ECF en cours de formation
Bachelor developpeur Web 22-24

Documentation technique

Réflexions initiales technologiques sur le sujet

Le maquettage sera réalisé avec AdobeXD car d'un point de vue pratique c'est le logiciel que je maîtrise le mieux. Le fait qu'il soit payant ne doit pas être un obstacle, le coût horaire d'un développeur finance 6 mois d'abonnement ! Mieux vaut donc un outil performant et maîtrisé que de perdre du temps !

Concernant le développement je vais utiliser l'IDE VS code tout a fait adapté au développement WEB (HTML,CSS,PHP).

Le backend sera réalisé en PHP qui est parfaitement adapté pour un site WEB dynamique,

Concernant l'accès depuis un mobile, j'ai fait le choix de construire le site WEB avec un « responsive design », ce qui permettra l'accès à l'application depuis la plupart des modèles de terminaux (ordinateur, tablette, téléphone Android ou iOS …) pour peu qu'ils disposent d'un browser.

Pour l'accès aux données j'utiliserais PDO dans PHP qui permet une grande portabilité au travers de l'univers des SGBDR SQL. Le développement étant fait sur XAMPP, il est en effet important de minimiser l'effort d'adaptation et de retest sur l'environnement d'exploitation.

Le front est évidemment en HTML et CSS, avec Bootstrap pour utiliser un menu responsive sans avoir besoin de le redévelopper.

Pour la « responsivité » de l'application elle même, j'ai décidé de construire les pages sous formes de boites assemblées dans un container flex. Ceci permet une adaptation très simple du positionnement relatif de ces boites en fonction de l'orientation de l'écran.

J'utiliserais aussi XAMPP en tant que plate forme de développement et de test, avec sa BDD relationnelle et PHPMyAdmin qui facilite la mise au point des requêtes les plus complexes.

Configuration de votre environnement de travail

Le développement sera effectué sur un PC sous Windows 10.

L'environnement de travail est composé des outils suivants :

Adobe XD : logiciel commercial destiné au maquettage d'applications. Il est installé sur mon PC, et permet un stockage sur le cloud ainsi que le partage des mockups pour remarques ou validation.

Vscode : Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. C'est une aide précieuse pour quiconque ne maîtrise pas parfaitement toutes les propriétés CSS !

XAMPP : est un serveur Apache local permettant la mise au point d'applications mySQL/PHP avant mise en ligne. C'est un logiciel open source.

Concernant la gestion de projet, j'utiliserais JIRA, logiciel en ligne qui permet notamment la création et la gestion de tableau Kanban. Il reste gratuit pour de petits projets comme celui qui fait l'objet de mon ECF.

Enfin, j'utiliserais GitHub pour le versionning et l'archivage, même si le fait d'être seul a travailler sur ce projet en réduit l'intérêt.

Un déploiement en environnement réel est également effectué sur l'hébergeur Planet Hoster. Malheureusement, la version gratuite souffre d'un choix de nom de domaine assez peu explicite. Néanmoins, cet environnement permet de disposer de PHP, d'un SGBDR mariaDB et de PhpMyAdmin comme sur l'environnement de développement XAMPP.

J'ai quand même rencontré 2 problèmes (de débutant!) :

1) sur XAMPP, les noms de tables supportent les majuscules, mais pas sur l'environnement de production, j'ai donc dû tout remettre en minuscules, heureusement que toutes les requêtes sont dans un même fichier (bdd.php)
2) un problème de gestion des caractères accentués se produisait sur les données lues depuis la BDD de production. J'ai réglé le problème en ajoutant `;charset=utf8mb4` dans le DSN de production.

Le site est accessible avec l'url suivante :

https://ecf-fv.go.yj.fr/acceuil.php

ou plus simplement https://ecf-fv.go.yj.fr

car j'ai créé un fichier index.php qui redirige sur la page d'acceuil (et qui évite au passage de pouvoir faire une lecture des fichiers sources du répertoire.

(désolé pour le nom, mais le site n'est pas destiné à vivre longtemps!)

Modèle conceptuel de données (MCD)

La lecture du cahier des charges permet le lister l'ensemble des informations qui devront être gérées par l'application

tout d'abord les données utilisateur :

- nom
- prénom
- société
- email
- mdp
- conseillé clientèle (*)
- matricule (*)
- rôle

l'email devra être unique dans la BDD

(*) ces informations seront optionnelles selon le rôle de l'utilisateur (conseillé client, utilisateur, administrateur

les utilisateurs sont associés à un conseiller clientèle, il faudra donc gérer cette association

bien évidemment, les mots de passe ne seront pas stockés dans la base de données. On y stockera leur empreinte. On stockera également l'empreinte du token de session et sa date de fin de validité (qui sera initialisée à la connexion à t+2 heures, et qui sera remie a jour à t+2 heures à chaque nouvelle page lue lorsqu'on est connecté. Ainsi, le token s'éteindra 2 heures après la dernière activité de l'utilisateur sur le site).

Il faudra également disposer d'une information permettant de savoir si le mot de passe doit être changé lors de la prochaine connexion.

concernant les produits :

- ils sont classés par des catégories dont le nom est unique (il pourra donc servir de clef)

chaque produit est caractérisé par

- un libellé
- une image
- une description
- un prix
- un volume en stock
- et est rattaché à une catégorie

les produits peuvent être commandés au sein de commandes avec des quantités et prix associés. Même si ca n'était pas demandé, on ajoutera une date d'émission des commandes.

Le volume en stock n'est pas géré par l'application, je l'ai ajouté pour mon exemple de transaction SQL.

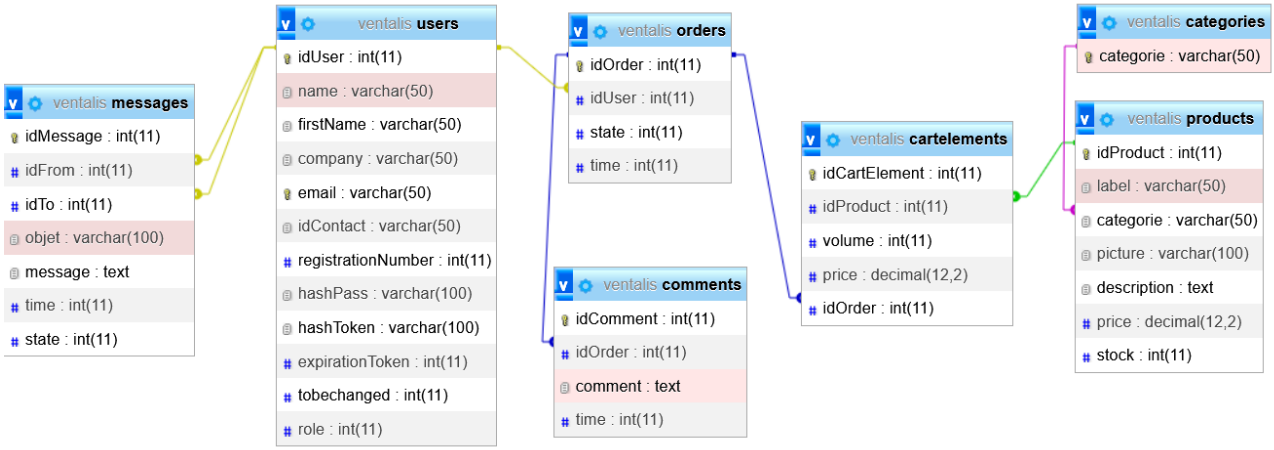
enfin les conseillers peuvent mettre des commentaires lors du traitement des commandes. On ajoutera là aussi une date d'ajout du commentaire.

Et finalement, il reste a gérer la messagerie interne. Les informations nécessaires sont :

- l'identité de l’émetteur
- l'identité du destinataire
- la date d'envoi du message
- son objet
- le texte du message
- son etat (non géré par l'application a ce stade, car pas dans le cahier des charges (ca pourra faire l'objet d'un avenant, conformément aux us et coutumes de la profession), mais ca permettrait de marquer les messages comme lus et de les effacer)

Le modèle conceptuel de donnée peut donc être schématisé selon le diagramme relationnel suivant (un grand merci a phpmyadmin!!!):

Une fois implémenté, PhpMyAdmin nous donne le diagramme suivant :



Cette structure permet d'effectuer l'ensemble des taches prévues au cahier des charges.

Par exemple, rechercher le conseiller clientèle (role=2) ayant le moins de client :

```
      $statement1 = $pdo->prepare('SELECT users.idUser,count(utilisateurs.idUser) as C from users
                    left join users as utilisateurs on users.idUser=utilisateurs.idContact
                    where users.role=2
                    group by users.idUser
                    order by C asc Limit 1');
      $statement1->execute();
      $result= $statement1->fetch(PDO::FETCH_ASSOC);;
      $idcontact=$result["idUser"];
```

ou rechercher les commandes d'un utilisateur :

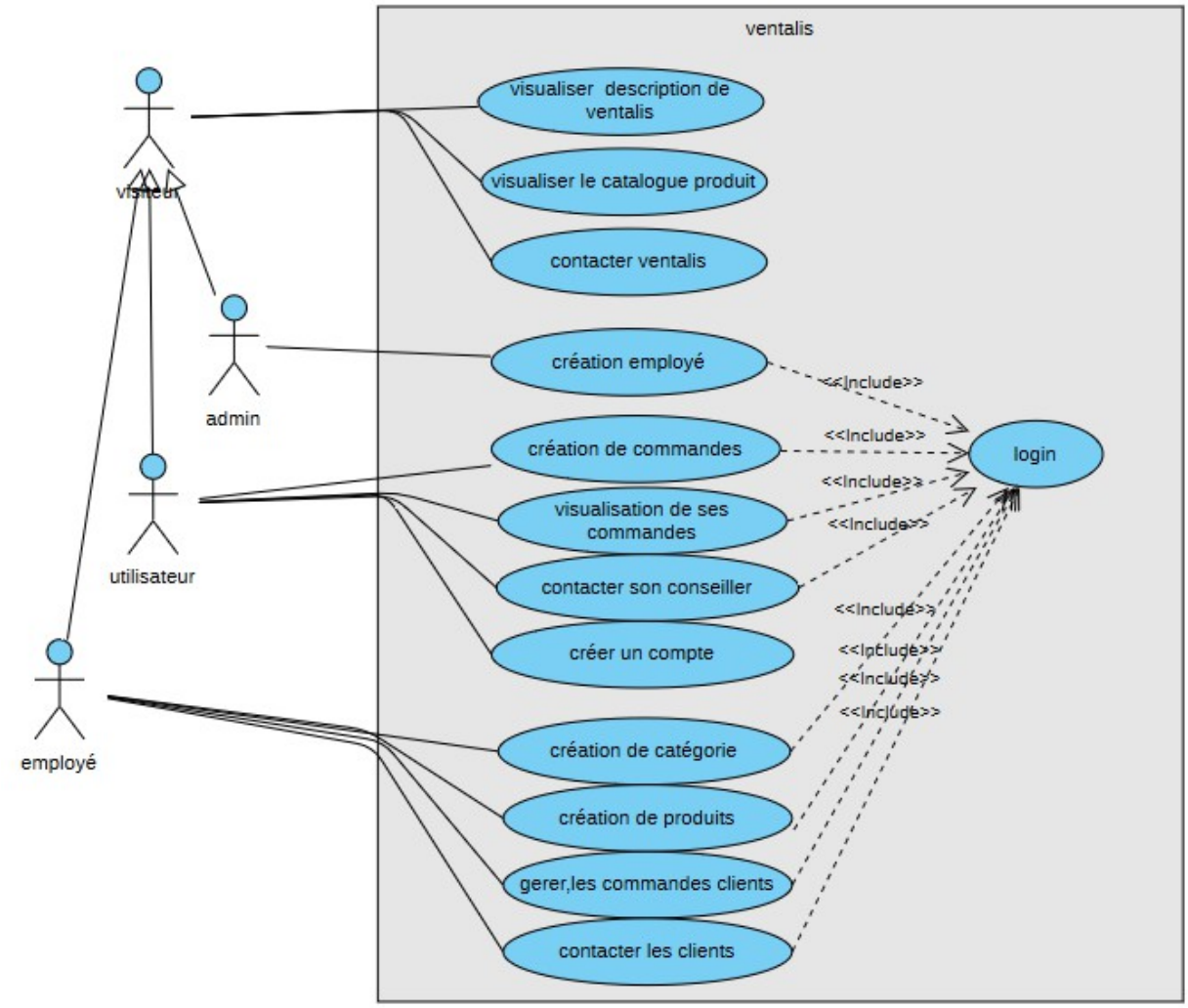
```
    $checkorder = $pdo->prepare('SELECT users.iduser,orders.idorder,orders.state,cartelements.idproduct,cartelements.volume,cartelements.price,products.label
                  FROM users
                  JOIN orders ON users.iduser=orders.iduser
                  JOIN cartelements ON orders.idorder=cartelements.idorder
                  JOIN products ON cartelements.idproduct=products.idproduct
                  WHERE hashtoken=:hashtoken
                  ORDER BY orders.idorder DESC');
```

l'utilisateur sera retrouvé par le hash de son token de connexion.

L'ensemble des fonctions d'accès à la base de donnée se trouve dans le fichier bdd.php sous la racine du site.

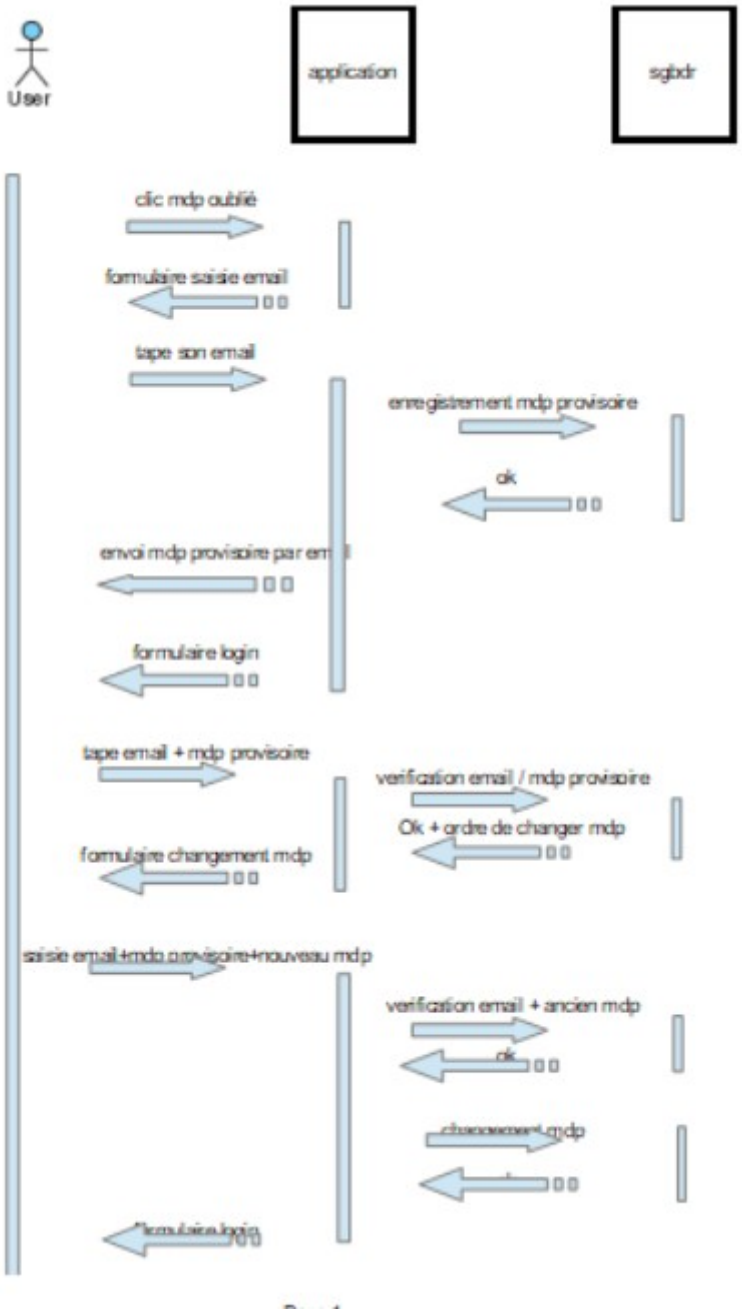
Diagramme d’utilisation, diagramme de séquence

Pour le diagramme de cas d'utilisation, j'ai utilisé l'outil en ligne visual paradigm. C'est plutot assez intuitif.



Pour illustrer au mieux les diagrammes de séquence, j'ai choisi de représenter la procédure « mot de passe oublié », car c'est lde mon point de vue a procédure la plus complexe de l'application.

Nous aurons 3 acteurs, l'utilisateur, l'application et le base de donnée. Le diagramme ci dessous illustre la procédure dans laquelle il n'y a aucune erreur de saisie.



Explication de votre plan de test

Ces explications sont dans le document principal copiearendre_ecf_finale_dirrecte.pdf

Transaction SQL

Pour illustrer ce qu'est une transaction SQL, j'ai choisi d'écrire une transaction utilisant les données de la bdd de l'application ventalis consistant à vértifier le stock des produits dans le panier, puis, si le stock est suffisant, d'enregistrer la commande.

Ce fichier SQL se trouve dans le git dans le répertoire sql et se nomme « exemple transaction.sql »

En fait ca n'était pas si simple que ca, car dans l'environnement utilisé (xampp mariaDB, phpmyadmin), je n'ai pas réussi a faire fonctionner un COMMIT dans une procédure stockée, ni a utiliser des instructions IF en dehors d'une procédure stockée.
Mes recherches sur internet ne m'ont pas permis de trouver de solution, j'ai juste appris que ces comportements étaient assez diverses en fonction du SGBDR utilisé, et que de ce point de vue, les opensources n'étaient pas les mieux disants …

mais il n'y a pas de problème sans solution. J'ai donc créé une procédure stockée qui fait les différents tests puis passe les ordres de mise a jour des tables, et j'ai encadré l'appel de la procédure dans un START TRANSACTION / COMMIT.