Plan de tests

Module méthodologie de tests

## Groupe

- Cellier Matthieu

- Devillepoix Maxime

- De Sousa Enzo

## Mail si incident avec 360 learning

yann.coornaert1@mail-formateur.net

# 

# Résumé

L’entreprise Game Freak souhaite développer un nouveau jeu Pokémon.

Ce jeu, qui est une web app, doit fonctionner sur PC (ou Mac) mais aussi sur les appareils Android 10 ou plus.

Les utilisateurs ne peuvent avoir qu’un compte par adresse mail. Ils peuvent aussi se connecter grâce à leur compte Facebook ou Google, dans ce cas le compte est rattaché au mail lié au service externe.

Les utilisateurs sont des dresseurs qui possèdent des Pokémons et des objets. Les dresseurs peuvent se battre les uns contre les autres et voir les pokémons et objets qu’ils possèdent.

Le jeu contient un classement. Une victoire rapporte 10 points + le nombre de Pokémon restant invaincu, une défaite en fait perdre 10 + le nombre de Pokémon restant invaincu.

Chaque dresseur peut avoir 5 pokémons aléatoires par jour et doit forcément remplacer 1 des pokémons de son équipe, mais pas le plus récent.

Un utilisateur obtient 1 objet par jour, et jusqu’à 3 autres en gagnant des matchs.

La partie front-end est développée en Javascript via la librairie React, tandis que le back-end est développé en Python avec le framework FastAPI. La base de données utilisée est PostgreSQL.

L’application sera déployée dans un conteneur via la google cloud platform.

# Ressources impliquées

Définir les besoins pour effectuer les diverses activités de tests

**Ressources pour les Tests Unitaires :**

Framework de Tests pour FastAPI (Back-End) :

* Utilisation d'outils comme Pytest pour les tests unitaires Back-End.

### **Ressources pour les Tests d'Intégration :**

### Base de Données de Test (PostgreSQL) :

### Utilisation d'une instance PostgreSQL dédiée pour les tests d'intégration.

### **Ressources pour les Tests de Système**

Outils de Tests de Performance :

* Utilisation d'outils comme Locust pour les tests de charge.

**Ressource pour effectuer les tests**

### **Authentification :**

* Authentification via Facebook et Google

### **Développement Back-End (FastAPI) :**

### - FastAPI

### - Python

### - Gestionnaire de Base de Données (PostgreSQL)

### **Base de Données :**

### - PostgreSQL

### **Outils de Développement et Collaboration :**

### **Git et GitHub:**

### Utilisation de Git pour la version de nos tests et de GitLab pour la collaboration.

### **Éditeurs de Code :**

### Utilisation d'un éditeur de code tel que Visual Studio Code.

### **Documentation et Communication :**

**Documentation du Projet :**

* Utilisation d'outils comme Confluence, GitBook ou Markdown pour documenter le projet.

**Communication d'Équipe :**

* Utilisation de plates-formes de communication comme Discord.

### 

# 

# 

# 

# 

# 

# KPIs

Définir 3 KPIs permettant de suivre l'avancée des tests

Voici les différents types de KPI mis en oeuvre pour voir l’avancée des tests

* Taux de Réussite des Connexions:
  + Mesurer le pourcentage de réussite des connexions des utilisateurs sur la web app, en particulier en se connectant via leur adresse e-mail, compte Facebook ou compte Google. Un taux élevé indique une bonne stabilité et fonctionnalité de la fonction de connexion.
* Taux de Victoires/Défaites dans les Combats:
  + Suivre le pourcentage de victoires et de défaites des dresseurs lors des combats. Cela permettra d'évaluer l'équilibre du jeu, la justesse des règles et la satisfaction des joueurs. Un ajustement des mécaniques de combat peut être nécessaire en fonction de ces données.
* Temps de Réponse du Serveur:
  + Mesurer le temps de réponse du serveur backend lors des interactions avec la web app. Un temps de réponse rapide est crucial pour offrir une expérience utilisateur fluide. Tout ralentissement peut affecter négativement l'expérience des joueurs, et cela peut être identifié grâce à ce KPI.

# 

# 

# 

# Budget

Augmentation capacité serveur si la capacité de prise en charge n’est pas suffisante

Ajout de nouvelle fonctionnalité

Modification fonctionnalité ou infrastructure de l’appli

Maintenance des scripts de tests

Licences logicielles éventuellement nécessaire pour certains outils

Financement de formations à de nouvelles technologies.

# Cas de tests

Au moins 4 cas de tests dont 1 négatif

### **Cas de test 1 : Connexion avec un compte email**

* Description : L'utilisateur tente de se connecter en utilisant une adresse email valide.
* Étapes :
  + Accéder à l'écran de connexion.
  + Saisir une adresse email existante.
  + Saisir le mot de passe correct.
  + Appuyer sur le bouton de connexion.
* Résultat attendu : L'utilisateur doit être connecté avec succès.

### **Cas de test 2 : Connexion avec un compte Facebook**

* Description : L'utilisateur tente de se connecter en utilisant son compte Facebook.
* Étapes :
  + Accéder à l'écran de connexion.
  + Sélectionner l'option de connexion via Facebook.
  + Autoriser l'application à accéder au compte Facebook.
* Résultat attendu : L'utilisateur doit être connecté avec succès en utilisant son compte Facebook.

### **Cas de test 3 : Remplacement quotidien de Pokémon**

* Description : L'utilisateur remplace un Pokémon de son équipe.
* Étapes :
  + Accéder à l'écran de gestion d'équipe.
  + Sélectionner l'option de remplacement quotidien.
  + Choisir un Pokémon à remplacer.
* Résultat attendu : Le Pokémon est remplacé avec succès dans l'équipe de l'utilisateur.

### **Cas de test 4 : Tentative de connexion avec des identifiants incorrects**

* Description : L'utilisateur tente de se connecter avec des identifiants incorrects.
* Étapes :
  + Accéder à l'écran de connexion.
  + Saisir une adresse email incorrecte.
  + Saisir un mot de passe incorrect.
  + Appuyer sur le bouton de connexion.
* Résultat attendu : Un message d'erreur doit s'afficher, indiquant que les identifiants sont incorrects.

### **Cas de test négatif : Tentative de connexion avec un compte inexistant**

* Description : L'utilisateur tente de se connecter avec une adresse email qui n'est pas enregistrée.
* Étapes :
  + Accéder à l'écran de connexion.
  + Saisir une adresse email inexistante.
  + Saisir un mot de passe.
  + Appuyer sur le bouton de connexion.
* Résultat attendu : Un message d'erreur doit s'afficher, indiquant que le compte n'existe pas.

# 

# Evaluation des risques

1 matrice de risque, possible de fournir un sheet externe, préciser le nom de l’annexe

# Critères d’entrée et de sortie

## **Critère d’entrée**

Code source

Environnement de test (application)

Données de test

## **Critère de sortie**

Résultats/rapports des tests

Stabilités du codes

# Planification

Fournir 1 calendrier + priorisation et dépendance de vos cas de tests

# Facteurs d’effort de test

1 paragraphe indiquant différents facteurs entrant en jeu

Les facteurs d’effort pouvant entrer en jeu sont tout d'abord les ressources humaines et matérielles. Ensuite il faut prendre en compte les délais impartis pour livrer le produit final et la stabilité de celui-ci.  
Pour finir il y a certains outils spécialisés qui pourront être nécessaires au développement ou aux tests et qui nécessiteront donc certaines formations.