03/05/2021

BRIAN.GL

CFPT

Flight Club

Travail pratique individuel

Table des matières

[2 Table des versions 2](#_Toc71018725)

[3 Introduction 2](#_Toc71018726)

[4 Rappel de l’énoncé 2](#_Toc71018727)

[4.1 Organisation 2](#_Toc71018728)

[4.2 Livrables 3](#_Toc71018729)

[4.3 Matériel et logiciels 3](#_Toc71018730)

[4.3.1 Matériel et logiciels à disposition 3](#_Toc71018731)

[4.3.2 Matériel utilisé 3](#_Toc71018732)

[4.4 Description de l’application 4](#_Toc71018733)

[5 Méthodologie 4](#_Toc71018734)

[5.1 S’informer 4](#_Toc71018735)

[5.2 Planifier 4](#_Toc71018736)

[5.3 Décider 5](#_Toc71018737)

[5.4 Réaliser 5](#_Toc71018738)

[5.5 Contrôler 5](#_Toc71018739)

[5.6 Évaluer 5](#_Toc71018740)

[6 Planification 5](#_Toc71018741)

[6.1 Product backlog 5](#_Toc71018742)

[7 Analyse fonctionnelle 8](#_Toc71018743)

[7.1 Navigation du site 8](#_Toc71018744)

[7.2 Design général 10](#_Toc71018745)

[8 Fonctionnalités 10](#_Toc71018746)

[8.1 Fonctionnalités du site 10](#_Toc71018747)

[8.1.1 Utilisateurs anonymes 10](#_Toc71018748)

[8.1.2 Utilisateurs connectés 11](#_Toc71018749)

[8.2 Fonctionnalités majeures 11](#_Toc71018750)

[8.2.1 Inscription 11](#_Toc71018751)

[8.2.2 Modification du profil 12](#_Toc71018752)

[8.2.3 Déconnexion 13](#_Toc71018753)

[8.2.4 Connexion 13](#_Toc71018754)

[9 Généralités concernant l’implémentation 14](#_Toc71018755)

[9.1 Base de données 14](#_Toc71018756)

[9.1.1 Le modèle de données final 14](#_Toc71018757)

[9.1.2 Dictionnaire de données 14](#_Toc71018758)

[10 Librairies et outils externes 16](#_Toc71018759)

[10.1 Git 16](#_Toc71018760)

[10.2 Bootstrap 16](#_Toc71018761)

[10.3 Bootstrap Studio 16](#_Toc71018762)

[10.4 Flaticon 16](#_Toc71018763)

[10.5 Expression régulière 16](#_Toc71018764)

[10.6 Katalon Recorder 17](#_Toc71018765)

[10.7 Trello 17](#_Toc71018766)

[11 Plan de tests et tests 17](#_Toc71018767)

[11.1 Périmètre des tests 17](#_Toc71018768)

[11.2 Équipement de test 17](#_Toc71018769)

[11.2.1 Environnement 17](#_Toc71018770)

[11.3 Scénarios de test 17](#_Toc71018771)

[12 Glossaire 24](#_Toc71018772)

[12.1 Termes techniques 24](#_Toc71018773)

[12.1.1 User stories (ou récit utilisateur) 24](#_Toc71018774)

[12.1.2 Importation (Anglais : upload) 24](#_Toc71018775)

[12.1.3 Carnet de produit (Anglais: product backlog) 24](#_Toc71018776)

[12.1.4 Formulaire GET 24](#_Toc71018777)

# Table des versions

|  |  |  |
| --- | --- | --- |
| N° de version | Auteur | Changements apportés |
| 1.0 | Brian GOLAY  <brian.gl@eduge.ch> | Version finale du document pour le rendu du tpi |

# Introduction

Ce document est un rapport présentant différents aspects de la conception du projet Flight Club. Ce projet a été réalisé dans le cadre du Travail pratique individuel (TPI) durant la session de mai 2021. Il a pour but de valider mes compétences acquises pendant la formation Informaticien CFC dispensée à l’école d’informatique du CFPT au Petit-Lancy. Flight Club a pour but de permettre aux pilotes réels et virtuels d’enregistrer leurs vols, et de les partager.

# Rappel de l’énoncé

*Les informations suivantes sont extraites du cahier de charges du TPI.*

## Organisation

|  |
| --- |
| Élève |
| Brian GOLAY  <brian.gl@eduge.ch> |

|  |
| --- |
| Maître d’apprentissage |
| Mathieu Michaël  <edu-mathieum@eduge.ch> |

|  |  |
| --- | --- |
| Experts | |
| Borys Folomietow  <borys@folomietow.ch> | Jacques Ortola  <jacques.ortola@gmail.com> |

## Livrables

* Planning
* Rapport au format PDF avec le code source en annexe
* FAQ présente sur le site avec des références dans le site sur des ancres de la FAQ
* Résumé du rapport du TPI
* Journal de travail – carnet de bord
* Script pour la création de la base de données
* Script pour l’initialisation des données (compte administrateur, permissions)

## Matériel et logiciels

### Matériel et logiciels à disposition

* 1 ordinateur avec Windows 10 et 2 écrans
* Suite Microsoft Office
* EasyPHP ou WAMP ou un serveur équivalent
* PHP
* MySql
* Bibliothèque pour un affichage responsive (par exemple bootstrap)
* Bibliothèque pour la génération de PDF en PHP (par exemple TCPDF)
* Bibliothèques utilitaires (par exemple jQuery)
* Repo Git (par exemple github ou gitlab)
* Outils pour des tests automatiques comme Katalon
* MySQL Workbench

### Matériel utilisé

* 1 ordinateur avec Windows 10 et 2 écrans
* Suite Microsoft Office
* EasyPHP
* MySQL Workbench
* Visual Studio code
* Git (Github)
* Google chrome
* Bootstrap
* TCPDF

## Description de l’application

L’application sera développée à l’aide des langages web suivants : PHP, HTML et CSS. Les fonctionnalités suivantes doivent être implémentées :

* Création d’un utilisateur
* Pour un utilisateur connecté
  + Enregistrer ses vols
  + Modifier ses vols enregistrés
  + Changer son mot de passe
  + Créer un groupe
  + Rejoindre un groupe
  + Télécharger ses vols au format .csv et .pdf

# Méthodologie

Pour planifier mon projet de TPI, je me suis basé sur la **méthode en 6 étapes** qui nous a été enseignée au CFPT.



## S’informer

La toute première étape de mon projet était la lecture en profondeur de mon énoncé pour comprendre toutes les fonctionnalités qu’il était nécessaire d’implémenter. J’ai également demandé à mon formateur des clarifications sur des détails des spécifications lorsque c’était nécessaire.

## Planifier

Dès le début du projet, j’ai préparé un planning de travail pour savoir ce que j’avais besoin de faire et quand. Afin de pouvoir créer ce planning, j’ai eu besoin de découper mon travail en sous-tâches : j’ai donc divisé les divers points de l’énoncé sous forme d’user stories, qui présentent la tâche du point de vue de l’utilisateur final.

Pour chaque story, j’ai fixé une priorité pour m’aider ensuite à savoir ce qui doit être réalisé au plus vite (mes niveaux de priorité sont ! P0 : Bloquant, ‼ P1 : Critique, !!! P2 : Important et ? P3 : Secondaire).

Une fois les stories créées, je les ai stockées dans un product backlog. Ce fonctionnement est inspiré par la méthodologie Scrum (cependant, je n’ai pas appliqué entièrement cette méthodologie, car elle perd son utilité lorsqu’on travaille seul sur un projet).

J’ai ensuite créé un diagramme de Gantt, outil qui m’a permis de visualiser au quotidien ma progression dans le travail ainsi que les différences entre ma planification et mon avancement effectif.

## Décider

Au cours de l’avancement de mon travail, j’ai dû prendre de nombreuses décisions sur la manière de le réaliser. Lorsque je faisais des choix que je jugeais importants pour le projet, j’en ai parlé dans mon journal de bord en expliquant les raisons qui m’ont poussé à choisir ces options d’implémentation.

## Réaliser

Une fois les bonnes décisions prises, j’ai pu continuer sur l’implémentation de la fonctionnalité dans le code (ou la rédaction pour tout ce qui concerne la documentation).

## Contrôler

À chaque fois que je terminais une fonctionnalité, je l’ai testée dans différents cas d’usage pour être sûr qu’elle fonctionne comme prévu. J’ai documenté mes tests dans mon rapport TPI pour m’assurer qu’ils soient reproductibles par une personne externe au besoin.

J’ai aussi effectué à nouveau tous mes tests à la fin de chaque journée de travail pour éviter les régressions. J’ai ajouté à mon rapport TPI un tableau qui permet de visualiser l’avancement de l’état des tests au cours du projet.

Enfin, j’ai testé de fond en comble le programme à la fin de son implémentation sur d’autres plateformes (navigateurs et systèmes d’exploitation) pour m’assurer qu’il puisse bien fonctionner chez tout le monde.

## Évaluer

La dernière étape de la méthodologie en 6 étapes est l’évaluation, pour faire une rétrospective de ce que j’ai fait et avoir un regard critique sur ce qui pourrait être amélioré. Pour ce faire, j’ai écrit des bilans journaliers à chaque fin de journée de travail dans mon journal de bord. Pour avoir une auto-évaluation du travail en entier, j’ai également rédigé la partie Conclusion du rapport TPI.

# Planification

## Product backlog

|  |  |
| --- | --- |
| Nom | S1 : Utilisation d’un dépôt Git |
| Description (user story) | En tant que développeur, je peux utiliser Git sur le code source du projet. Je peux également accéder à un dépôt distant hébergé sur un site comme GitHub. |
| Critère d’acceptation | Un dépôt Git est configuré dans le dossier contenant le code source. Le code source du projet est accessible depuis le web à l’adresse indiquée dans la documentation. |
| Priorité | ! P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S2 : Configuration de la base de données |
| Description (user story) | En tant que développeur, je peux utiliser une base de données MySQL respectant le modèle proposé par l’énoncé du TPI. Pour ce faire, j’ai accès à une classe côté PHP qui me permet d’effectuer les opérations de base (CRUD: create, read, update, delete) sur la base de données. Je souhaite également avoir dans le dépôt un dump de la structure de base de données qui me permettrait de la recréer sur un autre environnement. |
| Critère d’acceptation | Les tables Tbl\_Flight, Tbl\_Picture, Tbl\_User-Flight, Tbl\_User, Tbl\_User-Group et Tbl\_Group ont été créées et sont utilisables au moyen d’une classe Database |
| Priorité | ! P0 : Critique |

|  |  |
| --- | --- |
| Nom | S3 : Affichage de la *home* page |
| Description (user story) | En tant qu’utilisateur non connecté, je ne peux pas encore accéder à mon profil : le site m’affiche une page présentant brièvement le projet et qui contient un bouton « Se connecter ». |
| Critère d’acceptation | Les tests concernant la fonction se connecter se passe sans problème. |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S4 : Enregistrements des vols |
| Description (user story) | En tant qu’utilisateur, je peux enregistrer mes vols |
| Critère d’acceptation | Les tests concernant l’enregistrement des vol pour un utilisateur connectés passent |
| Priorité | ! P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S5 : Téléchargement du carnet de vol |
| Description (user story) | En tant qu’utilisateur, je peux télécharger au format CSV et PDF mes heures de vols |
| Critère d’acceptation | L’utilisateur télécharge son carnet de vol dans le format demandé |
| Priorité | !!! P2 : Important |

|  |  |
| --- | --- |
| Nom | S6 : Création de compte |
| Description (user story) | En tant qu’utilisateur non connecté, je veux pouvoir m’enregistrer pour accéder aux fonctionnalités des utilisateurs connectés |
| Critère d’acceptation | L’utilisateur peut se créer un compte |
| Priorité | ! P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S7 : Connexion à son compte |
| Description (user story) | En tant qu’utilisateur non connecté, je veux pouvoir me connecter à mon compte pour accéder aux fonctionnalités des utilisateurs connectés |
| Critère d’acceptation | L’utilisateur se connecte avec succès |
| Priorité | ! P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S8 : Modification des vols |
| Description (user story) | En tant qu’utilisateur connecté, je veux modifier mes vols |
| Critère d’acceptation | Modification du vol sélectionné avec succès dans la base de donnée |
| Priorité | !!! P2 : Important |

|  |  |
| --- | --- |
| Nom | S9 : Modification de mot de passe |
| Description (user story) | En tant qu’utilisateur connecté, je veux modifier mon mot de passe |
| Critère d’acceptation | Le mot de passe est modifier avec succès dans la base de donnée |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S10 : Créer un groupe |
| Description (user story) | En tant qu’utilisateur connecté, je veux créer un groupe |
| Critère d’acceptation | Un groupe est ajouté à la base de donnée avec l’utilisateur qui la créer |
| Priorité | ! P0 Bloquant |

|  |  |
| --- | --- |
| Nom | S11 : Inviter des utilisateurs à mon groupe |
| Description (user story) | En tant qu’utilisateur connecté, je veux inviter un utilisateur dans mon groupe |
| Critère d’acceptation | Une invitation est créer dans la base de donnée |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S12 : Accepter une invitation à un groupe |
| Description (user story) | En tant qu’utilisateur connecté, je veux accepter une invitation pour rejoindre un groupe |
| Critère d’acceptation | Une invitation est modifiée dans la base de donnée |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S13 : Refuser une invitation à un groupe |
| Description (user story) | En tant qu’utilisateur connecté, je veux refuser une invitation pour rejoindre un groupe |
| Critère d’acceptation | L’invitation est supprimée dans la base de donnée |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S14 : Affichage des vols dans les groupes |
| Description (user story) | En tant qu’utilisateur connecté, je veux voir les vols des membres de mon groupe |
| Critère d’acceptation | Les vols des membres s’affichent |
| Priorité | !! P1 : Critique |

|  |  |
| --- | --- |
| Nom | S15 : Quitter un groupe |
| Description (user story) | En tant qu’utilisateur connecté, je veux quitter un groupe |
| Critère d’acceptation | L’utilisateur n’est plus dans le groupe dans la base de donnée et n’as plus accès aux informations du groupe |
| Priorité | ! P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S16 : Accès à la FAQ |
| Description (user story) | En tant qu’utilisateur non connecté, j’ai la permission, comme tout le monde, de voir la page FAQ. |
| Critère d’acceptation | L’utilisateur n’est plus dans le groupe dans la base de donnée et n’as plus accès aux informations du groupe |
| Priorité | ! P0 : Bloquant |

! P0 : Bloquant, ‼ P1 : Critique, !!! P2 : Important et ? P3 : Secondaire

# Analyse fonctionnelle

## Navigation du site

Le site Flight Club contient XX pages. Elles sont divisées en fonction des permissions de l’utilisateur, en plus de l’accueil et de la page FAQ. Il y a la partie utilisateur, la partie concernant les groupes et pièces jointes.

## Design général



Fig. Structure des pages

Pour le design du site, j’ai choisi d’utiliser Bootstrap, car il s’agit d’un framework simple d’utilisation qui permet de créer une interface esthétique et compatible à tous les navigateurs. Toutes les pages du site ont la même structure pour faciliter la lecture et ne pas se perdre. Elle comprend un titre, une barre de navigation et son contenu. Le design du site est réalisé avec l’outil Bootstrap Studio (<https://bootstrapstudio.io/>)

# Fonctionnalités

## Fonctionnalités du site

La fonctionnalité principale de mon site est de sauver ses vols dans un carnet disponible partout dans le monde et de pouvoir télécharger ses heures de vols en .pdf ou en .csv facilement.

### Utilisateurs anonymes

C’est le niveau de permission le plus bas, ils représentent tous les utilisateurs du site qui ne sont pas connectés. Leurs fonctionnalités sont décrites dans le cahier des charges, et sont les suivantes :

* Accès à la page home
* Accès à la page FAQ
* Créer un compte
* Se connecter

### Utilisateurs connectés

Ce sont des utilisateurs lambda qui ont juste créé un compte sur le site. Ils sont définis par une adresse mail. Les fonctionnalités sont les suivantes :

* Modifier leur profil
* Enregistrer un vol
* Créer un groupe
* Inviter des autres utilisateurs dans leurs groupes
* Voire les heures de vols des groupes
* Quitter leurs groupes
* Se déconnecter
* Accéder à la page FAQ

## Fonctionnalités majeures

### Inscription

Cela permet d’avoir accès aux fonctionnalités avancées du site, comme les groupes, ou l’ajout de nouveaux vols. Les utilisateurs doivent entrer leur email entrer un mot de passe pour pouvoir se connecter. Lorsque l’utilisateur appuiera sur le bouton « S’enregistrer », la partie serveur va vérifier le contenu des champs, et retourné les erreurs commises à l’utilisateur. Dans le cas où aucune erreur n’est retenue, l’utilisateur est ajouté dans la base de données. La partie du code pour le mot de passe est une partie que j’ai inventé. Elle se passe comme cela: je commence par le « *hasher* » en sha512, ensuite on ajoute un salt qui est fait en prenant quelques parties du hash et en les mettant au début et à la fin. Finalement, on le hash encore une fois en sha512.

//if all data are not empty

if (!empty($email) && !empty($password) && !empty($password2)) {

    //we check that the email is not already in our database

    if (userDAO::readEmail($email) == null) {

        //we check booth of the password are the same and the strength of the password

        if ($password === $password2) {

            if (password\_strength($password)) {

                //we hash the password

                $hashed = hash('sha512', $password);

                $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

                $hashed = hash('sha512', $salted);

                //we add a php generated salt

                $salt = password\_hash($hashed, PASSWORD\_BCRYPT);

                //we add it to the database

                userDAO::AddUsers($email, $hashed, $salt);

                $\_SESSION['user'] = userDAO::readUsersByEmail($email);

                //we redirect him to the login page

                header("Location: ./index.php?page=login");

                exit();

            } else {

                $message = "Le mot de passe doit contenir au moins une majuscule, une minuscule, un chiffre et un caractère spécial et faire 8 caractère ou plus!";

            }

        } else {

            $message = "Les mots de passes ne sont pas identiques !";

        }

    } else {

        $message = "L'email existe déjà";

    }

}

### Modification du profil

Cette fonctionnalité est accessible uniquement pour soi-même, seul l’utilisateur peut modifier son compte. L’informations pouvant être modifiée est son mot de passe. Si une erreur survient, l’utilisateur est averti avec un message d’erreur.

//we check that the email and password are filled

if (!empty($newPassword) && !empty($repeatNewPassword) && !empty($oldPassword)) {

    //we hash the user password

    $hashed = hash('sha512', $oldPassword);

    $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

    $hashed = hash('sha512', $salted);

    //we compare the hashed password of the user with the one in the database

    if (password\_verify($hashed, userDAO::getUserByEmail(UserDAO::getUserByID($\_SESSION['userID'])['Txt\_Email'])['Txt\_Password\_Salt'])) {

        //we check that booth of the new password are the same

        if ($newPassword == $repeatNewPassword) {

            //we hash the new password to make it work with the login

            $hashed = hash('sha512', $newPassword);

            $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

            $hashed = hash('sha512', $salted);

            //we create a salt with the new password

            $salt = password\_hash($hashed, PASSWORD\_BCRYPT);

            //we update the password in the database

            UserDAO::changePassword($\_SESSION['userID'], $hashed, $salt);

            //we display a success message to the user

            $messageSucces = "Mot de passe changé avec succès";

        }else {

                //we display an error message to the user

        $messageErreur = "Les 2 mot de passe saisis ne sont pas identiques!";

        }

    } else {

        //we display an error message to the user

        $messageErreur = "Mauvais mot de passe!";

    }

}

### Déconnexion

La page de déconnexion est simple. Elle supprime la session et tous les cookies utilisés par le site et redirige l’utilisateur vers la page d’accueil.

//start a session

session\_start();

$\_SESSION = array();

//remove all cookies and destroy the session

if (ini\_get("session.use\_cookies")) {

    setcookie(session\_name(), '', 0);

}

session\_destroy();

//redirect the user to the homepage

header("Location: ./index.php?page=homepage");

exit;

### Connexion

Pour cette fonctionnalité, le serveur va d’abord valider les entrées utilisateurs avant de vérifier si l’utilisateur possédant ces identifiants existe avant de finalement sauvegarder les données de connexion dans la session.

//we check that the email and password are filled

if (!empty($email) && !empty($password)) {

    //we hash the user password

    $hashed = hash('sha512', $password);

    $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

    $hashed = hash('sha512', $salted);

    //we compare the hashed password of the user with the one in the database

    if (password\_verify($hashed, userDAO::getUserByEmail($email)['Txt\_Password\_Salt'])) {

        //we connect the user in session

        $\_SESSION['connected'] = true;

        $\_SESSION['userID'] = userDAO::getUserByEmail($email)['Id\_User'];

        $message = "Succesfully logged";

        //we redirect the user to the homepage

        header("Location: ./index.php?page=homepage");

        exit();

    } else {

        $message = "Mauvais mot de passe ou email !";

    }

}

### Groupes

#### Inviter un utilisateur au groupe

Cette fonctionnalité est seulement disponible si on est membre d’un groupe. Elle envoie une invitation à un utilisateur avec son email. L’invitation est seulement visible sur le site web, pas d’email envoyé, pas de notifications. Nous affichons un message de succès ou d’erreur en fonction du résultat de l’invitation.

//we check that the email field is not empty

if (!empty($email)) {

    //we get all the user data and try to invite him

    $userData = UserDAO::getUserByEmail($email);

    try {

        if (!empty($userData)) {//if the user is existing we invite him and show a success message

            GroupDAO::CreateInvite($idGroup, $userData['Id\_User']);

            $successMessage = "Utilisateur invité!";

        } else {//if the user doesn't exists, we show an error message

            $errorMessage = "Oops, une erreur est survenue. Vérifiez l'adresse du destinataire";

        }

    } catch (\Throwable $th) {//if the user is already in the group or is already invited, we show an error message

        $errorMessage = "Oops, une erreur est survenue. Vérifiez l'adresse du destinataire";

    }

}

#### Créer un groupe

Cette fonctionnalité est disponible à tous les utilisateurs connectés. L’utilisateur nomme son groupe et clique sur le bouton « Créer ». Un message de succès s’affichera si le groupe a bien été créer. L’utilisateur seras automatiquement ajouté à son groupe fraichement créer.

//we check that the name is not empty

if (!empty($name)) {

    //we create the group and get his id

    GroupDAO::createGroup($name);

    $lastID =GroupDAO::getLastGroupId();

    //we make the user join his newly created group

    GroupDAO::joinCreatedGroup($lastID[0]['Id\_Group'], $\_SESSION['userID']);

    //we show him a success message

    $message = "Groupe créer avec succès!";

}

#### Liste des membres du groupe

Cette fonctionnalité est disponible à tous les membres du groupe concerné. Au menu du groupe, l’utilisateur clique sur « Liste des membres ». Un tableau les données de tous les membres avec pour chaque membre l’email et la date à laquelle il a rejoint le groupe s’affiche.

/\*\*

 \* function that show all group members

 \*

 \* @return string

 \*/

function showGroupMembers($idGroup)

{

    //we get all users of the selected group in a variable

    $groupMember = GroupDAO::getAllOfGroupUser($idGroup);

    //we write every user of the selected group

    foreach ($groupMember as $key => $value) {

        echo "<tr><td>" . $value['Txt\_Email'] . "</td><td>" . $value['Dttm\_Membership'] . "</td></tr>";

    }

}

#### Quitter un groupe

Cette fonctionnalité est disponible à tous les membres d’un groupe. En cliquant sur Mes groupes, une liste des groupes s’affiche avec chacun deux boutons. En cliquant sur la croix « X » rouge, l’utilisateur quitte le groupe et ses heures de vols disparaissent du carnet de vol du groupe.

//we check if the user has clicked on the leave button

$leave = filter\_input(INPUT\_POST, "delete", FILTER\_SANITIZE\_STRING);

//remove the current user from a group

if ($leave) {

    GroupDAO::deletePendingInvite($leave, $\_SESSION['userID']);

}

#### Accepter ou refuser une invitation

Cette fonctionnalité est disponible à tous les utilisateurs connectés. Au menu du groupe, l’utilisateur clique sur « Mes invitations ». Un tableau apparait alors en lui montrant toutes les invitations en attente ou un message lui disant qu’il n’a pas reçu d’invitations.

//check if the user clicked on any of these buttons

$accept = filter\_input(INPUT\_POST, "accept");

$deny = filter\_input(INPUT\_POST, "delete");

//if he accepted an invite, we update the database

if ($accept) {

    GroupDAO::acceptPendingInvite($accept, $\_SESSION['userID']);

}

//if he deny an invite, we remove it from the database

if ($deny) {

    GroupDAO::deletePendingInvite($deny, $\_SESSION['userID']);

}

/\*\*

 \* function that show all invites for the current user

 \*

 \* @return string

 \*/

function showInvites()

{

    //we get all invites in a variable

    $pendingInvites = GroupDAO::getAllPendingInvites();

    //we initialize a counter

    $countUserInvite = 0;

    //we check every invite  of the database

    foreach ($pendingInvites as $key => $value) {

        //if an invite is for our user, we show it to him

        if ($value['Id\_User'] == $\_SESSION['userID']) {

            //we increment our counter

            $countUserInvite++;

            echo "<tr><td>" . GroupDAO::getGroupById($value['Id\_Group'])['Nm\_Group'] . " vous a invité a rejoindre son groupe!</td><td><button class=\"btn btn-success\" type=\"submit\" value=\" " . $value['Id\_Group'] . "\" name=\"accept\">✓</button></td><td><button class=\"btn btn-danger\" type=\"submit\" value=\" " . $value['Id\_Group'] . "\" name=\"delete\">X</button></td></tr>";

        }

    }

    //if our user have 0 invites, we show him a message telling him to comme back later to check if he got a new invite

    if ($countUserInvite == 0) {

        echo "<tr><td>Aucune invitation pour le moment... </td></tr><tr><td>Revenez plus tard</td></tr>";

    }

}

## Sécurité du carnet de vol

### Carnet personnel

Le carnet personnel est sécurisé avec la session php. On ne prend que les vols de l’utilisateur actuellement connecté.

//we check if the user is already loged-in

if (!isset($\_SESSION['userID'])) {

    header("Location: ./index.php?page=homepage");

    exit();

}

//we get all flight from the current user

$userFlight = FlightDAO::getAllUserFlightByUserId($\_SESSION['userID']);

/\*\*

 \* Show all flight of the current user

 \*

 \* @param array[mixed] $flight

 \* @return void

 \*/

function showFlight($flight)

{

    //if there is at least 1 flight, we show it

    if (!empty($flight)) {

        foreach ($flight as $key => $value) {

            echo "<tr><td>" . $value['No\_Flight'] . "</td><td>" . $value['Dt\_Departure'] . "</td><td>" . $value['Dt\_Arrival'] . "</td><td>" . $value['Tm\_Departure'] . "</td><td>" . $value['Tm\_Arrival'] . "</td><td>" .

            $value['Tm\_Engine\_On'] . "</td><td>" . $value['Tm\_Engine\_Off'] . "</td><td>" . $value['Nm\_Plane'] . "</td><td>" . $value['No\_Plane'] . "</td><td>" .

            $value['Cd\_ICAO\_Departure'] . "</td><td>" . $value['Cd\_ICAO\_Arrival'] . "</td><td>" . $value['Cd\_Flight\_Type'] . "</td><td>" . $value['Cd\_Flight\_Mode'] .

             "</td><td>" . $value['Cd\_Role'] . "</td><td><a href='?page=logbookDetail&id=" . $value['Id\_Flight'] . "'>Détails</a></td><td><a href='?page=logbookEdit&id=" . $value['Id\_Flight'] . "'>Editer le vol</a></td></tr>";

        }

    }else {//else we show "/"

            echo "<tr><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td><td>/</td></tr>";

    }

}

### Carnet de groupe

Le carnet de groupe fonctionne comme le croquis ci-dessous :

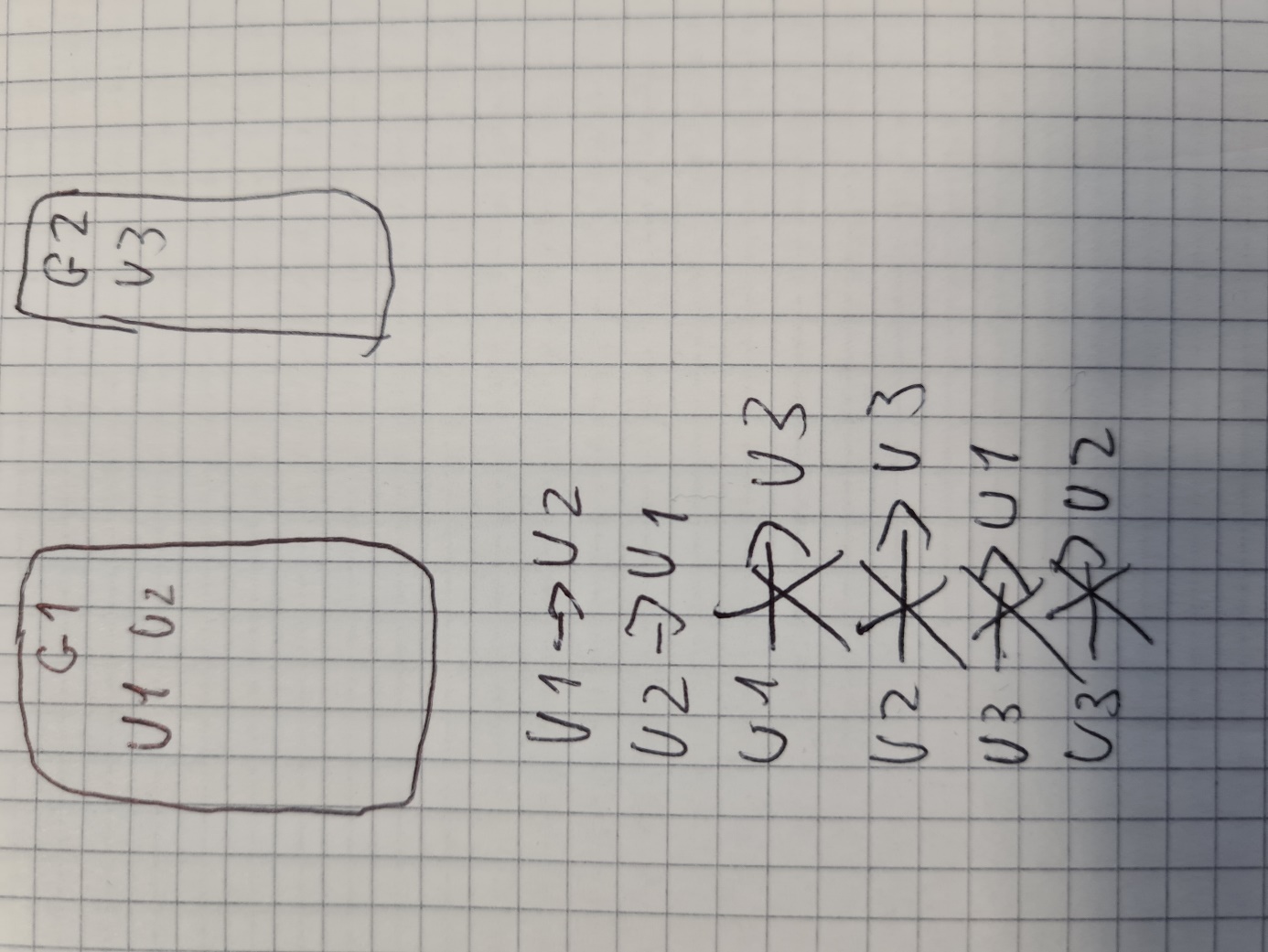


Fig. droits sur les carnets de groupes

G1 = groupe 1, G2 = Groupe 2, U1 = utilisateur1, U2 = utilisateur2 et U3 = utilisateur3

Le carnet de groupe est sécurisé via l’URL. Imaginons que l’utilisateur 1 à enregistrer un vol avec l’id 1. L’URL du vol vas être: « index.php?page=logbookDetail&id=1 ». Si l’utilisateur 2 essaie cette URL, il va avoir accès aux données du vol. Mais si l’utilisateur 3 décide de modifier l’url d’un de ses vol pour mettre l’id 1 du vol de l’utilisateur 1, le site va lui afficher un 404.

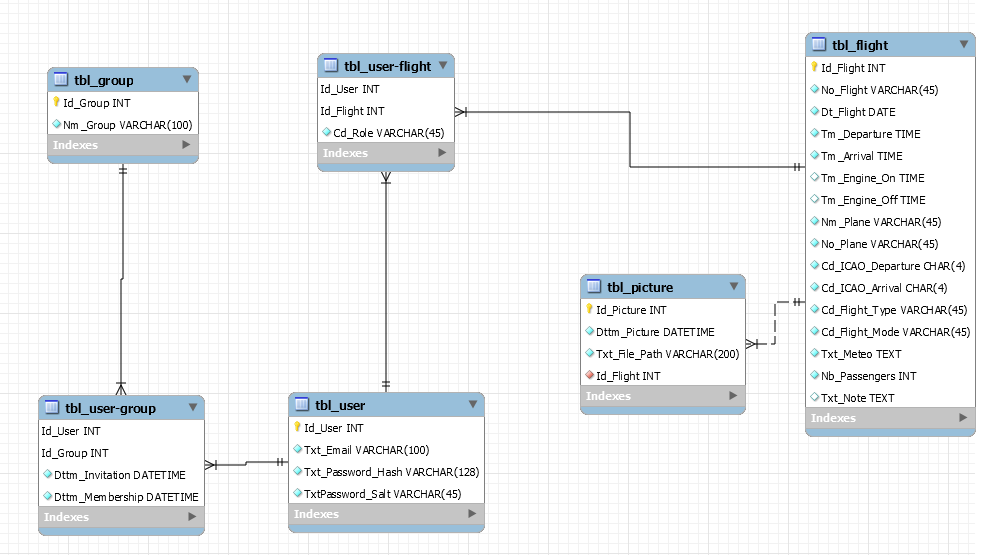
# Généralités concernant l’implémentation

## Base de données

J’utilise une base de donnée MySQL 5.7.17 pour stocker toutes les données du site. J’ai utiliser InnoDB comme moteur de stockage et utf8\_general\_ci comme interclassement.

Le modèle de données à utiliser était fourni dans l’énoncé (Il a légèrement été modifier comme discuté en classe avec M.Mathieu et expliqué par e-mail).

### Le modèle de données final



J’ai dû créer un script SQL permettant de créer les tables nécessaires, lors du premier jour du TPI. Ce dernier se trouve dans le répertoire suivant : /sql/ il se nomme structure.sql

### Dictionnaire de données

#### Table Tbl\_Flight

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_Flight | Int | Non |  | Clef primaire |
| No\_Flight | Varchar(45) | Non |  |  |
| Dt\_Flight | Date | Non |  |  |
| Tm\_Departure | Time | Non |  |  |
| Tm\_Arrival | Time | Non |  |  |
| Tm\_Engine\_On | Time | Oui |  |  |
| Tm\_Engine\_Off | Time | Oui |  |  |
| Nm\_Plane | Varchar(45) | Non |  |  |
| No\_Plane | Varchar(45) | Non |  |  |
| Cd\_ICAO\_Departure | Char(4) | Non |  |  |
| Cd\_ICAO\_Arrival | Char(4) | Non |  |  |
| Cd\_Flight\_Type | Varchar(45) | Non |  |  |
| Cd\_Flight\_Mode | Varchar(45) | Non |  |  |
| Txt\_Meteo | Text(512) | Non |  |  |
| Nb\_Passengers | Int | Non |  |  |
| Txt\_Note | Text(1024) | Oui |  |  |

#### Table Tbl\_Picture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_Picture | Int | Non |  | Clef primaire |
| Dttm\_Picture | DateTime | Non |  |  |
| Txt\_File\_Path | Text(200) | Non |  |  |
| Id\_Flight | Int | Non |  | Clef étrangère |

#### Table Tbl\_User-Flight

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_User | Int | Non |  | Clef étrangère |
| Id\_Flight | Int | Non |  | Clef étrangère |
| Cd\_Role | Varchar(45) | Non |  |  |

#### Table Tbl\_User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_User | Int | Non |  | Clef primaire |
| Txt\_Email | Varchar(100) | Non |  |  |
| Txt\_Password\_Hash | Varchar(128) | Non |  |  |
| Txt\_Password\_Hash | Varchar(60) | Non |  |  |

#### Table Tbl\_User-Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_User | Int | Non |  | Clef étrangère |
| Id\_Group | Int | Non |  | Clef étrangère |
| Dttm\_Invitation | DateTime | Oui |  |  |
| Dttm\_Membership | DateTime | Oui |  |  |

#### Table Tbl\_Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaire |
| Id\_Group | Int | Non |  | Clef primaire |
| Nm\_Group | Varchar(100) | Non |  |  |

# Librairies et outils externes

## Git



Le logiciel de gestion de versions Git a été utilisé tout au long de la création du projet afin de garder un historique des modifications effectuées, de synchroniser en permanence les dernières versions du code source et d’avoir une sauvegarde externe en cas de défaillance technique ou de corruptions de données. Un dépôt distant contenant le code source du projet est disponible sur GitHub (droits d’accès nécessaires) : <https://github.com/BRIANGL/Flight-Club>

## Bootstrap



Bootstrap est un framework open-source que j’ai utilisé durant mon projet afin d’obtenir un design agréable, car il contient des classes pré-faites pour les formulaires, les boutons et d’autres composants d’interface. <https://getbootstrap.com>

## Bootstrap Studio



Bootstrap studio est un outil payant que j’ai utilisé durant mon projet afin de me simplifier le design du site web. Il permet de créer un design utilisables et modifiable a souhait rapidement. Il permet aussi d’exporter le travail en un projet html, css et JavaScript.

## Flaticon

Un moteur de recherche utile pour trouver des icônes libres de droits

## Expression régulière

Les expressions régulières, plus communément appelées regex sont utilisées en informatique afin de faciliter les procédures de contrôles d’entrées, de lecture, d’analyse ou encore de modification. Dans le cadre de mon application, je les utilise dans le simple but de vérifier la force d’un mot de passe lors de l’enregistrement (une majuscule, une minuscule, un chiffre et un caractère spécial). <https://regex101.com/>

## Katalon Recorder

C’est une extension de navigateur, visant à aider les développeurs pour les tests et le débogage des applications. C’est une extension qui m’a aidé recommandé par mon formateur, et qui m’a servi tout au long de mon travail, afin de vérifier à chaque étape, qu’il n’y avait pas de régression.



## Trello

Trello est un outil de planification très utile pour savoir rapidement ce que l’on doit faire dans notre programme ou ce que nous avons déjà fait.

# Plan de tests et tests

## Périmètre des tests

Pour le projet Flight Club, j’ai choisi d’écrire des protocoles de test pour les actions qu’un utilisateur λ (lambda) utilisant un navigateur moderne et répandu pourrait raisonnablement effectuer, pour ce qui est des questions que l’utilisateur pourrait se poser, elles sont disponibles sur la page faq.php et présentent certaines contraintes actuelles de mon implémentation.

## Équipement de test

### Environnement

Lors du développement du projet, j’ai utilisé l’environnement suivant pour effectuer les tests : Google Chrome version 90.0.4430.93 (64 bits) sur Windows 10, version 10.0.19041 build 19041 Côté serveur, apache 2.4.25 a été utilisé avec PHP 7.1.3 et MySQL 5.7.17. Le système utilisé comme serveur est la machine Windows citée ci-dessus.

## Scénarios de test

Lors du premier jour du TPI, j’ai rédigé les scénarios de tests suivants qui m’ont permis par la suite de m’assurer que les besoins exprimés dans l’énoncé sont remplis. Les scénarios de tests sont précis dans le but d’être facilement reproductibles par une autre personne si nécessaire. Pour ce qui est de ce rapport de test, avec un camarade, on s’est dit que d’avoir un regard extérieur sur les tests serait une bonne idée, alors nous avons décidé de s’échanger nos sites, et de chacun tester celui de l’autre. Ce n’est donc pas moi, mais Yvelin RAY qui a fait le rapport de test. En revanche, pour l’évolution des tests, il s’agit bien de mes tests.

|  |  |
| --- | --- |
| Nom | 1.1 Affichage de la *home* page |
| User story | S3 : affichage de la home page |
| Situation | Étant donné que je suis un utilisateur non connecté Quand je me rends sur le site Alors une home page s’affiche. Cette page me présente brièvement le projet et contient un bouton « Se connecter » qui permet de se connecter au site. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 2.1 Connexion (compte existant) |
| User story | S7 : Connexion à son compte |
| Situation | Étant donné que je suis un utilisateur non connecté, qui possède déjà un compte sur le site. Quand je clique sur le bouton « Se connecter » sur la page de login et que je rentre les bons identifiants. Alors, je suis redirigé vers ma page profil, en étant connecté à mon compte existant. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 2.2 Connexion (erreur) |
| User story | S7 : Connexion à son compte |
| Situation | Étant donné que je suis un utilisateur non connecté, qui possède déjà un compte sur le site. Quand je clique sur le bouton “Se connecter” sur la page de login et que je rentre les mauvais identifiants. Alors, je reçois un message d’erreur. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 3.1 Inscription |
| User story | S6 : Création de compte |
| Situation | Étant un utilisateur anonyme, ne possédant pas encore de compte, je souhaite me créer un compte. Quand je suis sur la page inscription, il me suffit de remplir les champs requis. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 3.2 Inscription - erreur de saisie |
| User story | S6 : Création de compte |
| Situation | Étant un utilisateur anonyme, ne possédant pas encore de compte, je souhaite me créer un compte. Quand je suis sur la page inscription, si je mets un nom/prénom ou une adresse mail trop longue, ou que j’utilise des caractères non alphanumériques pour les champs noms et prénoms, lorsque je clique sur le bouton « register », je reçois un message d’erreur me demandant de vérifier mes entrées. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 4.1 Importation de pièce jointe |
| User story | S4 : Enregistrements des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je peux importer une photos de mon vols quand j’enregistre un de mes vols. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 4.2 Erreur Importation de pièce jointe |
| User story | S4 : Enregistrements des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J’enregistre une image d’un type pas accepté ou supérieur à 5MO lors de l’enregistrement d’un vol et un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 5.1 Enregistrement de vol – Pas d’erreur de saisie |
| User story | S4 : Enregistrements des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J’enregistre mon vol sur le site avec succès. Un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 5.2 Enregistrement de vol – Erreur de saisie |
| User story | S4 : Enregistrements des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J’enregistre mon vol avec des informations invalides. Un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 6.1 Modification de vol – Pas d’erreur de saisie |
| User story | S8 : Modification des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J’ai enregistrer un vol erroné. Je le corrige et enregistre la modification. Un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 6.2 Modification de vol – Erreur de saisie |
| User story | S8 : Modification des vols |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J’ai enregistrer un vol erroné. Je le corrige avec des données invalides et enregistre la modification. Un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 7.1 FAQ |
| User story | S16 : Accès à la FAQ |
| Situation | Lorsqu’un utilisateur veut en savoir plus sur le site, il lui suffit de cliquer sur le lien menant à la FAQ, page sur laquelle il apprendra quelques fonctionnalités et limites de l’implémentation actuelle du site. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 8.1 Téléchargement en CSV |
| User story | S5 : Téléchargement du carnet de vol |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le bouton télécharger en .CSV et le fichier se télécharge dans le format sélectionnés. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 8.2 Téléchargement en PDF |
| User story | S5 : Téléchargement du carnet de vol |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le bouton télécharger en .PDF et le fichier se télécharge dans le format sélectionnés. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 9.1 Modification du mot de passe - Succès |
| User story | S9 : Modification de mot de passe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de modification de mot de passe et je clique sur modifier mon mot de passe. Un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 9.2 Modification du mot de passe - Erreur |
| User story | S9 : Modification de mot de passe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de modification de mot de passe qui ne passe pas les tests de sécurité minimum et je clique sur modifier mon mot de passe. Un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 10.1 Création de groupe - Succès |
| User story | S10 : Créer un groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de création de groupe et je clique sur créer un groupe. Un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 10.2 Création de groupe - Erreur |
| User story | S10 : Créer un groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de création de groupe avec des informations invalides et je clique sur créer un groupe. Un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 11.1 Inviter des utilisateur à mon groupe - Succès |
| User story | S11 : Inviter des utilisateurs à mon groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs d’invitation de groupe et je clique sur inviter au groupe. Un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 11.2 Inviter des utilisateur à mon groupe - Erreur |
| User story | S11 : Inviter des utilisateurs à mon groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs d’invitation de groupe avec des données invalides et je clique sur inviter au groupe. Un message d’erreur apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 12.1 Accepter une invitation de groupe |
| User story | S12 : Accepter une invitation à un groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur accepter l’invitation et un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 13.1 Refuser une invitation de groupe |
| User story | S13 : Refuser une invitation à un groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur refuser l’invitation et un message de succès apparait. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 14.1 Affichage des vols dans les groupes auxquels j’appartiens |
| User story | S14 : Affichage des vols dans les groupes |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le groupe et tous les vols des utilisateurs du groupe apparaissent. |
| Résultats obtenus |  |
| Statut |  |

|  |  |
| --- | --- |
| Nom | 15.1 Quitter un groupe |
| User story | S15 : Quitter un groupe |
| Situation | Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur Quitter le groupe et un message de succès apparait. Je n’ai plus accès aux vols de ce groupe |
| Résultats obtenus |  |
| Statut |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N° test | J0 | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 | J9 | J10 | J11 |
| 1.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 2.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 2.2 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 3.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 4.1 | X | X | X | X | X |  |  |  |  |  |  |  |
| 4.2 | X | X | X | X | X |  |  |  |  |  |  |  |
| 5.1 | X | X | X | ✓ | ✓ |  |  |  |  |  |  |  |
| 5.2 | X | X | X | ✓ | ✓ |  |  |  |  |  |  |  |
| 6.1 | X | X | X | ✓ | ✓ |  |  |  |  |  |  |  |
| 6.2 | X | X | X | ✓ | ✓ |  |  |  |  |  |  |  |
| 7.1 | X | X | X | X | X |  |  |  |  |  |  |  |
| 8.1 | X | X | X | X | X |  |  |  |  |  |  |  |
| 8.2 | X | X | X | X | ✓ |  |  |  |  |  |  |  |
| 9.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 9.2 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 10.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 10.2 | X | X | X | X | ✓ |  |  |  |  |  |  |  |
| 11.1 | X | X | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 11.2 | X | X | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 12.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 13.1 | X | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 14.1 | X | X | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |
| 15.1 | X | X | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |

# Glossaire

## Termes techniques

### User stories (ou récit utilisateur)

Description d’une fonctionnalité utilisée dans la méthodologie agile, qui se présente sous cette forme: En tant que <qui>, je veux <quoi> afin de <pourquoi>.

### Importation (Anglais : upload)

Fait de pouvoir importer des fichiers depuis son ordinateur.

### Carnet de produit (Anglais: product backlog)

Ensemble des besoins recueillis pour créer le produit.

### Formulaire GET

Un formulaire GET, permet de recevoir les paramètres depuis l’URL.