

Documentation technique

AVRIL-MAI 2023

Billegas Lucas

I.DA-P4B

Maître d’apprentissage : M. Garchery

ShareGames

# Table des matières

[Table des matières 3](#_Toc133418455)

[Versions 5](#_Toc133418456)

[Introduction 5](#_Toc133418457)

[Résumé du cahier des charges 5](#_Toc133418458)

[But de l’application 5](#_Toc133418459)

[Fonctionnalité à réaliser 5](#_Toc133418460)

[Matériel et logiciels Nécessaires 5](#_Toc133418461)

[Livrables 5](#_Toc133418462)

[Méthodologie 6](#_Toc133418463)

[S’informer 6](#_Toc133418464)

[Planifier 6](#_Toc133418465)

[Décider 6](#_Toc133418466)

[Réaliser 6](#_Toc133418467)

[Contrôler 6](#_Toc133418468)

[Evaluer 6](#_Toc133418469)

[Planification 7](#_Toc133418470)

[Product backlog 7](#_Toc133418471)

[Planning prévisionnel 7](#_Toc133418472)

[Planning réel 7](#_Toc133418473)

[Analyse fonctionnelle 8](#_Toc133418474)

[Fonctionnalités 8](#_Toc133418475)

[Interfaces 8](#_Toc133418476)

[Structure du projet 9](#_Toc133418477)

[Base de données 10](#_Toc133418478)

[Librairies et outils externes 10](#_Toc133418479)

[Bootstrap 10](#_Toc133418480)

[Visual Studio Code 10](#_Toc133418481)

[WSL 2 (Ubuntu) 10](#_Toc133418482)

[Pecee/simple-router 11](#_Toc133418483)

[Github 11](#_Toc133418484)

[Composer 11](#_Toc133418485)

[Autoload 11](#_Toc133418486)

[Périmètre de test 12](#_Toc133418487)

[Équipement de test 12](#_Toc133418488)

[Plan de test 13](#_Toc133418489)

[Évolution des tests 14](#_Toc133418490)

[Rapport de test 15](#_Toc133418491)

[Conclusion 16](#_Toc133418492)

[Planification 16](#_Toc133418493)

[Améliorations possibles 16](#_Toc133418494)

[Bilan personnel 16](#_Toc133418495)

[ANNEXES 17](#_Toc133418496)

[Bibliographie 18](#_Toc133418497)

# 

# Versions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N. de version |  | Auteur | Date | Changements |
|  | 1.0 | Billegas Lucas | xx.xx.2023 | Maquette |

# Introduction

Ce document est un rapport décrivant toute la conception du projet réalisé dans le cadre du Travail pratique individuel (TPI) se déroulant du 27 avril au 17 Mai 2023. Ce travail a pour but de valider mes compétences techniques acquises durant toute ma formation d’informaticien en développement d’applications.

# Résumé du cahier des charges

Les informations ci-dessous sont extraites du cahier des charges du TPI :

|  |  |
| --- | --- |
| Élève | Maître d’apprentissage |
| Billegas Lucas  lucas.bllgs@eduge.ch | Garchery Stéphane  [stephane.garchery@edu.ge.ch](mailto:stephane.garchery@edu.ge.ch) |

|  |  |
| --- | --- |
| Experts | |
| Bouille Robin  [robin.bouille@gmail.com](mailto:robin.bouille@gmail.com) | Bizard Yohan  [yohan.bizard@gmail.com](mailto:yohan.bizard@gmail.com) |

### But de l’application

### Fonctionnalité à réaliser

### Matériel et logiciels Nécessaires

### Livrables

# Méthodologie

Afin d’avoir une bonne gestion de mon projet, j’ai utilisé la méthode de planification en 6 étapes. Le schéma ci-dessus montre le déroulement de la méthodologie et appuie sur le fait qu’il faut respecter chacune des étapes.

### S’informer

Se renseigner est la première des 6 étapes de la méthodologie. Elle consiste à comprendre les différentes attentes et à bien cerner le sujet afin de se faire une idée de l’objectif et se procurer systématiquement les bonnes informations.

### Planifier

Planifier est le meilleur moyen d’accomplir son travail dans un temps défini. Pour ce faire, il faut transformer les fonctionnalités demandées en tâches et estimer le temps nécessaire pour effectuer chacune d’entre elles.

### Décider

Après avoir récupérer les informations et planifiées les tâches, il faut décider de l’importance de chaque tâche, ce qui permet d’avoir un ordre de réalisation.

### Réaliser

Une fois les tâches triées par importance, il faut réaliser ces dernières. Pour cela, il faut suivre la planification et à la décision qui ont été faites précédemment.

### Contrôler

Une fois la tâche terminée, il faut vérifier qu’elle fonctionne correctement. Il faut donc mettre en place un scénario de test afin de tester efficacement la fonctionnalité qui a été implémenté.

### Evaluer

La tâche étant terminé, il faut passer en revu cette dernière afin de voir comment elle a été faite, s’il y a des choses qui laissent à désirer ainsi que ce qui pourrait être amélioré pour les prochaines tâches.

# Planification

### Product backlog

|  |  |
| --- | --- |
| Nom | **S1 : Utilisation d’un dépôt Git** |
| Description (user story) | En tant que développeur, je dois utiliser un dépôt Git afin d’enregistrer et versionner mon projet. |
| Critère d’acceptation | Un dépôt Git est créé et est en lien avec GitHub Desktop. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S2 : Installation de la base de données** |
| Description (user story) | En tant que développeur, je dois installer une base de données qui respectent le MCD qui m’a été donné dans l’énoncé. |
| Critère d’acceptation | La base de données est créée ainsi que les tables et un user spécifique à cette base de données. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S3 : Création du style du site** |
| Description (user story) | En tant que développeur, je dois créer le visuel de chaque page. |
| Critère d’acceptation | Le visuel de chaque page correspond à la description de l’énoncé. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S4 : Création d’une fonction PHP pour la connexion à la base de données** |
| Description (user story) | En tant que développeur, je dois créer une connexion à la base de données pour pouvoir récupérer les données. |
| Critère d’acceptation | Les données peuvent être récupérées, ajoutées, supprimées et éditées depuis des fonctions PHP. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S5 : Installation et configuration de composer et des librairies externes** |
| Description (user story) | En tant que développeur, je dois mettre en place un routeur pour gérer les url et mieux gérer le code. |
| Critère d’acceptation | Le test 1.1 passe. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S6 : Mise en place d’un Virtual Host** |
| Description (user story) | En tant que développeur, je dois créer un Virtual host qui me permettra de mettre le routeur dessus. |
| Critère d’acceptation | Le site est fonctionnel depuis un navigateur. |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S7 : Affichage des jeux sur la page d’accueil** |
| Description (user story) | En tant qu’utilisateur, je dois voir la liste des 10 meilleurs jeux selon leur moyenne de notes. Je dois aussi pouvoir voir les 5 jeux les plus récents selon leur date de saisie / modification. |
| Critère d’acceptation |  |
| Priorité | P2 : Critique |

|  |  |
| --- | --- |
| Nom | **S8 : Recherche de jeux** |
| Description (user story) | En tant qu’utilisateur, je dois pouvoir rechercher des jeux par le titre ou la description. |
| Critère d’acceptation |  |
| Priorité | P2 : Critique |

|  |  |
| --- | --- |
| Nom | **S9 : Affichage et filtrage des jeux sur une page** |
| Description (user story) | En tant qu’utilisateur, je dois pouvoir filtrer les jeux par date de saisie / modification ou par note. |
| Critère d’acceptation |  |
| Priorité | P2 : Critique |

|  |  |
| --- | --- |
| Nom | **S10 : Inscription** |
| Description (user story) | En tant qu’utilisateur, je dois pouvoir m’inscrire en saisissant un pseudo, un mot de passe et un avatar en option. |
| Critère d’acceptation |  |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S11 : Connexion** |
| Description (user story) | En tant qu’utilisateur, je dois pouvoir me connecter à mon compte avec mes informations. |
| Critère d’acceptation |  |
| Priorité | P1 : Bloquant |

|  |  |
| --- | --- |
| Nom | **S12 : Gestion des fiches de jeux** |
| Description (user story) | En tant qu’administrateur, je peux ajouter, supprimer, voir et modifier une fiche de jeu. |
| Critère d’acceptation |  |
| Priorité | P2 : Critique |

|  |  |
| --- | --- |
| Nom | **S13 : Déconnexion** |
| Description (user story) | En tant qu’utilisateur identifié, je peux me déconnecter de mon compte. |
| Critère d’acceptation |  |
| Priorité | P4 : Secondaire |

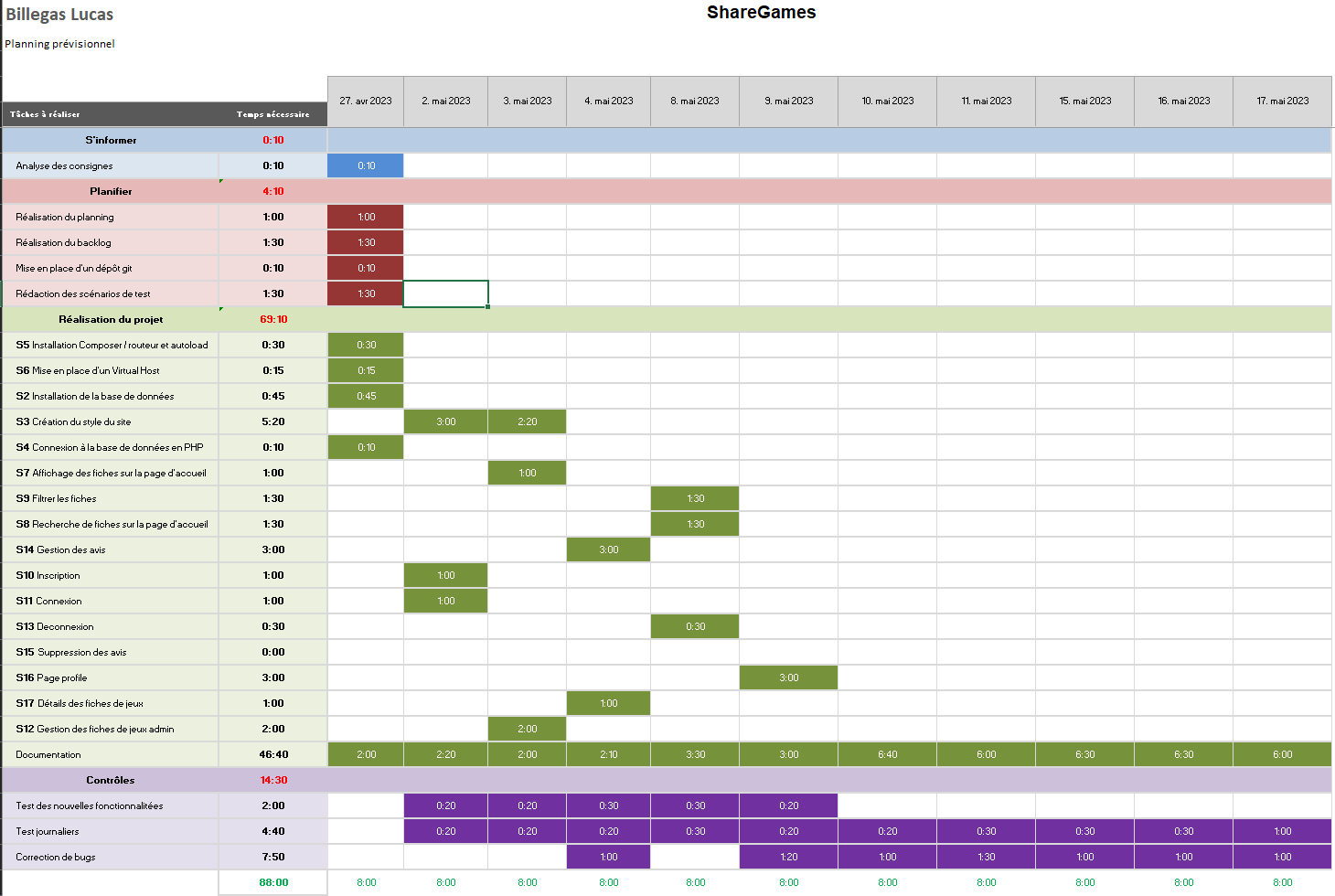
|  |  |
| --- | --- |
| Nom | **S14 : Gestion des avis** |
| Description (user story) | En tant qu’utilisateur identifié, je peux modifier, supprimer, voir et créer un avis. |
| Critère d’acceptation |  |
| Priorité | P2 : Critique |

|  |  |
| --- | --- |
| Nom | **S15 : Suppression des avis** |
| Description (user story) | En tant qu’administrateur, je peux supprimer un avis. |
| Critère d’acceptation |  |
| Priorité | P3 : Important |

|  |  |
| --- | --- |
| Nom | **S16 : Page profile** |
| Description (user story) | En tant qu’utilisateur identifié, je peux modifier mon avatar et mon mot de passe. J’ai aussi accès à toutes les fiches de jeux dans lesquels j’ai posté un ou plusieurs avis. |
| Critère d’acceptation |  |
| Priorité | P3 : Important |

|  |  |
| --- | --- |
| Nom | **S17 : Détails des fiches de jeux** |
| Description (user story) | En tant qu’utilisateur, je peux voir une fiche de jeu avec ces commentaires trier par ordre chronologique sous la fiche. |
| Critère d’acceptation |  |
| Priorité | P3 : Important |

### Planning prévisionnel



### Planning réel

# Analyse fonctionnelle

### Fonctionnalités

### Interfaces

Analyse organique

### Structure du projet

La structure de l’application a été réalisé en MVC (modèle, vue contrôleur). Chaque fichier est classé dans le dossier correspondant.

public/index.php

Le fichier principal qui démarre le routeur et redirige sur routes.php.

routes/routes.php

Le fichier qui contient toutes les routes et qui redirige en fonction de la page à afficher.

src/Views/

Le dossier qui contient toutes les vues.

src/Models/

Le dossier qui contient toutes les fonctions et la logique liées à la base de données.

src/Controllers/

Le dossier qui contient un controlleur par vue qui permet de faire le lien entre le modèle et la vue.

public/assets/

Le dossier contient toute la partie médias, design. Il contient toute la css, tout le javascript ainsi que toutes les images de l’application.

src/vendor/

Le dossier contient toutes les librairies installées. C’est le dossier généré par Composer.

src/vendor/composer

Le dossier contient tous les fichiers générés par l’installation de autoload.

src/vendor/pecee/simple-router

Le dossier contient les fichiers du routeur qui ont été généré par l’installation de simple-router.

composer.json

Fichier de configuration qui permet de spécifier les dépendances requises pour le projet.

### Base de données

# Librairies et outils externes

### Bootstrap

Bootstrap est un framework CSS utilisé pour faciliter la création visuelle de l’application. J’ai choisi ce framework car c’est celui avec lequel je travaille depuis ma 1ère année et il est facile d’utilisation.

### Une image contenant texte, clipart Description générée automatiquementVisual Studio Code

Visual studio code est un environnement de développement facile à utiliser et complet. Il permet de coder dans différents langages et beaucoup d’extensions sont disponibles.

Je l’ai utilisé pour développer mon site web.

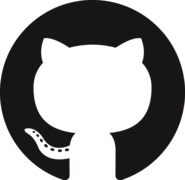
### WSL 2 (Ubuntu)

Une image contenant texte, clipart

Description générée automatiquementWSL 2 est un sous-système Linux dans Windows. Il permet en quelque sorte d’avoir une machine virtuelle Linux tout en étant relié à la machine Windows. J’ai utilisé ubuntu 20.04 pour me connecter.

### Pecee/simple-router

Pecee simple-router est une librairie fortement inspirée de la façon dont Laravel gère ses routages. Il est rapide et puissant et il est facile de l’intégrer dans un projet. C’est un outil utile pour gérer le routage dans une application web PHP et il offre une manière simple et flexible de définir les routes dans une application.



### Github

Github est un outil de versionnage GitHub est une plate-forme d'hébergement de code pour le contrôle de version et la collaboration. Il permet de travailler à plusieurs sur des projets et d’avoir un historique de versions.

### Composer (logiciel) — WikipédiaComposer

Composer est un logiciel de gestionnaire de dépendances libre écrit en PHP. Il permet d’installer des bibliothèques dans un projet. C’est grâce à composer que j’ai pu inclure Pecee/simple-router et autoload.

### Autoload

L'autoload est un mécanisme qui permet de charger automatiquement les classes PHP sans avoir à les inclure manuellement avec la fonction «include» ou « require ». Il sert aussi à définir un namespace et choisir le dossier cible qui contient les classes. En résumé, il facilite la gestion des classes dans les applications PHP et réduit les erreurs liées aux inclusions manuelles de fichiers.

# Périmètre de test

# Équipement de test

# Plan de test

Un plan de test des principales fonctionnalités de l’application est disponible ci-dessous.

|  |  |
| --- | --- |
| Nom | 1.1 **Routage** |
| User story | S5 **Installation et configuration de composer et des librairies externes** |
| Situation | Quand on arrive sur le site une page d’accueil s’affiche. |
| Résultat attendu | Le système de routage est fonctionnel et les pages s’affichent en fonction de leur url. |

# Évolution des tests

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N.  test | J1  jeu. 27.04 | J2  mar. 02.05 | J3 mer.  03.05 | J4 je.  04.05 | J5 lun.  08.05 | J6  mar.  09.05 | J7 mer.  10.05 | J8 jeu.  11.05 | J9 lun.  15.05 | J10 mar.  16.05 | J11 mer.  17.05 |
| 1.1 | ✔ |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | ❌ |
|  |  |  |  |  |  |  |  |  |  |  | ✔ |

# Rapport de test

|  |  |
| --- | --- |
| Nom | 1.1 **Routage** |
| Date | 27.04.2023 |
| Résultat obtenu | La page d’accueil c’est affiché. |
| Statut | OK |

# Conclusion

### Planification

### Améliorations possibles

### Bilan personnel

# ANNEXES

# Bibliographie

Template :

https://themewagon.com/themes/blogy/