

DOCUMENTATION TECHNIQUE  
SOS INFOBOBO

Jonathan Borel-Jaquet. 23.05.2019. V.1.0



# TABLE DES MATIÈRES

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Rappel du cahier des charges</b>	<b>4</b>
2.1	Objectifs	4
2.2	Spécifications	4
2.3	Restrictions	4
2.4	Environnement	5
2.5	Organisation	5
2.6	Livrable	5
<b>3</b>	<b>Analyse fonctionnelle</b>	<b>6</b>
3.1	Fonctionnalités	6
3.1.1	Créer une demande de réparation informatique	6
3.1.2	Changer le statut d'une demande de réparation	6
3.1.3	Créer un avis sur le service effectué	6
3.1.4	Valider la publication d'un avis	6
3.1.5	Enregistrer un rendez-vous	6
3.1.6	Modifier les informations personnelles	7
3.1.7	Consulter les statistiques de réparations effectuées	7
3.1.8	Consulter les différentes pages du site	7
3.1.9	Envoyer un email lors d'un changement de statut d'une demande	7
3.1.10	Envoyer un email lors de la création d'une nouvelle demande	7
3.1.11	Se connecter à l'application	7
3.2	Interfaces	8
3.2.1	Barre de navigation	8
3.2.2	Page d'accueil	9
3.2.3	Page de contact	10
3.2.4	Page d'avis	11
3.2.5	Page de connexion	12
3.2.6	Page calendrier	13
3.2.7	Page d'administration des demandes de réparation informatique	14
3.2.8	Page de modification des informations personnelles	15
3.2.9	Page d'administration des avis	16
3.2.10	Page de statistiques	17
3.3	Cas d'utilisation	18
3.3.1	Diagramme des cas d'utilisation	18

<b>4</b>	<b>Analyse organique</b>	<b>19</b>
<b>4.1</b>	<b>Base de données</b>	<b>19</b>
4.1.1	Modèle physique de données finales	19
4.1.2	L'accès à la base de données	20
4.1.3	Détails des tables	21
<b>4.2</b>	<b>Architecture du code</b>	<b>23</b>
4.2.1	Arborescence de fichiers	23
4.2.2	Classe POO	24
4.2.3	Classes utilisées	25
<b>4.3</b>	<b>Outils externes</b>	<b>35</b>
4.3.1	Bootstrap 4	35
4.3.2	GitHub	36
4.3.3	Font Awesome	36
4.3.4	FullCalendar	37
4.3.5	Swift Mailer	37
<b>5</b>	<b>Tests</b>	<b>37</b>
<b>5.1</b>	<b>Utilisateur non-connecté</b>	<b>37</b>
5.1.1	Connexion et déconnexion	37
5.1.2	Page de création d'une demande de réparation informatique	39
5.1.3	Page de création d'avis	40
5.1.4	Page d'affichage des informations personnelles du réparateur	40
<b>5.2</b>	<b>Utilisateur connecté</b>	<b>41</b>
5.2.1	Page de modification des informations personnelles du réparateur	41
5.2.2	Page d'administration des demandes de réparation informatique	42
5.2.3	Page d'administration des avis	42
5.2.4	Page d'affichage des statistiques	43
5.2.5	Page d'affichage et planification des rendez-vous	44
<b>6</b>	<b>Conclusion</b>	<b>45</b>
<b>6.1</b>	<b>Améliorations possibles</b>	<b>45</b>
<b>6.2</b>	<b>Comparaison plannings</b>	<b>45</b>
<b>6.3</b>	<b>Bilan personnel</b>	<b>45</b>
<b>7</b>	<b>Bibliographie</b>	<b>46</b>
<b>7.1</b>	<b>Codes repris</b>	<b>46</b>
<b>7.2</b>	<b>Sites utilisés</b>	<b>46</b>
<b>8</b>	<b>Planning</b>	<b>47</b>
<b>8.1</b>	<b>Planning prévisionnel</b>	<b>47</b>

8.2	Planning effectif	48
9	<i>Table des illustrations</i>	49
9.1	Figures	49
9.2	Tableaux	49
10	<i>Annexes</i>	50
11	<i>Glossaire</i>	50

# 1 INTRODUCTION

Cette documentation a pour but de détailler les étapes nécessaires à la réalisation de l'application web « Sos Infobobo », réalisé dans le cadre du TPI (Travail Pratique Individuel).

« Sos Infobobo » est une application WEB qui permet de principalement gérer des rendez-vous entre un réparateur informatique qui sera l'unique utilisateur connecté et des potentiels clients qui seront des utilisateurs non-connectés. L'application permet également aux clients de poster des avis sur les potentielles réparations effectuées qui devront d'abord être validées par le réparateur. Le réparateur, une fois connecté, a quant à lui accès à un calendrier avec ses différents rendez-vous planifiés ainsi qu'aux statistiques des réparations effectuées. Il aura également la possibilité de modifier ses informations personnelles dans une page dédiée pour cela.

## 2 RAPPEL DU CAHIER DES CHARGES

### 2.1 OBJECTIFS

Durant 11 jours de travail, soit 88 heures, le but est de concevoir une application web qui permettra à un utilisateur non-connecté de faire une demande de réparation informatique au réparateur du site lequel est l'unique utilisateur connecté du site. Le réparateur pourra alors lire sa demande et choisir si cette demande entre dans ses compétences ou non. Le réparateur devra valider les potentiels avis postés sur le site et aura accès à différentes statistiques et modifications du site web.

### 2.2 SPÉCIFICATIONS

L'application permettra de :

- Faire une demande de réparation informatique
- Visionner le statut et le détail de toutes les demandes de réparation informatique
- Poster un avis sur la potentielle réparation
- Valider un avis précédemment posté par un client pour sa publication
- Modifier les informations personnelles de la page d'accueil
- Inscrire différents rendez-vous planifiés dans un calendrier
- Consulter les statistiques des réparations effectuées
- Se connecter à l'application
- Informer par email le réparateur lors de la création d'une nouvelle demande
- Informer par email le client lors d'un changement de statut de sa demande

### 2.3 RESTRICTIONS

L'utilisateur non-connecté peut :

- Faire une demande de réparation informatique
- Poster un avis sur la potentielle réparation
- Informer par email le réparateur lors de la création d'une nouvelle demande

L'utilisateur connecté peut :

- Se connecter à l'application
- Visionner le statut et le détail de toutes les demandes de réparation informatique
- Valider un avis précédemment posté par un client pour sa publication
- Modifier les informations personnelles de la page d'accueil
- Inscrire différents rendez-vous planifiés dans un calendrier
- Consulter les statistiques des réparations effectuées
- Informer par email le client lors d'un changement de statut de sa demande

## 2.4 ENVIRONNEMENT

- Un PC standard école avec Windows 10, 2 écrans
- Serveur Web et SGBD à choix (EasyPHP, Wamp, Laragon, Adminr, phpMyAdmin, autre)
- IDE à choix (NetBeans, Visual Studio Code, PHPStorm, autre)
- Logiciel de création de schémas (Visio, Gliffy, autre)
- Outil de versionnage de code (Git, avec dépôt distant sur Github / Bitbucket / GitLab)
- Navigateur web (Mozilla Firefox/Google Chrome)
- Logiciel de création de maquettes d'interfaces utilisateur (Sketch, Adobe XD, Proto.io, autre)
- Outil bureautique à choix pour les documents (Google Docs, MSOffice, OpenOffice)

## 2.5 ORGANISATION

Élève :

- Jonathan Borel-Jaquet, **jonathan.brljq@eduge.ch**

Maître d'apprentissage :

- Jasmina Travnjak, **jasmina.travnjak@edu.ge.ch**

Experts :

- Carol Quarroz, **cquarroz@gmail.com**
- Nicolas Terrond, **nicolas@terrond.com**

## 2.6 LIVRABLE

Pour la fin du TPI, le 23 mai 2019 :

Pour les experts et la formatrice par email :

- Planning détaillé du projet
- Rapport de projet contenant le code source au format PDF
- Journal de bord
- Résumé du TPI (1 page A4)

Pour la formatrice uniquement :

- L'accès au repository distant du projet avec les droits de « clone »
- Un readme explicitant l'installation du projet en local
- Un dump de la base de données contenant la structure ainsi qu'un set de données de test

## 3 ANALYSE FONCTIONNELLE

### 3.1 FONCTIONNALITÉS

#### 3.1.1 CRÉER UNE DEMANDE DE RÉPARATION INFORMATIQUE

Cette fonctionnalité permet à un utilisateur non-connecté d'envoyer une demande de réparation au réparateur en signalant son problème informatique. L'utilisateur non-connecté devra remplir un formulaire avec les prérequis suivants :

- Nom
- Prénom
- Email
- Numéro de téléphone
- Description du problème informatique

#### 3.1.2 CHANGER LE STATUT D'UNE DEMANDE DE RÉPARATION

Cette fonctionnalité permettra au réparateur de modifier le statut d'une demande de réparation. Il pourra changer le statut d'une demande en :

- Traitée
- Refusée

#### 3.1.3 CRÉER UN AVIS SUR LE SERVICE EFFECTUÉ

Cette fonctionnalité permet à un utilisateur non-connecté de créer un avis sur la potentielle réparation effectuée qui sera affichée sur une page dédiée à cela.

#### 3.1.4 VALIDER LA PUBLICATION D'UN AVIS

Cette fonctionnalité permet à un utilisateur connecté de valider ou refuser un avis précédemment créé par un utilisateur non-connecté.

#### 3.1.5 ENREGISTRER UN RENDEZ-VOUS

Cette fonctionnalité permet à un utilisateur connecté d'inscrire une demande de réparation de type ouverte dans un calendrier, il peut choisir une date de début pour la récupération et une date de fin pour la reddition. Une fois le rendez-vous enregistré, le type de demande de réparation passe à « En cours ».

### 3.1.6 MODIFIER LES INFORMATIONS PERSONNELLES

Cette fonctionnalité permet à un utilisateur connecté de modifier ses informations personnelles de la page d'accueil. Il pourra modifier les champs suivants :

- Numéro de téléphone
- Email
- Tarif
- Description

### 3.1.7 CONSULTER LES STATISTIQUES DE RÉPARATIONS EFFECTUÉES

Cette fonctionnalité permet à un utilisateur connecté de consulter les statistiques des réparations qu'il a effectuées. Elles seront listées par mois et par année.

### 3.1.8 CONSULTER LES DIFFÉRENTES PAGES DU SITE

Cette fonctionnalité permet à un utilisateur non-connecté de consulter les différentes pages du site mises à sa disposition, comme les pages :

- Accueil (Information sur le réparateur)
- Contact (Formulaire de demande de réparation)
- Avis (Liste des avis de réparation)

### 3.1.9 ENVOYER UN EMAIL LORS D'UN CHANGEMENT DE STATUT D'UNE DEMANDE

Cette fonctionnalité permet à un utilisateur non-connecté d'être averti par email de tout changement de statut de sa demande (Réf : **3.1.2 Changer le statut d'une demande de réparation**).

### 3.1.10 ENVOYER UN EMAIL LORS DE LA CRÉATION D'UNE NOUVELLE DEMANDE

Cette fonctionnalité permet à un utilisateur connecté d'être averti par email de la création d'une nouvelle demande (Réf : 3.1.1 **Créer une demande de réparation informatique**).

### 3.1.11 SE CONNECTER À L'APPLICATION

Cette fonctionnalité permet à un utilisateur qui possède un compte de pouvoir se connecter et avoir accès aux différentes fonctionnalités d'administration.



## 3.2 INTERFACES

### 3.2.1 BARRE DE NAVIGATION

#### 3.2.1.1 UTILISATEUR NON-CONNECTÉ



Fig. 1 - Barre de navigation d'un utilisateur non-connecté

1. Le lien « Accueil » permet à l'utilisateur d'aller à la page d'accueil (voir **Fig. 3 – Page d'accueil**).
2. Le lien « Contact » permet à l'utilisateur d'aller sur la page de contact (voir **Fig. 4 - Page de contact**).
3. Le lien « Avis » permet à l'utilisateur d'aller sur la page d'avis (voir **Fig. 5 - Page d'avis**).
4. Le lien « Connexion » permet à l'utilisateur d'aller sur la page de connexion du site (voir **Fig. 6 - Page de connexion**).

#### 3.2.1.2 UTILISATEUR CONNECTÉ

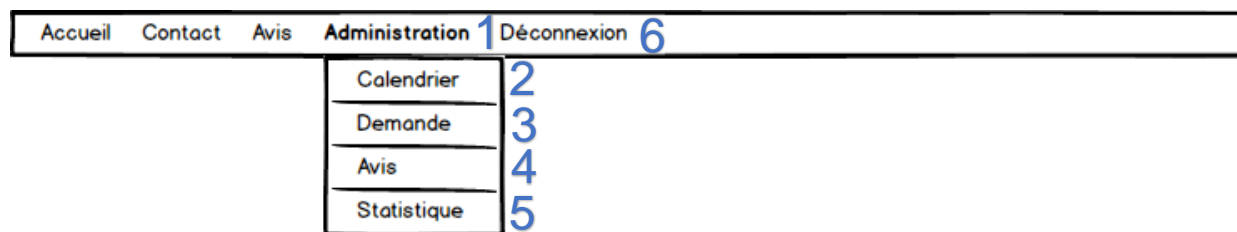


Fig. 2 - Barre de navigation d'un utilisateur connecté

1. Le lien « Administration » permet à l'utilisateur d'ouvrir le menu déroulant d'administration (voir **Fig. 2 - Barre de navigation d'un utilisateur connecté**).
2. Le lien « Calendrier » permet à l'utilisateur d'accéder à la page de planification de rendez-vous (voir **Fig. 7 - Page calendrier**).
3. Le lien « Demande » permet à l'utilisateur d'accéder à la page répertoriant toutes les demandes de réparation informatique (voir **Fig. 8 - Page d'administration des demandes de réparation informatique**).
4. Le lien « Avis » permet à l'utilisateur d'accéder à la page listant les avis en attente de validation (voir **Fig. 10 - Page d'administration des avis**).
5. Le lien « Statistique » permet à l'utilisateur d'accéder à la page de statistiques (voir **Fig. 11 - Page de statistiques**).
6. Le lien « Déconnexion » permet à l'utilisateur de se déconnecter du site.

### 3.2.2 PAGE D'ACCUEIL

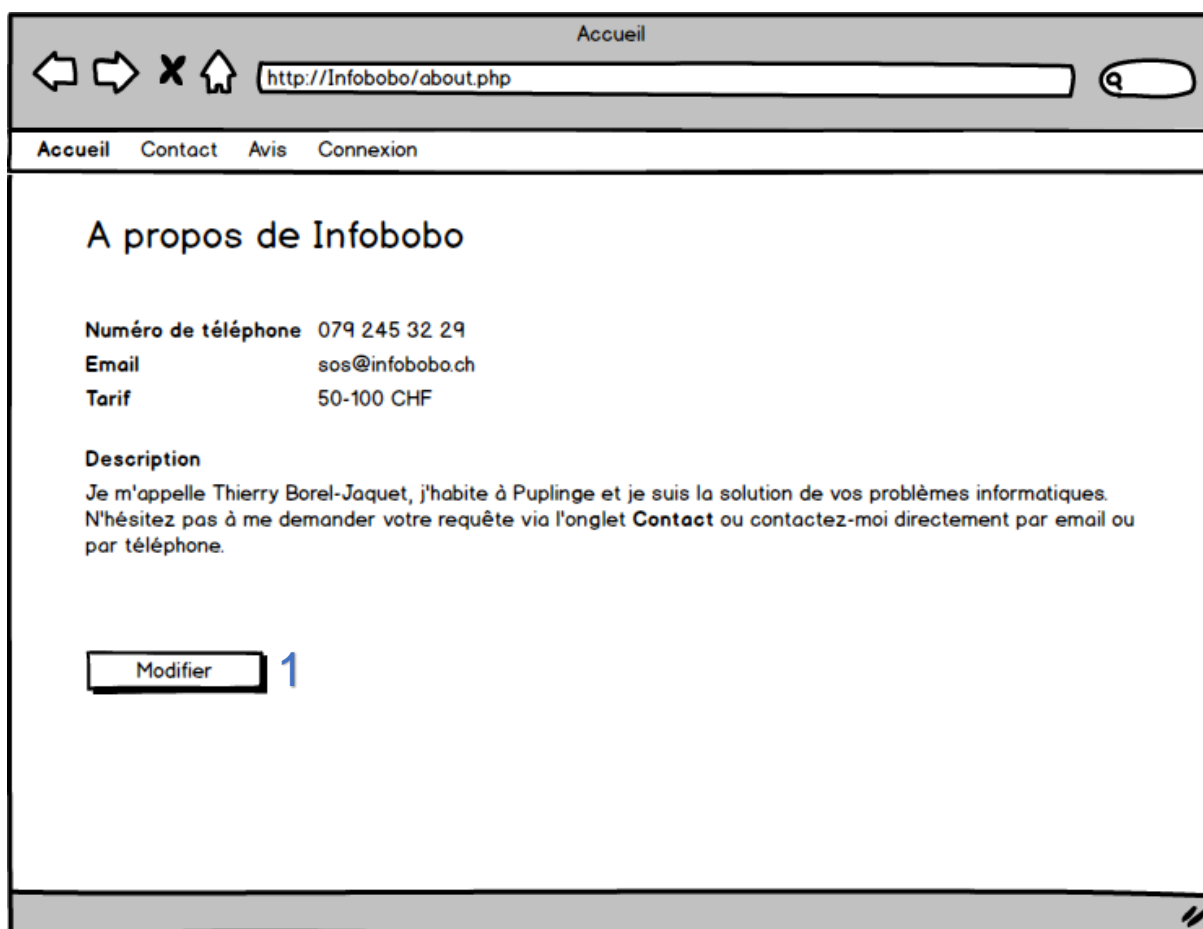


Fig. 3 - Page d'accueil

La page d'accueil accueille l'utilisateur et lui permet de consulter les informations personnelles du réparateur.

1. Le bouton « Modifier » apparaît uniquement si le réparateur est connecté. Il permet au réparateur d'aller sur la page de modification des informations personnelles (Voir **Fig. 9 - Page de modification des informations personnelles**).

### 3.2.3 PAGE DE CONTACT

Contact

http://Infobobo/contact.php

Accueil **Contact** Avis Connexion

### Faire une demande de réparation

Nom

Prénom

Email

Téléphone

Description du problème

Bonjour,

Prenez-vous en charge la réparation d'écran ? Si oui, mon écran d'ordinateur ne fonctionne plus et je souhaite le réparer.

Henry Durand

1

Fig. 4 - Page de contact

La page de contact permet à l'utilisateur de créer une demande qui sera soumise au réparateur.

1. Le bouton « Envoyer » permet d'envoyer la demande au réparateur, un pop-up confirme si l'envoi de la demande a bien été effectué ou non.

### 3.2.4 PAGE D'AVIS

Avis

http://Infobobo/opinion.php

Accueil Contact **Avis** Connexion

## Avis de réparation

Votre avis est important :

1

### Précédent avis

Très satisfait de la réparation que Monsieur Borel-Jaquet a effectuée. 13/05/2019

Service propre et pas cher ! 08/05/2019

Je connais Monsieur Borel-Jaquet depuis longtemps et je n'ai jamais été déçu de ses services. Je le recommande ! 08/05/2019

Fig. 5 - Page d'avis

La page d'avis permet à l'utilisateur de soumettre un avis sur la réparation effectuée. Il peut également consulter les précédents avis postés sur le site.

1. Le bouton « Soumettre l'avis » permet d'envoyer l'avis à l'administrateur qui devra ensuite le valider afin d'être affiché, un pop-up confirme si l'envoi de l'avis a bien été effectué ou non.

### 3.2.5 PAGE DE CONNEXION

Connexion

http://Infobobo/connection.php

Accueil Contact Avis Connexion

Connexion

Pseudo

Mot de passe

Connexion 1

Fig. 6 - Page de connexion

La page de connexion permet à un utilisateur de se connecter à l'application.

1. Le bouton « Connexion » permet de connecter l'utilisateur à l'application, un pop-up confirme si la connexion a réussi ou si le pseudo ou le mot de passe est incorrect.

### 3.2.6 PAGE CALENDRIER

Calendrier

http://Infobobo/adminCalendar.php

Accueil Contact Avis Administration Déconnexion

## Calendrier de rendez-vous

◀ MAY 2019 ▶

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

Date de récupération

Date de reddition

Planifier

Id	Nom	Prénom	Description
3	Borel	Jonathan	Bonjour, Mon téléphone ne fonctionne plus !
5	Alfieri	Elena	Mon ordinateur ne s'allume plus

Fig. 7 - Page calendrier

La page « Calendrier » permet au réparateur de choisir une date de début pour la récupération, une date de fin pour la reddition et une demande de réparation informatique afin de planifier un rendez-vous.

1. Une fois les dates sélectionnées, le champ de récupération (réf. 2) est rempli avec la date de début et le champ de reddition (réf. 3) est rempli avec la date de fin.
2. Zone de texte non modifiable contenant la date de récupération choisie par le réparateur.
3. Zone de texte non modifiable contenant la date de reddition choisie par le réparateur.
4. Le bouton « Planifier » permet au réparateur de planifier un rendez-vous une fois les informations bien sélectionnées (réf. 2, réf. 3, réf. 5).
5. Liste des demandes de statut « ouverte » : le réparateur doit cliquer sur l'une des demandes pour la sélectionner.

### 3.2.7 PAGE D'ADMINISTRATION DES DEMANDES DE RÉPARATION INFORMATIQUE

Administration des demandes

http://Infobobo/adminContact.php

Accueil Contact Avis **Administration** Déconnexion

#### Liste des demandes de réparation

Nom	Prénom	Email	Téléphone	Statut
Borel-Jaquet	Jonathan	jonathan@gmail.com	077 421 29 29	Ouverte
<div>Bonjour, Mon téléphone ne fonctionne plus, est-ce dans vos capacités ?</div>				<div>Modifier ▼ Refusée Traitée</div>
Alfieri	Elena	elena@gmail.com	078 324 21 21	En cours
<div>Bonjour, Mon ordinateur chauffe énormément, pouvez-vous m'aider ?</div>				<div>Modifier ▼</div>

Fig. 8 - Page d'administration des demandes de réparation informatique

La page d'administration des demandes de réparation informatique permet au réparateur de consulter toutes ces demandes et de modifier leur statut.

1. Le bouton « Modifier » permet au réparateur de choisir un nouveau statut pour la demande (Traitée ou Refusée).

### 3.2.8 PAGE DE MODIFICATION DES INFORMATIONS PERSONNELLES

The screenshot shows a web browser window titled 'Modification d'informations'. The address bar displays 'http://Infobobo/aboutUpdate.php'. The navigation menu includes 'Accueil', 'Contact', 'Avis', 'Administration', and 'Déconnexion'. The main content area is titled 'Modification de mes informations' and contains a form with the following fields:

- Numéro de téléphone:** 079 245 32 29
- Email:** sos@infobobo.ch
- Tarif:** 50-100 CHF
- Description:** Je m'appelle Thierry Borel-Jaquet, j'habite à Puplinge et je suis la solution de vos problèmes informatiques. N'hésitez pas à me demander votre requête via l'onglet **Contact** ou contactez-moi directement par email ou par téléphone.

At the bottom of the form is an 'Enregistrer' button, which is highlighted with a blue '1'.

Fig. 9 - Page de modification des informations personnelles

La page de modification des informations personnelles permet au réparateur de modifier les informations de la page d'accueil du site (voir **Fig. 3 - Page d'accueil**).

1. Le bouton « Enregistrer » permet au réparateur de modifier ses informations personnelles pour la page d'accueil du site.



### 3.2.9 PAGE D'ADMINISTRATION DES AVIS

Avis

http://Infobobo/adminOpinion.php

Accueil Contact Avis Administration Déconnexion

#### Avis en attente de validation

Super réparation, rien à dire !	17/05/2019	Valider 1	Refuser 2
Très satisfait du service ! merci Thierry	15/05/2019	Valider	Refuser
Au top !	14/05/2019	Valider	Refuser

Fig. 10 - Page d'administration des avis

La page d'administration des avis permet au réparateur de valider ou refuser un avis en attente de publication.

1. Le bouton « Valider » permet au réparateur de valider l'avis.
2. Le bouton « Refuser » permet au réparateur de refuser l'avis.

### 3.2.10 PAGE DE STATISTIQUES

Mois	Nombre de réparations effectuées
Janvier	5
Février	2
Mars	3
Avril	2
Mai	11
Juin	13
Juillet	0
Août	0
Septembre	0

Fig. 11 - Page de statistiques

La page de statistiques permet au réparateur de consulter le nombre de réparations faites par mois durant une année sélectionnée.

1. Le numérique « Up Down » permet au réparateur de sélectionner l'année dans laquelle il veut consulter le nombre de réparations.
2. Le bouton « Rechercher » permet au réparateur d'afficher le nombre de réparations par mois de l'année précédemment sélectionnée.

### 3.3 CAS D'UTILISATION

#### 3.3.1 DIAGRAMME DES CAS D'UTILISATION

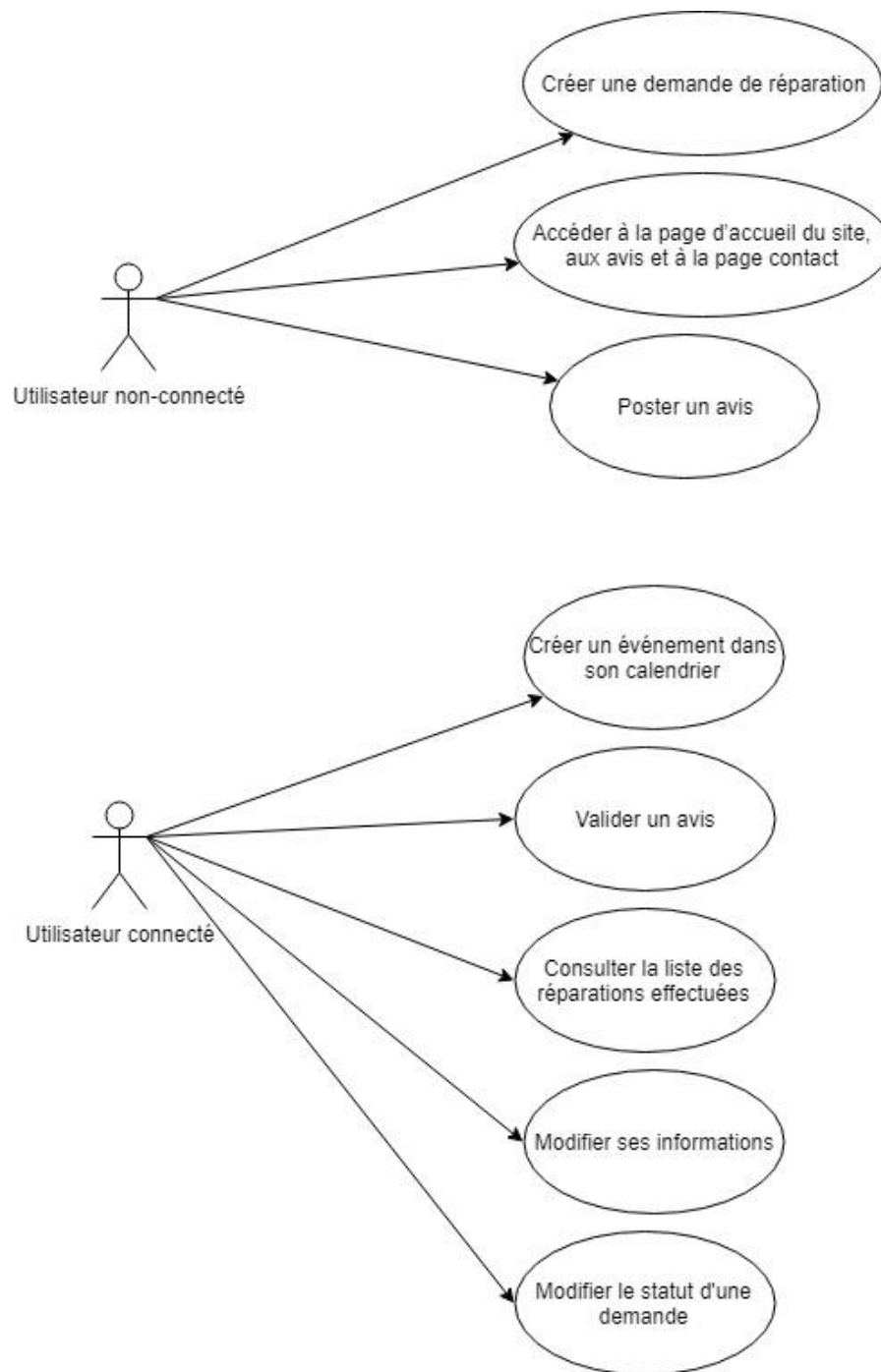


Fig. 12 - Diagramme des cas d'utilisation

## 4 ANALYSE ORGANIQUE

### 4.1 BASE DE DONNÉES

#### 4.1.1 MODÈLE PHYSIQUE DE DONNÉES FINALES

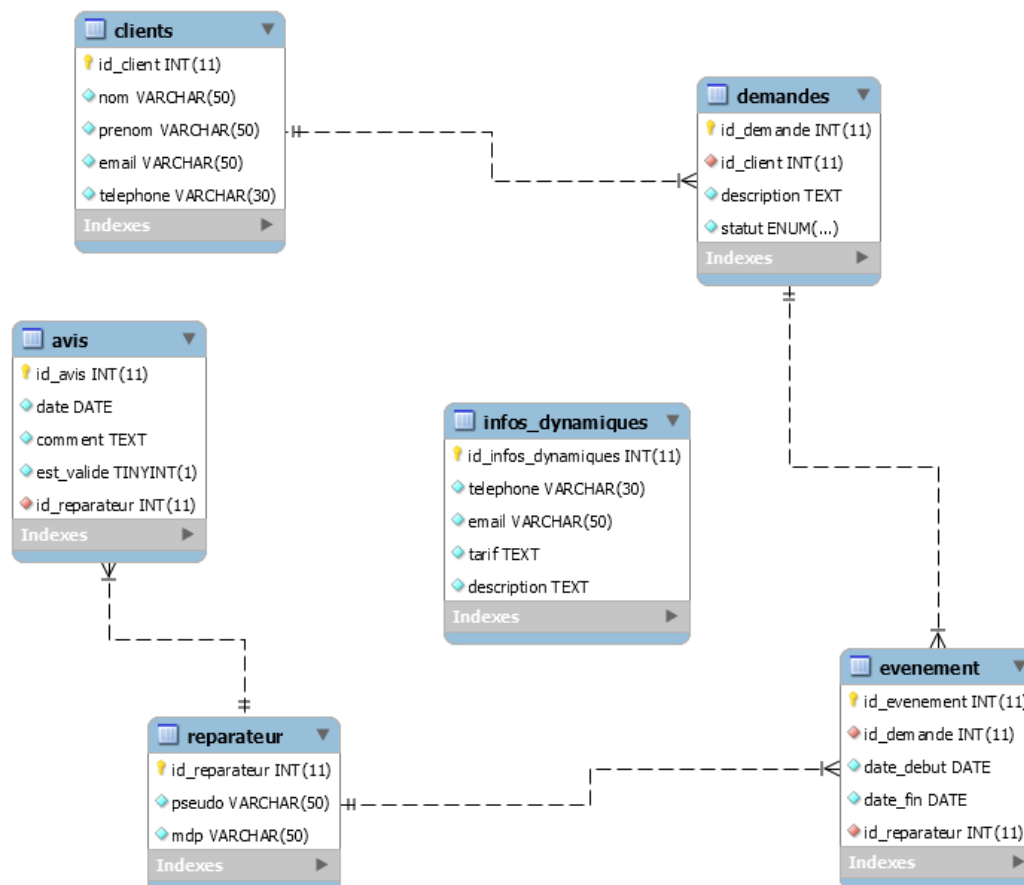


Fig. 13 - Modèle physique de données finales

Lors de mon développement, j'ai rencontré différents problèmes avec ma base de données initiale fournie par mon maître d'apprentissage.

Ma première modification a été sur la table « infos\_dynamiques ». En effet, elle ne contenait pas d'identifiant et de ce fait, la modification de ses champs était impossible pour cause de problème de sécurité MySQL empêchant la modification de champs sans la présence d'une colonne d'identifiant. J'ai donc ajouté un identifiant pour régler le problème.

Ma deuxième modification a été sur la table « evenement ». Afin de faciliter son utilisation et sa cohérence avec la librairie FullCalendar, j'ai supprimé les champs « date », « type » et « horaire » pour y ajouter les champs « date\_debut » et « date\_fin » : la date de début correspondant au jour de récupération du matériel informatique et la date de fin correspondant au jour de sa reddition. Pour ce qui est de l'horaire, j'ai pensé que cette donnée n'était pas la plus importante et qu'elle était superflue pour le réparateur.

## 4.1.2 L'ACCÈS À LA BASE DE DONNÉES

Dans le cadre de ce TPI j'ai utilisé une classe permettant l'accès à la base de données développée par Monsieur AIGROZ. J'ai utilisé cette classe durant ma troisième année (École Entreprise). Cette classe se nomme Database et retourne une instance PDO. Elle utilise le Design Pattern <sup>1</sup>Singleton.

Nom de fonction	Description
<i>private</i> <b>__construct()</b>	Constructeur privé pour que personne puisse créer une nouvelle instance via ' <code>= new Database();</code> '
<i>private</i> <b>__clone()</b>	Clone privé pour que personne ne puisse cloner l'instance
<i>public static</i> <b>getInstance()</b>	Retourne l'instance de la Database ou créer une connexion initiale
<i>final public static</i> <b>__callStatic</b> (\$chrMethod, \$arrArguments)	<p>Est lancée lorsque l'on invoque des méthodes inaccessibles dans un contexte statique</p> <p>Le paramètre <b>\$chrMethod</b> est le nom de la méthode appelée</p> <p>Le paramètre <b>\$arrArguments</b> est un tableau contenant les paramètres passés à la méthode <b>\$chrMethod</b></p>

Tableau 1 - Fonction de la classe Database

---

<sup>1</sup> Source Wikipédia : [https://fr.wikipedia.org/wiki/Patron\\_de\\_conception](https://fr.wikipedia.org/wiki/Patron_de_conception)

### 4.1.3 DÉTAILS DES TABLES

Abréviation utilisée	Signification
PK	Clé primaire
NN	Non Nul (Empêche la valeur NULL)
UQ	Unique (Empêche que la même valeur soit deux fois dans la table)
AI	Auto incrément

Tableau 2 - Légende des propriétés des attributs

#### 4.1.3.1 TABLE « CLIENTS »






Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_client	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(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 prenom	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 email	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 telephone	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 14 - Table « clients »

#### 4.1.3.2 TABLE « AVIS »






Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_avis	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"/>
 date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 comment	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 est_valide	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 id_reparateur	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 15 - Table « avis »

#### 4.1.3.3 TABLE « DEMANDES »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_demande	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"/>
 id_client	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 description	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 statut	ENUM('OUVERTE', 'ENCOURS', 'TRAITEE', 'REFUSEE')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 16 - Table « demandes »

## 4.1.3.4 TABLE «REPARATEUR »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_reparateur	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"/>
 pseudo	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 mdp	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 17 - Table « reparateur »

## 4.1.3.5 TABLE « EVENEMENT »







Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_evenement	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"/>
 id_demande	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 type	ENUM('RECUP', 'REDD')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 horaire	TIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 id_reparateur	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 18 - Table « evenement »

## 4.1.3.6 TABLE «INFOS\_DYNAMIQUES »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 telephone	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 email	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 tarif	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 description	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 19 - Table « infos\_dynamiques »

## 4.2 ARCHITECTURE DU CODE

### 4.2.1 ARBORESCENCE DE FICHIERS

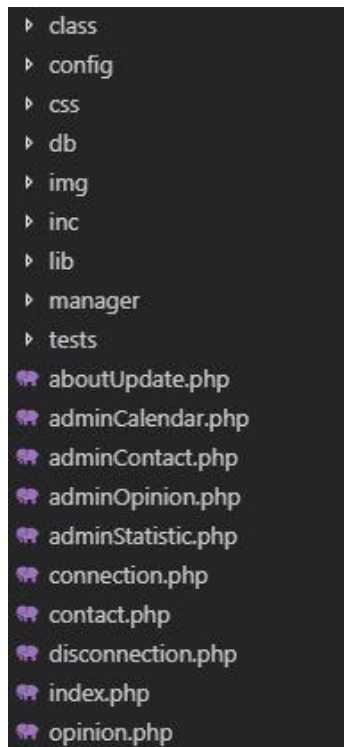


Fig. 20 - Arborescence de fichiers

- Le dossier « class » contient mes classes me servant de conteneur permettant à mon code de ne pas être dépendant de la base de données.
- Le dossier « config » contient mes fichiers de configuration pour la base de données ainsi que de Swift Mailer<sup>2</sup>.
- Le dossier « css » contient mon unique feuille de style.
- Le dossier « db » contient ma connexion à la base de données.
- Le dossier « img » contient les images utilisées de mon site.
- Le dossier « inc » contient mes fichiers à inclure dans toutes mes pages.
- Le dossier « lib » contient mes librairies externes utilisées, comme FullCalendar<sup>3</sup> et Swift Mailer.
- Le dossier « manager » contient tous mes managers contenant mes différentes fonctions.
- Le dossier « tests » contient tous mes fichiers de test pour chaque fonction utilisée
- Toutes les différentes vues du site sont mises à la racine.

---

<sup>2</sup> **Librairie externe** : Swift Mailer <https://swiftmailer.symfony.com/>

<sup>3</sup> **Librairie externe** : FullCalendar <https://fullcalendar.io/>



## 4.2.2 CLASSE POO

Dans le cadre de mon développement, j'ai décidé de créer des classes avec les tables dont j'avais besoin de ressortir des informations : il s'agit d'une sorte de conteneur. Ces classes contiennent uniquement des informations venant de la base de données, elles ne contiennent aucune méthode permettant d'accéder ou manipuler l'objet. Elles permettent à mes fonctions de ne pas être trop dépendantes de la base de données. Imaginons une modification de nom de champs en base, il faudra que je change ce nom à chaque utilisation de celui-ci. Mes conteneurs vont au contraire, permettre de modifier uniquement les fonctions les utilisant. Je décrirai cette pratique comme une sorte de dissociation entre mon code et la base de données.

J'ai également utilisé des managers (classe statique) qui contiennent uniquement des fonctions. Ce sont des classes qui permettent uniquement l'interaction avec les données pour lesquelles elles sont destinées. Une fonction permettant de créer un avis se trouvera dans le manager « OpinionManager » par exemple. Elles permettent de structurer mes différentes fonctions afin de les rendre plus facilement réutilisables pour moi ou pour une personne se greffant à mon projet.

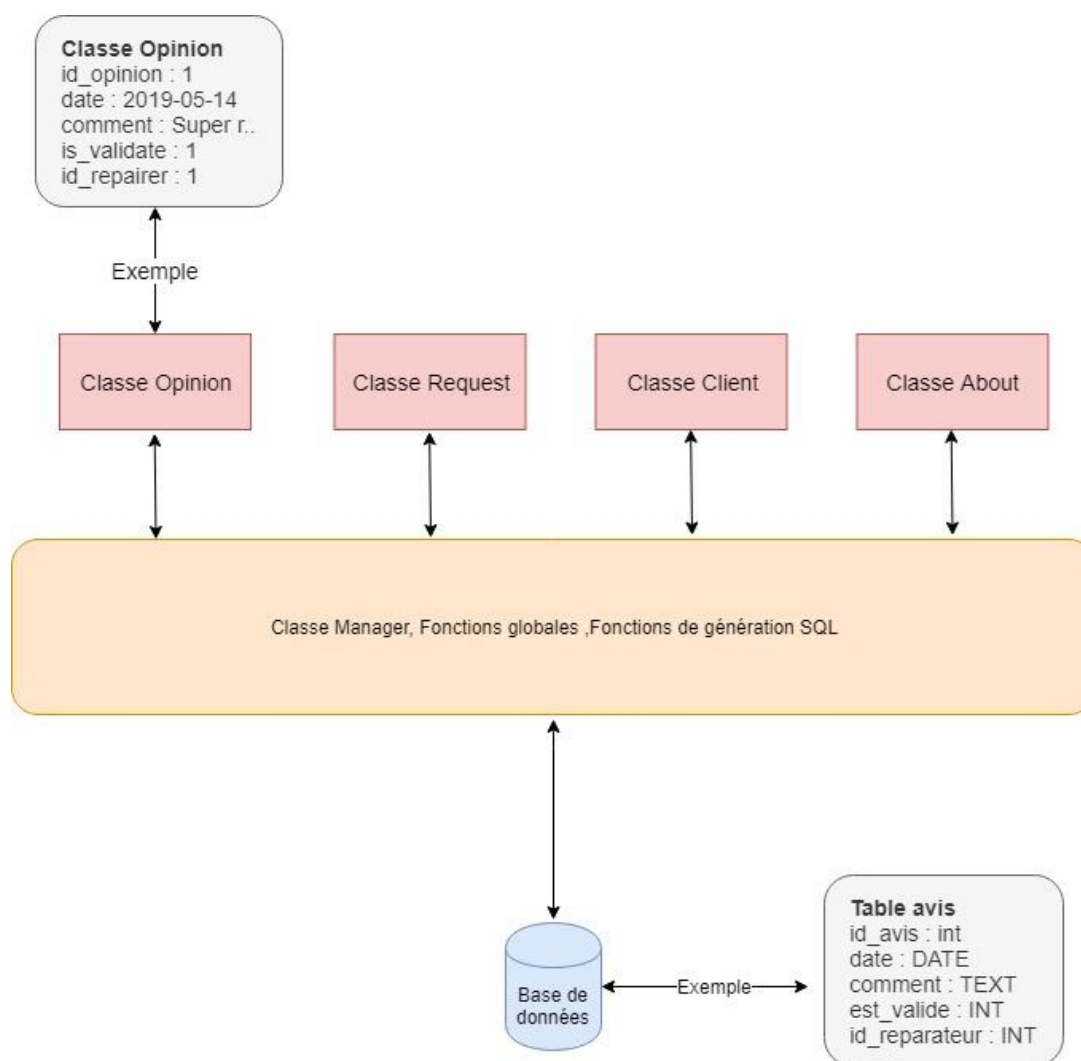


Fig. 21 - Diagramme du fonctionnement de l'interaction avec la base de données

## 4.2.3 CLASSES UTILISÉES

### 4.2.3.1 CLASSE ABOUT

Cette classe permet de stocker les informations de la table « infos\_dynamiques ».

Class About		
Variable	Type	Description
id_about	int	L'identifiant des informations dynamiques
phoneNumber	string	Le numéro de téléphone du réparateur
email	string	L'email du réparateur
price	string	Le prix du potentiel service
description	string	La description du réparateur et de ses services

Tableau 3 - Classe About

#### Exemple d'utilisation 1

```
$a = new About();  
$a->id_about = 1  
$a->phoneNumber = "077 421 34 20";  
$a->email = "sos@infobobo.ch";  
$a->price = "50 - 100 CHF";  
$a->description = "Bonjour, je m'appelle Thierry Borel-Jaquet et je suis la  
solut..";
```

#### Exemple d'utilisation 2

```
$a = new About(1,"077 421 34 20","sos@infobobo.ch","50 - 100 CHF","Bonjour,  
je m'appelle Thierry Borel-Jaquet et je suis la solut..");
```

#### 4.2.3.2 CLASSE CLIENT

Cette classe permet de stocker les informations de la table « clients ».

Class Client		
Variable	Type	Description
id_client	int	L'identifiant du client
firstName	string	Le nom du client
secondName	string	Le prénom du client
email	string	L'email du client
phoneNumber	string	Le numéro de téléphone du client

Tableau 4 - Classe Client

##### Exemple d'utilisation 1

```
$c = new Client();  
$c->id_client = 2;  
$c->firstName = "Coptère";  
$c->secondName = "Ellie";  
$c->email = "elli.coptere@gmail.com";  
$c->phoneNumber = "077 421 34 20";
```

##### Exemple d'utilisation 2

```
$c = new Client(2, "Coptère", "Ellie", "elli.coptere@gmail.com", "077 421 34 20");
```

#### 4.2.3.3 CLASSE OPINION

Cette classe permet de stocker les informations de la table « avis ».

Class Opinion		
Variable	Type	Description
id_opinion	int	L'identifiant de l'avis
date	string	La date de création de l'avis
comment	string	Le commentaire de l'avis
is_validate	bool	Si l'avis est validé ou non
id_repairer	int	L'identifiant du réparateur concerné

Tableau 5 - Classe Opinion

##### Exemple d'utilisation 1

```
$o = new Opinion();  
$o->id_opinion = 2;  
$o->date = "27/03/19";  
$o->comment = "Super réparation de Monsieur Borel-Jaquet, au top !";  
$o->is_validate = FALSE;  
$o->id_repairer = 1;
```

##### Exemple d'utilisation 2

```
$o = new Opinion(2, "27/03/19", "Super réparation de Monsieur Borel-Jaquet,  
au top !", FALSE, 1);
```

#### 4.2.3.4 CLASSE REQUEST

Cette classe permet de stocker les informations de la table « demandes ».

Class Request		
Variable	Type	Description
id_request	int	L'identifiant de la demande
id_client	int	L'identifiant du client ayant fait la demande
description	string	Description du problème
status	string	Statut de la demande

Tableau 6 - Classe Request

##### Exemple d'utilisation 1

```
$r = new Request();  
$r->id_request = 2;  
$r->id_client = 4;  
$r->description = "Bonjour, mon ordinateur à fréquemment des écrans bleus,  
est-ce dans vos compétences ?";  
$r->status = "OUVERTE";
```

##### Exemple d'utilisation 2

```
$r = new Request(1, 4, "Bonjour, mon ordinateur à...", "OUVERTE");
```

#### 4.2.3.5 ABOUTMANAGER

La classe « AboutManager » contient les fonctions en rapport avec les informations personnelles du réparateur, plus précisément de la table « infos\_dynamiques ».

Nom de la fonction	Description
GetAboutInformation()	Récupère les informations personnelles du réparateur dans un objet About
UpdateAboutInformation( <b>\$phoneNumber</b> , <b>\$email</b> , <b>\$price</b> , <b>\$description</b> )	<p>Modifie les informations personnelles du réparateur avec de nouvelles valeurs passées en paramètre</p> <p>Le paramètre <b>\$phoneNumber</b> est le numéro de téléphone du réparateur Le paramètre <b>\$email</b> est l'adresse email du réparateur Le paramètre <b>\$price</b> est le tarif de réparation du réparateur Le paramètre <b>\$description</b> est la description du réparateur</p>

Tableau 7 - Fonction de la classe AboutManager

#### 4.2.3.6 CLIENTMANAGER

La classe « ClientManager » contient les fonctions en rapport avec les informations personnelles du client, plus précisément de la table « clients ».

Nom de la fonction	Description
AddClient( <b>\$firstName</b> , <b>\$secondName</b> , <b>\$email</b> , <b>\$phoneNumber</b> )	<p>Enregistre les informations d'un client dans la base de données</p> <p>Le paramètre <b>\$firstName</b> est le nom du client Le paramètre <b>\$secondName</b> est le prénom du client Le paramètre <b>\$email</b> est l'adresse email du client Le paramètre <b>\$phoneNumber</b> est le numéro de téléphone du client</p>

Tableau 8 - Fonction de la classe ClientManager

#### 4.2.3.7 EVENTMANAGER

La classe EventManager contient les fonctions en rapport avec les événements créés par le réparateur, plus précisément de la table « evenement ».

Nom de la fonction	Description
AddEvent(\$id_request, \$dateEventStart, \$dateEventEnd)	Enregistre un événement et modifie le statut d'une demande dans la base de données  Le paramètre <b>\$id_request</b> est l'identifiant de la demande Le paramètre <b>\$dateEventStart</b> est la date de début de l'événement Le paramètre <b>\$dateEventEnd</b> est la date de fin de l'événement
GetAllEventFormatJSON()	Récupère tous les événements en format JSON destiné au calendrier JavaScript FullCalendar

Tableau 9 - Fonction de la classe EventManager

#### 4.2.3.8 MAILERMANAGER

La classe MailerManager contient les fonctions permettant d'envoyer un email en utilisant la librairie externe Swift Mailer.

Nom de la fonction	Description
<b>SendMailToRepairer(\$contactFirstName, \$contactSecondName, \$contactEmail, \$contactPhoneNumber, \$contactDescription)</b>	<p>Envoie un email au réparateur décrivant les informations de la demande du client en utilisant la librairie Swift Mailer</p> <p>Le paramètre <b>\$contactFirstName</b> est le nom du client Le paramètre <b>\$contactSecondName</b> est le prénom du client Le paramètre <b>\$contactEmail</b> est l'adresse email du client Le paramètre <b>\$contactPhoneNumber</b> est le numéro de téléphone du client Le paramètre <b>\$contactDescription</b> est la description du problème informatique du client</p>
<b>SendMailToClient(\$newStatus, \$client, \$request)</b>	<p>Envoie un email au client décrivant les informations de sa demande et le nouveau statut attribué à celle-ci en utilisant la librairie Swift Mailer</p> <p>Le paramètre <b>\$newStatus</b> est le nouveau statut de la demande Le paramètre <b>\$client</b> est un objet Client Le paramètre <b>\$request</b> est un objet Request</p>

Tableau 10 - Fonction de la classe MailerManager



#### 4.2.3.9 OPINIONMANAGER

La classe OpinionManager contient les fonctions en rapport avec les avis créés par les clients, plus précisément de la table « avis ».

Nom de la fonction	Description
AddOpinion( <b>\$comment</b> )	Enregistre un avis non validé dans la base de données  Le paramètre <b>\$comment</b> est le commentaire de l'avis
GetOpinionValidate()	Récupère tous les avis validés de la base de données
GetOpinionNotValidate()	Récupère tous les avis non validés de la base de données
ValidateOpinionById( <b>\$id_opinion</b> )	Valide un avis dans la base de données en utilisant son identifiant  Le paramètre <b>\$id_opinion</b> est l'identifiant de l'avis
DeleteOpinionById( <b>\$id_opinion</b> )	Supprime un avis dans la base de données en utilisant son identifiant  Le paramètre <b>\$id_opinion</b> est l'identifiant de l'avis

Tableau 11 - Fonction de la classe OpinionManager

#### 4.2.3.10 REQUESTMANAGER

La classe RequestManager contient les fonctions en rapport avec les demandes de réparation informatique créées par les clients, plus précisément de la table « demandes ».

Nom de la fonction	Description
AddRequest( <b>\$firstName</b> , <b>\$secondName</b> , <b>\$email</b> , <b>\$phoneNumber</b> , <b>\$description</b> )	<p>Enregistre une demande de réparation informatique et le client l'ayant créé dans la base de données</p> <p>Le paramètre <b>\$firstName</b> est le nom du client Le paramètre <b>\$secondName</b> est le prénom du client Le paramètre <b>\$email</b> est l'email du client Le paramètre <b>\$phoneNumber</b> est le numéro de téléphone du client Le paramètre <b>\$description</b> est la description du problème informatique du client</p>
GetAllRequest()	Récupère toutes les demandes de réparation informatique de la base de données
GetRequestById( <b>\$id_request</b> )	<p>Récupère une demande de réparation informatique avec les informations du client l'ayant créé de la base de données grâce à son identifiant</p> <p>Le paramètre <b>\$id_request</b> est l'identifiant de la demande</p>
UpdateRequestStatusById( <b>\$id_request</b> , <b>\$status</b> )	<p>Modifie le statut d'une demande de réparation informatique grâce à son identifiant.</p> <p>Le paramètre <b>\$id_request</b> est l'identifiant de la demande Le paramètre <b>\$status</b> est le nouveau statut de la demande</p>

GetOpenRequest()	Récupère toutes les demandes de réparation informatique de statut "Ouverte" de la base de données
GetProcessedRequestOrderByMonthAndYear( <b>\$year</b> )	Récupère toutes les demandes de réparation informatique de statut "Traitée" de la base de données triée par mois et par année  Le paramètre <b>\$year</b> est l'année choisie

Tableau 12 - Fonction de la classe RequestManager

#### 4.2.3.11 STYLEMANAGER

La classe StyleManager contient les fonctions permettant de rendre le site plus esthétique et lisible.

Nom de la fonction	Description
ColorStatus( <b>\$status</b> )	Retourne la bonne couleur pour un statut  Le paramètre <b>\$status</b> est le statut choisi
SqlDateToWritten( <b>\$date</b> )	Retourne le bon format d'une date récupérée en base de données  Le paramètre <b>\$date</b> est la date à modifier
ShowAlert( <b>\$type</b> , <b>\$message</b> )	Affiche un message d'erreur ou de confirmation avec Bootstrap  Le paramètre <b>\$type</b> est le type de message à afficher Le paramètre <b>\$message</b> est le message à afficher

Tableau 13 - Fonction de la classe StyleManager

#### 4.2.3.12 USERMANAGER

La classe UserManager contient une fonction permettant de connecter l'utilisateur au site web.

Nom de la fonction	Description
Connection(\$pseudo, \$pwd)	Valide la connexion d'un réparateur au site web
VerificateRoleUser()	Vérifie si l'utilisateur est connecté ou non

Tableau 14 - Fonction de la classe UserManager

### 4.3 OUTILS EXTERNES

#### 4.3.1 BOOTSTRAP 4

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) 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<sup>4</sup>.

Dans mon cas, j'ai utilisé Bootstrap pour ne pas trop me soucier de l'aspect esthétique du site WEB. Il m'a également permis de rendre mon site responsive<sup>5</sup> grâce à son système de cadrillage et à y ajouter quelques aspects graphiques. Pour l'inclure dans mon projet, j'ai utilisé les CDNs<sup>6</sup> disponibles sur le site de Bootstrap.

Pour le CSS :

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
```

Fig. 22 - CDNs Bootstrap CSS

---

<sup>4</sup> Source Wikipédia : [https://fr.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

<sup>5</sup> **Responsive** : Le site web s'adapte à tous les types de support afin de garder le confort visuel pour le visiteur.

<sup>6</sup> **CDN (réseau de diffusion de contenu) en français** : Ce sont des serveurs qui proposent du contenu (exemple : Bootstrap) afin d'améliorer la bande passante de notre serveur car le contenu serait téléchargé depuis le CDN.

Pour le JavaScript (jQuery, Popper.js, plugins Javascript de Bootstrap) :

```
<script
src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPiPm49"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

Fig. 23 - CDNs Bootstrap Javascript

### 4.3.2 GITHUB

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git<sup>7</sup>.

J'ai utilisé GitHub afin d'enregistrer chaque jour le développement de mon TPI. Il me sert à avoir un Backup et une trace de ce que j'effectue chaque jour.

### 4.3.3 FONT AWESOME

Font Awesome est une police d'écriture et un outil d'icônes qui se base sur CSS, LESS et SASS. Elle a été créée par Dave Gandy afin d'être utilisée avec Bootstrap<sup>8</sup>. Avant la sortie de Bootstrap 4, Font Awesome était intégré à Bootstrap. Cependant, elle a été enlevé de la V4 de Bootstrap.

J'ai utilisé Font Awesome pour afficher mes icônes de barre de navigation de mon site. Pour inclure Font Awesome j'utilise le CDN disponible sur leur site.

```
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+Pj2yciDJxpsK10YPAYjqT085Qq/1cq5FLXAZQ7Ay"
crossorigin="anonymous">
```

Fig. 24 - CDN Font Awesome

---

<sup>7</sup> Source Wikipédia : <https://fr.wikipedia.org/wiki/GitHub>

<sup>8</sup> Source Wikipédia : [https://fr.wikipedia.org/wiki/Font\\_Awesome](https://fr.wikipedia.org/wiki/Font_Awesome)

#### 4.3.4 FULLCALENDAR

FullCalendar est une librairie permettant d'utiliser un calendrier en JavaScript.

J'ai utilisé FullCalendar afin de rendre la planification de rendez-vous du réparateur plus pratique et esthétique. Pour inclure FullCalendar dans mon projet, j'ai téléchargé et inclus la librairie manuellement dans mon projet.

#### 4.3.5 SWIFT MAILER

Swift Mailer est un outil permettant d'envoyer des courriers électroniques à partir d'application PHP.

J'ai utilisé Swift Mailer pour avertir le réparateur qu'une nouvelle demande de réparation informatique a été créée et également d'avertir le client du changement de statut de sa demande de réparation informatique. Pour inclure Swift Mailer dans mon projet, j'ai téléchargé et inclus la librairie manuellement dans ce dernier.

## 5 TESTS

### 5.1 UTILISATEUR NON-CONNECTÉ

#### 5.1.1 CONNEXION ET DÉCONNEXION

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page de connexion	Accéder à la page sans être connecté	L'utilisateur non-connecté a accès à la page de connexion	Ok	Ø
2	Connexion au site	Remplir le formulaire de connexion correctement et cliquer sur le bouton « Connexion »	L'utilisateur est redirigé vers la page d'accueil et les boutons d'administration et de déconnexion sont affichés dans la barre de navigation	Ok	Ø
3	Mauvais pseudo et/ou mot de passe	Remplir le formulaire de connexion avec les mauvaises informations	Un message d'erreur « Mot de passe ou pseudo incorrect » est affiché	Ok	Ø

4	Déconnexion	Cliquer sur le bouton « Déconnexion » de la barre de navigation	L'utilisateur est redirigé vers la page de connexion et le bouton de connexion est affiché dans la barre de navigation	Ok	∅
5	Pseudo Sticky	Si l'utilisateur rentre des mauvaises informations de compte, le pseudo saisi est réaffiché	Le pseudo est bien réaffiché	Ok	∅
6	Si l'utilisateur ne remplit pas les champs obligatoires	Vérifier si les champs obligatoires sont remplis	Un message d'erreur « Tous les champs n'ont pas été remplis » est affiché	Ok	∅

*Tableau 15 - Tests de connexion et déconnexion*

## 5.1.2 PAGE DE CRÉATION D'UNE DEMANDE DE RÉPARATION INFORMATIQUE

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page de création d'une demande de réparation informatique	Accéder à la page sans être connecté	L'utilisateur non-connecté a accès à la page de création d'une demande de réparation informatique	Ok	∅
2	Envoyer une demande de réparation informatique au réparateur	Remplir le formulaire de demande de réparation informatique correctement et cliquer sur le bouton « Envoyer »	Un message de confirmation « Votre demande de réparation informatique a été envoyée au réparateur » est affiché	Ok	∅
3	Envoyer un email au réparateur	Création d'une demande de réparation informatique	Un email est envoyé au réparateur l'avertissant d'une nouvelle demande de réparation informatique	NOT OK	Le réseau de l'école bloque l'envoi d'email  <b>Solution :</b> Passer sur le réseau rouge de l'école
4	Si l'utilisateur ne remplit pas les champs obligatoires	Vérifier si les champs obligatoires sont remplis	Un message d'erreur « Tous les champs n'ont pas été remplis » est affiché	Ok	∅
5	Formulaire Sticky	Si l'utilisateur oublie un champ, les autres informations saisies sont réaffichées	Les autres informations de la demande de réparation informatique sont bien affichées	Ok	∅

Tableau 16 - Tests de création d'une demande de réparation informatique



### 5.1.3 PAGE DE CRÉATION D'AVIS

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page de création d'avis	Accéder à la page sans être connecté	L'utilisateur non-connecté a accès à la page de création d'avis	Ok	∅
2	Poster un avis sur la potentielle réparation informatique du réparateur	Remplir le formulaire d'avis correctement et cliquer sur le bouton « Envoyer »	Un message de confirmation « Votre avis est en attente de validation » est affiché	Ok	∅
3	Affichage des avis validés	Validation de l'avis par le réparateur	Les avis validés sont affichés	Ok	∅
4	Si l'utilisateur ne remplit pas le champ obligatoire	Vérifier si le champ commentaire est rempli	Un message d'erreur « Le champ commentaire n'a pas été rempli » est affiché	Ok	∅

Tableau 17 - Tests de création d'un avis

### 5.1.4 PAGE D'AFFICHAGE DES INFORMATIONS PERSONNELLES DU RÉPARATEUR

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page d'affichage des informations personnelles du réparateur	Accéder à la page sans être connecté	L'utilisateur non-connecté a accès à la page d'affichage des informations personnelles du réparateur	Ok	∅
2	Affichage des informations personnelles du réparateur	Chargement de la page	Les informations personnelles actuelles du réparateur sont affichées	Ok	∅

Tableau 18 - Tests d'affichage des informations personnelles du réparateur

## 5.2 UTILISATEUR CONNECTÉ

### 5.2.1 PAGE DE MODIFICATION DES INFORMATIONS PERSONNELLES DU RÉPARATEUR

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page de modification des informations personnelles du réparateur	Accéder à la page sans être connecté	Une redirection sur la page d'accueil doit avoir lieu	Ok	∅
2	Accès au bouton « Modifier »	Se connecter à l'application	Le bouton « modifier est visible » sur la page des informations personnelles du réparateur et il redirige vers la page de modification	Ok	∅
3	Affichage des informations personnelles du réparateur	Chargement de la page	Les informations personnelles actuelles du réparateur sont affichées	Ok	∅
4	Modifier les informations personnelles du réparateur	Remplir correctement le formulaire et appuyer sur le bouton « Modifier »	Un message de confirmation « Informations bien modifiées » est affiché	Ok	∅
5	Si l'utilisateur ne remplit pas les champs obligatoires	Vérifier si les champs obligatoires sont remplis	Un message d'erreur « Tous les champs n'ont pas été remplis » est affiché	Ok	∅

Tableau 19 - Tests de modification des informations personnelles du réparateur

## 5.2.2 PAGE D'ADMINISTRATION DES DEMANDES DE RÉPARATION INFORMATIQUE

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page d'administration des demandes de réparation informatique	Accéder à la page sans être connecté	Une redirection sur la page d'accueil doit avoir lieu	Ok	∅
2	Affichage de toutes les demandes de réparation informatique	Création d'une demande de réparation informatique sur la page de contact	Toutes les demandes de réparation informatique sont affichées	Ok	∅
3	Modifier le statut en « Traitée » ou « Refusée » d'une demande de réparation informatique	Cliquer sur le bouton « Modifier » puis cliquer sur le statut désiré	Le statut choisi devient le nouveau statut de la demande informatique et un message de confirmation « Statut de la demande bien modifié » est affiché	Ok	∅
4	Envoyer un email au client	Modifier le statut d'une demande de réparation informatique	Un email est envoyé au client l'avertissant du nouveau statut attribué à sa demande	<b>NOT OK</b>	Le réseau de l'école bloque l'envoi d'email  <b>Solution :</b> Passer sur le réseau rouge de l'école

Tableau 20 - Tests d'administration des demandes de réparation informatique

## 5.2.3 PAGE D'ADMINISTRATION DES AVIS

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page d'administration des avis	Accéder à la page sans être connecté	Une redirection sur la page d'accueil doit avoir lieu	Ok	∅

2	Affichage des avis en attente de validation	Création d'un avis sur la page de création d'avis	Tous les avis en attente de validation sont affichés	Ok	Ø
3	Valider un avis	Cliquer sur le bouton « Valider »	L'avis est affiché sur la page de création d'avis et un message de confirmation « Avis validé » est affiché	Ok	Ø
4	Refuser un avis	Cliquer sur le bouton « Refuser »	L'avis est supprimé et un message de confirmation « Avis refusé » est affiché	Ok	Ø

Tableau 21 - Tests d'administration des avis

#### 5.2.4 PAGE D'AFFICHAGE DES STATISTIQUES

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page d'affichage des statistiques	Accéder à la page sans être connecté	Une redirection sur la page d'accueil doit avoir lieu	Ok	Ø
2	Affichage du nombre de réparations informatiques effectuées par mois de l'année actuelle	Chargement de la page	Le nombre de réparations informatiques effectuées par mois de l'année actuelle est affiché	Ok	Ø
3	Affichage du nombre de réparations informatiques effectuées par mois d'une année sélectionnée	Sélectionner une année et cliquer sur le bouton « Rechercher »	Le nombre de réparations informatiques effectuées par mois de l'année sélectionnée est affiché	Ok	Ø

Tableau 22 - Tests de l'affichage des statistiques

## 5.2.5 PAGE D’AFFICHAGE ET PLANIFICATION DES RENDEZ-VOUS

N°	A tester	Comment ?	Résultat attendu ?	Résultat	Bug ?
1	Accès à la page d’affichage et planification des rendez-vous	Accéder à la page sans être connecté	Une redirection sur la page d’accueil doit avoir lieu	Ok	∅
2	Affichage des demandes de réparation informatique de statut « Ouverte »	Chargement de la page	Les demandes de réparation informatique de statut « Ouverte » sont affichées	Ok	∅
3	Affichage du calendrier avec les rendez-vous planifiés	Chargement de la page	Les rendez-vous planifiés sont affichés dans le calendrier	Ok	∅
4	Planification d’un rendez-vous	Sélectionner une date de début et de fin dans le calendrier, une demande de réparation informatique et cliquer sur le bouton « Planifier »	Le nouveau rendez-vous s’affiche dans le calendrier, son statut est modifié en « En cours » et un message « Rendez-vous créé » est affiché	Ok	∅
5	Envoyer un email au client	Planifier un rendez-vous	Un email est envoyé au client l’avertissant du nouveau statut attribué à sa demande	<b>NOT OK</b>	Le réseau de l’école bloque l’envoi d’email  <b><u>Solution :</u></b> Passer sur le réseau rouge de l’école

Tableau 23 - Tests d’affichage et planification des rendez-vous

## 6 CONCLUSION

### 6.1 AMÉLIORATIONS POSSIBLES

A première vue il existe plusieurs améliorations possibles. Tout d'abord, je trouve que tous mes formulaires sont trop sensibles aux bots et/ou spams. L'ajout d'un captcha simple mais efficace est une amélioration envisageable à mes yeux.

Ensuite, il serait intéressant de pouvoir demander aux clients quelle date leur conviendrait pour la récupération et la reddition de l'ordinateur. Une fois cette date envoyée au réparateur, il pourra valider ou proposer une autre date. Tant qu'une date n'est pas validée par le client et le réparateur, le rendez-vous n'est pas planifié.

Pour terminer, permettre la création de plusieurs réparateurs pourrait être intéressante. Les réparateurs pourraient tous modifier leurs informations personnelles visibles sur le site, afin d'avoir les descriptions de chaque réparateur. Le client pourrait également choisir avec quel réparateur organiser un rendez-vous et poster un avis sur celui-ci.

### 6.2 COMPARAISON PLANNINGS

Dans l'ensemble de mon TPI, je pense avoir été dans les temps. Mon planning est organisé de manière à avoir les fonctionnalités les plus simples au début et les plus compliquées à la fin. L'on pourrait croire que cela est à première vue risqué, mais cela m'a permis de ne pas perdre de temps sur des points bloquants qui nécessitait plus de temps.

Bien évidemment mon planning prévisionnel ne respecte pas mon planning effectif, mais c'est tout à fait normal car certaines tâches ont été plus difficiles à effectuer alors que d'autres plus faciles. Mon planning étant détaillé en fonctionnalité, m'a permis d'être mieux informé sur l'avancement de mon projet au fil du temps.

### 6.3 BILAN PERSONNEL

Je suis très content du résultat final au bout de ces 11 jours de TPI. Il y a eu bien évidemment quelques difficultés et imprévus par rapport au temps planifié pour la réalisation de ce TPI. En effet, certaines tâches m'ont données du retard alors que d'autres de l'avance. J'ai vraiment apprécié travailler sur ce TPI ; suggérer une idée à son maître d'apprentissage est génial car l'on produit quelque chose que l'on aime et qui nous motive. Dans mon cas, il s'agissait de la réalisation d'un site destiné à un hobby de mon père : réaliser des petites réparations informatiques pour particulier. J'ai donc décidé de produire ce site dans le cadre de mon TPI et de ne pas le faire les années précédentes de ma formation, car je souhaitais pouvoir donner mon maximum et montrer toutes mes compétences acquises durant ses quatre dernières années. Pour conclure, je suis très satisfait de mon travail et de ce que j'ai appris durant ces 10 jours malgré le temps mis à ma disposition. Le travail de réflexion, planification, réalisation et rédaction a été à mon égard très intéressant et agréable. Il permet une meilleure organisation afin de rendre le projet réutilisable par la suite.

## 7 BIBLIOGRAPHIE

### 7.1 CODES REPRIS

- Classe Database, pris des projets École Entreprise de la 3<sup>ème</sup> année
- Fonction SqlDateToWritten(**\$date**), pris du projet ProVélo de la 3<sup>ème</sup> année

### 7.2 SITES UTILISÉS

- Documentation Bootstrap : <https://getbootstrap.com/>
- Documentation Swift Mailer : <https://swiftmailer.symfony.com/>
- Documentation FullCalendar : <https://fullcalendar.io/>
- **Documentation Font Awesome** : <https://fontawesome.com/>
- Site stackoverflow : <https://stackoverflow.com/>
- Site W3Schools : <https://www.w3schools.com/>
- Documentation PHP : <http://www.php.net/>
- Documentation SQL : <http://sql.sh/>

## 8 PLANNING

### 8.1 PLANNING PRÉVISIONNEL

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
Lecture de l'énoncé	00:30	00:30											00:30
Planification	03:30	03:30											03:30
<b>Maquette statique, vues</b>													00:00
Création d'un template Web Responsive	01:30	01:30											01:30
<b>Page Connexion</b>													00:00
Création de la vue	01:00		01:00										01:00
Affichage du formulaire de connexion	01:00		01:00										01:00
Connexion et déconnexion du réparateur à l'application	02:00		02:00										02:00
Test / Débuggage	01:00		01:00										01:00
<b>Page Avis</b>													00:00
Création de la vue	01:00		00:30	00:30									01:00
Affichage du formulaire d'avis	01:00			01:00									01:00
CRUD d'un avis	02:00			02:00									02:00
Test / Débuggage	01:00			01:00									01:00
<b>Page À propos</b>													00:00
Création de la vue	01:00			01:00									01:00
Affichage des informations du réparateur	01:00				01:00								01:00
Edition des informations du réparateur	01:00				01:00								01:00
CRU des informations du réparateur	01:00				01:00								01:00
Test / Débuggage	01:00				01:00								01:00
<b>Page Contact</b>													00:00
Création de la vue	01:00				01:00								01:00
Affichage du formulaire de demande	01:00				00:30	00:30							01:00
CRU d'une demande	02:00					02:00							02:00
Avertissement par mail au réparateur pour la création d'une demande	02:00					02:00							02:00
Test / Débuggage	01:00					01:00							01:00
<b>Administration</b>													00:00
<b>Page Demande</b>													00:00
Création de la vue	01:00						01:00						01:00
Afficher les demandes des clients avec leur statut (ouverte / en cours / traitée / refusée)	02:00						02:00						02:00
Modification du statut d'une demande	02:30						02:30						02:30
Avertissement par mail au client pour la modification de la demande	02:00							02:00					02:00
Test / Débuggage	01:00							01:00					01:00
<b>Page Avis</b>													00:00
Création de la vue	01:00							1:00:00					01:00
Affichage des avis pas encore validé	02:00							02:00					02:00
Valider ou refuser un avis	02:00								02:00				02:00
Test / Débuggage	01:00								01:00				01:00
<b>Page Calendrier</b>													00:00
Création de la vue	02:00								02:00				02:00
Afficher un calendrier vide	03:00								01:00	02:00			03:00
Saisir des rendez-vous de type récupération / rédition	03:00									03:00			03:00
Afficher les rendez-vous déjà saisi	01:00									01:00			01:00
CRU d'une rendez-vous	02:00									01:00	01:00		02:00
Test / Débuggage	01:00										01:00		01:00
<b>Page Statistique</b>													00:00
Création de la vue	01:00										01:00		01:00
Affichage des réparations traitées dans l'année ou par mois	02:00										02:00		02:00
Test / Débuggage	01:00										01:00		01:00
<b>Analyse</b>													00:00
	08:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00				08:00
<b>Documentation</b>													00:00
Doc technique + Journal de bord	20:00	01:30	01:30	01:30	01:30	01:30	01:30	01:00	01:00	01:00	02:00		06:00
Résumé	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

Fig. 25 - Planning prévisionnel



## 8.2 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
Lecture de l'énoncé	00:30	00:30											00:30
Planification	03:30	03:30											03:30
<b>Maquette statique, vues</b>													00:00
Création d'un template Web Responsive	01:30	01:00	01:00										02:00
<b>Page Connexion</b>													00:00
Création de la vue	01:00		00:30										00:30
Affichage du formulaire de connexion	01:00		00:30										00:30
Connexion et déconnexion du réparateur à l'application	02:00		01:00										01:00
Test / Débuggage	01:00		00:30										00:30
<b>Page Avis</b>													00:00
Création de la vue	01:00		00:30	01:00									01:30
Affichage du formulaire d'avis	01:00			01:00									01:00
CRUD d'un avis	02:00			01:00									01:00
Test / Débuggage	01:00			01:00									01:00
<b>Page À propos</b>													00:00
Création de la vue	01:00			00:45									00:45
Affichage des informations du réparateur	01:00			00:30									00:30
Edition des informations du réparateur	01:00			00:30									00:30
CRU des informations du réparateur	01:00			00:30	01:00								01:30
Test / Débuggage	01:00				01:00								01:00
<b>Page Contact</b>													00:00
Création de la vue	01:00				01:00								01:00
Affichage du formulaire de demande	01:00				01:00								01:00
CRU d'une demande	02:00				01:00								01:00
Avertissement par mail au réparateur pour la création d'une demande	02:00							01:00		01:00			02:00
Test / Débuggage	01:00				00:30								00:30
<b>Administration</b>													00:00
<b>Page Demande</b>													00:00
Création de la vue	01:00				00:30								00:30
Afficher les demandes des clients avec leur statut (ouverte / en cours / traitée / refus)	02:00				00:30	01:00							01:30
Modification du statut d'une demande	02:30					01:00							01:00
Avertissement par mail au client pour la modification de la demande	02:00							01:00		01:00			02:00
Test / Débuggage	01:00					01:00							01:00
<b>Page Avis</b>													00:00
Création de la vue	01:00					01:00							01:00
Affichage des avis pas encore validé	02:00					01:00	01:00						02:00
Valider ou refuser un avis	02:00						01:00						01:00
Test / Débuggage	01:00						01:00						01:00
<b>Page Calendrier</b>													00:00
Création de la vue	02:00							02:00					02:00
Afficher un calendrier vide	03:00							02:00					02:00
Saisir des rendez-vous de type récupération / rédition	03:00								02:00	01:00	02:00		05:00
Afficher les rendez-vous déjà saisi	01:00								01:00				01:00
CRU d'un rendez-vous	02:00								01:00	02:00			03:00
Test / Débuggage	01:00									01:00			01:00
<b>Page Statistique</b>													00:00
Création de la vue	01:00										01:00		01:00
Affichage des réparations traitées dans l'année ou par mois	02:00										02:00		02:00
Test / Débuggage	01:00												00:00
<b>Analyse</b>													00:00
	08:00	01:00	01:00	01:00	00:30	01:00	01:00	01:00	02:00				08:30
<b>Documentation</b>													00:00
Doc technique + Journal de bord	20:00	02:00	03:00	00:45	01:00	02:00	04:00	01:00	02:00	02:00	03:00	06:00	26:45
Résumé	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

Fig. 26 - Planning effectif

## 9 TABLE DES ILLUSTRATIONS

### 9.1 FIGURES

FIG. 1 - BARRE DE NAVIGATION D'UN UTILISATEUR NON-CONNECTÉ.....	8
FIG. 2 - BARRE DE NAVIGATION D'UN UTILISATEUR CONNECTÉ.....	8
FIG. 3 - PAGE D'ACCUEIL .....	9
FIG. 4 - PAGE DE CONTACT.....	10
FIG. 5 - PAGE D'AVIS.....	11
FIG. 6 - PAGE DE CONNEXION .....	12
FIG. 7 - PAGE CALENDRIER .....	13
FIG. 8 - PAGE D'ADMINISTRATION DES DEMANDES DE RÉPARATION INFORMATIQUE .....	14
FIG. 9 - PAGE DE MODIFICATION DES INFORMATIONS PERSONNELLES.....	15
FIG. 10 - PAGE D'ADMINISTRATION DES AVIS.....	16
FIG. 11 - PAGE DE STATISTIQUES.....	17
FIG. 12 - DIAGRAMME DES CAS D'UTILISATION .....	18
FIG. 13 - MODÈLE PHYSIQUE DE DONNÉES FINALES.....	19
FIG. 14 - TABLE « CLIENTS » .....	21
FIG. 15 - TABLE « AVIS » .....	21
FIG. 16 - TABLE « DEMANDES ».....	21
FIG. 17 - TABLE « REPARATEUR » .....	22
FIG. 18 - TABLE « EVENEMENT » .....	22
FIG. 19 - TABLE « INFOS_DYNAMIQUES » .....	22
FIG. 20 - ARBORESCENCE DE FICHIERS .....	23
FIG. 21 - DIAGRAMME DU FONCTIONNEMENT DE L'INTERACTION AVEC LA BASE DE DONNÉES.....	24
FIG. 22 - CDNS BOOTSTRAP CSS .....	35
FIG. 23 - CDNS BOOTSTRAP JAVASCRIPT.....	36
FIG. 24 - CDN FONT AWESOME.....	36
FIG. 25 - PLANNING PRÉVISIONNEL.....	47
FIG. 26 - PLANNING EFFECTIF .....	48

### 9.2 TABLEAUX

TABLEAU 1 - FONCTION DE LA CLASSE DATABASE.....	20
TABLEAU 2 - LÉGENDE DES PROPRIÉTÉS DES ATTRIBUTS .....	21
TABLEAU 3 - CLASSE ABOUT.....	25
TABLEAU 4 - CLASSE CLIENT .....	26
TABLEAU 5 - CLASSE OPINION.....	27
TABLEAU 6 - CLASSE REQUEST .....	28
TABLEAU 7 - FONCTION DE LA CLASSE ABOUTMANAGER .....	29
TABLEAU 8 - FONCTION DE LA CLASSE CLIENTMANAGER .....	29
TABLEAU 9 - FONCTION DE LA CLASSE EVENTMANAGER .....	30
TABLEAU 10 - FONCTION DE LA CLASSE MAILERMANAGER .....	31
TABLEAU 11 - FONCTION DE LA CLASSE OPINIONMANAGER .....	32
TABLEAU 12 - FONCTION DE LA CLASSE REQUESTMANAGER.....	34
TABLEAU 13 - FONCTION DE LA CLASSE STYLEMANAGER .....	34
TABLEAU 14 - FONCTION DE LA CLASSE USERMANAGER .....	35
TABLEAU 15 - TESTS DE CONNEXION ET DÉCONNEXION.....	38
TABLEAU 16 - TESTS DE CRÉATION D'UNE DEMANDE DE RÉPARATION INFORMATIQUE .....	39
TABLEAU 17 - TESTS DE CRÉATION D'UN AVIS.....	40
TABLEAU 18 - TESTS D'AFFICHAGE DES INFORMATIONS PERSONNELLES DU RÉPARATEUR .....	40

TABLEAU 19 - TESTS DE MODIFICATION DES INFORMATIONS PERSONNELLES DU RÉPARATEUR .....	41
TABLEAU 20 - TESTS D'ADMINISTRATION DES DEMANDES DE RÉPARATION INFORMATIQUE .....	42
TABLEAU 21 - TESTS D'ADMINISTRATION DES AVIS.....	43
TABLEAU 22 - TESTS DE L'AFFICHAGE DES STATISTIQUES .....	43
TABLEAU 23 - TESTS D'AFFICHAGE ET PLANIFICATION DES RENDEZ-VOUS.....	44

## 10 ANNEXES

- Code source
- Résumé
- Journal de bord

## 11 GLOSSAIRE

Mot	Définition
CSS	Langage informatique de mise en forme
FullCalendar	Librairie externe pour le calendrier
HTML	Langage de balisage utilisé pour la création de pages web
PHP	PHP Hyper Protocole
POO	Programmation orienté objet
Swift Mailer	Librairie externe pour l'envoi d'email