger les données sensibles de ses utilisateurs, éviter le vol d'informations confidentielles et prévenir les attaques de pirates informatiques. Les principales raisons de sécuriser un site web sont les suivantes :

- Protéger les données personnelles des utilisateurs, telles que les noms, adresses, numéros de carte de crédit, etc.
- Éviter les attaques de pirates informatiques qui peuvent causer des dommages irréparables à l'application, notamment la perte de données.
- Garantir la confidentialité et l'intégrité des données échangées entre le navigateur de l'utilisateur et le serveur web.

Pour assurer la sécurité de l'application e-commerce H2Eau, nous recommandons l'utilisation de TLS (Transport Layer Security), un protocole de sécurité qui permet de chiffrer les données échangées entre le navigateur de l'utilisateur et le serveur web, assurant ainsi leur confidentialité et leur intégrité. Ce protocole empêche les attaques de type "man-in-the-middle".

Pour mettre en place cette protection, il est nécessaire d'avoir un certificat TLS valide, émis par une autorité de certification reconnue. Ce certificat est utilisé pour chiffrer les données échangées entre le navigateur de l'utilisateur et le serveur web. Le certificat est associé à une clé privée qui permet de déchiffrer les données chiffrées à l'aide d'une clé publique associée au certificat.

Les versions préconisées pour TLS sont TLS 1.2 et TLS 1.3, qui offrent une meilleure sécurité que les versions précédentes. Il est important de noter que les versions plus anciennes de TLS, comme la version 1.0 et 1.1, présentent des vulnérabilités connues et ne sont donc plus considérées comme sécurisées.