



**DOCUMENTATION TECHNIQUE  
SOS INFOBOBO**Jonathan Borel-Jaquet. 21.05.2019. V.1.0

**Table des matières**

[1 Introduction 3](#_Toc9344021)

[2 Rappel du cahier des charges 3](#_Toc9344022)

[2.1 Objectifs 3](#_Toc9344023)

[2.2 Spécifications 3](#_Toc9344024)

[2.3 Restrictions 3](#_Toc9344025)

[2.4 Environnement 4](#_Toc9344026)

[2.5 Organisation 4](#_Toc9344027)

[2.6 Livrable 4](#_Toc9344028)

[3 Analyse fonctionnelle 5](#_Toc9344029)

[3.1 Fonctionnalités 5](#_Toc9344030)

[3.1.1 Créer une demande de réparation informatique 5](#_Toc9344031)

[3.1.2 Changer le statut d’une demande de réparation 5](#_Toc9344032)

[3.1.3 Créer un avis sur le service effectué 5](#_Toc9344033)

[3.1.4 Valider la publication d’un avis 5](#_Toc9344034)

[3.1.5 Enregistrer un rendez-vous 5](#_Toc9344035)

[3.1.6 Modifier les informations personnelles 5](#_Toc9344036)

[3.1.7 Consulter les statistiques de réparations effectuées 6](#_Toc9344037)

[3.1.8 Consulter les différentes pages du site 6](#_Toc9344038)

[3.1.9 Envoyer un email lors d’un changement de statut d’une demande 6](#_Toc9344039)

[3.1.10 Envoyer un email lors de la création d’une nouvelle demande 6](#_Toc9344040)

[3.1.11 Se connecter à l’application 6](#_Toc9344041)

[3.2 Interfaces 6](#_Toc9344042)

[3.2.1 Barre de navigation 6](#_Toc9344043)

[3.2.2 Page d’accueil 8](#_Toc9344044)

[3.2.3 Page de contact 9](#_Toc9344045)

[3.2.4 Page d’avis 10](#_Toc9344046)

[3.2.5 Page de connexion 11](#_Toc9344047)

[3.2.6 Page calendrier 12](#_Toc9344048)

[3.2.7 Page d’administration des demandes de réparation informatique 13](#_Toc9344049)

[3.2.8 Page de modification des informations personnelles 14](#_Toc9344050)

[3.2.9 Page d’administration des avis 15](#_Toc9344051)

[3.2.10 Page de statistiques 16](#_Toc9344052)

[3.3 Cas d’utilisation 17](#_Toc9344053)

[3.3.1 Diagramme des cas d’utilisation 17](#_Toc9344054)

[4 Analyse organique 18](#_Toc9344055)

[4.1 Base de données 18](#_Toc9344056)

[4.1.1 Modèle physique de données 18](#_Toc9344057)

[4.1.2 Détails des tables 18](#_Toc9344058)

[4.2 Architecture du code 20](#_Toc9344059)

[4.2.1 Arborescence de fichier 20](#_Toc9344060)

[4.2.2 Classe POO 21](#_Toc9344061)

[4.2.3 Classes utilisées 22](#_Toc9344062)

[4.3 Outils externes 28](#_Toc9344063)

[4.3.1 Bootstrap 4 28](#_Toc9344064)

[4.3.2 GitHub 29](#_Toc9344065)

[4.3.3 Font Awesome 29](#_Toc9344066)

[4.3.4 FullCalendar 30](#_Toc9344067)

[4.3.5 Swift Mailer 30](#_Toc9344068)

[5 Tests 30](#_Toc9344069)

[5.1 Utilisateur non-connecté 30](#_Toc9344070)

[5.1.1 Connexion et déconnexion 30](#_Toc9344071)

[5.1.2 Page de création d’une demande de réparation informatique 32](#_Toc9344072)

[5.1.3 Page de création d’avis 32](#_Toc9344073)

[5.1.4 Page d’affichage des informations personnelles du réparateur 33](#_Toc9344074)

[5.2 Utilisateur connecté 33](#_Toc9344075)

[5.2.1 Page de modification des informations personnelles du réparateur 33](#_Toc9344076)

[5.2.2 Page d’administration des demandes de réparation informatique 34](#_Toc9344077)

[5.2.3 Page d’administration des avis 35](#_Toc9344078)

[5.2.4 Page d’affichage des statistiques 35](#_Toc9344079)

[5.2.5 Page d’affichage et planification des rendez-vous 36](#_Toc9344080)

[6 Conclusion 36](#_Toc9344081)

[6.1 Améliorations possibles 36](#_Toc9344082)

[6.2 Bilan personnel 37](#_Toc9344083)

[7 Bibliographie 37](#_Toc9344084)

[7.1 Codes repris 37](#_Toc9344085)

[7.2 Sites utilisés 37](#_Toc9344086)

# 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 au client de poster des avis sur la potentielle réparation effectuée qui devra d’abord être validée 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.

# Rappel du cahier des charges

## 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.

## 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 d’une demande

## 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 d’une demande

## 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)

## Organisation

Élève :

* Jonathan Borel-Jaquet, [jonathan.brljq@eduge.ch](mailto:jonathan.brljq@eduge.ch)

Maître d’apprentissage :

* Jasmina Travnjak, [jasmina.travnjak@edu.ge.ch](mailto:jasmina.travnjak@edu.ge.ch)

Experts :

* Carol Quarroz, [cquarroz@gmail.com](mailto:cquarroz@gmail.com)
* Nicolas Terrond, [nicolas@terrond.com](mailto:nicolas@terrond.com)

## 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

# Analyse fonctionnelle

## Fonctionnalités

### 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

### 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

### 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.

### 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.

### 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 ».

### 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

### 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.

### 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)

### 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](#_Changer_le_statut)).

### 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](#_Créer_une_demande)).

### 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.

## Interfaces

### Barre de navigation

#### Utilisateur non-connecté



4

3

2

1D

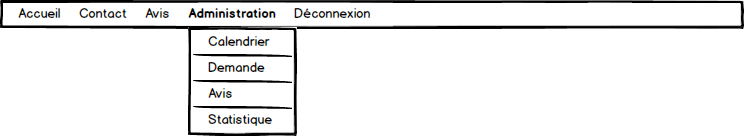
Fig. - Barre de navigation d’un utilisateur non-connecté

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).

#### Utilisateur connecté

6D

1D



4D

5D

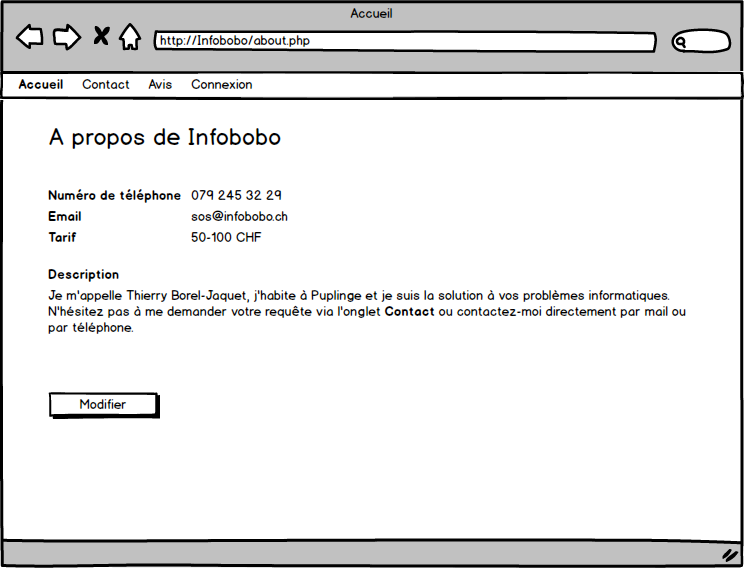
3

2D

Fig. - 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 (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 la page de statistiques (voir Fig. 11 - Page de statistiques).
6. Le lien « Déconnexion » permet à l’utilisateur de se déconnecter du site.

### Page d’accueil



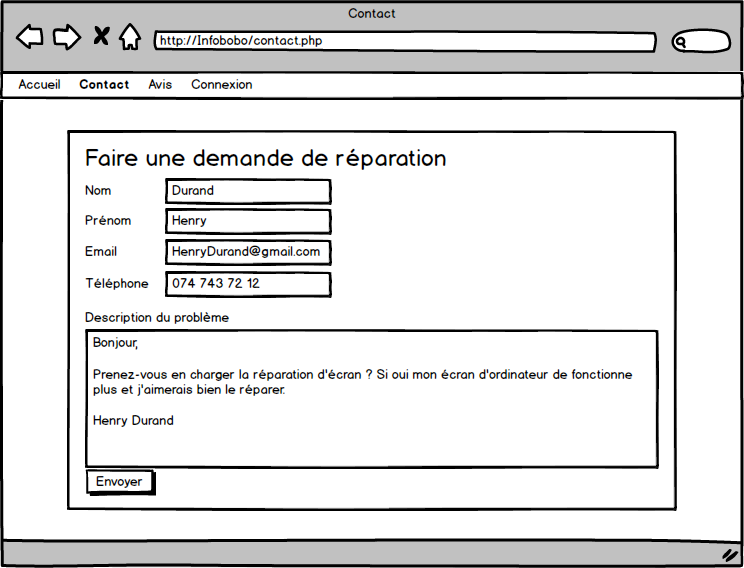
1D

Fig. - Page d’accueil

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

1. Le bouton « Modifier » apparait 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).

### Page de contact



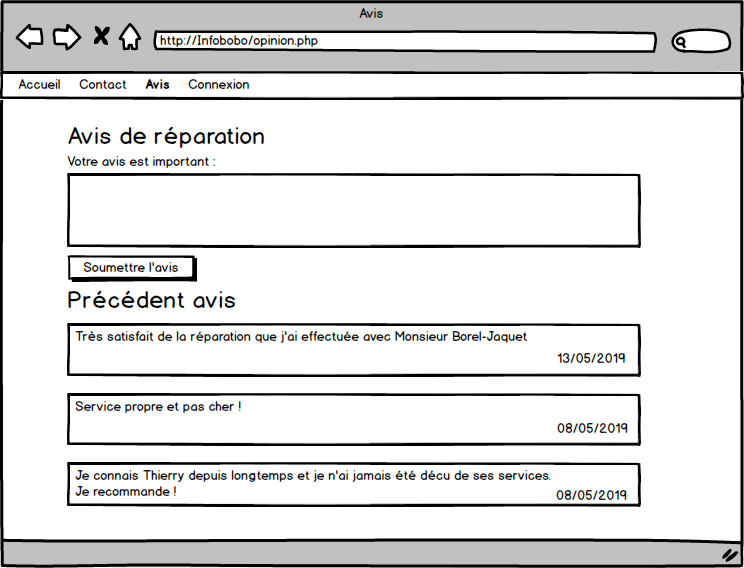
1D

Fig. - Page de contact

La page « 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, une pop-up confirme si l’envoi de la demande à bien été effectué ou non.

### Page d’avis



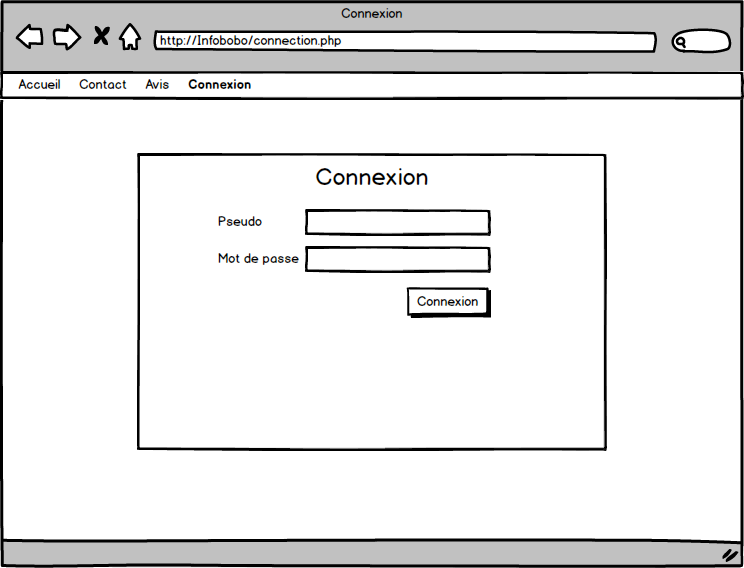
1D

Fig. - Page d’avis

La page « 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.

### Page de connexion



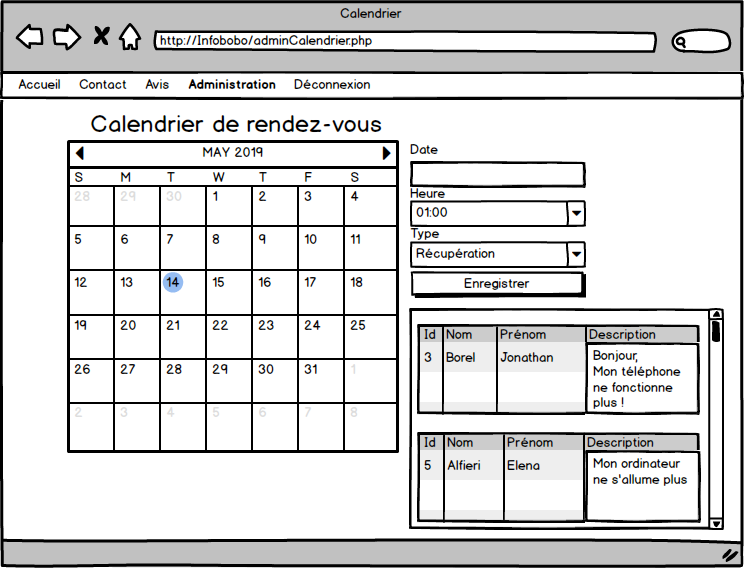
1D

Fig. - 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.

### Page calendrier



2

3D

5D

4D

6D

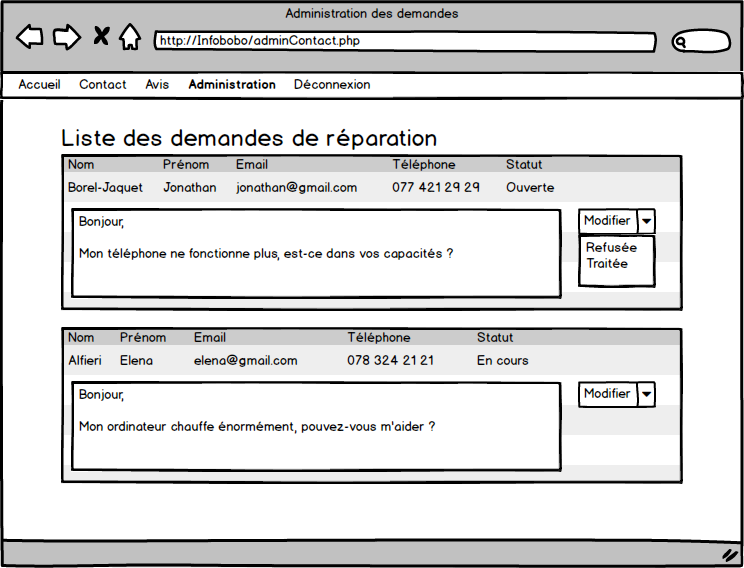
1D

Fig. - Page calendrier

La page « Calendrier » permet au réparateur de choisir une date de début pour la récupération et une date de fin pour la reddition.

1. Une fois la date sélectionnée, le champ date (réf .2) est rempli avec la date choisie.
2. Zone de texte non modifiable contenant la date choisie par le réparateur.
3. Liste de choix pour l’heure du rendez-vous à planifier.
4. Liste de choix pour le type de rendez-vous (Récupération ou reddition).
5. Le bouton « Enregistrer » permet au réparateur d’enregistrer un rendez-vous une fois les informations bien sélectionnées (réf .2 réf .6).
6. Liste des demandes de statut « ouverte » : le réparateur doit cliquer sur l’une des demandes pour la sélectionner.

### Page d’administration des demandes de réparation informatique



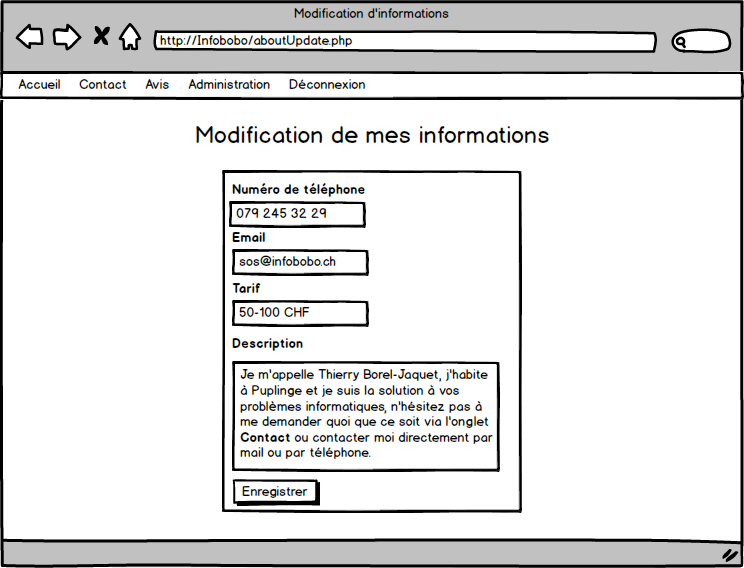
1D

Fig. - Page d’administration des demandes de réparation informatique

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

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

### Page de modification des informations personnelles



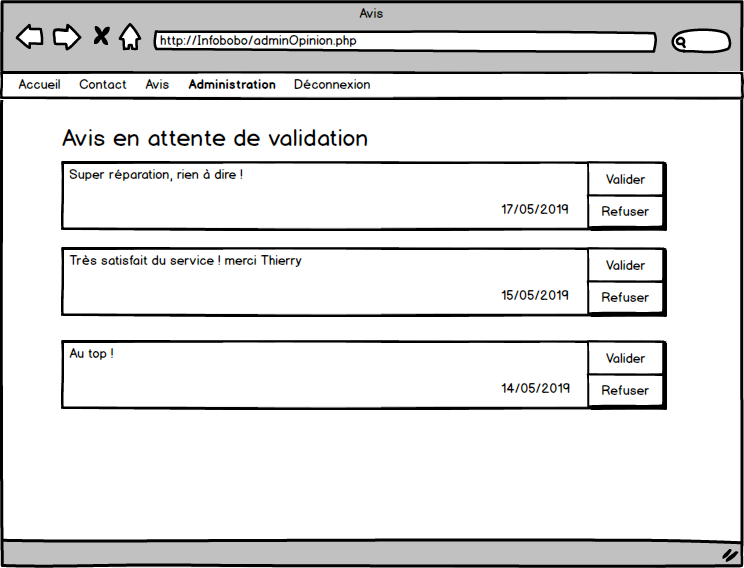
1D

Fig. - 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.

### Page d’administration des avis



2D

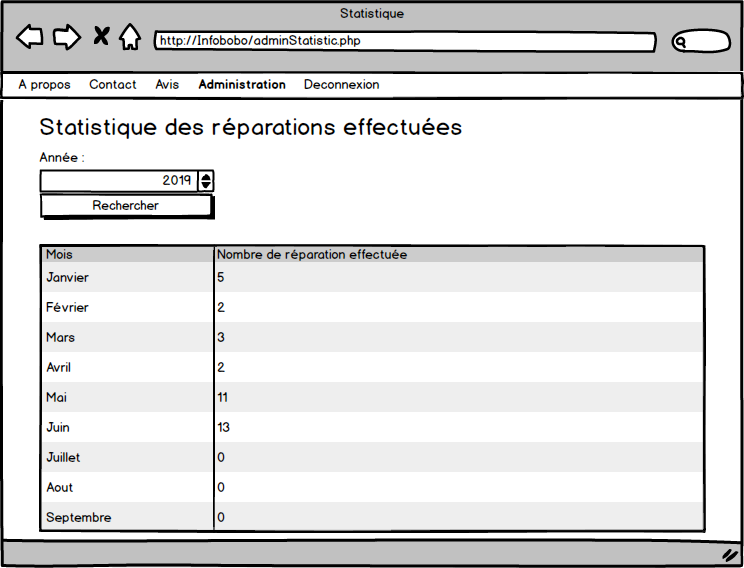
1D

Fig. - 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.

### Page de statistiques



1D

2D

Fig. - 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.

## Cas d’utilisation

### Diagramme des cas d’utilisation

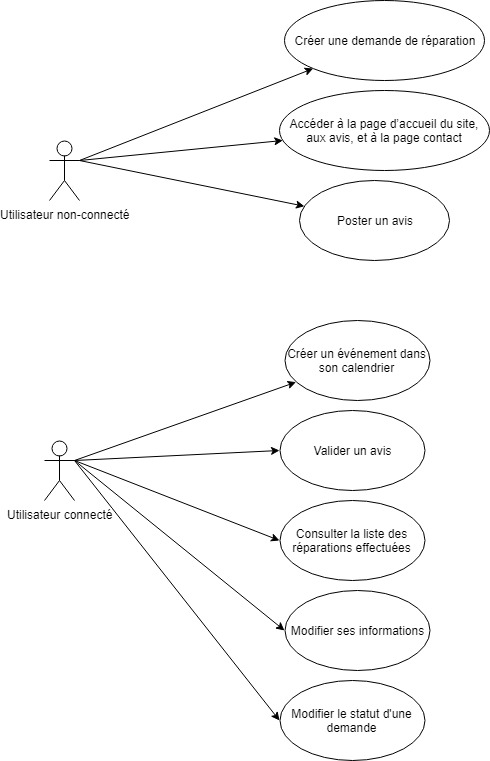


Fig. - Diagramme des cas d'utilisation

# Analyse organique

## Base de données

### Modèle physique de données

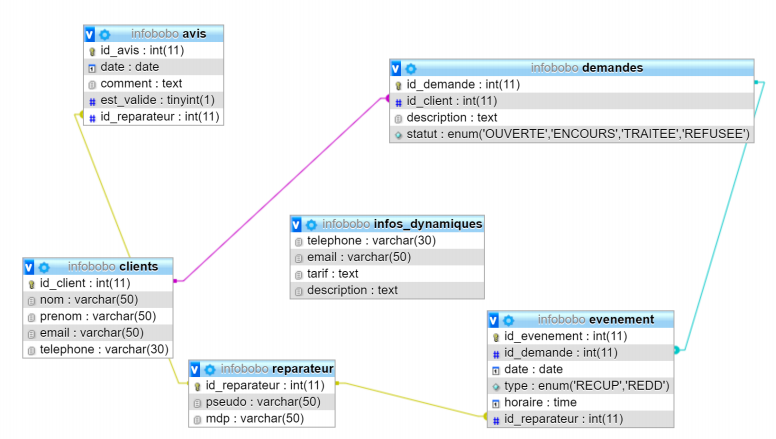


Fig. - Modèle physique de données initial

### 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 - Légende des propriétés des attributs

#### Table « clients »

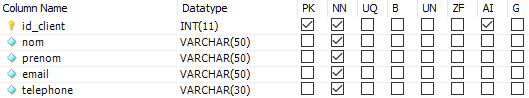


Fig. - Table « clients »

#### Table « avis »

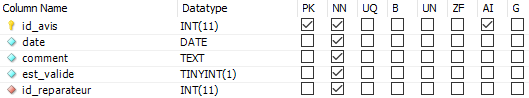


Fig. - Table « avis »

#### Table « demandes »



Fig. - Table « demandes »

#### Table « reparateur »

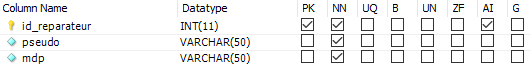


Fig. - Table « reparateur »

#### Table « evenement »

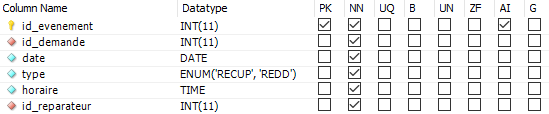


Fig. - « Table evenement »

#### Table « infos\_dynamiques »

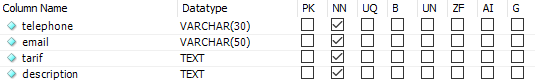


Fig. - Table « infos\_dynamiques »

## Architecture du code

### Arborescence de fichier

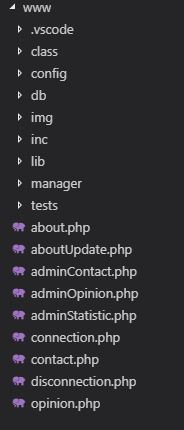


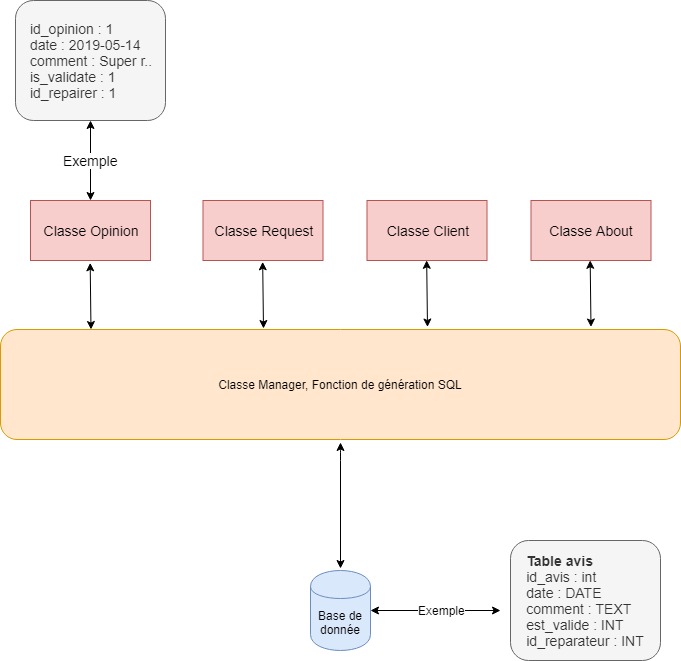
Fig. - Arborescence de fichier

* 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[[1]](#footnote-1).
* 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 FullCalandar[[2]](#footnote-2) 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.

### 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. Ce sont des classes qui contiennent uniquement des informations venant de la base de données, elles contiennent aucune méthode permettant d’accéder ou manipuler l’objet. Elles permettent à mes fonctions de ne pas être trop dépendante de la base de données. Imaginons une modification de champs en base, il faudra que je change le nom de chaque utilisation de ces anciens champs. Mes conteneurs vont au contraire, permettre de modifier uniquement les fonctions les utilisant. Je décrirais 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 ne sont des classes qui permettent uniquement l’interaction avec les données pour lesquelles elles sont destiné. Une fonction permettant de créer un avis se trouvera dans le manager « OpinionManager » par exemple. Elles permettent de structurer mes fonctions afin de les rendre plus facilement réutilisables pour moi ou pour une personne se greffant à mon projet.

   
  
Fig. – Diagramme du fonctionnement de l’interaction avec la base de données

### Classes utilisées

#### 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 - 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..");

#### 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 - 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");

#### 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 - 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);

#### 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 - 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");

#### 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() | Retourne un objet About contenant les informations personnelles du réparateur |
| UpdateAboutInformation(**$phoneNumber**, **$email**, **$price**, **$description**) | Modifie les informations personnelles du réparateur avec de nouvelles valeurs passées en paramètre |

Tableau - Fonction de la classe AboutManager

#### ClientManager

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

|  |  |
| --- | --- |
| **Nom de la fonction** | **Description** |
| AddClient(**$firstName**, **$secondName**, **$email**, **$phoneNumber**) | Enregistre les informations d’un client dans la base de données |

Tableau - Fonction de la classe ClientManager

#### 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(**$id\_request**, **$dateEvent**, **$typeEvent**, **$hourEvent**) | Enregistre un événement et modifie le statut d'une demande dans la base de données |
| GetAllEventFormatJSON() | Récupère tous les événements en format JSON destiné au calendrier JavaScript FullCalendar |

Tableau - Fonction de la classe EventManager

#### MailerManager

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

|  |  |
| --- | --- |
| **Nom de la fonction** | **Description** |
| SendMail**($desinataire**, **$subject**, **$textMessage**) | Envoie un email en utilisant la librairie Swift Mailer |

Tableau - Fonction de la classe MailerManager

#### 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(**$comment**) | Enregistre un avis non validé dans la base de données |
| GetOpinionValidate() | Récupère tous les avis validés de la base de données |
| GetOpinionNotValidate() | Récupère tous les avis non validés de la base de données |
| ValidateOpinionById(**$id\_opinion**) | Valide un avis dans la base de données en utilisant son identifiant |
| DeleteOpinionById(**$id\_opinion**) | Supprime un avis dans la base de données en utilisant son identifiant |

Tableau - Fonction de la classe OpinionManager

#### RequestManager

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

|  |  |
| --- | --- |
| **Nom de la fonction** | **Description** |
| AddRequest(**$firstName**, **$secondName**, **$email**, **$phoneNumber**, **$description**) | Enregistre une demande de réparation et un client dans la base de données |
| GetAllRequest() | Récupère toutes les demandes de réparation de la base de données |
| UpdateRequestStatusById(**$id\_request**, **$status**) | Modifie le statut d'une demande de réparation |
| GetOpenRequest() | Récupère toutes les demandes de réparation de statut "ouverte" de la base de données |
| GetProcessedRequestOrderByMonthAndYear(**$year**) | Récupère les demandes de statut "traitée" de la base de données triée par mois et par année |

Tableau - Fonction de la classe RequestManager

#### StyleManager

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

|  |  |
| --- | --- |
| **Nom de la fonction** | **Description** |
| ColorStatus(**$status**) | Retourne la bonne couleur pour un statut |
| SqlDateToWritten(**$date**) | Retourne le bon format d'une date récupérée en base de données |

Tableau - Fonction de la classe StyleManager

#### 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. |

Tableau - Fonction de la classe UserManager

## Outils externes

### 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[[3]](#footnote-3).

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[[4]](#footnote-4) grâce à son système de cadrillage et à y ajouter quelques aspects graphiques. Pour l’inclure dans mon projet, j’ai utilisé les CDNs [[5]](#footnote-5) disponibles sur le site de Bootstrap.

Pour le CSS :

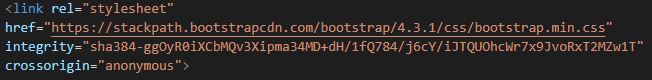


Fig. – CDNs BootStrap CSS

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



Fig. - CDNs BootStrap JavaScript

### GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git[[6]](#footnote-6).

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.

### 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[[7]](#footnote-7). Avant la sortie de Bootstrap 4, Font Awesome était intégré à Bootstrap, cependant il 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.

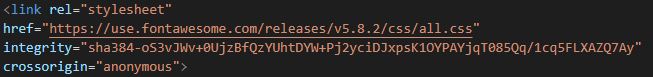


Fig. - CDN Font Awesome

### FullCalendar

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

J’ai utilisé FullCalendar afin de rentre 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.

### 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 afin 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.

# Tests

## Utilisateur non-connecté

### 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 à accès à la page de connexion | Ok | ∅ |
| **2** | Connexion au site | Remplir le formulaire de connexion correctement | 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 | ∅ |
| **3** | Mauvais pseudo 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 | ∅ |
| **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 - Tests de connexion et déconnexion

### 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 | 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’envoie d’ email |
| **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 - Tests de création d'une demande de réparation informatique

### 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 à 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 | 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 - Tests de création d'un avis

### 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 à accès à la page d’affichage des informations personnel du réparateur | Ok | ∅ |
| **2** | Affichage des informations personnel du réparateur | Chargement de la page | Les informations personnelles actuelles du réparateur sont affichées | Ok | ∅ |

Tableau - Tests d'affichage des informations personnelles du réparateur

## Utilisateur connecté

### 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 | 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 - Tests de modification des informations personnelles du réparateur

### 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 toute 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 d’une demande de réparation informatique | Cliquer sur le bouton « modifier » | 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 aux clients | Modifier le statut d’une demande de réparation informatique | Un email est envoyé au client l’avertissant du nouveau statut attribué à sa demande | **NOT OK** | Le réseau de l’école bloque l’envoie d’ email |

Tableau - Tests d'administration des demandes de réparation informatique

### 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 | ∅ |
| **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 | ∅ |
| **4** | Refuser un avis | Cliquer sur le bouton « Refuser » | L’avis est supprimé et un message de confirmation « Avis refusé » est affiché | Ok | ∅ |

Tableau - Test d'administration des avis

### Page d’affichage des statistiques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat** | **Bug ?** |
| **1** | Accès à la page des statistiques | Accéder à la page sans être connecté | Une redirection sur la page d’accueil doit avoir lieu | Ok | ∅ |
| **2** | Affichage du nombre de réparation informatique effectué par mois de l’année actuelle | Chargement de la page | Le nombre de réparation informatique effectué par mois de l’année actuelle est affiché | Ok | ∅ |
| **2** | Affichage du nombre de réparation informatique effectué par mois d’une année sélectionnée | Sélectionner une année et cliquer sur le bouton « Rechercher » | Le nombre de réparation informatique effectué par mois de l’année sélectionné est affiché | Ok | ∅ |

Tableau - Test de l'affichage des statistiques

### Page d’affichage et planification des rendez-vous

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat** | **Bug ?** |
| **1** | Accès à la page calendrier | Accéder à la page sans être connecté | Une redirection sur la page d’accueil doit avoir lieu | Ok | ∅ |
| **2** | Affichage des demandes de réparations informatiques de statut « ouverte » | Chargement de la page | Les demandes de réparation informatique de statut ouverte est affichée | Ok | ∅ |
| **3** | Affichage du calendrier avec les rendez-vous planifiés | Chargement de la page | Les rendez-vous planifiés sont affichés dans le calendrier | - | - |
| **4** | Planification d’un rendez-vous | Sélectionner une date de début et de fin dans le calendrier, une heure, une demande de réparation informatique et cliquer sur le bouton « Planifier » | Le nouveau rendez-vous s’affiche dans le calendrier | - | - |

Tableau - Test d'affichage et planification des rendez-vous

# Conclusion

## Améliorations possibles

A première vue il existe plusieurs améliorations possibles. Tout d’abord, je pense à la publication d’avis. Empêcher les bots et spam de créer un avis est bien, mais à mon sens filtrer les avis des clients n’est pas une bonne pratique. Le réparateur pourrait très bien supprimer tous les avis négatifs et laisser que les positifs.

Ensuite, il serait intéressant de pouvoir demander aux clients quelle date leur conviendrait pour la récupération 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, rendre le site plus esthétique et professionnel serait intéressant car cela permettrait de montrer le professionnalisme du réparateur.

## Bilan personnel

Je suis très content du résultat final au bout de ces 10 jours de TPI, j’ai

# Bibliographie

## Codes repris

* Classe Database, pris des projets École Entreprise de la 3ème année
* Fonction SqlDateToWritten(**$date**), pris du projet ProVélo de ka 3ème année

## Sites utilisés

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

1. **Librairie externe** : Swift Mailer <https://swiftmailer.symfony.com/> [↑](#footnote-ref-1)
2. **Librairie externe** : FullCalandar <https://fullcalendar.io/> [↑](#footnote-ref-2)
3. Source Wikipédia : <https://fr.wikipedia.org/wiki/Bootstrap_(framework)> [↑](#footnote-ref-3)
4. **Responsive** : Le site web s’adapte à tous les types de support afin de garder le confort visuel pour le visiteur. [↑](#footnote-ref-4)
5. **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. [↑](#footnote-ref-5)
6. Source Wikipédia : <https://fr.wikipedia.org/wiki/GitHub> [↑](#footnote-ref-6)
7. Source Wikipédia : <https://fr.wikipedia.org/wiki/Font_Awesome> [↑](#footnote-ref-7)