

# DOCUMENTATION TECHNIQUE LOCAMOTO

V.1.0

LUDOVIC JACOT-DIT-MONTDANDON

CFPT

## Table des matières

INTRODUCTION .....	3
RAPPEL DU CAHIER DES CHARGES .....	4
OBJECTIFS DU PROJET .....	4
MATÉRIEL ET LOGICIELS À DISPOSITION .....	4
PLANNIFICATION .....	4
LIVRABLES.....	4
ANALYSE FONCTIONNELLE .....	5
FONCTIONNALITÉS .....	5
S'INSCRIRE SUR LE SITE.....	5
SE CONNECTER À L'APPLICATION.....	5
MODIFIER SON PROFIL .....	5
LOUER UNE MOTO .....	5
DONNER SON AVIS .....	5
VALIDER UNE LOCATION .....	5
ENVOI DE MAIL.....	5
INTERFACES .....	6
BARRE DE NAVIGATION.....	6
PAGE D'ACCUEIL.....	7
PAGE DE LOCATION.....	9
PAGE DE GESTION DES UTILISATEURS.....	10
PAGE DE GESTION DU PARKING.....	11
PAGE DE GESTION DES LOCATIONS.....	12
FORMULAIRE DE SAUVEGARDE D'UTILISATEURS.....	13
FORMULAIRE DE SAUVEGARDE DE MOTOS.....	14
FORMULAIRE DES AVIS.....	15
LOGIN.....	16
CONFIRMATION DE SUPPRESSION .....	16
CAS D'UTILISATION.....	17
VISITEUR ANONYME.....	17
UTILISATEUR AUTHENTIFIÉ.....	18
ADMINISTRATEUR .....	19
ANALYSE ORGANIQUE .....	22
BASE DE DONNÉES .....	22
DIAGRAMME DE BASE DE DONNÉES.....	22
DÉTAILS DES TABLES.....	23

ARBORESCENCE DE FICHER .....	24
OUTILS EXTERNES .....	25
Bootstrap : .....	25
GitHub : .....	25
Swift Mailer : .....	25
FullCalendar : .....	25
DESCRIPTION DES FONCTIONS .....	26
FICHER « dbconnection.php » .....	26
FICHER « users.php » .....	26
FICHER « motos.php » .....	27
FICHER « images.php » .....	28
FICHER « locations.php » .....	28
CALENDRIER .....	30
ENVOIE DE MAIL .....	31
TESTS ET PROTOCOLE DE TESTS .....	32
PLAN DE TESTS .....	32
RAPPORT DE TESTS .....	36
CONCLUSION .....	38
BILAN PERSONNEL .....	38
PROBLEMES RENCONTRES .....	38
AMELIORATIONS POSSIBLES .....	39
BIBLIOGRAPHIE .....	39
CODE REPRIS .....	39
SITE UTILISES .....	39
PLANNING .....	39
PLANNING PREVISIONNEL .....	39
PLANNING EFFECTIF .....	40
TABLE DES ILLUSTRATIONS .....	41
FIGURES .....	41
GLOSSAIRE .....	42
ANNEXES .....	42

## INTRODUCTION

Cette documentation détaille les étapes réalisées pour le site web intitulé « Locamoto », réalisé dans le cadre du Travail Pratique Individuel nécessaire à l'obtention du CFC d'informaticien.

Le projet « Locamoto » a pour but de créer un site web permettant aux utilisateurs de consulter des motos avec des variantes de marques et de cylindrée, et ce dans le but de louer différents modèles. Les motos mentionnées sont de marque fictives.

Les motos sont dans un parc qui peut être modifié selon les besoins par l'administrateur : entrée d'un nouveau modèle, correction d'une cylindrée, etc.

Le visiteur anonyme ou l'utilisateur authentifié peuvent consulter le parc de véhicules depuis la page d'accueil et y faire des recherches par marques ou par cylindrée.

L'utilisateur qui crée un compte sur le site peut, après s'être connecté :

- Modifier ses informations
- Louer une moto à la journée
- Donner un avis sur une moto louée

L'administrateur a la possibilité de gérer l'ensemble du site :

- Gérer un utilisateur :
  1. Création d'un utilisateur
  2. Modification d'un utilisateur
  3. Suppression d'un utilisateur
- Gérer le parking :
  1. Création d'une moto
  2. Modification d'une moto
- Gérer les locations
  1. Validation d'une location
  2. Gestion d'un calendrier

Le calendrier pour louer une moto est un libraire. J'ai choisi une solution JavaScript de calendrier nommé « Fullcalendar », qui offre une documentation assez simple. Il est disponible gratuitement à cette adresse : <https://fullcalendar.io/>

Le style de ce site est inspiré d'un template Bootstrap pour rendre le site adapté à l'écran de l'utilisateur, que ce soit sur un mobile, sur un ordinateur, sur une tablette, etc. qui se nomme : <https://getbootstrap.com/docs/4.2/examples/blog/>

## RAPPEL DU CAHIER DES CHARGES

### OBJECTIFS DU PROJET

La solution est fonctionnelle. Il est donc possible de s'inscrire et de se connecter en tant qu'utilisateur et ainsi effectuer une location. La validation des locations est gérée et le prix de ces dernières est affiché.

Le code est découpé suivant le paterne Modèle-Vue-Contrôleur et le site est responsif.

Les fonctionnalités de gestion par l'administrateur des motos, des locations et du calendrier sont implémentées.

La recherche sur la page d'accueil est implémentée.

L'utilisateur peut donner des avis sur la moto précédemment louée.

### MATERIEL ET LOGICIELS A DISPOSITION

- Un Pc standard école avec 2 écrans
- Windows 10
- EasyPHP ou autre WAMP
- Visual code
- Suite Office

### PLANNIFICATION

- Analyse : 12h
- Implémentation : 36h
- Test : 16h
- Documentation : 24h

### LIVRABLES

- Une planification initiale
- Un exemplaire format PDF de son mémoire avec annexe
  1. Documentation technique format PDF
  2. Code source
- Un accès aux sources du projet ainsi qu'une version exécutable
- Résumé du travail (page A4 pdf)

## ANALYSE FONCTIONNELLE

### FONCTIONNALITES

#### S'INSCRIRE SUR LE SITE

Cette fonctionnalité permet à un visiteur anonyme de pouvoir créer un compte en y rentrant son nom, son prénom, un pseudo, un email, un mot de passe et une date de naissance.

#### SE CONNECTER A L'APPLICATION

Cette fonctionnalité permet à un utilisateur qui possède un compte de pouvoir se connecter afin de pouvoir effectuer une réservation.

#### MODIFIER SON PROFIL

Sur le site, quand on possède un compte, il est possible de consulter son profil et d'en changer les informations.

#### LOUER UNE MOTO

Cette fonctionnalité offre la possibilité à un utilisateur authentifié de pouvoir louer une moto. Il peut consulter un calendrier et entrer la date du début de sa location souhaitée ainsi que la date de fin. Le prix pour cette location est affiché en conséquence du nombre de jour choisi.

#### DONNER SON AVIS

L'utilisateur peut laisser un commentaire sur une des réservations qu'il a effectuées.

#### VALIDER UNE LOCATION

Cette fonctionnalité permet à l'administrateur d'accepter ou refuser une réservation demandée par un utilisateur.

#### ENVOIE DE MAIL

Quand une réservation est validée par l'administrateur, un email de confirmation est envoyé à l'utilisateur qui l'a effectué.

## INTERFACES

### BARRE DE NAVIGATION

#### BARRE DE NAVIGATION D'UN VISITEUR ANONYME



Figure 1: BARRE DE NAVIGATION D'UN VISITEUR ANONYME

Le lien « Accueil » est la seule page disponible pour le visiteur (voir Figure 4 : Page d'accueil en tant que visiteur).

Le lien « Connexion » permet au visiteur de s'identifier ou de créer un compte (voir ).

#### BARRE DE NAVIGATION D'UN UTILISATEUR AUTHENTIFIE



Figure 2: BARRE DE NAVIGATION D'UN UTILISATEUR AUTHENTIFIE

Le lien « Accueil » permet à l'utilisateur d'aller à la page principale (voir Figure 5 : Page d'accueil en tant qu'utilisateur connecté).

Le lien « Profil » permet à l'utilisateur de consulter son profil et d'en modifier les informations (voir Figure 10 : Formulaire de sauvegarde d'utilisateurs).

Le lien « Déconnexion » permet à l'utilisateur de se déconnecter et ainsi passer à simple visiteur.

#### BARRE DE NAVIGATION D'UN ADMINISTRATEUR



Figure 3: BARRE DE NAVIGATION D'UN ADMINISTRATEUR

Le lien « Utilisateurs » permet à l'administrateur d'accéder à la page de gestion des utilisateurs (voir Figure 7 : Page de gestion des utilisateurs).

Le lien « Parking » permet à l'administrateur d'accéder à la page de gestion des motos (voir Figure 8 : Page de gestion du parking).

Le lien « Locations » permet à l'administrateur d'accéder à la page de gestion des réservations (voir Figure 9 : Page de gestion des locations).

## PAGE D'ACCUEIL

## PAGE D'ACCUEIL EN TANT QUE VISITEUR

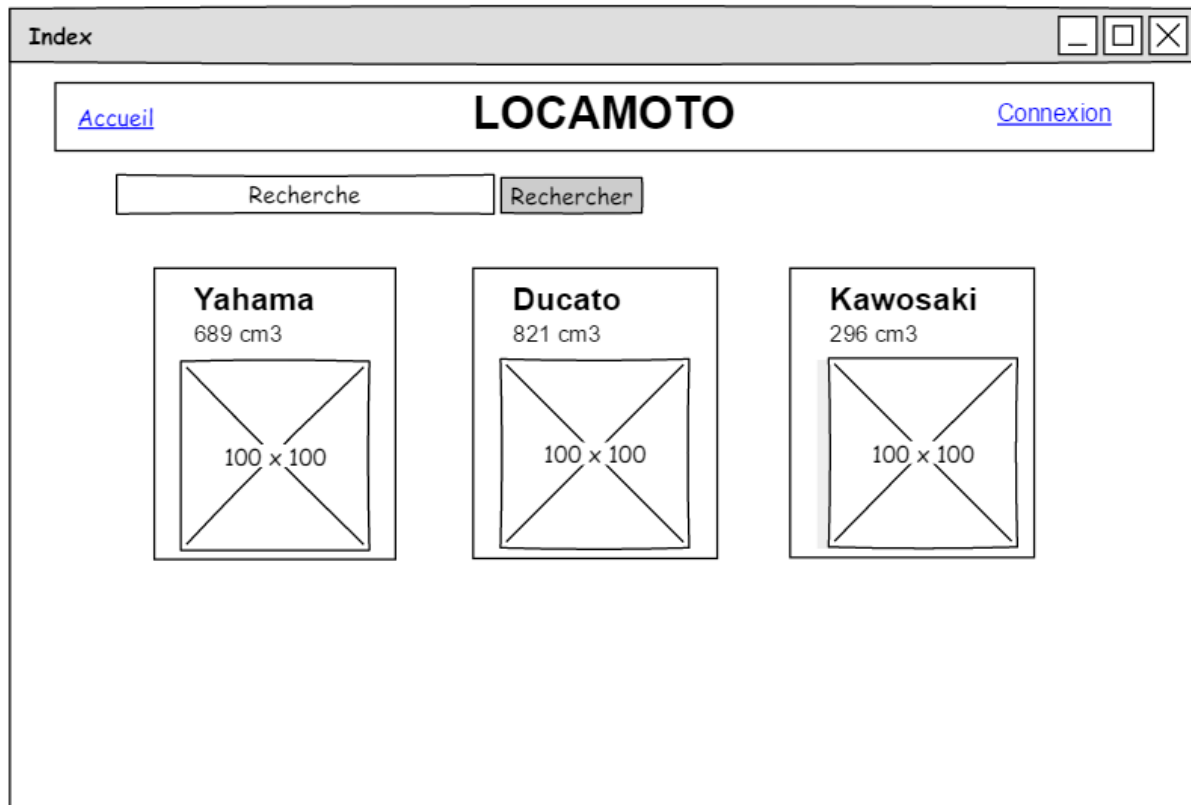
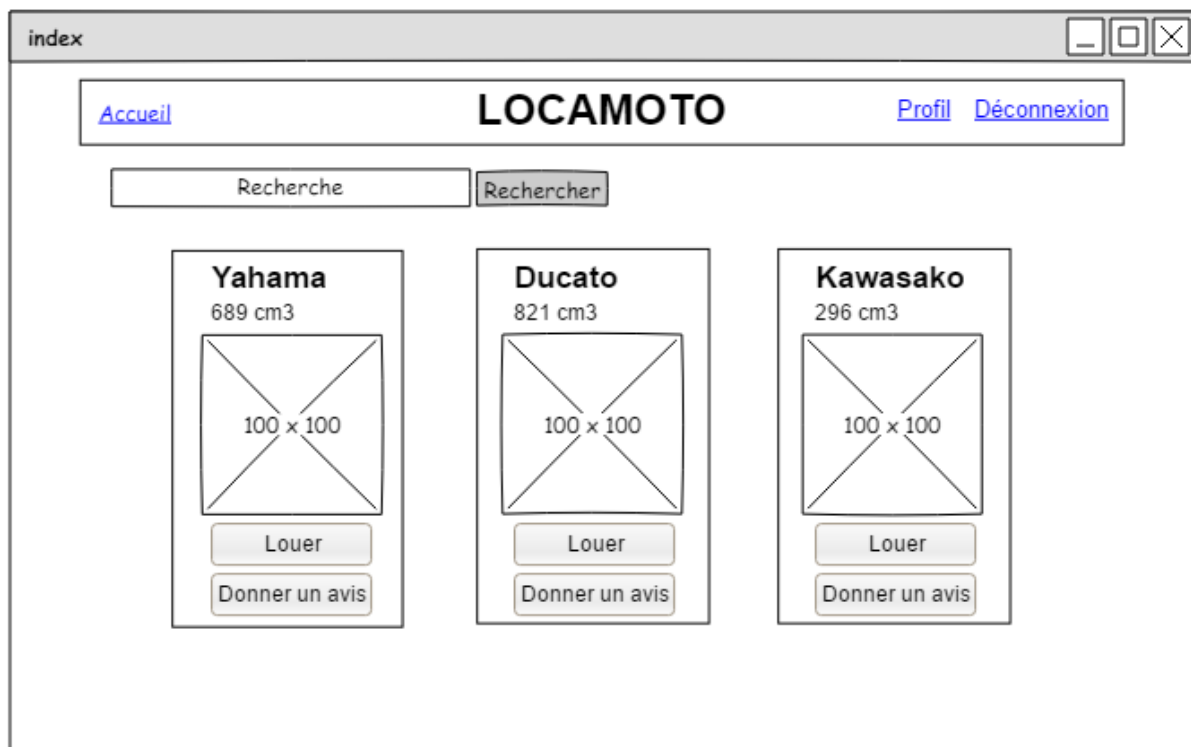


Figure 4: PAGE D'ACCUEIL EN TANT QUE VISITEUR

La page d'accueil en tant que visiteur affiche toutes les motos enregistrées sur le site avec leurs marques, leurs cylindrées et leurs photos.

La barre de recherche invite à écrire les mots-clés que l'on souhaite trouver. Une fois le bouton « rechercher » pressé, seul les noms de marque ou cylindrée correspondant à la recherche sont affichés.



*PAGE D'ACCUEIL EN TANT QU'UTILISATEUR CONNEXE**Figure 5 : PAGE D'ACCUEIL EN TANT QU'UTILISATEUR CONNEXE*

La page d'accueil en tant que visiteur affiche toutes les motos enregistrées sur le site avec leurs marques, leurs cylindrées et leurs photos.

La barre de recherche invite à écrire les mots-clés que l'on souhaite trouver. Une fois le bouton « rechercher » pressé, seul les noms de marque ou cylindre correspondant à la recherche sont affichés.

Le bouton « Louer » envoie sur la page de réservation (voir Figure 6 : Page de location).

Le bouton « Donner un avis » envoie sur la page d'ajout d'avis seulement si on a déjà réalisé une réservation pour cette moto.

## PAGE DE LOCATION

The screenshot shows a web browser window titled 'location'. The page has a header with 'Accueil' on the left, 'LOCAMOTO' in the center, and 'Profil' and 'Déconnexion' on the right. The main content area is divided into two sections. On the left is a calendar for August 2016, with days of the week (S, M, T, W, T, F, Sa) as column headers. The calendar shows dates from 31st of the previous month to 10th of the next month. The 1st of August is highlighted in grey. On the right is the motorcycle details section for a 'Yamaha'. It lists '683 cm3', 'Bleue', and '26.06.2000'. Below this are input fields for 'Date début:' (07-05-2019) and 'Date fin:' (23-05-2000). A 'Louer' button is present. At the bottom, the price is shown as 'Prix : 500 CHF'.

S	M	T	W	T	F	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

**Yamaha**  
683 cm3  
Bleue  
26.06.2000

Date début: 07-05-2019  
Date fin: 23-05-2000

Louer

Prix : 500 CHF

Figure 6: PAGE DE LOCATION

La page de location affiche toute les informations sur la moto voulue (marque, cylindrée, couleur, date d'immatriculation) sur la droite.

Le prix évolue en fonction du nombre de jour loué.

Le bouton « louer » réserve les dates indiquées avec cette moto et il devra être validé par l'administrateur (voir Figure 9 : Page de gestion des locations).

## PAGE DE GESTION DES UTILISATEURS

Nom	Prenom	Pseudo	MotDePasse	Email	DateNaissance	Edition
Jacot	Ludovic	Ludo	*****	ludovic.jctdt@eduge.ch	26.06.2000	
Vert	Olive	Oli	*****	olive.vert@eduge.ch	23.02.1990	
Justice	Helen	Ash	*****	HelenLJustice@dayrep.co	08.09.1940	
Reynolds	Adrian	Arcand	*****	Adrianeynolds@gmail.com	05.04.1967	
Bousquet	Cheney	Ruellan	*****	CheneyBousquet@gmail.c	20.12.1972	

Figure 7: PAGE DE GESTIONS DES UTILISATEURS

La page de gestion des utilisateurs est accessible uniquement par l'administrateur. Elle affiche un tableau avec tous les utilisateurs inscrits sur le site.

Le bouton plus « + » permet de créer un nouvel utilisateurs (voir Figure 10 : Formulaire de sauvegarde d'utilisateurs).

L'icône du crayon « » permet d'éditer l'utilisateur correspondant à l'aide du formulaire prévu à cet effet (voir Figure 10 : Formulaire de sauvegarde d'utilisateurs).

L'icône de la croix « » permet de supprimer l'utilisateur correspondant après une confirmation (voir Figure 13 : Confirmation de suppression).

## PAGE DE GESTION DU PARKING




noPlaque	Marque	Cylindree	Couleur	Date immatriculation	Edition
123456	Yahama	689 cm3	bleue	26.06.2000	
234567	Yahama	660 cm3	Noire	23.02.1990	
345678	Ducato	821 cm3	Rouge	08.09.1940	
823457	Kawosaki	296 cm3	Verte	12.12.2012	

Figure 8: PAGE DE GESTION DU PARKING

La page de gestion du parking est accessible uniquement par l'administrateur. Elle affiche un tableau de toutes les motos disponibles à la location dans le parking.

Le bouton plus « + » permet de créer une nouvelle moto (voir Figure 11 : Formulaire de sauvegarde de motos).

L'icône du crayon «  » permet d'éditer la moto correspondante à l'aide du formulaire prévu à cet effet (voir Figure 11 : Formulaire de sauvegarde de motos).

## PAGE DE GESTION DES LOCATIONS

Date de la Reservation	Date debut	Date de fin	Validation
24.08.2019	24.08.2019	26.09.2019	✓ ✕
10.02.2019	11.02.2019	12.02.2019	✓ ✕
08.09.2019	10.09.2019	13.09.2019	✓ ✕
10.05.2019	15.06.2019	12.07.2019	✓ ✕

Figure 9: PAGE DE GESTION DES LOCATIONS

La page gestion des locations est accessible uniquement par l'administrateur. Elle affiche un tableau de toutes les locations en attente d'être validée.

L'icône du vu « ✓ » existe pour confirmer la réservation correspondante et ainsi, un email est envoyé à l'utilisateur.

L'icône de la croix « ✕ » refuse la location et supprime celle-ci après une validation (voir Figure 13 : Confirmation de suppression).

## FORMULAIRE DE SAUVEGARDE D'UTILISATEURS

The screenshot shows a web browser window titled 'saveUser'. Inside the window, there is a header bar with a link 'Accueil' on the left and the text 'LOCAMOTO' in the center. Below the header, the title 'Edition d'utilisateurs' is displayed. The form contains seven input fields, each with a label to its left: 'Nom', 'Prenom', 'Pseudo', 'Mot de passe', 'Confirmation mot de passe', 'Email', and 'Date de naissance'. Each label is followed by a text input field. Below these fields is a button labeled 'Enregistrer'.

Figure 10: FORMULAIRE DE SAUVEGARDE D'UTILISATEURS

Le formulaire indique sept champs obligatoires « Nom », « Prénom », « Pseudo », « Mot de passe », « Confirmation de mot de passe », « Email » et « Date de Naissance ». Il utilise une méthode « POST ».

Le nom, le prénom, le pseudo sont des inputs de type « text ».

Le mot de passe et la confirmation de mot de passe sont des inputs de type « password ».

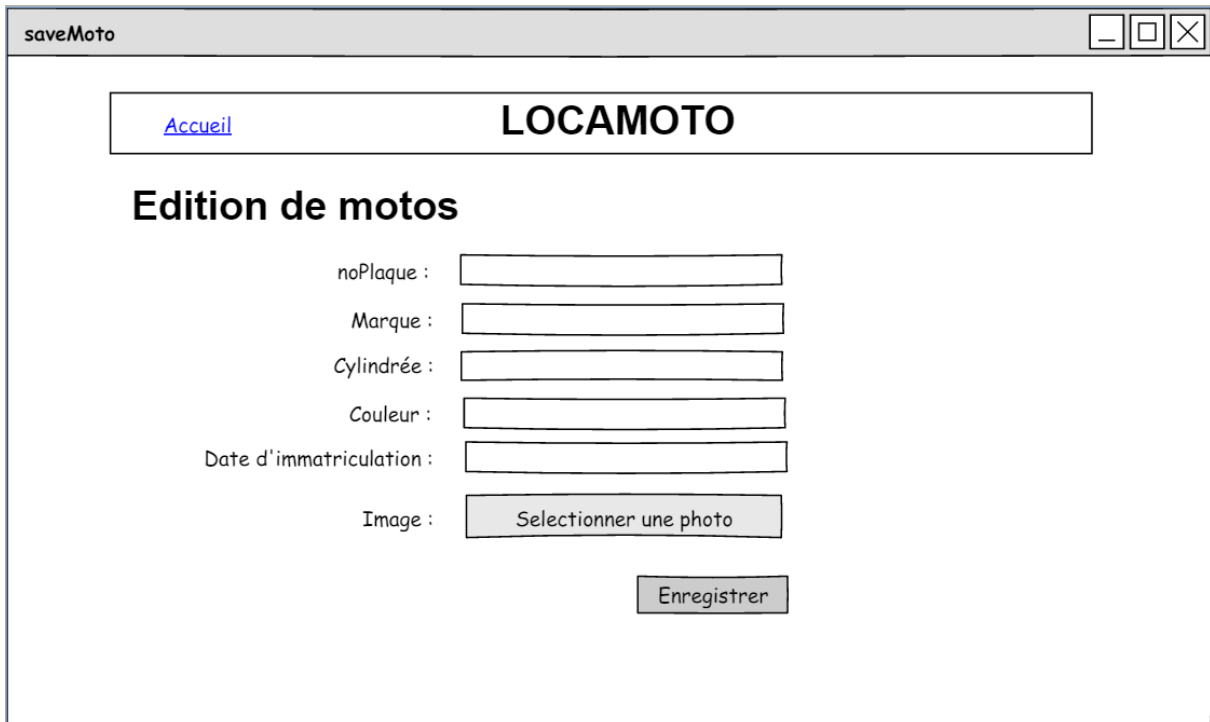
L'email est un input de type « email ».

La date est un input de type « date »

Le bouton enregistrer est de type « submit ».

Il est utilisé pour l'inscription ainsi que pour l'édition d'un utilisateur. En cas d'édition, les informations sont automatiquement remises dans les champs correspondants.

## FORMULAIRE DE SAUVEGARDE DE MOTOS



The screenshot shows a web browser window titled 'saveMoto'. Inside the window, there is a header bar with a blue link 'Accueil' on the left and the text 'LOCAMOTO' in the center. Below the header, the main heading is 'Edition de motos'. The form contains six input fields: 'noPlaque', 'Marque', 'Cylindrée', 'Couleur', 'Date d'immatriculation', and 'Image'. The 'Image' field is a file selection button labeled 'Selectionner une photo'. At the bottom right of the form is a button labeled 'Enregistrer'.

Figure 11: FORMULAIRE DE SAUVEGARDE DE MOTOS

Le formulaire est composé de 6 champs « noPlaque », « Marque », « Cylindrée », « Couleur », « Date d'immatriculation » et « Image ». Il utilise une méthode « POST ».

Le numéro de plaque est un input de type « number »

La marque, la cylindrée et la couleur sont des inputs de type « text ».

La date d'immatriculation est un input de type « date »

L'image est un input de type « file ».

Le bouton enregistrer est de type « submit ». Il ajoute ou met à jour une moto dans la base de données. Si on édite une moto existante, les informations sont automatiquement insérées dans les champs.

## FORMULAIRE DES AVIS

The screenshot shows a web browser window titled 'saveMoto'. Inside the browser, there is a navigation bar with a link 'Accueil' and the title 'LOCAMOTO'. Below the navigation bar, the main heading is 'Rediger un avis'. Under this heading, there is a label 'Redigez votre avis :' followed by a large text input field. Below the text field, there is a label 'Selectionnez la réservation correspondante :'. Under this label, there are three buttons, each representing a reservation: 'Kawasako le 06.09.2019', 'Kawasako le 10.12.2018', and 'Kawasako le 21.03.2019'. Below these buttons, there is a button labeled 'Enregistrer'.

Figure 12: FORMULAIRE DES AVIS

Cette page est accessible uniquement si l'on a déjà fait une réservation sur une moto spécifique.

Le formulaire est composé d'une textarea et de check box.

L'utilisateur est invité à rédiger son avis dans la zone de texte et ensuite sélectionner la réservation sur laquelle il aimerait l'affilié.

Le bouton « Enregistrer » est de type « submit ». Cela met à jour le champs « avis » dans la base de données sur la réservation concernée.



## LOGIN




The screenshot shows a web browser window with the title 'login'. The page content includes a header with a link 'Accueil' and the title 'LOCAMOTO'. Below this is a section titled 'Connexion'. It contains two input fields: 'Pseudo:' and 'Mot de passe:'. There is a blue link 'S'inscrire' and a grey button labeled 'Connexion'.

Figure 13: LOGIN

Le formulaire utilise une méthode « POST ». Le login comporte deux champs « Pseudo » et « Mot de passe ».

Le bouton connexion est de type « submit »

## CONFIRMATION DE SUPPRESSION



The screenshot shows a web browser window with the title 'deleteUser'. The page content includes a section titled 'Confirmer la suppression'. It contains the text 'Etes-vous sûr de vouloir supprimer ?' followed by two radio buttons: 'Oui' and 'Non'. The 'Non' radio button is selected. There is a grey button labeled 'Confirmer'.

Figure 14: CONFIRMATION DE SUPPRESSION

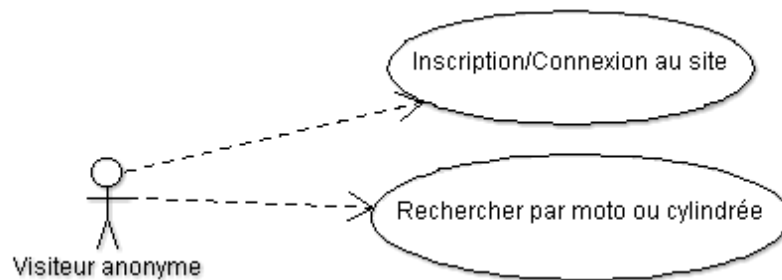
Ce n'est pas une page à proprement parlé, mais une pop-up qui apparaît au moment où l'on souhaite supprimer un utilisateur, une moto ou une réservation.

Le formulaire est de type « POST ».

Le bouton confirmer est type « submit ». Il supprime définitivement l'utilisateur ou la location.

## CAS D'UTILISATION

### VISITEUR ANONYME



**Nom cas d'utilisation :** Se connecter

**Acteur principal :** Visiteur anonyme

**Déclencheur :** L'utilisateur accède au site

**Flot principal :**

1. Le système affiche la page d'accueil.
2. L'utilisateur clique sur « Connexion ».
3. Le système redirige l'utilisateur sur le formulaire de connexion qui contient deux champs : « Pseudo » et « Mot de passe ».
4. L'utilisateur entre les informations demandées et clique sur « Connexion ».
5. Le système affiche la page principale.

**Flot alternatif :**

- 4a. L'utilisateur ne possède pas de compte et clique sur « inscription ».
  - 4a1. Le système redirige l'utilisateur sur le formulaire d'édition d'utilisateur qui contient sept champs : « Nom », « Prénom », « Pseudo », « Mot de passe », « Confirmation de mot de passe », « Email » et « Date de naissance ».
  - 4a2. L'utilisateur renseigne tous les champs et clique sur « Enregistrer ».
  - 4a3. Le système affiche la page d'accueil.

**Nom cas d'utilisation :** Rechercher par moto ou cylindrée

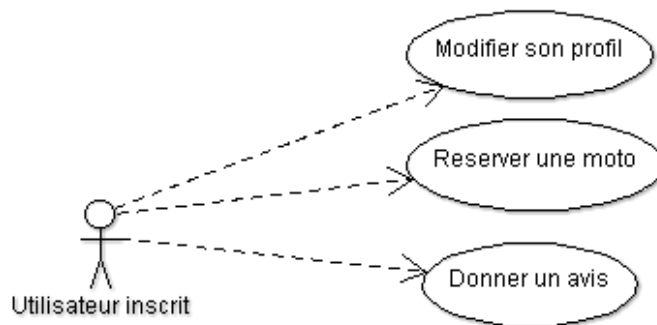
**Acteur principal :** Visiteur anonyme et utilisateur connecté

**Déclencheur :** L'utilisateur accède au site

**Flot principal :**

1. Le système affiche la page d'accueil.
2. L'utilisateur entre les mots-clés qu'il souhaite atteindre et clique sur rechercher.
3. La page affiche uniquement les résultats qui correspondent.

## UTILISATEUR AUTHENTIFIE



**Nom cas d'utilisation :** Modifier son profil

**Acteur principal :** Utilisateur inscrit

**Déclencheur :** L'utilisateur clique sur « Profil »

**Flot principal :**

1. L'utilisateur clique sur « profil ».
2. Le système redirige l'utilisateur sur le formulaire d'édition d'utilisateur qui contient sept champs : « Nom », « Prénom », « Pseudo », « Mot de passe », « Confirmation de mot de passe », « Email » et « Date de naissance ». Ils sont tous remplis avec les informations correctes.
3. L'utilisateur édite les informations qu'il souhaite et clique sur « enregistrer ».
4. Le système affiche la page d'accueil.

**Nom cas d'utilisation :** Réserver une moto

**Acteur principal :** Utilisateur inscrit

**Déclencheur :** L'utilisateur consulte les motos.

**Flot principal :**

1. Le système affiche la page d'accueil avec les différentes motos.
2. L'utilisateur clique sur « louer ».
3. Le système redirige l'utilisateur sur la page de location. Elle affiche un calendrier.
4. L'utilisateur renseigne la date du début de la location ainsi que la fin de location et clique sur « Louer ».
5. Le système affiche la page d'accueil.

**Nom cas d'utilisation :** Donner un avis

**Acteur principal :** Utilisateur inscrit

**Déclencheur :** L'utilisateur clique sur « Laisser un avis »

**Flot principal :**

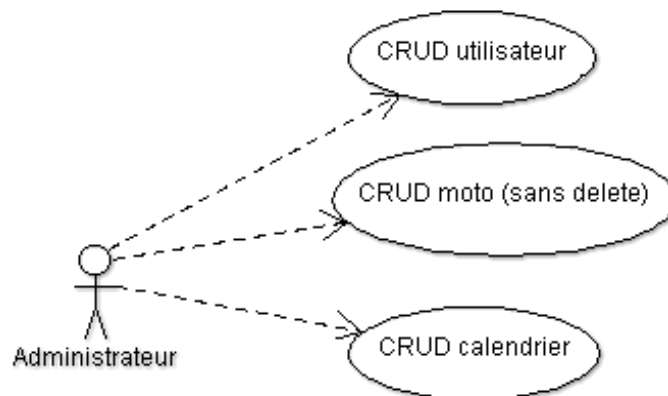
1. Le système affiche la page d'accueil avec les différentes motos.
2. L'utilisateur clique sur « Laisser un avis »
3. Le système affiche une page avec une zone de texte et toutes les réservations effectuées par l'utilisateur sur cette moto.
4. L'utilisateur entre un message, sélectionne la location qui correspond et clique sur « Envoyer »
5. Le système redirige sur la page d'accueil.

**Flot secondaire :**

3a. L'utilisateur n'a jamais fait de réservation sur cette moto

3a1. Le système affiche un message d'erreur.

## ADMINISTRATEUR



**Nom cas d'utilisation :** Gérer les utilisateurs

**Acteur principal :** Administrateur

**Déclencheur :** L'administrateur clique sur « Utilisateurs »

**Flot principal :**

1. L'administrateur clique sur « Utilisateurs »
2. Le système affiche un tableau avec tous les utilisateurs inscrits sur le site.
3. L'administrateur choisit un profil et clique sur éditer.
4. Le système redirige l'administrateur sur le formulaire d'édition d'utilisateur qui contient sept champs : « Nom », « Prénom », « Pseudo », « Mot de passe », « Confirmation de mot de passe », « Email » et « Date de naissance ». Ils sont tous remplis avec les informations correctes.
5. L'administrateur édite les informations qu'il souhaite et clique sur « enregistrer ».

6. Le système affiche la page principale.

**Flot alternatif :**

- 3a. L'administrateur choisi un profil et clique sur supprimer.
  - 3a1. Le système affiche un formulaire de confirmation.
  - 3a2. L'utilisateur clique sur « Confirmer ».
  - 3a3. Le système supprime l'utilisateur et affiche la page d'accueil.
- 3b. L'administrateur clique sur le bouton plus « + ».
  - 3b1. Le système redirige l'administrateur sur le formulaire d'édition d'utilisateur qui contient sept champs : « Nom », « Prénom », « Pseudo », « Mot de passe », « Confirmation de mot de passe », « Email » et « Date de naissance ». Ils sont tous rempli avec les informations correctes.
  - 3b2. L'administrateur édite les champs souhaités et clique sur enregistrer.
  - 3b3. Le système affiche la page d'accueil.

**Nom cas d'utilisation :** Gérer les motos

**Acteur principal :** Administrateur

**Déclencheur :** L'administrateur clique sur « Parking »

**Flot principal :**

1. L'administrateur clique sur « Parking »
2. Le système affiche un tableau avec toutes les motos inscrites sur le site.
3. L'administrateur clique sur le bouton plus « + ».
4. Le système redirige l'administrateur sur le formulaire d'édition de moto. Les champs à remplir sont : « noPlaque », « Marque », « Cylindrée », « Couleur » et « Date d'immatriculation ».
5. L'administrateur entre les champs stipulés et clique sur enregistrer
6. Le système affiche la page précédente.

**Flot alternatif :**

- 3a. L'administrateur choisi une moto et clique sur éditer.
  - 3a1. Le système redirige l'administrateur sur le formulaire d'édition de moto. Les champs à remplir sont : « noPlaque », « Marque », « Cylindrée », « Couleur » et « Date d'immatriculation ».
  - 3a2. L'administrateur édite les champs souhaités et clique sur enregistrer.
  - 3a3. Le système affiche la page précédente.

**Nom cas d'utilisation :** Gérer les locations

**Acteur principal :** Administrateur

**Déclencheur :** L'administrateur clique sur « Locations »

**Flot principal :**

1. L'administrateur clique sur « Locations »
2. Le système affiche un tableau avec toutes les réservations demandées par les utilisateurs.
3. L'administrateur choisi de validé une réservation en cliquant sur le vu « ✓ ».
4. Le système la valide, envoie un mail à l'utilisateur correspondant et recharge la page.

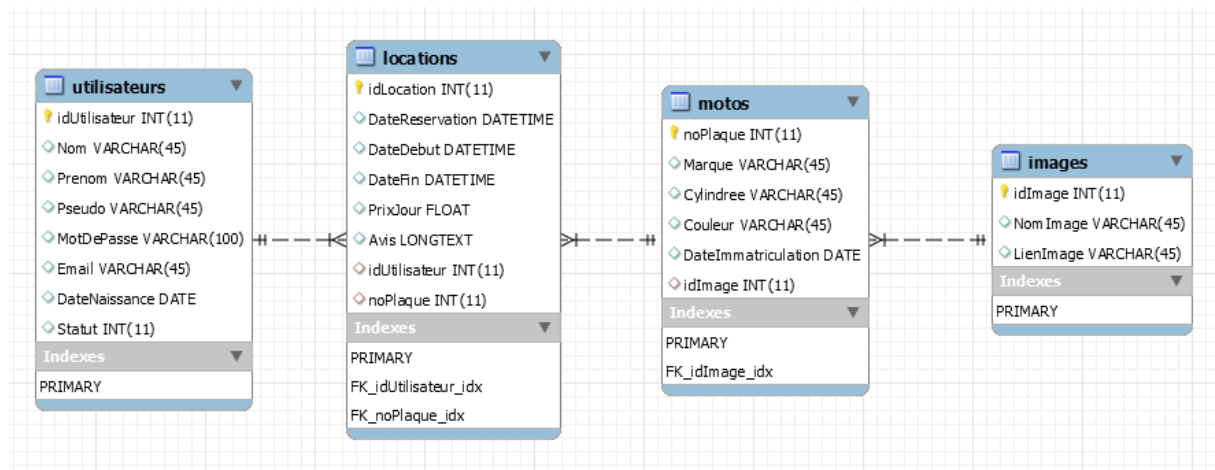
**Flot alternatif :**

- 3a. L'administrateur choisi de refuser une réservation en cliquant sur la croix « ✕ ».
  - 3a1. Le système affiche un formulaire de confirmation.
  - 3a2. L'utilisateur clique sur « Confirmer ».
  - 3a3. Le système supprime l'utilisateur et affiche la page précédente.

## ANALYSE ORGANIQUE

### BASE DE DONNEES

#### DIAGRAMME DE BASE DE DONNEES



Le site comporte quatre tables dans la base de données.

La table utilisateurs contient tous les champs soumis par l'utilisateur ainsi que son identifiant et son statut. « idUtilisateurs » est donc la clé primaire.

La table locations contient la date à laquelle elle a été faite ainsi que les dates auxquelles la réservation est effectuée. Les avis y sont aussi stockés avec le prix. « idLocation » est la clef primaire. La table possède aussi deux clefs étrangères, l'identifiant de l'utilisateur qui l'a effectué et le numéro de plaque de la moto réservée.

La table motos contient toutes les informations relatives à une moto. Le numéro de plaque « noPlaque » est la clef primaire. Elle possède aussi une clef étrangère, l'identifiant de l'image qui lui est relié.

La table image possède une clef primaire qui est l'identifiant de cette dernière. Le nom et le lien de l'image y sont stockés.

## DETAILS DES TABLES

Abréviations utilisées	Définition
PK	Clefs primaire
NN	Non nul (Impossible d'être vide – NULL)
UQ	Unique (Impossible d'avoir deux fois la même valeur pour cette occurrence)
AI	Auto incrémentation

TABLE « UTILISATEURS »









Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idUtilisateur	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 Nom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Prenom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Pseudo	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 MotDePasse	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Email	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 DateNaissance	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Statut	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'

Figure 15: TABLE UTILISATEURS

TABLE LOCATIONS









Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idLocation	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 DateReservation	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 DateDebut	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 DateFin	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 PrixJour	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Avis	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 idUtilisateur	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 noPlaque	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 16: TABLE LOCATIONS

TABLE MOTOS







Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 noPlaque	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Marque	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Cylindree	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Couleur	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 DateImmatriculation	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 idImage	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 17: TABLE MOTOS

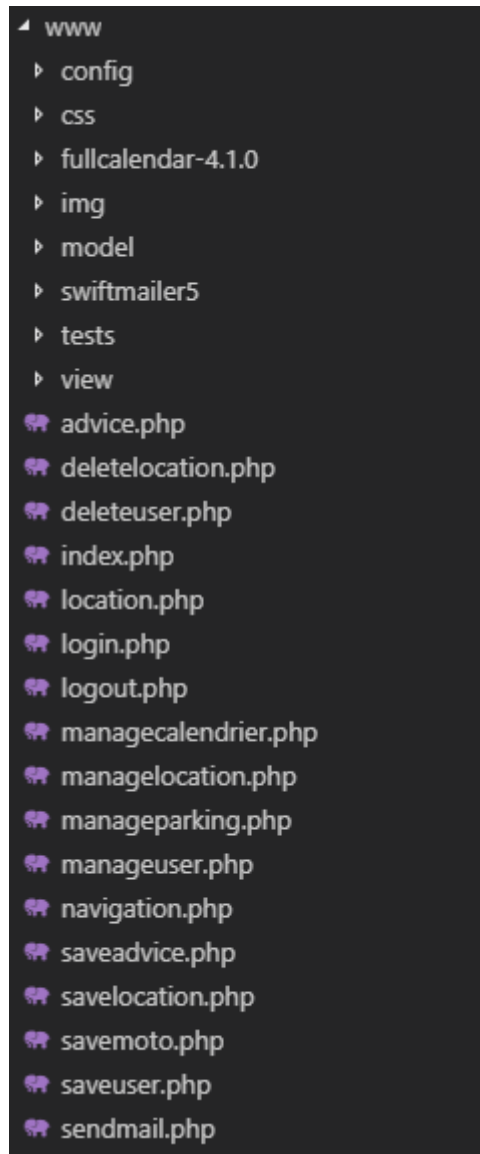
TABLE IMAGES

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idImage	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 NomImage	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 LienImage	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 18: TABLE IMAGES



## ARBORESCENCE DE FICHIER



- Le dossier « config » contient un fichier avec des constantes utiles à la connexion à la base de données et un fichier avec des constantes utiles à l'envoi de mail.
- Le dossier « fullcalendar-4.1.0 » contient les fichiers du plugin FullCalendar
- Le dossier « css » contient les fichiers bootstrap ainsi qu'un fichier style.css
- Le dossier « img » contient toutes images visibles sur le site.
- Le dossier « model » contient tous les fichiers de CRUD exercés sur les tables de la base de données ainsi qu'un fichier de connexion à la base de données.
- Le dossier « swiftmailer5 » contient les fichiers du plugin Swift Mailer 5.
- Le dossier test contient les
- Le dossier « view » contient les pages du site.
- A la racine il y a tous les contrôleurs

J'utilise une architecture MVC (modèle-vue-contrôleur), comme demandé dans le cahier des charges. Le modèle MVC est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs. Un modèle contient les données à afficher. Une vue contient la présentation de l'interface graphique. Un contrôleur contient la logique concernant les actions effectuées par l'utilisateur.

## OUTILS EXTERNES

### BOOTSTRAP :

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

J'ai utilisé Bootstrap pour que le site soit facilement responsive.

### GITHUB :

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

J'ai utilisé GitHub pour la gestion de fichier de mon site. Effectivement chaque soir, je sauvegardais tous les changements.

### SWIFT MAILER :

Swift Mailer s'intègre à n'importe quelle application Web écrite en PHP, offrant une approche souple et élégante orientée objet pour envoyer des courriels avec une multitude de fonctionnalités.

J'ai utilisé Swift Mailer pour envoyer des mails aux utilisateurs.

### FULLCALENDAR :

FullCalendar est un calendrier JavaScript. Il est idéal pour afficher des événements, mais ce n'est pas une solution complète pour la gestion du contenu des événements. Au-delà du déplacement d'un événement à une heure / un jour différent, on ne peut pas modifier le nom d'un événement ni d'autres données associées. C'est à nous d'ajouter cette fonctionnalité via l'API de FullCalendar.

J'ai utilisé FullCalendar pour avoir un calendrier sur le site.

## DESCRIPTION DES FONCTIONS

### FICHIER « dbconnection.php »

#### *Fonction « connectDB »*

Cette fonction est déclarée dans le fichier « dbconnection.php » et sert à établir la connexion à la base de données. Elle ne reçoit pas de paramètre mais utilise cinq constantes déclarée dans le fichier « config/confparam.php ». Ces constantes sont « DB\_DBTYPE » pour le type de base de données, « DB\_HOST » pour l'hôte du serveur, « DB\_DBNAME » pour le nom de la base de données, « DB\_USER » pour l'utilisateur qui se connecte et « DB\_PASS » pour le mot de passe utilisé lors de la connexion. Cette fonction retourne une instance de l'objet PDO.

### FICHIER « users.php »

#### *Fonction « userExist »*

Cette fonction permet de savoir si un couple pseudo de l'utilisateur / mot de passe est correct et existe dans la base. Elle est utilisée lors de la connexion d'utilisateur (classique ou administrateur). Elle reçoit deux paramètres, le pseudo de l'utilisateur ainsi que le mot de passe, qu'elle utilise pour construire une requête SQL (un SELECT avec le nom et le mot de passe dans la clause WHERE) qui lui permet de savoir si le couple pseudo d'un utilisateur / mot de passe est correct, et donc si le login est correct. Effectivement, si le SQL retourne une ligne, c'est que les données correspondent. Elle retourne un tableau contenant tous les champs de la table coïncidant avec ces paramètres (des données qui pourront être réutilisées sur le site) si le login est juste, sinon « False ».

#### *Fonction « getAllUsers »*

Cette fonction permet de récupérer tous les utilisateurs inscrit dans le base de données. Elle est utilisée pour la page « manageuser.php ». Elle ne reçoit aucun paramètre, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « utilisateurs ». Elle retourne un tableau à deux dimensions, dont chaque ligne contient l'identifiant de l'utilisateur, son nom, son prénom, son mot de passe encrypté, son email, sa date de naissance.

#### *Fonction « getUserById »*

Cette fonction permet de récupérer les informations sur un utilisateur d'après son identifiant. Elle est utilisée sur la page « saveuser.php », lorsqu'il faut modifier un utilisateur. Elle est donc très similaire à la fonction précédente (Fonction « getAllUsers »), sauf que celle-ci reçoit en paramètre l'identifiant. Celui-ci est utilisé dans la clause WHERE de la requête SQL. Elle retourne un tableau qui contient l'identifiant de l'utilisateur, son nom, son prénom, son mot de passe encrypté, son email, sa date de naissance.

#### *Fonction « getUserId »*

Cette fonction, en cas de succès, retourne l'identifiant d'un utilisateur d'après son pseudo. Elle est utilisée dans le contrôleur « login.php » pour mettre en session l'identifiant de la personne qui vient de se connecter. Elle reçoit en paramètre le pseudo qui vient de se connecter qui est utilisé dans la clause WHERE de la requête SQL. Elle retourne l'identifiant de l'utilisateur.

#### *Fonction « getUserStatut »*

Cette fonction, en cas de succès, retourne le statut d'un utilisateur d'après son pseudo. Elle est utilisée dans le contrôleur « login.php » pour mettre en session le statut de la personne qui vient de se connecter. Elle reçoit en paramètre le pseudo qui vient de se connecter qui est utilisé dans la clause WHERE de la requête SQL. Elle retourne le statut de l'utilisateur.

*Fonction « addUser »*

Cette fonction permet d'ajouter un nouvel utilisateur à la base de données. Elle est utilisée dans le contrôleur « saveuser.php » pour entrer toutes les informations du formulaire dans la base de données. Elle reçoit en paramètre chaque champ nécessaire pour un utilisateur : nom, prénom, pseudo, mot de passe, email, date de naissance. Ils sont tous utilisés comme valeur à insérer. Elle retourne l'identifiant de l'utilisateur à l'instant ajouté ou « False » en cas de problème.

*Fonction « updateUser »*

Cette fonction permet de mettre à jour un utilisateur d'ores et déjà présent dans la base de données. Elle est utilisée dans le contrôleur « saveuser.php » pour entrer toutes les nouvelles informations du formulaire dans la base de données. Elle reçoit en paramètre chaque champ possible pour un utilisateur : nom, prénom, pseudo, email, date de naissance. Ils sont tous utilisés comme valeur à insérer. Elle retourne « True » en cas de succès ou « False » en cas de problème.

*Fonction « deleteUser »*

Cette fonction permet de supprimer définitivement un utilisateur de la base de données. Elle est utilisée sur la page « deleteuser.php » où, après confirmation, il est justement possible pour l'administrateur de supprimer un utilisateur. Elle reçoit en paramètre l'identifiant de l'utilisateur à supprimer qui est utilisé dans la clause WHERE du DELETE. Elle retourne « True » en cas de succès ou « False » en cas de problème.

*FICHER « motos.php »**Fonction « getMotos »*

Cette fonction permet de récupérer toutes les motos présentes dans la base de données. Elle est utilisée pour la page « manageparking.php ». Elle ne reçoit aucun paramètre, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « motos ». Elle retourne un tableau à deux dimensions, dont chaque ligne contient le numéro de plaque d'une moto, sa marque, sa cylindrée, sa couleur, sa date d'immatriculation.

*Fonction « getMotosImages »*

Cette fonction permet de récupérer toutes les motos présentes dans la base de données avec leurs images assignées. Elle est utilisée pour la page « index.php ». Elle est donc très similaire à la fonction précédente (Fonction « getMotos ») mais fait une jointure entre la table « motos » et la table « images ». Elle ne reçoit aucun paramètre, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « motos » en plus de la table « images ». Elle retourne un tableau à deux dimensions, dont chaque ligne contient le numéro de plaque d'une moto, sa marque, sa cylindrée, sa couleur, sa date d'immatriculation, le nom de son image.

*Fonction « getMotosSearch »*

Cette fonction permet de récupérer toutes les motos présentes dans la base de données avec leurs images assignées correspondant à la recherche demandée. Elle effectue le même SELECT que la fonction précédente (Fonction « getMotosImages ») mais utilise en plus l'opérateur LIKE entre la marque et le paramètre ou la cylindrée et le paramètre. Elle retourne un tableau à deux dimensions, dont chaque ligne contient le numéro de plaque d'une moto, sa marque, sa cylindrée, sa couleur, sa date d'immatriculation, le nom de son image.

*Fonction « getMotoByPlaque »*

Cette fonction permet de récupérer toutes les informations d'une moto par rapport à son numéro de plaque. Elle est utilisée dans le fichier « savemoto.php ». Elle effectue un SELECT de « Marque »,

« Cylindree », « Couleur », « DateImmatriculation ». Elle reçoit donc le numéro de plaque « noPlaque » qui est la clé primaire en paramètre. Elle retourne un tableau avec chaque champ.

#### *Fonction « addMoto »*

Cette fonction permet d'ajouter une nouvelle moto à la base de données. Elle est utilisée dans le contrôleur « manageparking.php » pour entrer toutes les informations du formulaire dans la base de données. Elle reçoit en paramètre chaque champ nécessaire pour une moto : noPlaque, marque, cylindrée, couleur, date immatriculation. Ils sont tous utilisés comme valeur à insérer. Elle retourne « True » en cas de succès ou « False » en cas de problème.

#### *Fonction « updateMoto »*

Cette fonction permet de mettre à jour une moto d'ores et déjà présente dans la base de données. Elle est utilisée dans le contrôleur « manageparking.php » pour entrer toutes les nouvelles informations du formulaire dans la base de données. Elle reçoit en paramètre chaque champ possible pour une moto : noPlaque, marque, cylindrée, couleur, date immatriculation. Ils sont tous utilisés comme valeur à insérer. Elle retourne « True » en cas de succès ou « False » en cas de problème.

#### FICHER « images.php »

##### *Fonction « addImage »*

Cette fonction permet d'ajouter une image dans la base de données. Elle est utilisée dans le contrôleur « savemoto.php ». Elle reçoit en paramètre le nom de l'image et le lien de l'image. Ils sont utilisés comme valeur à insérer. Elle renvoie le dernier identifiant entré ou « FALSE ».

##### *Fonction « updatelImage »*

Cette fonction permet de mettre à jour des informations concernant une moto existante. Elle est utilisée dans le contrôleur « manageparking.php » pour changer les informations en fonction du formulaire. Elle reçoit en paramètre le nom de l'image et le lien de l'image. Les deux champs sont utilisés comme valeur à mettre à jour. Elle retourne « True » en cas de succès ou « False » en cas d'erreur.

#### FICHER « locations.php »

##### *Fonction « getLocationsJSON »*

Cette fonction permet de récupérer toutes les locations présentes dans la base de données. Elle est utilisée pour la page « managelocation.php ». Elle ne reçoit aucun paramètre, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « locations ». Elle retourne une transformation d'un tableau à deux dimensions, dont chaque ligne contient la date de réservation, la date de début, la date de fin, le prix, l'avis, l'identifiant de l'utilisateur et le numéro de plaque en fichier JSON.

##### *Fonction « getLocationsJSONByPlaque »*

Cette fonction permet de récupérer toutes les locations présentes dans la base de données. Elle est utilisée pour la page « managelocation.php ». Elle reçoit en paramètre le numéro de plaque, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « locations ». Elle retourne une transformation d'un tableau à deux dimensions, dont chaque ligne contient la date de réservation, la date de début, la date de fin, le prix, l'avis, l'identifiant de l'utilisateur et le numéro de plaque en fichier JSON.

##### *Fonction « getLocationsUserMoto »*

Cette fonction récupère tous les enregistrements de la table « locations » ainsi que le pseudo de l'utilisateur qui a réservé et la marque de la moto qui viennent de la table utilisateurs et motos. Elle est utilisée sur la page « managelocation.php ». Elle ne reçoit pas de paramètre, c'est un SELECT avec une jointure entre l'identifiant de l'utilisateur et une seconde jointure entre le numéro de plaque.

Elle retourne un tableau à deux dimensions, dont chaque ligne contient la date de réservation, la date de début, de date de début, le prix, l'avis, l'identifiant de l'utilisateur, le numéro de plaque, le pseudo, l'email et la marque.

#### *Fonction « getLocationsByPlaque »*

Cette fonction permet de récupérer toutes les locations présentes dans le base de données. Elle est utilisée pour la page « managelocation.php ». Elle reçoit en paramètre le numéro de plaque, vu qu'il s'agit simplement d'un SELECT général effectué sur la table « locations ». Elle retourne un tableau à deux dimensions, dont chaque ligne contient la date de réservation, la date de début, la date de fin, le prix, l'avis, l'identifiant de l'utilisateur et le numéro de plaque.

#### *Fonction « addLocation »*

Cette fonction ajoute un enregistrement dans le table « locations ». Elle est utilisée avec la page « savelocation.php ». Elle reçoit en paramètre chaque champ nécessaire pour une location : la date de réservation, la date de début, la date de fin, le prix, l'avis, l'identifiant de l'utilisateur et le numéro de plaque. Ils sont tous utilisé comme valeur à insérer. Elle retourne « True » en cas de succès ou « False » en cas de problème.

#### *Fonction « addAvis »*

Cette fonction ajoute un avis sur une réservation dans la table locations. Elle est utilisée dans le fichier « saveadvice.php ». Elle reçoit en paramètre l'identifiant de la location et l'avis à inséré. Ils sont utilisés comme valeur à mettre à jour. Elle renvoie « TRUE » en cas de succès ou « FALSE ».

#### *Fonction « updateLocation »*

Cette fonction permet de mettre à jour des informations concernant une location existante. Elle est utilisée dans le contrôleur « managelocation.php » pour en changer les données. Elle reçoit en paramètre le nom de l'image et le lien de l'image. Les dates sont utilisées comme valeur à mettre à jour. Elle retourne « True » en cas de succès ou « False » en cas d'erreur.

#### *Fonction « deleteLocation »*

Cette fonction permet de supprimer un enregistrement de la table « locations ». Elle est dans le contrôleur « deletelocation.php » où, après confirmation, il est justement possible pour l'administrateur de supprimer une réservation. Elle reçoit en paramètre l'identifiant de la location à supprimer qui est utilisé dans la clause WHERE du DELETE. Elle retourne « True » en cas de succès ou « False » dans le cas contraire.

#### *Fonction « alreadyLocationExist »*

Cette fonction permet de contrôler si l'utilisateur qui souhaite effectuer une réservation n'en a pas déjà une en cours. Elle est utilisée dans le contrôleur « savelocation.php » au moment d'ajouté une réservation, elle effectue un contrôle à ce moment-ci. Elle reçoit en paramètre l'identifiant de l'utilisateur. Ce dernier est utilisé dans la clause WHERE du SELECT sur la table « locations ». Le SELECT récupère la date de fin des réservations correspondant à l'identifiant de l'utilisateur. Dans cette fonction, une variable \$today est initialisé avec la date du jour. La fonction compare finalement si la date de fin intervient avant ou après la date du jour et retourne « TRUE » ou « FALSE » en conséquence

## CALENDRIER

Mois		Semaine		mai 2019							<< < > >>	
lun.	mar.	mer.	jeu.	ven.	sam.	dim.						
29	30	1	2	3	4	5						
6	7	8	9	10	11	12	00 h Reservé par: Admin Moto : Yamaha 123732					
13	14	15	16	17	18	19	Reservé par: Admin Moto : Yamaha 123732					
			00 h Reservé par: Ludo Moto : Ducato 345678									
20	21	22	23	24	25	26	Reservé par: Ludo Moto : Ducato 345678					
			00 h Reservé par: Admin Moto : Yamaha 123732									
27	28	29	30	31	1	2	Reservé par: Admin Moto : Yamaha 123732					
3	4	5	6	7	8	9	00 h Reservé par: Admin Moto : Kawasaki 123456					

Pour générer le calendrier, j'utilise Full Calendar. Cette librairie crée un calendrier suivant les paramètres que l'on définit.

Pour l'utiliser, il faut placer le dossier fullcalendar-4.1.0 à la racine de l'espace de travail et ajouté différents liens et scripts dans la balise <head> de notre page. On a parmi ceux-ci des plugins css pour un affichage correct des plugins utilisés, des plugins JS pour le fonctionnement et des plugins pour sélectionner une date dans le calendrier et pour mettre le calendrier en français.

Le calendrier est créé dans une balise <div> comprenant un identifiant « calendar ».

Un script JavaScript est codé en dehors du <body> pour que le HTML soit chargé en premier. Il va créer le calendrier dans la balise <div> qui contient l'identifiant « calendar ». L'entête est paramétré avec une vue par mois ou par semaine à gauche, le mois en cours ou la semaine en cours au milieu comme titre et les différents boutons de navigation à droite. La grille par mois est définie par défaut avec la langue en français et la possibilité de sélectionner un jour activée. La fonction « dateClick » place la date du jour sur lequel on vient de cliquer dans le formulaire. La fonction « select » place les dates sélectionnées avec un appui de souris constant dans le formulaire, logiquement le premier jour dans le champs « date de début » et le dernier jour dans le champs « date de fin ». Elle calcule aussi la différence entre le dernier jour et le premier pour actualisé le prix en conséquence. La fonction « selectAllow » empêche de sélectionner des jours déjà passés.

## ENVOIE DE MAIL

Le site utilise Swift Mailer 5 pour envoyer des emails. Swift Mailer est une solution de publipostage à base de composants pour PHP 7.

Une adresse mail a été créée exclusivement pour le site Locamoto : [locamoto.noreply@gmail.com](mailto:locamoto.noreply@gmail.com) avec le mot de passe : Super2018.

Pour utiliser Swift Mailer, le dossier contenant tous ses fichiers « swiftmailer5 » est placé à la racine du site. Un fichier « mailparam.php » contenant toutes les constantes relative à l'envoi de mail, c'est-à-dire, l'adresse mail qui envoie, son mot de passe, le serveur (SMTP : Simple Mail Transfer Protocol) etc.

Lorsque l'admin valide une réservation, le contrôleur « sendmail.php » va s'occuper d'inclure le fichier de configuration et de swiftmailer. Les constantes sont paramétrées et le corps de mail, donc son contenu se décide ici.

L'utilisateur va donc recevoir un mail de la part de [locamoto.noreply@gmail.com](mailto:locamoto.noreply@gmail.com) semblable à « Votre réservation du véhicule : Yamaha du : 29.05.2019 jusqu'au : 01.05.2019 a été approuvée au prix de : 25.- CHF.

Exemple :

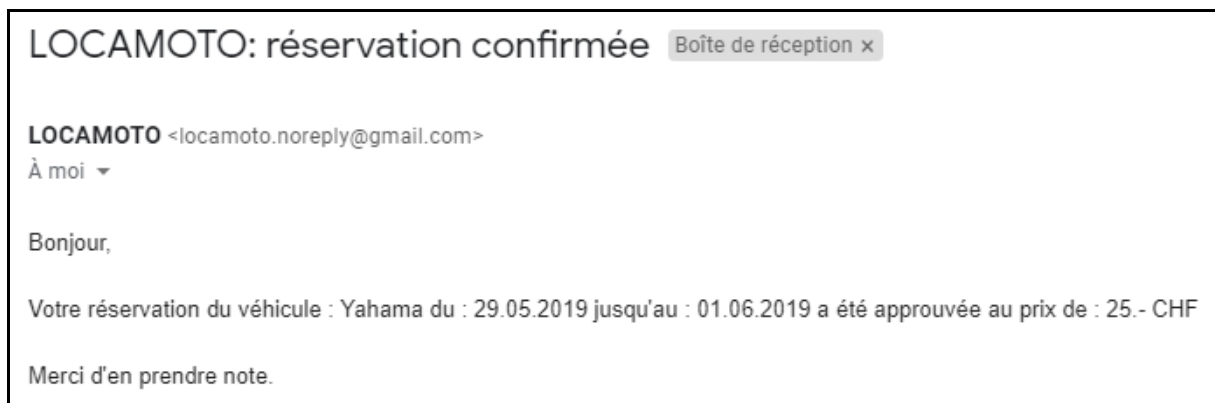


Figure 19: MAIL

Nota Bene : Le réseau de l'école ne permet pas d'envoyer des mails, pour implémenter et tester cette fonctionnalité le partage de connexion d'un smartphone a été utilisé.



## TESTS ET PROTOCOLE DE TESTS

## PLAN DE TESTS

*Tests sur la page « index.php »*

N°	A tester	Scénario	Résultat attendu
01	Recherche	Inscrire « Ya » dans le barre de recherche et cliquer sur rechercher.	Seules les motos « Yamaha » s'affiche sur la page.
02	Location	Se connecter et cliquer sur « Location ». Sélectionner du 22 mai ou 25 mai et cliquer sur « Louer ».	Notification « Réservation effectuée avec succès. » + La location est enregistrée dans le base de données.
03	Avis	Se connecter et cliquer sur « Laisser un avis ». Entrez « test » dans la zone de texte, cocher sur la première réservation et cliquer sur « Envoyer ».	Notification « Avis envoyé avec succès » + L'avis est enregistrer dans la base de données.
04	Retour en haut	Scroller jusqu'en bas du site et clique sur « Retour en haut ».	La page s'affiche tout en haut.

*Tests connexion et inscription*

N°	A tester	Scénario	Résultat attendu
05	Vérifier que le mot de passe et la confirmation de mot de passe correspondent.	Entrer « Super2018 » dans le champ mot de passe et « Super2019 » dans le champ confirmation de mot de passe.	Notification : « Les mots de passe ne correspondent pas ».
06	Vérifier que le mot de passe est crypté en sha1	Entrer toutes les informations demandées : - Prénom : Olivier - Nom : Mine - Pseudo : Oli - Email : <a href="mailto:Olivier.Mine@gmail.com">Olivier.Mine@gmail.com</a> - Date d'anniversaire : 01.01.1999 - Mot de passe : Super Et cliquer sur enregistrer.	Dans la base de données le champs mot de passe doit être : « f6889fc97e14b42de c11a8c183ea791c5465b658 »
07	Vérifier que l'on se connecte uniquement si le pseudo et le mot de passe correspondent.	Entrer « Marie » dans le champ pseudo et « Super » dans le champ mot de passe et cliquer sur se connecter.	Notification : « Les identifiants n'existe pas ».
08	Obliger de remplir tous les champs du formulaire	Entrer « Marie » dans le champ pseudo et cliquer sur se connecter.	Notification : « Veuillez remplir tous les champs ».
09	Connexion réussi	Entrer « Olivier » dans le champ pseudo et « Super » dans le champ mot de passe et cliquer sur se connecter.	Notification : « Bienvenue ! Vous êtes connecté ». + La barre de navigation affiche « déconnexion et profil »
10	Déconnexion	Cliquer sur déconnecté	La barre de navigation affiche « Se connecter » à droite

N°	A tester	Scénario	Résultat attendu
11	Les informations sont entrées automatiquement dans le formulaire.	Cliquer sur « profil ».	Les informations de l'utilisateur connecté sont placées dans le formulaire.

*Tests sur la page « manageuser.php »*

N°	A tester	Scénario	Résultat attendu
12	La page « manageuser.php »	Accéder à la page sans être connecté en tant qu'administrateur	Redirection sur la page d'accueil.
13	Ajouter un utilisateur	Cliquer sur « nouvel utilisateur », entrer toutes les informations demandées : - Prénom : Olivier - Nom : Mine - Pseudo : Oli - Email : <a href="mailto:Olivier.Mine@gmail.com">Olivier.Mine@gmail.com</a> - Date d'anniversaire : 01.01.1999 - Mot de passe : Super Et cliquer sur enregistrer.	L'utilisateur s'affiche dans la liste de gestion des utilisateurs avec les mêmes informations et dans la base de données.
14	Modifier un utilisateur	Cliquer sur le crayon à la ligne de l'utilisateur « Olivier » et remplacer son prénom « Olivier » par « Marie ».	L'information est changée.
15	Supprimer un utilisateur	Cliquer sur la croix à la ligne de l'utilisateur « Olivier » puis sur confirmer.	Notification : « Utilisateur supprimé avec succès ».

*Tests sur la page « manageparking.php »*

N°	A tester	Scénario	Résultat attendu
16	La page « gestionparking.php »	Accéder à la page sans être connecté en tant qu'administrateur.	Redirection sur la page d'accueil.
17	Ajouter une moto	Cliquer sur « nouvelle moto », entrer toutes les informations demandées : - N° Plaque : 987654 - Marque : Triomphe - Cylindrée : 800 cm3 -Couleur : Noire - Date d'immatriculation : 01.01.2016 - Image : triomphe.png Et cliquer sur enregistrer.	La moto s'affiche dans la liste de gestion du parking ainsi que sur la page d'accueil.

18	Modifier une moto	Cliquer sur le crayon à la ligne « Triomphe » et changer la marque par « Harley Davidfils ». Cliquer sur enregistrer.	L'information est changée.
19	Images incorrect	Entrer une image avec un type de fichier inconnu	Notification : « Format de l'image incorrect ».
20	Tous les champs sont remplis dans le formulaire	Cliquer sur « nouvelle moto », entrer les informations suivantes (tous sauf marque) : - N° Plaque : 987654 - Marque : Harley Davidfils - Cylindrée : 610 cm3 - Couleur : Noire - Date d'immatriculation : 01.01.2017 - Image : Harleydavidson.png Et cliquer sur enregistrer.	Notification : « Veuillez remplir tous les champs. »

*Tests sur la page « managelocation »*

N°	A tester	Scénario	Résultat attendu
21	La page « managelocation »	Accéder à la page sans être connecté en tant qu'administrateur.	Redirection sur la page d'accueil.
22	Valider une réservation	Cliquer sur le vu à la ligne de « Ludo » puis sur valider.	Notification : « Réservation validée », Un email est envoyé.
23	Email	Partager le réseau d'un smartphone par câble USB. Se connecter à l'adresse mail « ludo.jctdt@eduge.ch ». Cliquer sur le vu à la ligne où le pseudo correspond à « Ludo » puis sur valider.	Un mail est reçu dans la boîte de réception de « ludo.jctdt@eduge.ch »
24	Refuser une réservation	Cliquer sur la croix à la ligne de « Oli » puis sur confirmer.	La réservation est supprimée.
25	Calendrier Administrateur	Cliquer sur le bouton « Calendrier »	Un calendrier du mois en cours affiche toutes les réservations effectuées.

*Tests sur la page « location.php »*

N°	A tester	Scénario	Résultat attendu
26	Sélectionner des jours passé dans le calendrier	Cliquer et maintenir du 2 mai 2019 au 5 mai 2019.	Impossible de cliquer sur les jours précédant aujourd'hui
27	Modifier le prix manuellement	Choisir deux jours. Essayer d'éditer le prix dans le formulaire.	Impossible de cliquer.
28	Remplissage des champs quand on clique sur 1 jour.	Cliquer sur le 13 mai.	Le champs « date de début » se met à jour avec la date du 13 mai et le champs « date de fin » se met à jour avec la date du 14 mai.
29	Remplissage des champs quand on maintient une sélection	Cliquer sur 13 mai et maintenir jusqu'au 16 mai.	Le champs « date de début » se met à jour à jour avec la date du 13 mai et le champs « date de fin » se met à jour avec la date du 17 mai.
30	Enregistrer une réservation	Cliquer sur 20 mai et maintenir jusqu'au 23 mai. Puis cliquer sur « Louer ».	La réservation s'ajoute dans la base de données.
31	Enregistrer deux réservation	Cliquer sur 20 mai et maintenir jusqu'au 23 mai. Puis cliquer sur « Louer ». Cliquer sur 25 mai et maintenir jusqu'au 27 mai. Puis cliquer sur « Louer ».	La deuxième ne s'insère pas dans la base de données + Notification : « Vous avez déjà une location en cours ».

## RAPPORT DE TESTS

*Tests sur la page « index.php »*

N°	Etat	Résultat obtenu
01	OK	Uniquement deux motos s'affichent portant le nom de « Yamaha ».
02	OK	Une notification est affichée et la réservation a été prise en compte et donc ajoutée dans la base de données.
03	OK	Une notification est affichée et l'avis s'est ajouté à la bonne location.
04	OK	La page affiche les premières motos.

*Tests connexion et inscription*

N°	Etat	Résultat obtenu
05	OK	Une notification rouge s'affiche avec le texte : « Les mots de passe ne correspondent pas ».
06	OK	Le mot de passe dans la base de données est bien : « f6889fc97e14b42de c11a8c183ea791c5465b658 »
07	OK	Une notification rouge s'affiche bien avec le texte : « Les identifiants n'existe pas ».
08	OK	Une notification rouge s'affiche bien avec le texte : « Veuillez remplir tous les champs ».
09	OK	Une notification verte s'affiche bien avec le texte : « Bienvenu ! Vous êtes connecté ».
10	OK	Une notification comme prévu, l'utilisateur peut désormais faire des locations et laisser des avis.
11	OK	Avec le compte « Ludo » le nom, prénom, mail, pseudo et date de naissance sont bien mis d'office dans le formulaire.

*Tests sur la page « manageuser.php »*

N°	Etat	Résultat obtenu
12	OK	Le statut ne correspond pas à celui de l'administrateur donc redirection sur la page d'accueil.
13	OK	L'utilisateur est ajouté dans la base de données et s'affiche dans la liste des utilisateurs à gérer.
14	OK	Le nom est changé sur le site et la base de données. Une notification verte affiche : « Utilisateur mis à jour ».
15	OK	L'utilisateur est supprimé de la base. Une notification rouge affiche « Utilisateur supprimé ».

*Tests sur la page « manageparking.php »*

N°	Etat	Résultat obtenu
16	OK	Le statut ne correspond pas à celui de l'administrateur donc redirection sur la page d'accueil.
17	OK	Toutes les informations entrées dans le formulaire sont correctement insérées dans la base de données entre les tables « motos » et « images ». Une notification affiche « Motos ajoutée ».
18	OK	Sur le site et la base de données l'utilisateur s'appelle désormais « test2 ».
19	OK	Notification : « lors du téléchargement de l'image »
20	OK	La moto n'est pas ajoutée et une notification indique qu'un champ est vide.

*Tests sur la page « managelocation »*

N°	Etat	Résultat obtenu
21	OK	Le statut ne correspond pas à celui de l'administrateur donc redirection sur la page d'accueil.
22	OK	Notification : « Réservation approuvée, Email envoyé » L'email s'envoie si on utilise un réseau autre que celui de l'état.
23	OK	Email reçu : « Bonjour,  Votre réservation du : 16.05.2019 jusqu'au : 24.05.2019 a été approuvée au prix de : 69.- CHF  Merci d'en prendre note. »
24	OK	La réservation disparaît de la liste sur le site et elle n'est plus présente dans la base de données
25	OK	On a le même calendrier que la page location mais avec toutes les réservations du site. On peut naviguer de mois en mois ou semaine par semaine.

*Tests sur la page « location.php »*

N°	Etat	Résultat obtenu
26	OK	Les jours précédents le jour d'aujourd'hui sont désactivés, il n'y a rien de possible à faire.
27	OK	L'input est en « readonly » on ne peut pas l'éditer.
28	OK	Les dates s'actualisent automatiquement quand on clique sur un jour ainsi que le prix.
29	OK	Les dates s'actualisent automatiquement quand on clique sur un jour ainsi que le prix.
30	OK	Les données du formulaire sont enregistrées dans la base de données.
31	OK	Les données du formulaire ne sont pas insérées et une notification nous prévient qu'il est impossible d'avoir deux réservations en même temps.

## CONCLUSION

### BILAN PERSONNEL

Le fait de réaliser ce projet m'a permis de me rendre compte que la gestion du temps de travail est primordiale. En effet, le fait de développer un site complet seul avec une documentation technique en parallèle est une tâche intéressante à vivre.

Le calendrier fût bien plus simple à implémenter que je ne le pensais. Effectivement, en découvrant le cahier des charges, il constituait la tâche qui me semblait le plus vague et complexe à penser et réaliser. Finalement, je suis heureux de voir le résultat et d'avoir appris rapidement à l'utiliser pour parvenir à mes fins.

J'ai grandement apprécié développer ce site et rédiger la documentation nécessaire, cela m'a été très instructif et passionnant. Venir chaque jour avec des objectifs bien définis et simplement suivre le programme préétabli dans la planification initiale fût très satisfaisant. Peu à peu le projet prenait de l'ampleur et toutes les fonctionnalités finissaient par être implantées. Voir ainsi le site aboutir à un résultat concret en ayant démarché et produit tout son code en totale autonomie est plaisant.

### COMPARAISON DE PLANIFICATION

Au niveau du planning, j'ai séparé les tâches par fichiers type de fichier (MVC) et par fonctionnalité. J'ai rangé les tâches partie, analyse, vues, modèles, contrôleur, test et documentation. L'ordre de ces dernières correspond avec leurs équivalences, j'entends par là que la page de login dans la vue se trouve au même niveau que le contrôleur du login dans l'ordre d'apparition des tâches.

J'ai pu progresser rapidement par rapport au temps que je m'étais octroyé pour certaines fonctionnalités. Effectivement, pour exemple, le calendrier me semblait être une tâche fort importante et complexe vu qu'au paravent je n'avais pas exercé ce genre de fonction. Finalement, comme je le disais précédemment, il a été très simple de comprendre le fonctionnement de FullCalendar, ce qui a grandement réduit le temps réel que cela m'a pris.

Le journal de bord fût très utile pour remplir au fur et à mesure le planning effectif. Entre autre, j'ai toujours bien fait attention à inscrire tout ce que je réalisais dans la journée et il était très simple de reporter mon activité dans ce dernier.

### PROBLEMES RENCONTRES

Au niveau des principales fonctionnalités, il n'y a pas eu de problèmes majeurs. De simples fautes de frappe ou oublis ont constitué les seuls obstacles au bon déroulement du projet mais rien de très important pour autant.

Le réseau de l'état ne permet pas d'envoyer des mails en SSL. J'ai dû comprendre que ce n'était pas mon code le problème mais qu'il suffisait de se mettre avec un réseau partagé de mon smartphone. Heureusement, j'avais assez d'avance pour me permettre de perdre du temps sur cette activité et donc comprendre et arranger le problème.

## AMELIORATIONS POSSIBLES

Il existe plusieurs améliorations envisageables. Premièrement, il me semble logique qu'il faudrait donner la possibilité à l'administrateur de pouvoir supprimer une moto. Effectivement, si à l'avenir un modèle est hors d'usage, il faudrait pouvoir l'enlever du site et ne plus la proposer à la location.

Ensuite, il serait intéressant de lier plusieurs images par modèle de moto et donc afficher sur la page d'accueil un carrousel, pour chacune des motos disponibles à location, qui permette de défiler entre les images.

Finalement, il me semble pertinent de créer une page pour l'utilisateur qui contiendrait un tableau avec toutes ses réservations, passées, en attente et validée. Cela permettrait à l'utilisateur d'avoir une vue d'ensemble sur son activité sur le site.

## BIBLIOGRAPHIE

### CODE REPRIS

- Style de la barre de navigation : <https://getbootstrap.com/docs/4.2/examples/blog/>
- Le tutoriel de monsieur Dominique Aigro pour SwiftMailer 5

### SITE UTILISES

- Documentation Bootstrap : <https://getbootstrap.com/>
- Site W3Schools : <https://www.w3schools.com/>
- Site Stack Overflow : <https://stackoverflow.com/>
- Documentation PHP : <http://www.php.net/>
- Documentation SQL : <https://sql.sh/>
- Documentation FullCalendar : <https://fullcalendar.io>



## PLANNING

## PLANNING PREVISIONNEL

Tâches à réaliser	Temps nécessaire	07.05.2019	08.05.2019	09.05.2019	13.05.2019	14.05.2019	15.05.2019	16.05.2019	20.05.2019	21.05.2019	22.05.2019	23.05.2019	Total
<b>Analyse</b>													
Lecture de l'énoncé	01:00	01:00											01:00
Analyse + Planification	02:00	02:00											02:00
Conversion MCD->MPD	02:00	02:00											02:00
Maquette écran	03:00	03:00											03:00
Calendrier (se documenter)	04:00					02:00	02:00						04:00
<b>Vues</b>													00:00
Page d'accueil	02:00		02:00										02:00
Page Login / Logout	02:00		02:00										02:00
Page modifier profil utilisateur	01:00			01:00									01:00
Page gestion utilisateur	02:00				02:00								02:00
Page gestion moto	02:00					02:00							02:00
Page locations (calendrier)	01:00						01:00						01:00
Page gestions locations	02:00							02:00					02:00
<b>Modèles</b>													00:00
connexion base de données	01:00		01:00										01:00
table utilisateurs (CRUD)	01:00			01:00									01:00
table motos (CRUD - pas de delete)	01:00				01:00								01:00
table images (CRUD)	01:00				01:00								01:00
table locations (CRUD)	01:00					01:00							01:00
<b>Contrôleurs</b>													00:00
Accueil	02:00		02:00										02:00
Navigation	01:00		01:00										01:00
Login / Logout	02:00			02:00									02:00
Edition utilisateurs (ajouter/modifier)	02:00			02:00									02:00
Gestion utilisateurs	02:00				02:00								02:00
Gestion motos (créer/modifier)	01:00					01:00							01:00
Enregistrer location	02:00						02:00						02:00
Locations (calendrier)	04:00						01:00	03:00					04:00
Gestion locations (validation)	01:00							01:00					01:00
Email	02:00								02:00				02:00
<b>Test</b>													00:00
Debuggage	08:00								02:00	04:00	02:00		08:00
Protocole de test	06:00								02:00	02:00	02:00		06:00
Rapport de test	02:00											02:00	02:00
<b>Documentation</b>													00:00
Documentation technique	22:00			02:00	02:00	02:00	02:00	02:00	02:00	02:00	04:00	04:00	22:00
Résumé (page A4)	02:00											02:00	02:00
	88:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	88:00

Figure 20: PLANNING PREVISIONNEL

## PLANNING EFFECTIF

Tâches à réaliser	Temps nécessaire	1er jour	2e jour	3e jour	4e jour	5e jour	6e jour	7e jour	8e jour	9e jour	10e jour	11e jour	Total
<b>Analyse</b>													
Lecture de l'énoncé	01:00	01:00											00:00
Analyse + Planification	02:00	02:00											01:00
Conversion MCD->MPD	02:00	01:00											02:00
Maquette écran	03:00	03:00											01:00
Use-case	01:00		01:00										03:00
Calendrier (se documenter)	04:00					02:00							01:00
<b>Vues</b>													02:00
Page d'accueil	02:00		00:15	01:00									01:15
Page Login / Logout	02:00		00:15										00:15
Page modifier profil utilisateur	01:00		00:30	00:30									01:00
Page gestion utilisateur	02:00				01:00								01:00
Page gestion moto	02:00				01:00								01:00
Page formulaire moto	00:30				00:30								00:30
Page locations (calendrier)	01:00					01:00							01:00
Page gestions locations	02:00					01:00							01:00
Page avis	01:00						01:00						01:00
<b>Modèles</b>													
connexion base de données	00:30		00:30										00:30
table utilisateurs (CRUD)	01:00			00:30									00:30
table motos (CRUD - pas de delete)	01:00			00:30									00:30
table images (CRUD)	01:00				00:30								00:30
table locations (CRUD)	01:00					01:00	01:00						02:00
<b>Contrôleurs</b>													
Accueil	02:00		00:30										00:30
Navigation	01:30			00:30									00:30
Login / Logout	02:00		01:00	00:30									01:30
Edition utilisateurs (ajouter/modifier/supprimer)	02:00			01:30									01:30
Gestion utilisateurs	02:00				01:00								01:00
Gestion motos (créer/modifier)	01:00				01:00								01:00
Edition motos (ajouter / modifier)	01:00				01:00								01:00
Enregistrer location	02:00					01:00				02:00			03:00
Locations (calendrier)	04:00						02:00						02:00
Gestion locations (validation/suppression)	01:00						01:00						01:00
Avis	02:00						02:00						02:00
Email	02:00							02:00					02:00
Fonction Rechercher	01:00			01:00									01:00
<b>Test</b>													
Debuggage	08:00							01:00	02:00	02:00	01:00	02:00	08:00
Protocole de test	06:00					00:30		02:30	03:00	01:00			07:00
Rapport de test	02:00								02:00	01:00	02:00		05:00
<b>Documentation</b>													
Documentation technique	22:00	01:00	04:00	02:00	02:00	01:30	01:00	02:30	01:00	02:00	02:00	03:00	22:00
Read.me												02:00	
Résumé (page A4)	02:00										03:00	01:00	04:00
	94:30	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	08:00	88:00

Figure 21: PLANNING EFFECTIF

## TABLE DES ILLUSTRATIONS

### FIGURES

Figure 1: BARRE DE NAVIGATION D'UN VISITEUR ANONYME .....	6
Figure 2: BARRE DE NAVIGATION D'UN UTILISATEUR AUTHENTIFIE .....	6
Figure 3: BARRE DE NAVIGATION D'UN ADMINISTRATEUR .....	6
Figure 4: PAGE D'ACCUEIL EN TANT QUE VISITEUR .....	7
Figure 5 : PAGE D'ACCUEIL EN TANT QU'UTILISATEUR CONNECTE .....	8
Figure 6: PAGE DE LOCATION .....	9
Figure 7: PAGE DE GESTIONS DES UTILISATEURS .....	10
Figure 8: PAGE DE GESTION DU PARKING .....	11
Figure 9: PAGE DE GESTION DES LOCATIONS .....	12
Figure 10: FORMULAIRE DE SAUVEGARDE D'UTILISATEURS .....	13
Figure 11: FORMULAIRE DE SAUVEGARDE DE MOTOS .....	14
Figure 12: FORMULAIRE DES AVIS .....	15
Figure 13: LOGIN .....	16
Figure 14: CONFIRMATION DE SUPPRESSION .....	16
Figure 15: TABLE UTILISATEURS .....	23
Figure 16: TABLE LOCATIONS .....	23
Figure 17: TABLE MOTOS .....	23
Figure 18: TABLE IMAGES .....	23
Figure 19: MAIL .....	31
Figure 20: PLANNING PREVISIONNEL .....	40
Figure 21: PLANNING EFFECTIF .....	41

### GLOSSAIRE

Mots	Définitions
Css	Langage informatique de mise en forme
dateClick	Événement lors d'un clic sur un événement du calendrier
MVC	Modèle-Vue-Contrôleur
HTML	Langage de balisage utilisé pour la création de pages web
JavaScript	Langage de script orienté objet
Readonly	Lecture seule
Input	Champs de saisie
INSERT	Commande SQL d'insertion
SELECT	Commande SQL de sélection
Textarea	Champs de saisie multi-ligne
Template	Exemple sur lequel on se base
UPDATE	Commande SQL de mise à jour

### ANNEXES

- Code sources
- Résumé
- Journal de bord manuscrit