# Travler

Rapport TPI et documentation technique

***Mai 2020***

***Travail pratique individuel (TPI)***

***Thomas Fujise***

***Maître d’apprentissage : Dominique Aigroz***

Table des matières

[Travler 1](file:///C:\Users\Thomas\Desktop\PHP\ProjectTPI\docs\technical%20docs\documentation.docx#_Toc42605039)

[Versions 4](#_Toc42605040)

[Introduction 4](#_Toc42605041)

[Rappel de l’énoncé 4](#_Toc42605042)

[Matériel et logiciels à disposition 4](#_Toc42605043)

[Description de l’application 4](#_Toc42605044)

[Livrables 5](#_Toc42605045)

[Acteurs 5](#_Toc42605046)

[Organisation 5](#_Toc42605047)

[Méthodologie 5](#_Toc42605048)

[Product backlog 6](#_Toc42605049)

[Planification 10](#_Toc42605050)

[Maquettage 10](#_Toc42605051)

[Visiteur 11](#_Toc42605052)

[Utilisateur 15](#_Toc42605053)

[Administrateur 23](#_Toc42605054)

[Plan de navigation 26](#_Toc42605055)

[Visiteur 26](#_Toc42605056)

[Utilisateur 27](#_Toc42605057)

[Administrateur 28](#_Toc42605058)

[Implémentation 29](#_Toc42605059)

[Base de données 29](#_Toc42605060)

[Structure du projet 35](#_Toc42605061)

[Container (PHP) 36](#_Toc42605062)

[Manager (PHP) 38](#_Toc42605063)

[Vues (PHP) 51](#_Toc42605064)

[Librairies et outils externes 53](#_Toc42605065)

[Mapquest 53](#_Toc42605066)

[Git 53](#_Toc42605067)

[StartBootstrap 53](#_Toc42605068)

[Tests 54](#_Toc42605069)

[Unitaire 54](#_Toc42605070)

[Utilisateur 54](#_Toc42605071)

[61](#_Toc42605072)

[Conclusion 63](#_Toc42605073)

[Difficultés rencontrées 63](#_Toc42605074)

[Améliorations possibles 63](#_Toc42605075)

[Bilan personnel 64](#_Toc42605076)

## Versions

|  |  |  |  |
| --- | --- | --- | --- |
| N° de version | Auteur | Date | Description |
| 1.0 | Thomas Fujise | 09.06.2020 | Version finale du document pour le rendu du TPI |

## Introduction

Ce document présente différents aspects de la réalisation du projet « Travler ». Ce projet a été réalisé dans le cadre de mon Travail Pratique Individuel (TPI) 2020. Il a comme objectif de valider toutes les compétences que j’ai acquises durant ma formation au CFPT en tant qu’Informaticien CFC en développement d’application.

« Travler » est une application web, destinée aux motards, permettant de partager des itinéraires avec les autres utilisateurs de l’application. Les utilisateurs peuvent utiliser plusieurs fonctionnalités implémentées sur le site afin de créer leurs propres itinéraires. Ils ont également la possibilité de commenter et de noter les itinéraires des autres utilisateurs.

## Rappel de l’énoncé

*Les informations suivantes sont tirées du cahier des charges.*

### Matériel et logiciels à disposition

* Un PC standard école avec Windows 10 ,2 écrans
* Serveur Web (Apache avec module PHP et MySQLServer)
* SGBD (MySQLWorkbench)
* IDE (Visual Studio Code avec extensions)
* Logiciel de création de diagramme (MySQLWorkbench, dbdiagram.io)
* Logiciel de création de maquettes d’interfaces (Balsamiq)
* Outil de versionnage de projet (Git avec dépôt distant sur GitHub)
* Outil de bureautique pour les documents (Suite Office)
* API pour gérer les maps ([MapQuest](https://developer.mapquest.com/))
* Navigateur Web (Google Chrome, Microsoft Edge)

### Description de l’application

Travler est une application web, développée à l’aide des langages suivants : PHP, HTML 5, CSS 3, SQL, Javascript et avec l’API « MapQuest » pour gérer les maps. Le but de l’application est de partager des itinéraires réalisables à moto avec les autres utilisateurs. Chaque utilisateur est libre de noter de 0 à 10 et de commenter les itinéraires des autres utilisateurs. Le créateur de l’itinéraire quant à lui reçoit un mail à chaque fois qu’un autre utilisateur commente son itinéraire. En tant qu’utilisateur non connecté, il ne peut accéder qu’à la page d’accueil où sont affichés tous les itinéraires avec un système de filtrage. Lorsque l’utilisateur clique sur le titre d’un itinéraire, tous les détails de celui-ci sont affichés avec les éventuelles images qui y sont attachées.

### Livrables

|  |  |
| --- | --- |
| Pour le responsable de projet et experts : | Pour le responsable de projet seulement : |
| * Planning | * Accès au dépôt distant sur GitHub |
| * Manuel utilisateur | * Modèles conceptuel et physique |
| * Journal de bord | * Protocole de test |
| * Résumé du projet |  |
| * Rapport du projet |  |

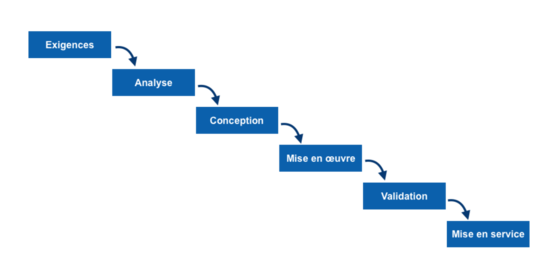
### Acteurs

|  |  |  |
| --- | --- | --- |
| **Élève** |  | **Maître d’apprentissage** |
| Thomas Fujise  <thomas.fjs@eduge.ch> |  | Dominique Aigroz  <edu-aigrozd@eduge.ch> |

|  |  |
| --- | --- |
| **Experts** | |
| Stéphane Monet  <stephane.monet@gmail.com> | Borys Folomietow  <borys@folomietow.ch> |

## Organisation

### Méthodologie

Pour planifier mon projet de TPI, j’ai utilisé la **méthode en cascade** qui m’a été demandé par mon maître d’apprentissage. Les 2 premières étapes de cette méthodologie ont été réalisé par mes experts et mon maître d’apprentissage.

1. **Exigences :** les exigences font l’objet d’une expression des besoins, cette étape à été réalisé lorsque mon maître d’apprentissage ma demandé dans quel domaine je voulais réaliser mon TPI.
2. **Analyse :** les exigences sont analysées pour établir un cahier des charges fonctionnel, cette étape à été réalisé par mon maître d’apprentissage et validé par les experts.
3. **Conception :** le produit est conçu et spécifié de sorte à pouvoir être réalisé, j’ai réalisé cette étape lors de ma planification du projet et la création des différents products backlog.
4. **Mise en œuvre :** le produit est réalisé sur la base des spécifications, j’ai réalisé cette étape lors du développement de l’application web en suivant le planning réalisé au préalable.
5. **Validation :** le produit est testé et vérifié et sa conformité aux exigences est validée, cette étape sera réalisée une fois mon TPI rendu et corriger pour vérifier s’il est conforme aux exigences.
6. **Mise en service :**  le produit est installé, les préparatifs pour sa mise en service sont organisés, puis le produit est utilisé, cette étape ne sera malheureusement pas réalisé dans le cadre de ce TPI car une fois que le projet a été présenté durant la défense oral, le projet prend fin.

### Product backlog

|  |  |
| --- | --- |
| **ID & Nom** | **01 : Mise en place d’un dépôt GIT** |
| **Description (user story)** | En tant que développeur, je souhaite pouvoir accéder à un dépôt distant hébergé sur GitHub. |
| **Critère d’acceptation** | Le dépôt Git est configuré dans le répertoire du projet. Le projet est accessible depuis GitHub à l’adresse indiquée dans la documentation technique |
| **Importance** | Bloquant |

|  |  |
| --- | --- |
| **ID & Nom** | **02 : Création des différentes maquettes d’interfaces utilisateur** |
| **Description (user story)** | En tant que développeur, je souhaite avoir plusieurs maquettes d’interface pour pouvoir développer l’application plus facilement plus tard. |
| **Critère d’acceptation** | Les maquettes sont créées et détaillées dans la documentations technique |
| **Importance** | Bloquant |

|  |  |
| --- | --- |
| **ID & Nom** | **03 : Implémentation de la base de données** |
| **Description (user story)** | En tant que développeur, je souhaite pouvoir utiliser une base de données MySQL. Je dois pouvoir utiliser un manager PHP me permettant d’effectuer des opérations sur la base de données. Je souhaite également avoir un dump de ma base de données sur le dépôt distant afin de pouvoir la recréer si nécessaire. |
| **Critère d’acceptation** | La base de données est créée et correspond au cahier des charges. Une classe Database est mise en place et permet d’effectuer les opérations de base sur la base. Le dump est accessible sur le dépôt. |
| **Importance** | Bloquant |

|  |  |
| --- | --- |
| **ID & Nom** | **04 : Connexion au site WEB** |
| **Description (user story)** | En tant qu’utilisateur non connecté, je peux me connecter au site avec mon compte et accéder aux fonctionnalité utilisateur. |
| **Critère d’acceptation** | Les test 1.X à 1.X sont validés |
| **Importance** | Bloquant |

|  |  |
| --- | --- |
| **ID & Nom** | **05 : Déconnexion** |
| **Description (user story)** | En tant qu’utilisateur connecté, je peux me déconnecter de l’application web et mettre fin à ma session d’utilisation. |
| **Critère d’acceptation** | Le test 2 est validé |
| **Importance** | Secondaire |

|  |  |
| --- | --- |
| **ID & Nom** | **06 : Inscription** |
| **Description (user story)** | En tant que visiteur du site web, je peux me créer un compte pour pouvoir accéder aux fonctionnalités utilisateur. |
| **Critère d’acceptation** | Les tests 3.X à 3.X sont validés |
| **Importance** | Critique |

|  |  |
| --- | --- |
| **ID & Nom** | **07 : Activation du compte par mail** |
| **Description (user story)** | En tant qu’utilisateur qui vient de s’inscrire, je peux activer mon compte à l’aide d’un lien de validation qui m’a été envoyé par mail. |
| **Critère d’acceptation** | Les tests 4.X à 4.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **08 : Envoi de mail après changement de statut** |
| **Description (user story)** | En tant qu’utilisateur, je veux recevoir un mail m’informant du changement de statut de mon compte et/ou d’un de mes itinéraires. |
| **Critère d’acceptation** | Les tests 5.X à 5.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **09 : Affichage des itinéraires existants** |
| **Description (user story)** | En tant que visiteur, je veux pouvoir voir tous les itinéraires créés par les utilisateurs avec un minimum de détail. |
| **Critère d’acceptation** | Les tests 6.X à 6.X sont validés |
| **Importance** | Critique |

|  |  |
| --- | --- |
| **ID & Nom** | **10 : Système de filtrage des itinéraires** |
| **Description (user story)** | En tant que visiteur, je peux filtrer la page d’accueil afin d’afficher les itinéraires d’après le filtre que j’ai mis en place. |
| **Critère d’acceptation** | Les tests 7.X à 7.X sont validés |
| **Importance** | Critique |

|  |  |
| --- | --- |
| **ID & Nom** | **11 : Création d’un itinéraire** |
| **Description (user story)** | En tant qu’utilisateur, je peux créer un itinéraire en complétant le formulaire de création d’itinéraire disponible en cliquant sur le bouton « Créer un itinéraire » |
| **Critère d’acceptation** | Les tests 8.X à 8.X sont validés |
| **Importance** | Critique |

|  |  |
| --- | --- |
| **ID & Nom** | **12 : Affichage des itinéraires de l’utilisateur** |
| **Description (user story)** | En tant qu’utilisateur, je peux afficher tous les itinéraires que j’ai créé en cliquant sur le bouton « Mes itinéraires » pour avoir une vue d’ensemble sur tous les itinéraires que j’ai créé. |
| **Critère d’acceptation** | Les tests 9.X à 9.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **13 : Modification d’un itinéraire** |
| **Description (user story)** | En tant qu’utilisateur, je peux modifier un itinéraire que j’ai créé afin de corriger certaine erreur. |
| **Critère d’acceptation** | Les tests 10.X à 10.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **14 : Suppression d’un itinéraire** |
| **Description (user story)** | En tant qu’utilisateur, je peux supprimer un itinéraire que j’ai créé en cliquant sur le bouton « Supprimer », sur la page de mon itinéraire, une confirmation est requise. |
| **Critère d’acceptation** | Les tests 11.X à 11.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **15 : Ajout/Modification des informations personnelles d’un utilisateur** |
| **Description (user story)** | En tant qu’utilisateur, je peux ajouter/modifier des informations personnelles depuis la page profil de mon compte pour que mon profil soit plus complet. |
| **Critère d’acceptation** | Les tests 12.X à 12.X sont validés |
| **Importance** | Secondaire |

|  |  |
| --- | --- |
| **ID & Nom** | **16 : Afficher informations itinéraire** |
| **Description (user story)** | En tant qu’utilisateur, je peux afficher les informations d’un itinéraire en détail en cliquant sur le nom de l’itinéraire afin de me renseigner sur l’itinéraire |
| **Critère d’acceptation** | Les tests 13.X à 13.X sont validés |
| **Importance** | Critique |

|  |  |
| --- | --- |
| **ID & Nom** | **17 : Affichage de commentaires** |
| **Description (user story)** | En tant qu’utilisateur, je peux voir les commentaires qui ont été posté par les autres utilisateurs afin de savoir quoi commenter. |
| **Critère d’acceptation** | Les tests 14.X à 14.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **18 : Ajout de commentaires** |
| **Description (user story)** | En tant qu’utilisateur, je veux pouvoir commenter un itinéraire afin de donner mon avis sur ce dernier. |
| **Critère d’acceptation** | Les tests 15.X à 15.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **19 : Système de notation des itinéraires** |
| **Description (user story)** | En tant qu’utilisateur, je veux pouvoir attribuer une note à un itinéraire afin de donner mon avis sur ce dernier |
| **Critère d’acceptation** | Les tests 16.X à 16.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **20 : Modification du statut de l’itinéraire** |
| **Description (user story)** | En tant qu’administrateur, je veux pouvoir changer le statut d’un itinéraire (Bloqué ou pas) afin de bloquer les itinéraires jugés invalides. |
| **Critère d’acceptation** | Les tests 17.X à 17.X sont validés |
| **Importance** | Important |

|  |  |
| --- | --- |
| **ID & Nom** | **21 : Modification du statut de l’utilisateur** |
| **Description (user story)** | En tant qu’administrateur, je veux pouvoir changer le statut de l’utilisateur afin de bloquer les utilisateurs jugés indésirables. |
| **Critère d’acceptation** | Les tests 19.X à 19.X sont validés |
| **Importance** | Important |

### Planification

Sur les tableaux suivants, vous trouverez ma planification prévisionnelle ainsi que le planning effectif du déroulement du projet.

Il y a eu pas mal de différence concernant l’ordre effectif des tâches. Ces différences sont dues au fait que j’ai négligé le temps que je prendrai pour développer l’application. Les quelques imprévus qui sont survenues ont suffi à me mettre en retard sur le planning.

De manière générale, j’ai eu un léger retard durant tous les projets. L’affichage d’itinéraire avec l’api mapquest m’a bloqué pendant un certain moment à cause du manque de documentation sur l’api. J’ai quand même réussi à rattraper ce retard dans les heures prévues pour et à terminer le projet à temps.

## C:\Users\Thomas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DA5BCC3F.tmpC:\Users\Thomas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7F826A49.tmpMaquettage

Planning prévisionnel 1

Planning effectif 1

Voici les différentes maquettes d’interfaces utilisateurs, dans l’application il existe 3 niveaux d’accès :

* Visiteur
* Utilisateur
* Administrateur

### Visiteur

En tant que visiteur, la page d’accueil est accessible avec tous les itinéraires afficher. Un visiteur peut également afficher les détails d’un itinéraire en cliquant dessus mais ne peut ni le noter, ni le commenter. Il ne peut pas non plus créer d’itinéraire.

###### Navbar

Voici la barre de navigation disponible sur l’application WEB en tant que visiteur.

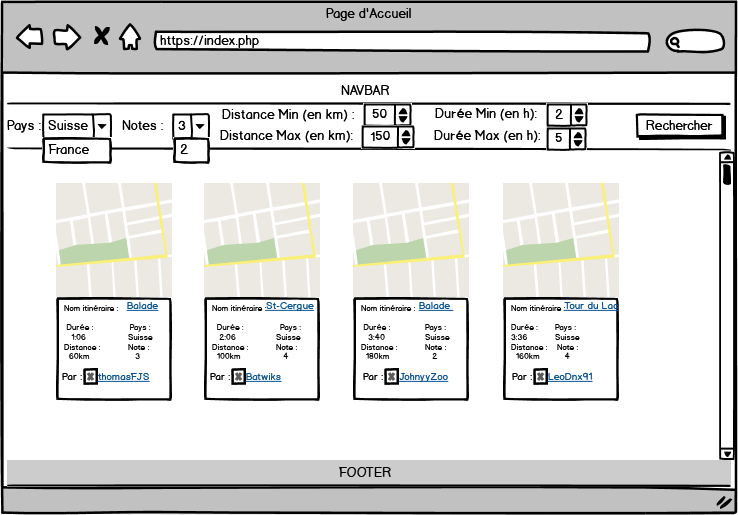
C

B

A

1. Lien redirigeant l’utilisateur à l’accueil (voir écran [*1) Accueil*](#_Accueil)*)*.
2. Lien redirigeant l’utilisateur sur l’écran de connexion (voir écran [4) Login](#_Login)).
3. Lien redirigeant l’utilisateur sur l’écran d’inscription (voir écran [*3) Inscription*](#_Inscription)).

#### Accueil

La page d’accueil de l’application. Elle répertorie les différents itinéraires qui ont été posté par les utilisateurs. Un système de filtrage en disponible en haut de la page. L’utilisateur peut voir les détails d’un itinéraire en cliquant sur le nom de celui-ci.

K

J

H

I

G

F

E

D

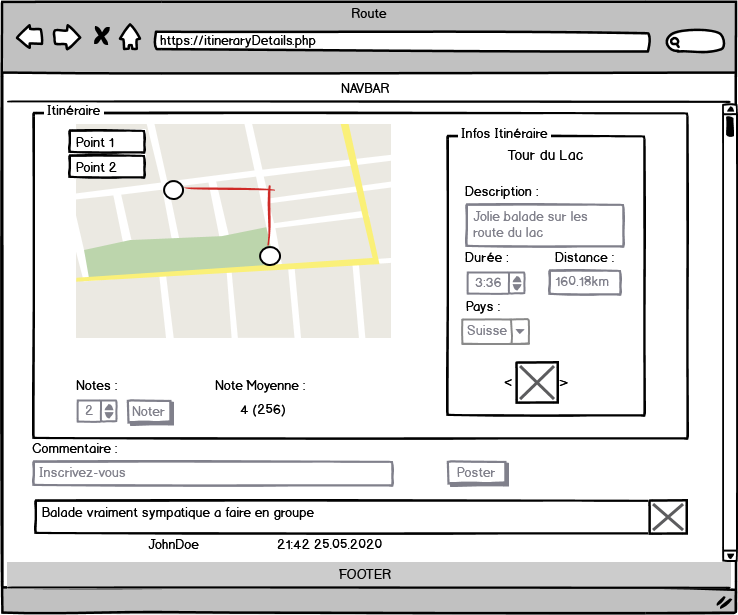
C

B

A

1. Filtrage du pays de départ de l’itinéraire.
2. Filtrage de la note moyenne minimum de l’itinéraire.
3. Filtrage de la distance désirée pour l’itinéraire, l’utilisateur peut choisir la distance minimum et maximum désirée (par 50km).
4. Filtrage de la durée désiré pour l’itinéraire, l’utilisateur peut choisir la durée minimum et maximum souhaitée (par 1h).
5. Bouton de recherche pour appliquer le filtre mis en place par l’utilisateur.
6. Le titre de l’itinéraire, un lien qui redirige sur la page itinéraire avec toutes les infos de l’itinéraire sélectionné.
7. La durée de l’itinéraire.
8. Le pays de départ de l’itinéraire.
9. La distance de l’itinéraire.
10. La note moyenne de l’itinéraire.
11. L’utilisateur qui a créé l’itinéraire.

#### Détail itinéraire

La page itinéraire qui permet d’afficher les détails d’un itinéraire, elle affiche également tous les commentaires publiés par les utilisateurs et la note moyenne de l’itinéraire.

F

D

C

E

G

L

K

J

I

B

A

H

1. Les différents points de l’itinéraire.
2. L’itinéraire affiché sur la carte.
3. Titre de l’itinéraire.
4. Description de l’itinéraire.
5. Durée de l’itinéraire en heure.
6. Distance de l’itinéraire en km.
7. Pays de départ de l’itinéraire
8. Les images publiées par l’utilisateur sur l’itinéraire
9. Système pour noter l’itinéraire (Désactivé en tant que visiteur non connecté)
10. Note moyenne de l’itinéraire avec le nombre d’utilisateur ayant voté
11. Zone de texte pour poster un commentaire (En tant que visiteur non connecté le champ est désactivez et il est proposé de s’inscrire au visiteur pour pouvoir poster un commentaire)
12. Commentaire posté par un autre utilisateur avec le nom de l’utilisateur, la date et l’heure à laquelle le commentaire a été posté ainsi que l’avatar ce dernier.

#### Inscription

Le formulaire d’inscription pour les visiteurs non connecté, le minimum y est demandé pour faciliter l’inscription de l’utilisateur.

E

F

D

C

A

B

1. Nickname désiré par l’utilisateur (Il doit être unique).
2. Email de l’utilisateur (Elle doit être unique).
3. Mot de passe de l’utilisateur.
4. Confirmation du mot de passe.
5. Bouton pour envoyer le formulaire d’inscription.
6. Lien de redirection si le visiteur possède déjà un compte.

### Utilisateur

En tant qu’utilisateur, tous les itinéraires sont affichés sur la page d’accueil. Il peut afficher les détails d’un itinéraire en cliquant dessus, il peut également le noter et le commenter. L’utilisateur peut créer des itinéraires.

###### Navbar

Voici la barre de navigation disponible sur l’application WEB en tant qu’utilisateur connecté. Elle reprend la redirection à l’accueil de la barre de navigation disponible en tant que visiteur (voir Navbar visiteur).

A

C

D

B

1. Lien redirigeant l’utilisateur sur l’écran de création d’itinéraire (voir écran [*6) Création d’itinéraire part. 1* ).](#_Création_d’itinéraire_part.)
2. Lien redirigeant l’utilisateur sur l’écran des itinéraires de l’utilisateur (voir écran [*1) Accueil*](#_Accueil)).
3. Lien redirigeant sur la page de profil de l’utilisateur connecté.
4. Lien permettant la déconnexion de l’utilisateur.

#### Login

La page de connexion qui permet aux utilisateurs de se connecter sur l’application WEB.

D

C

B

A

1. Nom d’utilisateur de l’utilisateur pour se connecter (Email ou Nickname)
2. Mot de passe de l’utilisateur
3. Bouton pour se connecter avec les infos saisit au-dessus
4. Lien qui redirige sur l’écran [3) Inscription](#_Inscription) si le visiteur ne possède pas de compte

#### Mes itinéraires

Voici la page ou tous les itinéraires créés par l’utilisateur connecté sont affiché il peut cliquer sur un de ses itinéraires pour voir les détails.

E

C

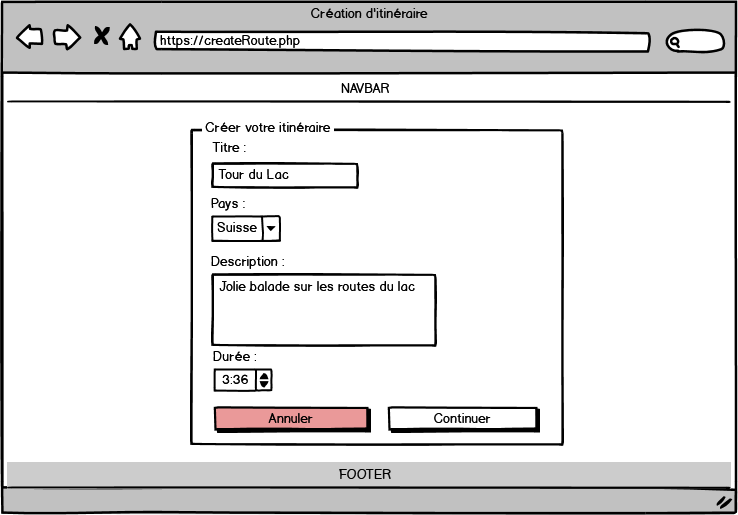
D

B

A

1. Titre de l’itinéraire (lien qui redirige sur l’écran [9) Détail mon itinéraire](#_Détail_mon_itinéraire)).
2. Durée de l’itinéraire.
3. Pays de départ de l’itinéraire.
4. Distance de l’itinéraire.
5. Note moyenne de l’itinéraire.

#### Création d’itinéraire part. 1

Voici la première partie de la création d’un itinéraire. Pour commencer l’utilisateur doit remplir un formulaire contenant les informations générales de l’itinéraire et pourra ensuite le placer sur une map.

F

E

D

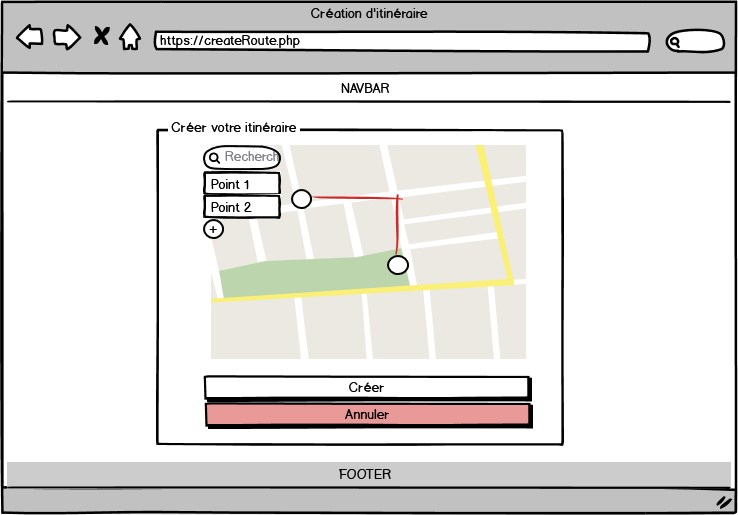
C

B

A

1. Titre de l’itinéraire
2. Pays de départ de l’itinéraire
3. Description de l’itinéraire
4. Durée de l’itinéraire.
5. Bouton pour soumettre la première partie du formulaire de création d’itinéraire.
6. Bouton pour annuler la création de l’itinéraire

#### Création d’itinéraire part. 2

Voici la deuxième partie de la création d’un itinéraire. Pour pouvoir finaliser la création de votre itinéraire il faut le placer sur la map en marquant les différents points de l’itinéraire. Une fois l’itinéraire placé et le formulaire envoyé l’itinéraire sera immédiatement créé.

E

F

D

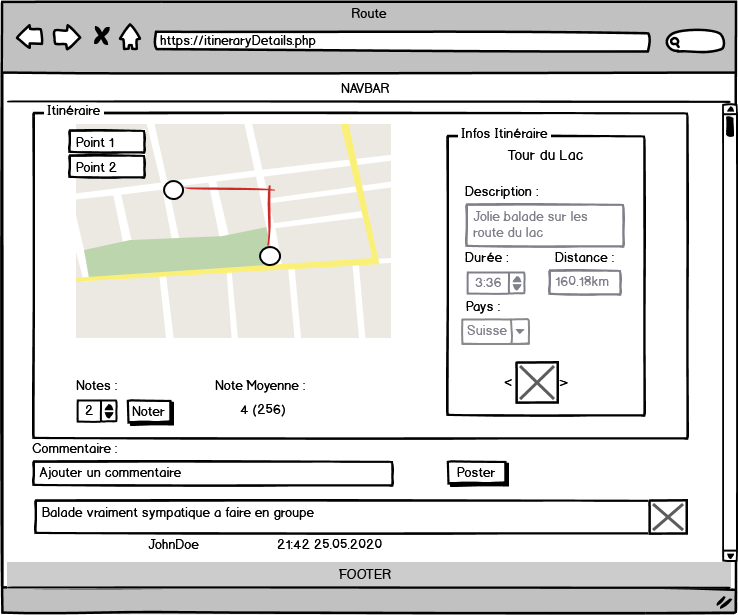
C

B

A

1. Barre de recherche pour se placer sur la map.
2. Les différents points placés dans l’itinéraire.
3. Bouton pour rajouter un point dans l’itinéraire.
4. Prévisualisation de l’itinéraire.
5. Bouton pour confirmer la création de l’itinéraire.
6. Bouton pour annuler la création de l’itinéraire.

#### Détail itinéraire

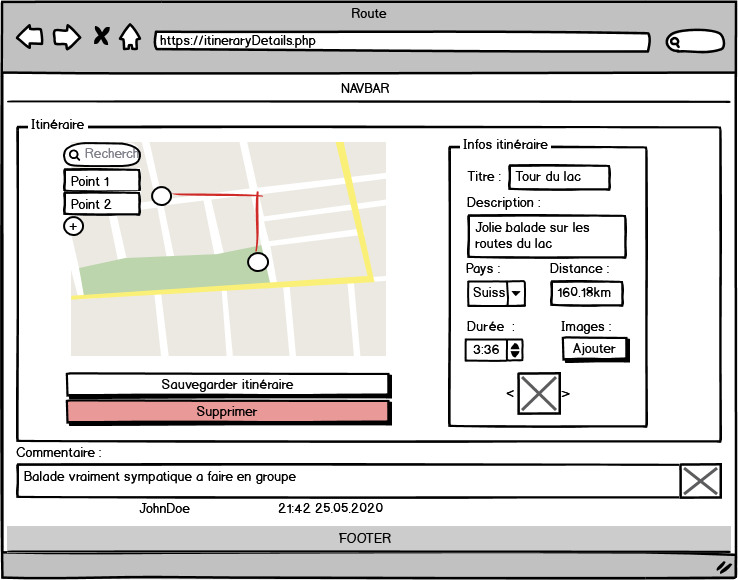
Page contenant les détails d’un itinéraire en étant connecté en tant qu’utilisateur. Cette page est là même que la page détail itinéraire en tant que visiteur (voir écran [*2) Détail itinéraire*](#_Détail_itinéraire_2)) avec l’espace commentaire qui est débloqué et l’utilisateur peut également noter l’itinéraire.

A

B

1. Note que l’utilisateur peut attribuer à l’itinéraire en appuyant sur le bouton « Noter »
2. Zone de texte pour ajouter un commentaire à l’itinéraire (Le commentaire est poster en cliquant sur le bouton « Poster »

#### Détail mon itinéraire

L’écran affiché lorsqu’un utilisateur clique sur un de ses itinéraires. Les détails sont affichés comme sur l’écran *8) Détail itinéraire*. L’utilisateur peut effectuer des modifications sur son itinéraire ou alors le supprimer.

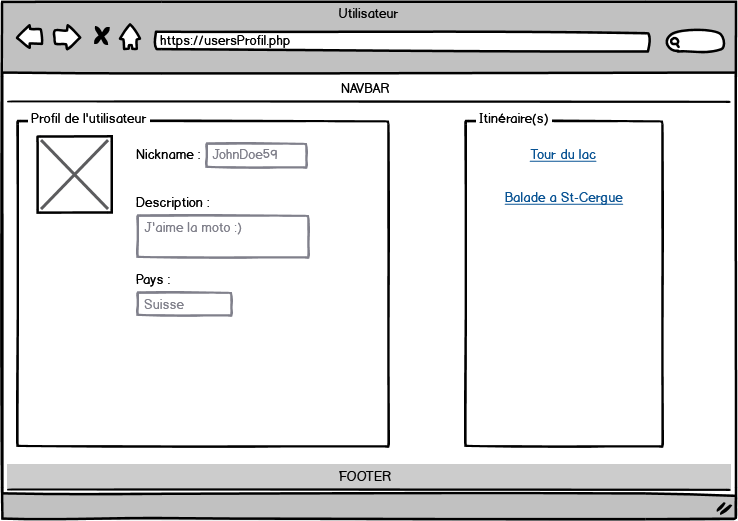
C

B

A

1. Bouton pour sauvegarder les modifications de l’itinéraire
2. Bouton pour supprimer l’itinéraire, message de confirmation si l’utilisateur clique.
3. Bouton pour ajouter des images à l’itinéraire

#### Profil utilisateur

La page profil qui affiche les infos d’un utilisateur ainsi que tous les itinéraires créés par ce dernier. Les itinéraires sont affichés sous forme de lien qui redirige directement sur la page de l’itinéraire concerné.

E

D

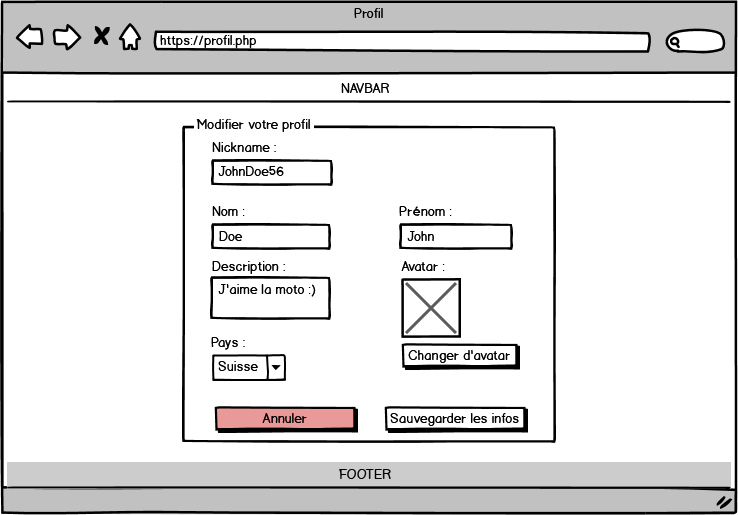
C

B

A

1. Avatar de l’utilisateur.
2. Nickname de l’utilisateur.
3. Description de l’utilisateur.
4. Pays de résidence de l’utilisateur.
5. Liste des itinéraires créés par l’utilisateur (Les titres des différents itinéraires sont des liens qui redirige sur l’écran [8) Détail itinéraire](#_Détail_itinéraire)).

#### Modification Profil

Voici la page de modification du profil de l’utilisateur qui est connecté, il peut modifier/compléter ses informations pour mieux détailler son profil.

H

G

F

E

D

C

B

A

1. Nickname de l’utilisateur.
2. Nom de l’utilisateur.
3. Prénom de l’utilisateur.
4. Description de l’utilisateur.
5. Avatar de l’utilisateur avec un bouton pour importer un nouvel avatar
6. Pays de résidence de l’utilisateur
7. Bouton pour sauvegarder les changements d’infos
8. Bouton pour annuler les changements effectués, si l’utilisateur clique il est redirigé à l’accueil voir écran *1) Accueil*

### Administrateur

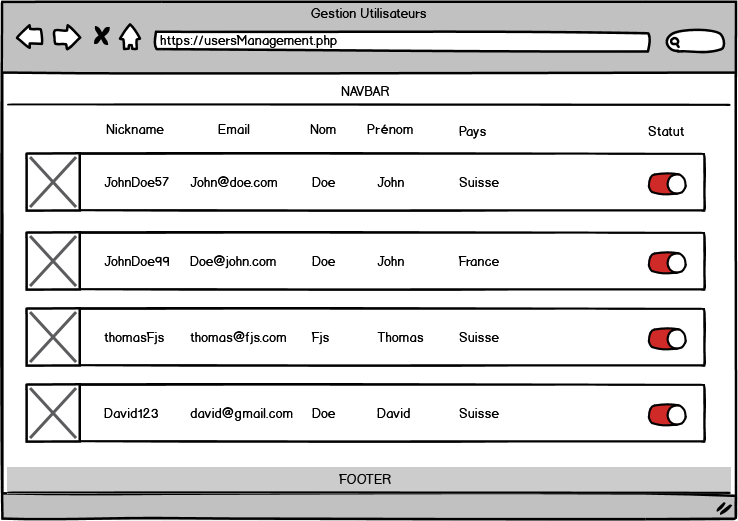
###### Navbar

Voici la barre de navigation disponible sur l’application WEB en tant qu’administrateur. Elle reprend la redirection à l’accueil de la barre de navigation disponible en tant que visiteur (voir Navbar visiteur) ainsi que le lien de déconnexion disponible en tant qu’utilisateur (voir Navbar utilisateur).

A

1. Redirection sur la page de gestion d’utilisateur. (Voir écran 14) Gestion utilisateurs)

#### Gestion utilisateurs

Page qui affiche une liste de tous les utilisateurs avec quelques infos, l’administrateur possède l’option de changer le statut des comptes utilisateurs.

E

D

G

F

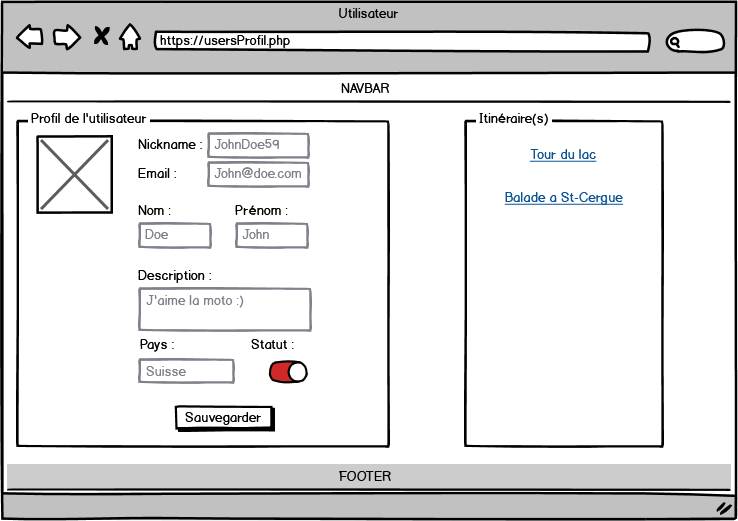
C

A

B

1. Avatar de l’utilisateur.
2. Nickname de l’utilisateur.
3. Email de l’utilisateur.
4. Nom de l’utilisateur.
5. Prénom de l’utilisateur.
6. Pays de résidence de l’utilisateur.
7. Statut du compte de l’utilisateur.

#### Profil utilisateur

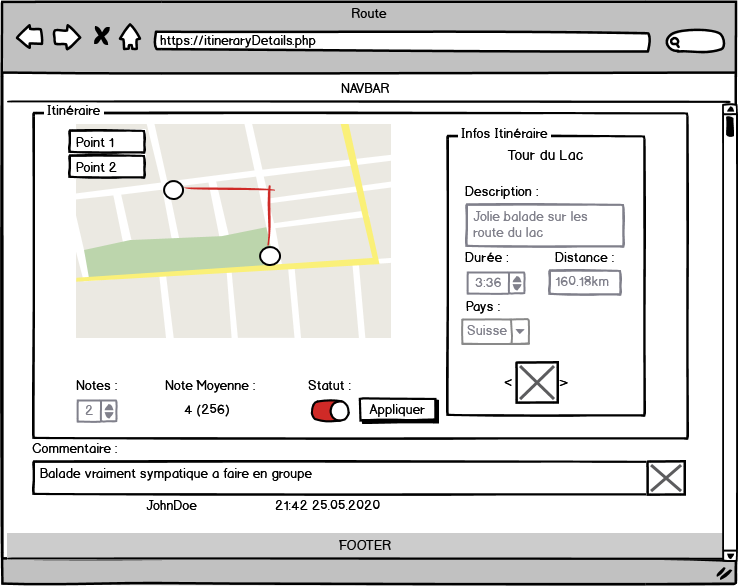
Page qui affiche le profil d’un utilisateur (voir écran *9) Profil utilisateur* ) avec l’option administrateur qui permet de changer le statut de l’utilisateur.

B

A

1. Switch (On/Off) qui permet à l’administrateur de changer le statut de l’utilisateur. (Pour valider le changement, l’administrateur doit appuyer sur le bouton « Sauvegarder »)
2. Liste des itinéraires créés par l’utilisateur (Les titres des différents itinéraires sont des liens qui redirige sur l’écran [14) Détail itinéraire](#_Détail_itinéraire_1))

#### Détail itinéraire

Lorsque l’administrateur clique sur un itinéraire il est redirigé sur cette page qui affiche les détails d’un itinéraire (voir *écran 8) Détail itinéraire*) avec l’option administrateur qui permet de changer le statut de l’itinéraire.

A

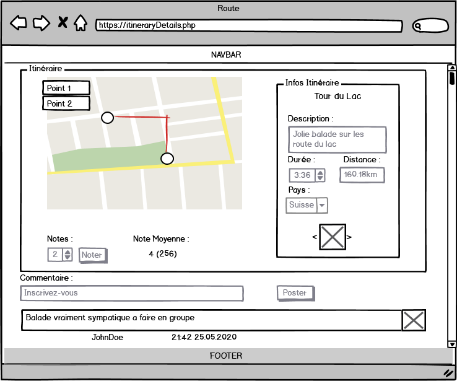
1. Switch (On/Off) qui permet à l’administrateur de changer le statut de l’itinéraire. (Pour valider le changement de statut, il faut appuyer sur le bouton « Appliquer »).

## Plan de navigation

Les différents plans de navigation pour les 3 niveaux d’accès :

### Visiteur

1

Le visiteur peut accéder à la page d’accueil (écran 1), la page de connexion (écran 4), la page d’inscription (écran 3) ainsi qu’à la page de détails d’un itinéraire (écran 2).

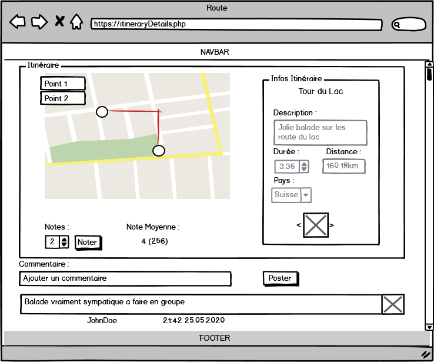
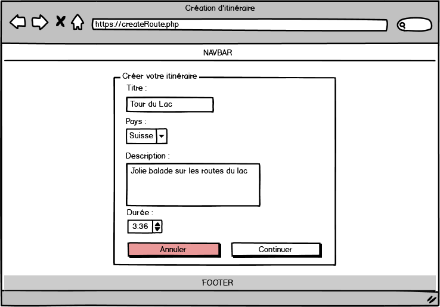
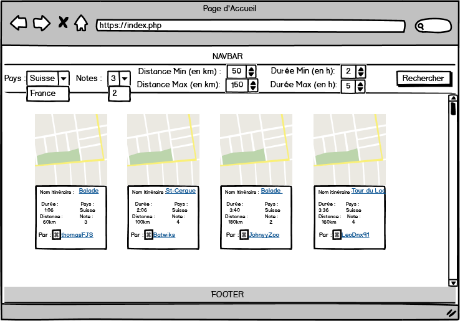
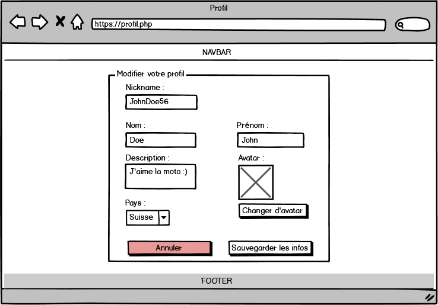
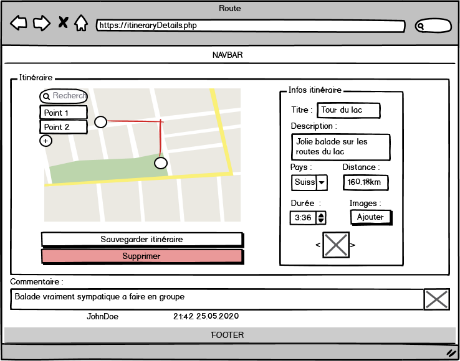
4

3

2

### Utilisateur

L’utilisateur connecté peut accéder à la page d’accueil (écran 1), la page de création d’itinéraire (écran 6), la page mes itinéraires (écran 5), la page mon profil (écran 11), la page détails de l’itinéraire d’un autre utilisateur (écran 8) ainsi que la page détails d’un itinéraire qu’il a créé (écran 9).



8

9

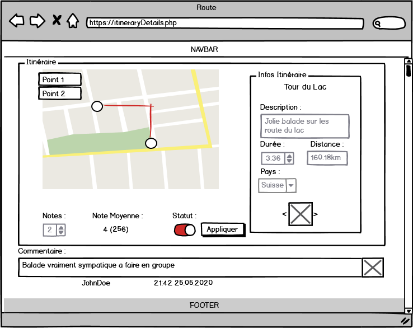
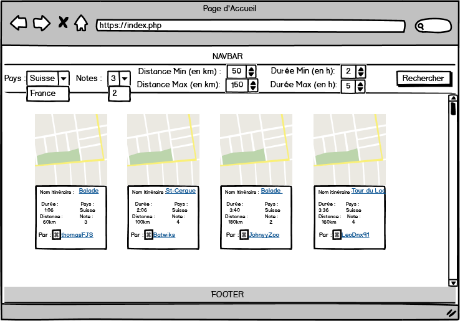
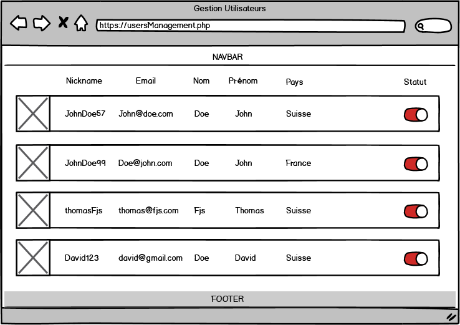
11

5

6

1

### Administrateur

L’administrateur peut accéder à la page d’accueil (écran 1), la page de gestion d’utilisateurs (écran 12) ainsi que la page détails d’itinéraire avec l’option administrateur (écran 14).

14

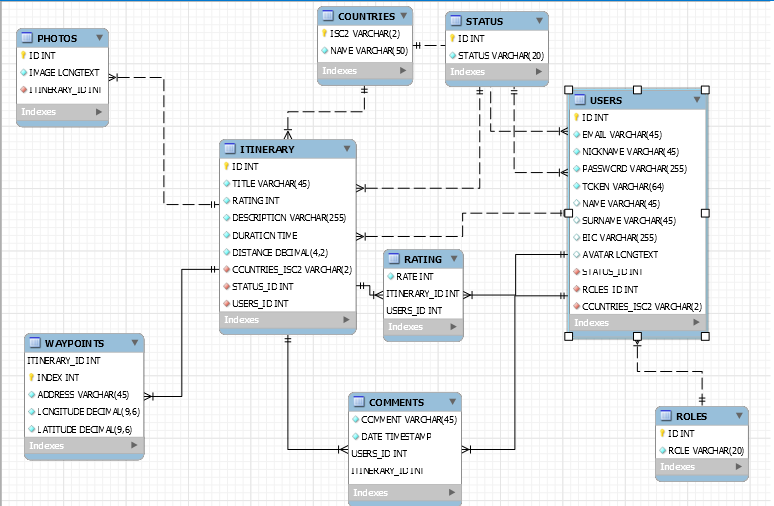
12

1

## Implémentation

### Base de données

L’application web, Travler, utilise une base de données MySQL afin de stocker les informations utiles pour l’application. J’ai utilisé la version 8.0.20 de MySQL pour le développement de mon projet. La base de données utilise le moteur de stockage par défaut : *InnoDB* et l’interclassement *utf8mb4\_0900\_ai\_ci*.

Voici le modèle physique de données :

Le modèle physique de données

Le script SQL permettant de créer toute la base de données a été exporté dans le dossier : **/docs/database/script/.**

#### Dictionnaire de données

Voici le dictionnaire de données de l’application *Travler*:

##### USERS

Table « USERS » pour stocker toutes les données concernant un utilisateur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ID (*Primaire*) | INT | L’id unique de l’utilisateur | Non | / |
| EMAIL | VARCHAR(45) | L’email de l’utilisateur | Non | / |
| NICKNAME | VARCHAR(45) | Le nickname de l’utilisateur | Non | / |
| PASSWORD | VARCHAR(255) | Le mot de passe de l’utilisateur | Non | / |
| TOKEN | VARCHAR(64) | Le token d’activation du compte | Non | / |
| NAME | VARCHAR(45) | Le nom de l’utilisateur | Oui | NULL |
| SURNAME | VARCHAR(45) | Le prénom de l’utilisateur | Oui | NULL |
| BIO | MEDIUMTEXT | La description de son profil | Oui | NULL |
| AVATAR | LONGTEXT | L’avatar de l’utilisateur | Non | NULL |
| STATUS\_ID (*Étrangère*) | INT | Le code qui définit le statut du compte | Non | / |
| ROLES\_ID (*Étrangère*) | INT | Le code qui définit le role de l’utilisateur | Non | / |
| COUNTRIES\_ISO2 (*Étrangère)* | VARCHAR(2) | Le code iso2 du pays de résidence de l’utilisateur | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ID\_USERS | Non |
| STATUS\_ID | BTREE | Non | STATUS\_ID | Non |
| ROLES\_ID | BTREE | Non | ROLES\_ID | Non |
| COUNTRIES\_ISO2 | BTREE | Non | COUNTRIES\_ISO2 | Non |

##### ITINERARY

Table « ITINERARY » pour stocker toutes les données concernant un itinéraire.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ID (*Primaire*) | INT | L’id unique de l’itinéraire | Non | / |
| TITLE | VARCHAR(45) |  | Non | / |
| RATING | DOUBLE(2,1) | La note moyenne de l’itinéraire | Oui | / |
| DESCRIPTION | MEDIUMTEXT | La description de l’itinéraire | Non | / |
| DURATION | TIME | La durée de l’itinéraire | Non | / |
| DISTANCE | DECIMAL(6,2) | La distance totale de l’itinéraire | Non | / |
| PREVIEW | LONGTEXT | L’image de l’itinéraire encodé en base 64 | Non | / |
| COUNTRIES\_ISO2 (*Étrangère*) | VARCHAR(2) | La code iso2 du pays de départ de l’itinéraire | Non | / |
| STATUS\_ID (*Étrangère*) | INT | Le code qui définit le statut de l’itinéraire | Non | / |
| USERS\_ID (*Étrangère)* | INT | L’id unique de l’utilisateur qui a créé l’itinéraire | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ID | Non |
| STATUS\_ID | BTREE | Non | STATUS\_ID | Non |
| USERS\_ID | BTREE | Non | USERS\_ID | Non |
| COUNTRIES\_ISO2 | BTREE | Non | COUNTRIES\_ISO2 | Non |

##### COMMENTS

Table « COMMENTS » pour stocker toutes les données concernant un commentaire.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| COMMENT | VARCHAR(255) | Le texte du commentaire | Non | / |
| COMMENT\_DATE | TIMESTAMP | La date du commentaire | Non | CURRENT\_TIMESTAMP |
| ITINERARY\_ID (*Primaire*)(*Étrangère*) | INT | L’id unique de l’itinéraire ou le commentaire a été posté | Non | / |
| USERS\_ID (*Primaire)*(*Étrangère*) | INT | L’id unique de l’utilisateur qui a posté le commentaire | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| ITINERARY\_ID | BTREE | Non | ITINERARY\_ID | Non |
| USERS\_ID | BTREE | Non | STATUS\_ID | Non |

##### WAYPOINTS

Table « WAYPOINTS » pour stocker toutes les données concernant un point de l’itinéraire.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ITINERARY\_ID (*Primaire*)(*Étrangère)* | INT | L’id unique de l’itinéraire | Non | / |
| INDEX  (*Primaire)* | INT | L’index du point dans l’itinéraire | Non | / |
| ADDRESS | VARCHAR(100) | L’adresse du point | Non | / |
| LONGITUDE | DECIMAL(9,6) | La longitude du point | Non | / |
| LATITUDE | DECIMAL(9,6) | La latitude du point | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| ITINERARY\_ID | BTREE | Non | ITINERARY\_ID | Non |
| INDEX | BTREE | Non | INDEX | Non |

##### RATING

Table « RATING » pour stocker les notes que les utilisateurs ont attribué aux itinéraires

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| RATE | INT(2) | La note que l’utilisateur a attribuée | Non | / |
| ITINERARY\_ID (*Étrangère*)(*Primaire)* | INT | L’id unique de l’itinéraire que l’utilisateur a noté | Non | / |
| USERS\_ID (*Étrangère*)(*Primaire)* | INT | L’id unique de l’utilisateur qui a noté | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| ITINERARY\_ID | BTREE | Non | ITINERARY\_ID | Non |
| USERS\_ID | BTREE | Non | USERS\_ID | Non |

##### PHOTOS

Table « Photos » pour stocker les photos importées par les utilisateurs pour les itinéraires.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ID (*Primaire*) | INT | L’id unique de la photo | Non | / |
| IMAGE | LONGTEXT | La photo encodée en base 64 | Non | / |
| ITINERARY\_ID (*Étrangère*) | INT | L’id unique de l’itinéraire sur lequel la photo est importé | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ID | Non |
| ITINERARY\_ID | BTREE | Non | ITINERARY\_ID | Non |

##### COUNTRIES

Table « COUNTRIES » pour stocker tous les pays disponibles pour différentes fonctionnalités de l’application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ISO2 (*Primaire*) | VARCHAR(2) | Le code iso2 du pays | Non | / |
| NAME | VARCHAR(50) | Le nom complet du pays | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ISO2 | Non |

##### ROLES

Table « ROLES » pour stocker les différents rôles pour l’application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ID (*Primaire*) | INT | L’id unique du rôle | Non | / |
| ROLE | VARCHAR(20) | Le titre du rôle | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ID | Non |

##### STATUS

Table « STATUS » pour stocker les différents statuts disponibles dans l’application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Description | Null | Défaut |
| ID (*Primaire*) | INT | L’id unique du statut | Non | / |
| STATUS | VARCHAR(20) | Le titre du statut | Non | / |

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Colonne | Null |
| PRIMARY | BTREE | Oui | ID | Non |

### C:\Users\Thomas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\2F2F96CA.tmpStructure du projet

La structure que j’ai utilisé pour l’arborescence de mon projet est la suivante :

* **/docs / :** Contient tous les fichiers de documentation sur l’application web.
* **/src/www/ :** Racine du serveur web, contient tous les fichiers accessibles directement par les utilisateurs.
* **/app/**: Contient tous les managers, modèles, vues et api interne.
  + **/app/api/**:Contient toutes les API interne
  + **/app/manager/**: Contient tous les managers PHP
  + **/app/model/**: Contient tous les modèles PHP
  + **/app/view/**: Contient toutes les vues PHP
* **/assets/**: Contient tous les fichiers sources front-end (JS et CSS).
  + **/assets/css/**: Feuilles de styles pour l’application.
  + **/assets/js/**: Tous les scripts Javascript de l’application.
  + **/assets/resources/images/**: Les images de l’application.
* **/config/**: Contient le fichier config.php, qui est utiliser pour configurer l’accès à la base de données et l’utilisation du mailer.
* **/constants/**: Contient les fichiers de constantes PHP et JavaScript.
* **/inc/**: Contient les différents fichiers PHP à inclure sur les pages de l’application.
  + **/inc/form/**: Les formulaires à inclure.
  + **/inc/inc.all/**: Contient le fichier inc.all.php qui permet d’inclure tous les fichiers PHP nécessaire.
  + **/inc/navbar/**: Les barres de navigation à inclure.
* **/swiftmailer5/**: Contient tous les dossiers/fichiers nécessaire pour l’utilisation du SwiftMailer
* **/test/unit/**: Contient tous les tests unitaires de l’application.

### Container (PHP)

Les objets conteneur permettent de stocker les informations sauvegardées en base de données.

###### FUser

Cette classe contient toutes les informations des différents utilisateurs.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Id | L’id de l’utilisateur | INT |
| Email | L’email de l’utilisateur | STRING |
| Nickname | Le pseudo de l’utilisateur | STRING |
| Name | Le nom de l’utilisateur | STRING |
| Surname | Le prénom de l’utilisateur | STRING |
| Bio | La description du profil de l’utilisateur | STRING |
| Avatar | L’avatar de l’utilisateur encodé en base 64 | STRING |
| Country | Le code iso2 du pays de résidence de l’utilisateur | STRING |
| Status | Code qui détermine le statut du compte (1 = **Inscrit** / 2 = **Activé** / 3 = **Bloqué**) | INT |
| Role | Code qui détermine le rôle du compte | INT |

###### FCountry

Cette classe contient le nom et le code ISO2 de tous les pays stockés dans la base de données.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Code | Le code ISO2 du pays | STRING |
| Name | Le nom complet du pays | STRING |

###### FWaypoint

Cette classe contient toutes les informations des différents points sur la map utilisé dans les itinéraires.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Index | L’index du waypoints | INT |
| Address | L’adresse du point sur la map | STRING |
| Longitude | La longitude du point | STRING |
| Latitude | La latitude du point | STRING |

###### FComment

Cette classe contient toutes les informations des commentaires postés sur les itinéraires.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Comment | Le commentaire posté par l’utilisateur | STRING |
| PostedAt | La date et l’heure à laquelle le commentaire a été posté | STRING |
| User | L’utilisateur qui a posté le commentaire | FUser |

###### FPhoto

Cette classe permet de stocker les photos des différents itinéraires.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Image | L’image encodé en base 64 | STRING |

###### FItinerary

Cette classe permet de stocker toutes les informations d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Variables | Description | Type |
| Id | L’id de l’itinéraire | INT |
| Title | Le titre de l’itinéraire | STRING |
| Rating | La note moyenne de l’itinéraire | STRING |
| Description | La description de l’itinéraire | STRING |
| Duration | La durée de l’itinéraire | TIMESTAMP |
| Distance | La distance totale de l’itinéraire | DOUBLE |
| Country | Le pays de départ de l’itinéraire | STRING |
| Status | Le statut de l’itinéraire | STRING |
| Waypoints[] | Tableau contenant tous les *waypoints* de l’itinéraire | ARRAY<FWaypoint> |
| Comments[] | Tableau contenant tous les commentaires de l’itinéraire | ARRAY<FComment> |
| Photos[] | Tableau contenant toutes les photos de l’itinéraire qui ont été ajouté par l’utilisateur. | ARRAY<FPhoto> |

### Manager (PHP)

#### FUserManager

Ce manager contient toutes les fonctions traitant et/ou récupérant des données en relation avec un utilisateur.

##### Login($username, $password)

Fonction qui permet de connecter l’utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| username | Le mail ou le nickname saisit par l’utilisateur | STRING |
| password | Le mot de passe saisit par l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  True si OK, False si un champ est incorrect |

##### Register($nickname, $email, $password)

Fonction qui permet d’inscrire l’utilisateur dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| nickname | le nickname de l’utilisateur | STRING |
| email | L’email de l’utilisateur | STRING |
| password | Le mot de passe saisit par l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  True si l’inscription est OK, False si une erreur est survenue lors de l’insertion. |

##### GetHashPassowrd($loginValue)

Fonction qui permet de récupérer le mot de passe hashé d’un utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| loginValue | L’email ou le nickname de l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  Le mot de passe hashé de l’utilisateur, False si une erreur est survenue. |

##### IsAllowed($user)

Fonction qui permet de savoir si un utilisateur aux droits administrateurs ou pas.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| user | L’objet Fuser de l’utilisateur a vérifier | FUser |

|  |
| --- |
| Valeur retournée  True si l’utilisateur à les droits, sinon False |

##### IsLogged()

Fonction qui permet de savoir si un utilisateur est connecté ou pas.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  True si l’utilisateur est connecté, sinon False |

##### ActivateAccount($token, $userEmail)

Fonction qui permet d’inscrire l’utilisateur dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| token | Le token d’activation que l’utilisateur a reçu par mail | STRING |
| userEmail | L’email du compte utilisateur a activer | STRING |

|  |
| --- |
| Valeur retournée  True si l’activation est OK, False si une erreur est survenue lors de l’update dans la bd. |

##### VerifyAccount($nicknmae)

Fonction qui permet de vérifier si le compte est activé ou pas.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| nickname | Le nickname de l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  True si le compte est activé, False si le compte est désactivé. |

##### GetById($id)

Fonction qui permet de récupérer les données d’un utilisateur avec son id.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| id | L’id de l’utilisateur | INT |

|  |
| --- |
| Valeur retournée  Un objet *FUser* contenant tous les détails de l’utilisateur |

##### GetByNickname($nickname)

Fonction qui permet de récupérer les données d’un utilisateur avec son nickname.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| nickname | Le pseudo de l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  Un objet *FUser* contenant tous les détails de l’utilisateur. |

##### GetAll()

Fonction qui permet de récupérer tous les utilisateurs stockés dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  Tableau contenant tous les itinéraires. (Array<FUser>) |

##### UpdateInfos($idUser, $nickname, $name, $surname, $bio, $country, $avatar)

Fonction qui permet de mettre à jour les informations personnelles d’un utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idUser | L’id de l’utilisateur | INT |
| nickname | Le pseudo de l’utilisateur | STRING |
| name | Le nom de l’utilisateur | STRING |
| surname | Le prénom de l’utilisateur | STRING |
| bio | La bio du profil de l’utilisateur | STRING |
| country | Le pays de résidence de l’utilisateur | STRING |
| avatar | L’avatar de l’utilisateur encodé en base 64 | STRING |

|  |
| --- |
| Valeur retournée  True si les infos ont été mises à jour, sinon False |

##### DisableAccount($nickname)

Fonction qui permet de désactiver un compte utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| nickname | Le pseudo de l’utilisateur | STRING |

|  |
| --- |
| Valeur retournée  True si le compte est désactivé, false si une exception est intervenue. |

#### FSessionManager

Ce manager contient toutes les fonctions traitant les sessions.

##### SetUserLogged($user)

Fonction qui permet de définir l’utilisateur connecté en variable de session.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| user | Le conteneur utilisateur de l’utilisateur actuellement connecté | FUser |

|  |
| --- |
| Valeur retournée  True si la variable de session est set, sinon False. |

##### GetUserLogged()

Fonction qui permet de récupérer l’utilisateur connecté en variable de session.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  L’objet Fuser stocké en session si l’utilisateur est connecté, sinon un objet Fuser qui définit un visiteur. |

##### Reset()

Fonction qui permet de reset la session actuelle.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  True si le reset de la session a réussi. |

##### SetItineraryFilter($filters)

Fonction qui permet de définir les filtres d’affichages d’itinéraire définit par l’utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| filters | Un tableau contenant tous les filtres à définir | array |

|  |
| --- |
| Valeur retournée  True si les filtres sont définit, sinon False. |

##### GetItineraryFilter()

Fonction qui permet de récupérer les filtres d’affichages d’itinéraire définit par l’utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  Le tableau avec tous les filtres définit, sinon False |

##### UnsetItineraryFilter()

Fonction qui permet d’effacer les filtres d’affichages d’itinéraires définit par l’utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  True si les filtres sont effacés, sinon false |

#### FDatabaseManager

Ce manager contient toutes les fonctions traitant la base de données.

##### GetDb()

Fonction qui permet récupérer l’instance PDO pour la connexion a la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  L’instance PDO. |

#### FMailerManager

Ce manager contient toutes les fonctions utiliser en relation avec le mailer mit en place avec *swiftMailer*.

##### Send($subject, $userMail, $body)

Fonction qui permet d’envoyer un mail avec le swiftmailer.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| subject | Le sujet du mail à envoyer | STRING |
| userMail | Tableau contenant le(s) email(s) concerné par le mail | ARRAY |
| body | Le texte qui sera envoyé dans le mail | STRING |

|  |
| --- |
| Valeur retournée  True si le mail est envoyé correctement, False si une exception survient. |

##### GetActivationMail($token)

Fonction qui permet de récupérer le corps du mail d’activation avec le lien permettant d’activer le compte.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| token | Le token d’activation du compte | STRING |

|  |
| --- |
| Valeur retournée  Le corps du mail d’activation contenant le token pour activer le compte. |

##### GetDisableAccountMail()

Fonction qui permet de récupérer le corps du mail d’avertissement pour la désactivation d’un compte utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  Le corps du mail d’avertissement pour le changement de statut d’un compte utilisateur. |

##### GetDisableItineraryMail($nickname, $title)

Fonction qui permet de récupérer le corps du mail d’avertissement pour le changement de statut d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| title | Le titre de l’itinéraire dont le statut a été changé | STRING |

|  |
| --- |
| Valeur retournée  Le corps du mail d’avertissement pour le changement de statut d’un itinéraire. |

##### GetCommentAddMail($comment, $nicknameUser, $itineraryTitle)

Fonction qui permet de récupérer le corps du mail d’avertissement pour un ajout de commentaire sur un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| comment | Le commentaire qui a été posté | STRING |
| nicknameUser | Le pseudo de l’utilisateur qui a posté le commentaire | STRING |
| itineraryTitle | Le titre de l’itinéraire | STRING |

|  |
| --- |
| Valeur retournée  Le corps du mail d’avertissement pour l’ajout d’un commentaire sur un itinéraire. |

#### FItineraryManager

Ce manager contient toutes les fonctions permettant de récupérer/traiter des itinéraires

##### GetAll()

Fonction qui permet de récupérer tous les itinéraires stockés dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| / | / | / |

|  |
| --- |
| Valeur retournée  Tableau contenant tous les itinéraires. (Array<*FItinerary*>) |

##### GetAllWithFilter($filters)

Fonction qui permet de récupérer tous les itinéraires stockés dans la base de données avec un filtre définit par l’utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| filters | Tableau contenant tous les filtres à appliquer | ARRAY |

|  |
| --- |
| Valeur retournée  Tableau contenant tous les itinéraires filtrés. (Array<*FItinerary*>) |

##### GetAllByUserId($idUser)

Fonction qui permet de récupérer tous les itinéraires stockés dans la base de données qui ont été créé par un utilisateur.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idUser | L’id de l’utilisateur | INT |

|  |
| --- |
| Valeur retournée  Tableau contenant tous les itinéraires créés par l’utilisateur. (Array<*FItinerary*>) |

##### GetById($id)

Fonction qui permet de récupérer les détails d’un itinéraire stocké dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| id | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  Un objet *FItinerary* contenant tous les détails de l’itinéraire |

##### GetByTitle($title)

Fonction qui permet de récupérer les détails d’un itinéraire grâce à son titre.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| title | Titre de l’itinéraire | STRING |

|  |
| --- |
| Valeur retournée  Un objet *FItinerary* contenant tous les détails de l’itinéraire |

##### Create($title, $description, $duration, $distance, $country, $waypoints[], $idUser, $photos[])

Fonction qui permet de créer un itinéraire et de l’enregistrer dans la base de données.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| title | Le titre de l’itinéraire | STRING |
| description | La description de l’itinéraire | STRING |
| duration | La durée de l’itinéraire | DATETIME(« hh.mm ») |
| distance | La distance de l’itinéraire | DOUBLE |
| country | Le pays de départ de l’itinéraire | STRING |
| waypoints[] | Tableau contenant tous les waypoints placés par l’utilisateur | ARRAY<FWaypoint> |
| idUser | L’id de l’utilisateur qui a créé l’itinéraire | INT |
| photos[] | Tableau contenant toutes les photos que l’utilisateur a importé pour l’itinéraire. | ARRAY<FPhoto> |

|  |
| --- |
| Valeur retournée  True si l’itinéraire est créé, sinon false |

##### Update($idUser, $idItinerary, $title, $description, $duration, $distance, $country, $waypoints[], $photos[])

Fonction qui permet de mettre à jour un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idUser | L’id de l’utilisateur | INT |
| idItinerary | L’id de l’itinéraire | INT |
| title | Le titre de l’itinéraire | STRING |
| description | La description de l’itinéraire | STRING |
| duration | La durée de l’itinéraire | DATETIME(« hh.mm ») |
| distance | La distance de l’itinéraire | DOUBLE |
| country | Le pays de départ de l’itinéraire | STRING |
| waypoints[] | Tableau contenant tous les waypoints placés par l’utilisateur | ARRAY<FWaypoint> |
| photos[] | Tableau contenant toutes les photos que l’utilisateur a importé pour l’itinéraire. | ARRAY<FPhoto> |

|  |
| --- |
| Valeur retournée  True si l’itinéraire est mit à jour, sinon false |

##### UpdateRating($newRating, $idItinerary)

Fonction qui permet d’ajouter un commentaire à un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| newRating | La nouvelle moyenne de note de l’itinéraire | FLOAT |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  True si le rating est mis à jours , False si exception |

##### Disable($idItinerary, $title)

Fonction qui permet de désactiver un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |
| title | Le titre de l’itinéraire | STRING |

|  |
| --- |
| Valeur retournée  True si l’itinéraire est désactivé, sinon False |

##### Delete($idItinerary)

Fonction qui permet de supprimer un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  True si l’itinéraire est supprimé, sinon False |

#### FCommentManager

Ce manager contient toutes les fonctions permettant de récupérer/traiter des commentaires

##### GetAllById($id)

Fonction qui permet de récupérer tous les commentaires d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| id | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  Array<FComment> avec tous les commentaires de l’itinéraire |

##### AddToItinerary($idItinerary, $comment, $idUser)

Fonction qui permet d’ajouter un commentaire à un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |
| comment | Le commentaire posté par l’utilisateur | STRING |
| idUser | L’id de l’utilisateur ayant posté le commentaire | INT |

|  |
| --- |
| Valeur retournée  True si le commentaire a bien été ajouté, False si exception |

#### FRateManager

Ce manager contient toutes les fonctions permettant de récupérer/traiter des notes

##### HasAlreadyRate($idItinerary, $idUser)

Fonction qui permet de savoir si un utilisateur a déjà attribué une note à un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |
| idUser | L’id de l’utilisateur qui a noté l’itinéraire. | INT |

|  |
| --- |
| Valeur retournée  True si l’utilisateur a déjà attribué une note, sinon False. |

##### AddToItinerary($idItinerary, $idUser, $rate)

Fonction qui permet d’ajouter une note à un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |
| rate | La note que l’utilisateur a saisit sur 10. | INT |
| idUser | L’id de l’utilisateur qui a noté l’itinéraire. | INT |

|  |
| --- |
| Valeur retournée  True si la note a été ajouté, false si une exception est intervenue. |

##### GetAvgByItinerary($idItinerary)

Fonction qui permet de récupérer la note moyenne d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  La note moyenne de l’itinéraire, False si une exception est intervenue. |

##### GetNbByItinerary($idItinerary)

Fonction qui permet de récupérer le nombre de notes attribués à un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  Le nombre de notes attribués à l’itinéraire, False si une exception est intervenue. |

#### FPhotosManager

Ce manager contient toutes les fonctions permettant de récupérer/traiter des photos

##### Create($idItinerary, $photos)

Fonction qui permet de créer une ou plusieurs photos chargées sur l’itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |
| photos | Tableau de *FPhoto* contenant les photos encodé en base 64 | ARRAY<FPhoto> |

|  |
| --- |
| Valeur retournée  True si les photos ont été ajouté, false si une exception est intervenue. |

##### GetAllById($idItinerary)

Fonction qui permet de récupérer toutes les photos d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  Tableau de *FPhoto* contenant toutes les photos de l’itinéraire, False si une exception est intervenue. |

#### FWaypointsManager

Ce manager contient toutes les fonctions permettant de récupérer/traiter les points des itinéraires.

##### GetAllById($idItinerary)

Fonction qui permet de récupérer tous les points d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  Un tableau de *FWaypoint* contenant tous les points de l’itinéraire, False si une exception est intervenue. |

##### Create($waypoints[], $idItinerary)

Fonction qui permet de créer tous les points d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| waypoints | Tableau de *FWaypoint* contenant tous les points de l’itinéraire à créer | ARRAY<FWAYPOINT> |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  True si les points ont été créé, sinon False |

##### Delete($idItinerary)

Fonction qui permet de supprimer les points d’un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| idItinerary | L’id de l’itinéraire | INT |

|  |
| --- |
| Valeur retournée  True si les points ont été supprimé, sinon False |

### Vues (PHP)

###### FItineraryView

Cette classe contient toutes les fonctions qui permettent d’afficher des itinéraires ou des éléments en liens avec.

##### DisplayItineraries(array $itineraries)

Fonction qui permet d’afficher plusieurs itinéraires.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| itineraries | Tableau de tous les itinéraires stockés en base de données | ARRAY<FItinerary> |

|  |
| --- |
| Valeur retournée  Le code html pour afficher tous les itinéraires. |

##### DisplayItinerary($itinerary)

Fonction qui permet d’afficher un itinéraire.

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| itinerary | Objet FItinerary qui contient tous les détails de l’itinéraire | FItinerary |

|  |
| --- |
| Valeur retournée  Le code html pour afficher l’itinéraire. |

##### DisplayPhotosItineraries($photos)

Fonction qui permet d’afficher les photos d’un itinéraire

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| photos | Tableau contenant toutes les photos à afficher | ARRAY |

|  |
| --- |
| Valeur retournée  Le code html pour afficher les photos. |

##### DisplayCommentsItineraries($comments)

Fonction qui permet d’afficher les commentaires d’un itinéraire

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| comments | Tableau contenant tous les commentaires à afficher | ARRAY |

|  |
| --- |
| Valeur retournée  Le code html pour afficher les commentaires. |

###### FAdminView

Cette classe contient toutes les fonctions qui permettent d’afficher des composants disponibles en mode admin.

##### DisplayUsers($users)

Fonction qui permet d’afficher tous les utilisateurs enregistrés sur l’application

|  |  |  |
| --- | --- | --- |
| Paramètre(s) | Description | Type |
| users | Tableau contenant tous les utilisateurs à afficher | ARRAY |

|  |
| --- |
| Valeur retournée  Le code html pour afficher les utilisateurs. |

## Librairies et outils externes

### C:\Users\Thomas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\295B3360.tmpMapquest

Mapquest est un service gratuit qui permet d’utiliser des maps à l’aide d’API et de SDK mit à disposition. Ce service a été initialement lancé en 1996, en tant que premier service de « web mapping » commerciale. Il a aujourd’hui de gros concurrents comme par exemple *Google Maps*.

Pour mon application j’ai utilisé le SDK *Mapquest.js* qui m’a permis d’utiliser les maps et d’y afficher les itinéraires créés par les utilisateurs.

### Git – WikipediaGit

Git est le logiciel de gestion de versions qui a été utilisé durant tout le projet afin de garder un historique des différentes modifications effectuées et de synchroniser les dernières versions du code source du projet.

Un dépôt distant contenant tout le projet est disponible sur GitHub :

<https://github.com/thomasFJS/ProjectTPI>

### Free Bootstrap Themes, Templates, Snippets, and Guides - Start ...StartBootstrap

StartBootstrap est un site créant des thèmes/templates de site web avec bootstrap qu’ils laissent en « free to use » pour tout le monde. Tous leurs exemples sont sous licence MIT, qui donne le droit de réutiliser l’exemple, de le modifier ou encore même de le publier (voir [Licence MIT](https://github.com/BlackrockDigital/startbootstrap-freelancer/blob/master/LICENSE)). Le site a été fondé par David Miller, un développeur web du centre de la Floride. Il a lancé le site le 19 aout 2013 depuis plus de 4'000'000 d’utilisateurs utilisent ses exemples pour leurs sites web. J’ai utilisé une de ces templates pour le projet Travler, la template « Freelancer » qui m’a permis de ne pas perdre énormément de temps sur le design du site web et de me concentrer surtout sur les fonctionnalités principales de l’application

### SwiftMailer

SwiftMailer est une librairie qui permet d’envoyer des mails dans une application en PHP. Elle est sous licence MIT (voir [Licence MIT](https://github.com/BlackrockDigital/startbootstrap-freelancer/blob/master/LICENSE)) et peut donc être utilisé par tout le monde dans n’importe quel but. J’ai utilisé la version 5 de SwiftMailer pour créer le système de notification de l’application.

## Tests

### Unitaire

Les différents tests unitaires qui ont été fait sont dans le dossier **/test/unit/ :**

* **FUserManagerTest.php :** Fichier testant unitairement toutes les fonctions du manager *FUserManager.*
* **FItineraryManagerTest.php :** Fichier testant unitairement les fonctions du manager *FItineraryManager.*

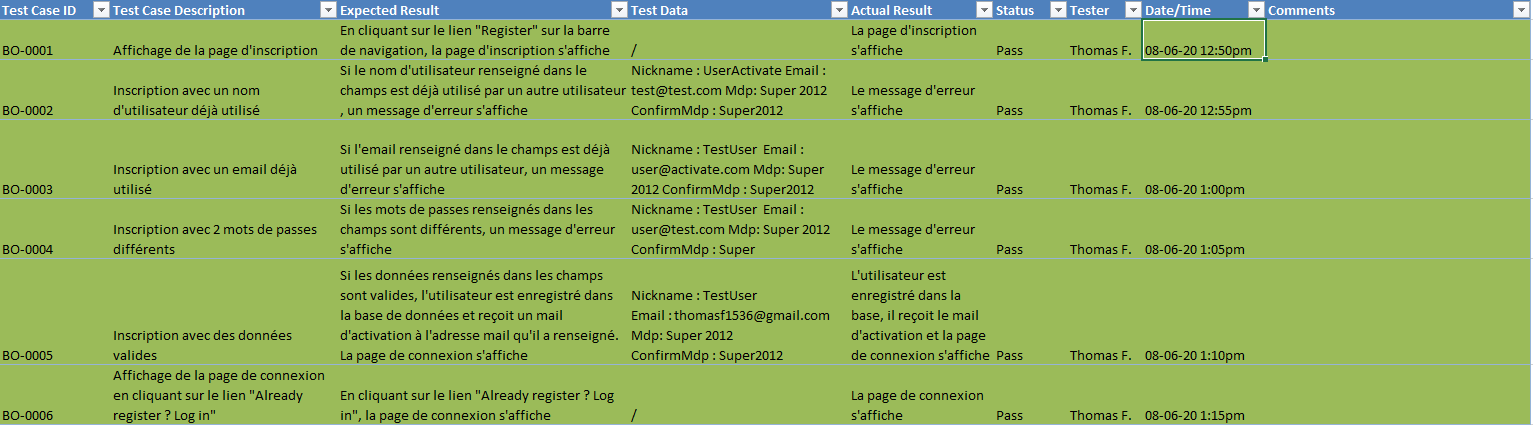
### Utilisateur

Les différents tests utilisateurs qui ont été effectuer pour le bon fonctionnement de l’application :

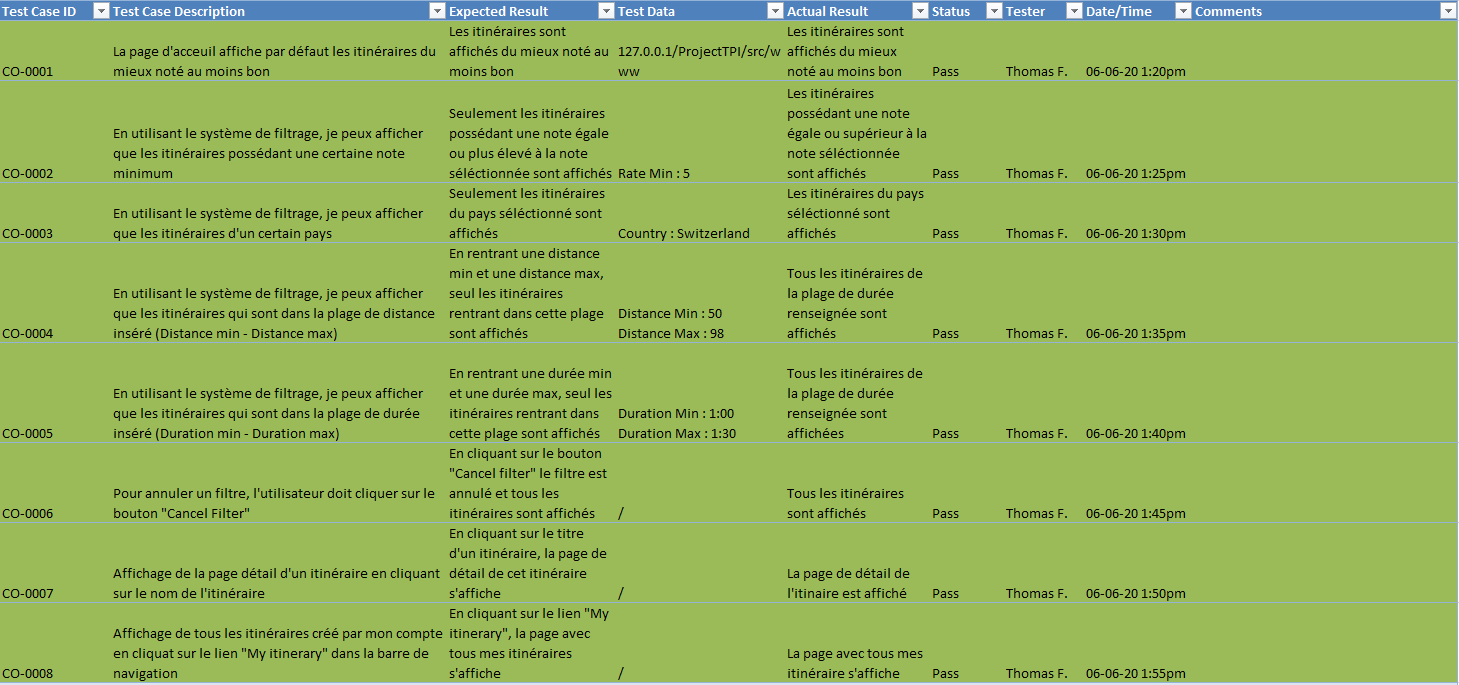
###### Connexion et déconnexion

Ces tests permettent de vérifier tous ce qui est en lien avec la connexion et la déconnexion de l’application web.

###### Inscription

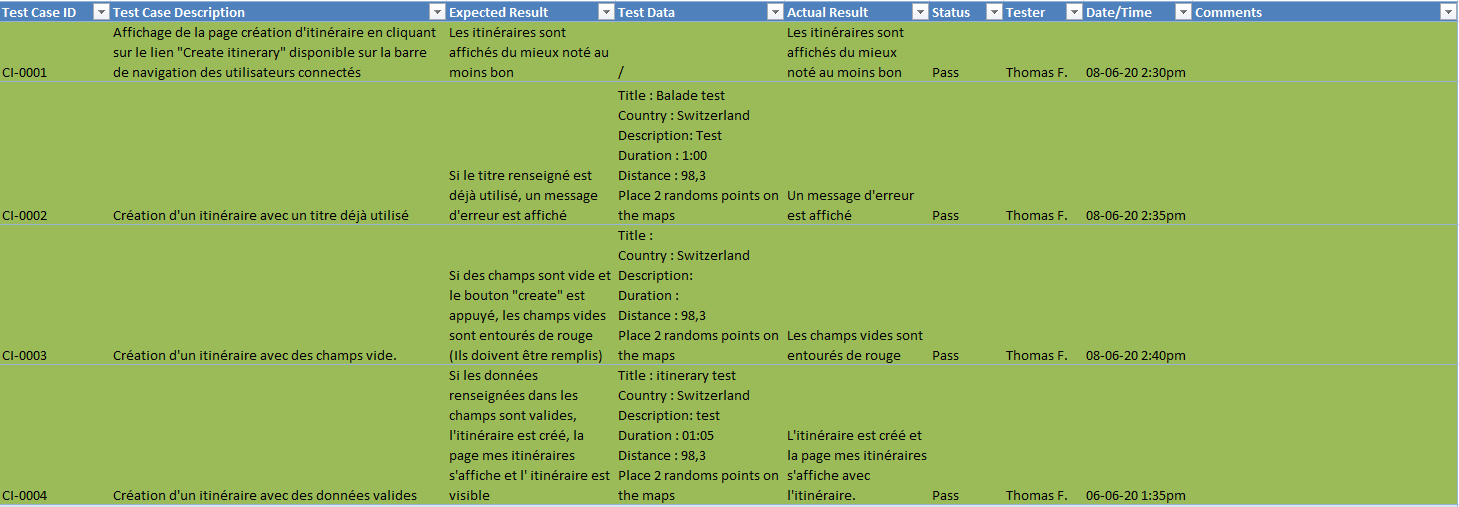
Ces tests permettent de vérifier tous ce qui est en lien avec l’inscription à l’application web.

###### Affichage

Ces tests permettent de vérifier tous ce qui est en lien avec l’affichage de l’application web.

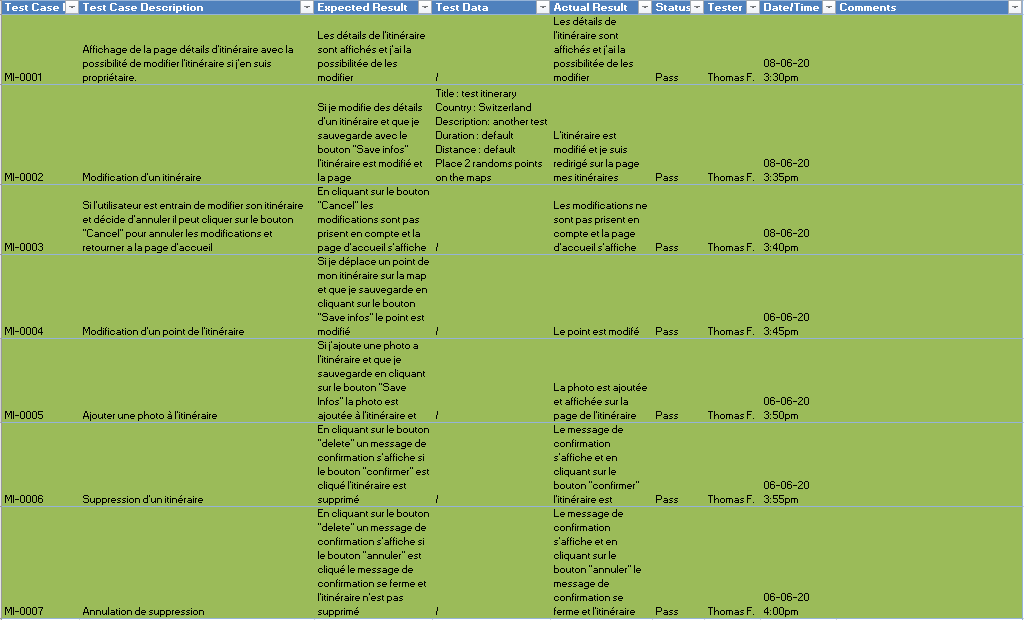
###### Création d’itinéraire

Ces tests permettent de vérifier tous ce qui est en lien avec la création d’itinéraire.



###### Modification et Suppression d’itinéraire

Ces tests permettent de vérifier tous ce qui est en lien avec la modification et la suppression d’itinéraire



###### Notes et commentaires

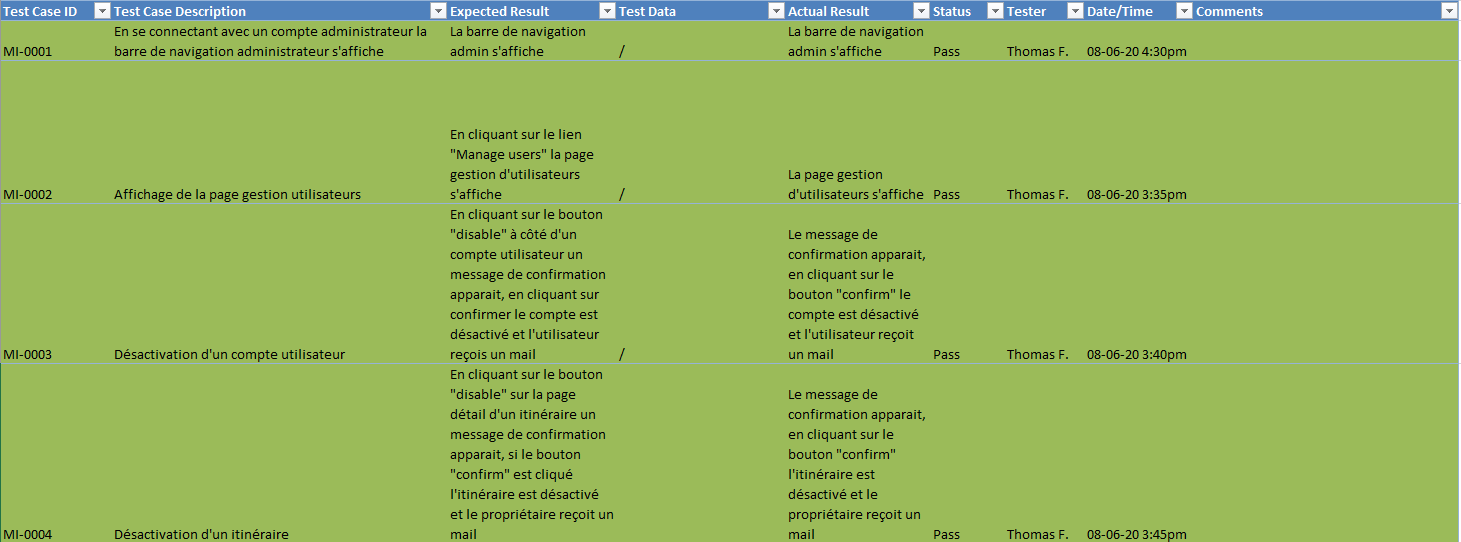
Ces tests permettent de vérifier tous ce qui est en lien avec une notes ou un commentaire.

###### Profil utilisateur

Ces tests permettent de vérifier tous ce qui est en lien avec un profil utilisateur.

## C:\Users\Thomas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D3BE9E39.tmp

###### Administration

Ces tests permettent de vérifier tous ce qui est en lien avec l’administration de l’application

## Conclusion

### Difficultés rencontrées

Durant l’élaboration du projet, je n’ai pas rencontré beaucoup d’imprévus, il y eu un seul problème réellement bloquant qui m’a retardé au cours de mon projet. Les imprévus principaux auxquels j’ai dû faire face ont été :

* L’affichage des itinéraires impossible à cause de l’api qui n’est pas capable d’afficher plusieurs itinéraires sur la même page.
  + Je n’ai malheureusement pas pu afficher tous les itinéraires sur la page d’accueil car cela n’était réellement pas possible avec l’api mapquest. Premièrement, j’ai voulu essayer d’utiliser une méthode permettant de convertir un élément html en image svg dans un canvas et de récupérer l’image encodé en base 64 pour afficher les plusieurs itinéraires. Cette méthode n’a pas marché car les maps étaient composé de plusieurs vingtaine d’éléments html, la conversion était donc trop longue. J’ai ensuite voulu essayer d’afficher les différents itinéraires dans des frames différentes pour qu’elles soient séparés les unes des autres. Malheureusement, cette méthode n’a pas fonctionné non plus. J’ai donc partiellement solutionné le problème en n’affichant que les marqueurs sur les différents points de l’itinéraire. Les itinéraires ne sont donc pas tracés sur les pages affichants plusieurs itinéraires mais les points des itinéraires sont marqués.
* Le manque de documentation disponible pour l’api mapquest, il y avait un gros manque de documentation sur leurs forums, presque aucun topic avec des questions d’utilisateurs. Ce manque ne m’a pas beaucoup aidé lors du gros problème rencontré (voir ci-dessus).

### Améliorations possibles

Au cours de mon projet, j’ai plusieurs fois remarqué des points que j’aurai pu améliorer mais que j’ai dû laisser tel quel par manque de temps et qu’ils n’étaient pas spécialement demandé dans l’énoncé. Les différents points qui auraient pu être amélioré dans l’application :

* Ajouter la possibilité, en tant qu’administrateur, de réactiver les comptes bloqués.
* Eventuellement changer d’api et utiliser la Google Maps API, qui est beaucoup plus documenté et facile d’utilisation. L’affichage des plusieurs itinéraires sur la page d’accueil serait possible.
* Utiliser l’api pour définir le pays de départ de l’itinéraire ainsi que la durée et la distance car actuellement l’utilisateur doit saisir ces informations manuellement et peut donc donner des informations erronées.

### Bilan personnel

Globalement, j’ai apprécié travailler sur Travler car ce projet a été mon premier gros projet concret à réaliser. J’ai regretté d’avoir choisit l’api Mapquest pour mon TPI, si je devais recommencer je choisirais clairement l’API Google Maps. Hormis cela, je suis vraiment fier du projet, malgré le gros problème rencontré pour l’affichage j’ai réussi à rendre le projet fonctionnel et à temps avec une documentation technique relativement complète.