Documentation

Date de création : 23.02.2024  
Version 3 du 22.02.2024

Orlando Espin

|  |  |  |
| --- | --- | --- |
|  |  |  |

Table des matières

[1 Introduction 2](#_Toc125302133)

[2 Domaine de la sécurité Erreur ! Signet non défini.](#_Toc125302134)

[3 Documentation Erreur ! Signet non défini.](#_Toc125302135)

[3.1 Structure d’une documentation Erreur ! Signet non défini.](#_Toc125302136)

[3.2 Etude de cas Erreur ! Signet non défini.](#_Toc125302137)

[3.2.1 Miele **Erreur ! Signet non défini.**](#_Toc125302138)

[3.2.2 Singer **Erreur ! Signet non défini.**](#_Toc125302139)

[3.3 Présentation Erreur ! Signet non défini.](#_Toc125302140)

[3.3.1 Informations utiles et pertinentes **Erreur ! Signet non défini.**](#_Toc125302141)

[3.3.2 Moyens possibles **Erreur ! Signet non défini.**](#_Toc125302142)

[3.4 Prestation Erreur ! Signet non défini.](#_Toc125302143)

[3.4.1 Cycle à 5 étapes **Erreur ! Signet non défini.**](#_Toc125302144)

[a) Mise au courant **Erreur ! Signet non défini.**](#_Toc125302145)

[b) Démonstration **Erreur ! Signet non défini.**](#_Toc125302146)

[c) Mise en situation **Erreur ! Signet non défini.**](#_Toc125302147)

[d) Evaluation du participant **Erreur ! Signet non défini.**](#_Toc125302148)

[e) Evaluation de la prestation **Erreur ! Signet non défini.**](#_Toc125302149)

[4 Le projet Erreur ! Signet non défini.](#_Toc125302150)

[4.1 Réalisation Erreur ! Signet non défini.](#_Toc125302151)

[4.1.1 Présentation personnelle **Erreur ! Signet non défini.**](#_Toc125302152)

[4.1.2 Mise au courant **Erreur ! Signet non défini.**](#_Toc125302153)

[4.1.3 Démonstration **Erreur ! Signet non défini.**](#_Toc125302154)

[4.1.4 Mise en situation **Erreur ! Signet non défini.**](#_Toc125302155)

[4.1.5 Evaluation du participant **Erreur ! Signet non défini.**](#_Toc125302156)

[4.2 Bilan post projet Erreur ! Signet non défini.](#_Toc125302157)

[5 Projet des camarades Erreur ! Signet non défini.](#_Toc125302158)

[5.1 Pour chaque projet, identification des plus et moins Erreur ! Signet non défini.](#_Toc125302159)

[5.1.1 Projet du groupe PVN **Erreur ! Signet non défini.**](#_Toc125302160)

[5.1.2 Projet du groupe East Blue **Erreur ! Signet non défini.**](#_Toc125302161)

[6 Conclusion Erreur ! Signet non défini.](#_Toc125302162)

[6.1 Ce que j’ai aimé Erreur ! Signet non défini.](#_Toc125302163)

[6.2 Ce que j’ai moins aimé Erreur ! Signet non défini.](#_Toc125302164)

# Introduction

Lors de ce projet je vais développer le backend ainsi que le frontend de la partie gestion des utilisateurs pour mon client Claude Meylan. Nous travaillons à deux sur ce site web, mais nous avons divisé les tâches avec qu’il soit en rapport avec ce module. Donc tandis que moi je gérerai les utilisateurs ainsi que toutes leurs actions possible, Adrien gérera le CMS et l’affichage des données sur le site.

# Analyse

## Présentation du projet

Le site web que nous développons est une boutique en ligne dans l’achat de montre de luxe basé à la vallée de joux. Mon rôle sera de faire en sorte que les visiteurs de notre site puissent créer un compte ou se connecter si c’est déjà fait, qu’ils aient accès ou non à certaines pages selon le statuts de leur compte. Ils pourront aussi s’inscrire à une newsletter ainsi que s’en désinscrire. Je dois faire attention à rester dans la légalité au niveau de la gestion des données, donc les clients devront pouvoir en tout temps modifier leurs données ainsi que supprimer leur compte. Je dois leur proposer une solution de suppression de compte efficace afin qu’il ne reste pas de trace d’eux. Chaque client pourra aussi commander des montres et accéder à leur historique des commandes.

## Uses Case

Une image contenant texte, diagramme, ligne, capture d’écran

Description générée automatiquement

## Diagramme activité

### Création de compte

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

### Inscription newsletter

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

## Diagramme de séquences systèmes

### Création compte

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

## Schéma ER



# Conception

## Diagrammes de classe

### Client

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

### Serveur

## Une image contenant texte, diagramme, ligne, Plan Description générée automatiquementSchéma relationnel

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Diagramme séquence interactions

Une image contenant texte, diagramme, Parallèle, Plan

Description générée automatiquement

## Conception des tests

|  |  |  |
| --- | --- | --- |
| Test | Résulat Attendu | Résultat obtenu |
| Connexion avec les bons credentials | Connexion réussie |  |
| Connexion avec les mauvais credentials | Connexion échouée |  |
| Créer un compte avec données valide | Création réussie |  |
| Créer un compte avec donnée non-valide | Création échouée | **LLe compte se crée même avec un mail non-valide** |
| Créer un compte avec un mail déjà utilisé | Création échouée |  |
| Acheter une montre | Achat réussi |  |
| Modifier des infos du compte avec des données valides | Modification réussie |  |
| Modifier des infos du compte avec des données non-valide | Modification échouée |  |
| Supprimer son compte | Suppression réussie de la db et déconnexion |  |
| S’inscrire à la newsletter en tant qu’invité | Ajout du mail à la table newsletter |  |
| S’inscrire à la newsletter en tant qu’user | Changement à true du champs newsletter dans la table user. |  |
| Se déconnecter de la newsletter en tant qu’user | Changement à false du champs newsletter dans la table user. |  |
| Se déconnecter de le newsletter en tant que visiteur | Suppression du mail dans la newsletter |  |
| Déconnexion | Suppression de la session et déconnexion coté client. |  |
| Injection SQL | Injection patch |  |
| Injection script/html | Injection patch |  |

# Implémentation

## Descente de code

### Récupération des requêtes

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    switch ($\_POST['action']) {

        case 'addMontre':

            // $imgBackground = $\_POST['imgBackground'];

            // $imgMontreShadow = $\_POST['imgMontreShadow'];

            // $imgMontre = $\_POST['imgMontre'];

            //Attention rajouter les images après ainisi que la collection id

            if (isset($\_POST['titre'], $\_POST['titreLigne2'], $\_POST['txtDescription\_fr'], $\_POST['txtDescription\_en'], $\_POST['taille'], $\_POST['epaisseur'], $\_POST['mouvement'], $\_POST['prix'], $\_POST['variante'])) {

                $titre = $\_POST['titre'];

                $titreLigne2 = $\_POST['titreLigne2'];

                $txtDescription\_fr = $\_POST['txtDescription\_fr'];

                $txtDescription\_en = $\_POST['txtDescription\_en'];

                $taille = $\_POST['taille'];

                $epaisseur = $\_POST['epaisseur'];

                $mouvement = $\_POST['mouvement'];

                $prix = $\_POST['prix'];

                $variante = $\_POST['variante'];

                //Attention ne contrôle pas que les values ne sont pas vide, a implémenter.

                $montre = new Montre($titre, $titreLigne2, $taille, $epaisseur, $mouvement, $prix, $variante);

                $montre->setTxtDescriptionFr($txtDescription\_fr);

                $montre->setTxtDescriptionEn($txtDescription\_en);

                if (isset($\_POST['collectionID'])) {

                    $collectionID = $\_POST['collectionID'];

                    $montre->setCollectionId($collectionID);

                    $ctrlCms->addMontreCollection($montre);

                } else {

                    $ctrlCms->addMontre($montre);

                }

            } else {

                http\_response\_code(400);

                break;

            }

            break;

### Ajout client

#### Main

        case 'addUser':

            if (isset($\_POST['nom'], $\_POST['prenom'], $\_POST['civilite'], $\_POST['mail'], $\_POST['newsletter'], $\_POST['motDePasse'])) {

                $nom = $\_POST['nom'];

                $prenom = $\_POST['prenom'];

                $civilite = $\_POST['civilite'];

                $mail = $\_POST['mail'];

                $newsletter = $\_POST['newsletter'];

                $motDePasse = $\_POST['motDePasse'];

                $user = new User($nom, $prenom, $civilite, $mail, $newsletter, $motDePasse);

                $ctrlUserManager->addUser($user);

                break;

            } else {

                http\_response\_code(400);

                echo 'ici';

            }

            break;

#### Ctrl

    public function addUser($user)

    {

        if (filter\_var($user->getMail(), FILTER\_VALIDATE\_EMAIL)) {

            if (!empty($user)) {

                if ($this->wrkManagerDBUser->addUser($user) == true) {

                    http\_response\_code(200);

                } else {

                    //code http pour accès non-autorisé

                    http\_response\_code(409);

                }

            } else {

            }

        }

    }

#### Wrk

    public function addUser($user)

    {

        $safeMail = htmlspecialchars($user->getMail(), ENT\_QUOTES, 'UTF-8');

        if ($this->connexion->selectSingleQuery('SELECT \* FROM t\_user WHERE mail = ?', [$safeMail]) == null) {

            if (

                $this->connexion->executeQuery('INSERT INTO `t\_user` (`nom`, `prenom`, `civilite`, `mail`, `newsletter`, `motDePasse`)

         VALUES (?,?,?,?,?,?);'

                    ,

                    [$user->getNom(), $user->getPrenom(), $user->getCivilite(), $user->getMail(), $user->getNewsletter(), password\_hash($user->getMotDePasse(), PASSWORD\_DEFAULT)]

                )

            ) {

                return true;

            } else {

                return false;

            }

        } else {

            return false;

        }

    }

### Connexion

#### Main

        case 'connectionUser':

            if (isset($\_POST['mail'], $\_POST['motDePasse'])) {

                $mail = $\_POST['mail'];

                $motDePasse = $\_POST['motDePasse'];

                $ctrlUserManager->connection($mail, $motDePasse);

            }

#### Ctrl

    public function connection($mail, $mdp)

    {

        $user = $this->wrkManagerDBUser->getUserByMail($mail);

        if ($user != false) {

            if (filter\_var($mail, FILTER\_VALIDATE\_EMAIL)) {

                if (password\_verify($mdp, $user->getMotDePasse())) {

                    $this->sessionManager->set("user", $user);

                    http\_response\_code(200);

                } else {

                    //code http pour accès non-autorisé

                    http\_response\_code(401);

                }

            } else {

                //code http pour une requête fausse

                http\_response\_code(400);

            }

        } else {

            //code http pour accès non-autorisé

            http\_response\_code(401);

        }

    }

#### Wrk

    public function getUserByMail($mail)

    {

        // Échappe les caractères spéciaux pour se protéger contre les XSS

        $safeMail = htmlspecialchars($mail, ENT\_QUOTES, 'UTF-8');

        $userArray = $this->connexion->selectSingleQuery('SELECT \* FROM `t\_user` WHERE `mail` = ?', [$safeMail]);

        if ($userArray == null) {

        } else {

            extract($userArray);

            $user = new User($nom, $prenom, $civilite, $mail, $newsletter, $motDePasse);

            if ($user != null) {

                $user->setId($pk\_user);

                array\_walk\_recursive($user, function (&$item) {

                    $item = htmlspecialchars($item, ENT\_QUOTES, 'UTF-8');

                });

                return $user;

            }

        }

    }

## Problèmes rencontrés

### Session réinitialisé entre chaques requêtes

Nous avons rencontré un problème qu’est que nous n’arrivons pas à stocker un user dans la session. Une fois stocké, lorsque nous refaisons une requête censer aller chercher l’user dans la session il est introuvable. Nous avons différentes hypothèses :

* Plusieurs instance de la class session ont été créé.
* La session se recrée automatiquement entre chaques requêtes ce qui supprime ce qui se trouve dans le cache
* Une ligne de code reset à en continue la key user.

## Tests fonctionnels

|  |  |  |
| --- | --- | --- |
| Test | Résulat Attendu | Résultat obtenu |
| Connexion avec les bons credentials | Connexion réussie |  |
| Connexion avec les mauvais credentials | Connexion échouée |  |
| Créer un compte avec données valide | Création réussie |  |
| Créer un compte avec donnée non-valide | Création échouée |  |
| Créer un compte avec un mail déjà utilisé | Création échouée |  |
| Acheter une montre | Achat réussi |  |
| Modifier des infos du compte avec des données valides | Modification réussie |  |
| Modifier des infos du compte avec des données non-valide | Modification échouée |  |
| Supprimer son compte | Suppression réussie de la db et déconnexion |  |
| S’inscrire à la newsletter en tant qu’invité | Ajout du mail à la table newsletter |  |
| S’inscrire à la newsletter en tant qu’user | Changement à true du champs newsletter dans la table user. |  |
| Se déconnecter de la newsletter en tant qu’user | Changement à false du champs newsletter dans la table user. |  |
| Se déconnecter de le newsletter en tant que visiteur | Suppression du mail dans la newsletter |  |
| Déconnexion | Suppression de la session et déconnexion coté client. |  |
| Injection SQL | Injection patch |  |
| Injection script/html | Injection patch |  |

## Hébergement

# Synthèse

## Présentation réalisation

## Différences entre planning et réalisation

# Conclusion