

2023

Projet Conception Agile
Master 1 Informatique

PLANNING POKER

RAPPORT

Kawther Djouad
Manar HAKOUME

SOMMAIRE

- 01** Introduction
- 02** Choix des technologies
- 03** Présentation des Pages de l'Application
- 04** Annexes

INTRODUCTION

Plongés dans l'univers agile du développement, nous avons conçu une application de planning poker alliant simplicité et efficacité. Guidés par notre choix de **Python** comme langage de programmation et le framework **Dash**, nous avons créé une solution à la fois puissante et conviviale. Optant délibérément pour une application **locale**, nos utilisateurs peuvent se réunir physiquement autour d'une table, favorisant ainsi une interaction directe.

Cette application, conçue pour la planification de projets, offre des fonctionnalités basiques telles que le choix du nombre de joueurs et la personnalisation des pseudonymes. Notre rapport détaillera ce voyage de conception, soulignant notre utilisation stratégique de design patterns pour assurer une structure robuste. De plus, nous partagerons notre approche de l'intégration continue, des tests automatisés et de la documentation, démontrant ainsi notre engagement envers la qualité technique.

CHOIX DES TECHNOLOGIES

1. Langage de Programmation Python :



Python a été choisi en raison de sa syntaxe claire, de sa lisibilité du code et de sa polyvalence, permettant un développement rapide et efficace. Sa vaste bibliothèque standard a facilité la mise en œuvre élégante du projet et favorisé une collaboration fluide au sein de l'équipe de développement.

2. Framework Dash



Dash, le framework web développé par Plotly, a été choisi pour son approche déclarative simple, permettant de créer des applications web interactives en utilisant uniquement Python. Il offre une flexibilité suffisante pour répondre aux besoins spécifiques de notre application Planning Poker, tout en intégrant naturellement des fonctionnalités visuelles avancées avec Plotly.

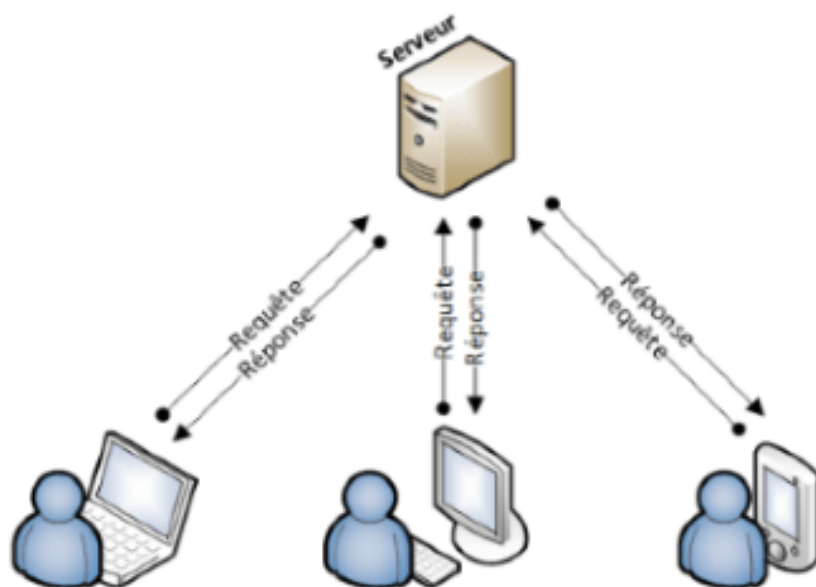
3. Firebase



Firebase, un ensemble de services cloud de Google, a été sélectionné pour sa base de données en temps réel, essentielle pour synchroniser les estimations des utilisateurs dans notre application Planning Poker. Les services d'authentification, de stockage de fichiers et autres fonctionnalités de Firebase ont simplifié la mise en œuvre des aspects avancés de l'application.

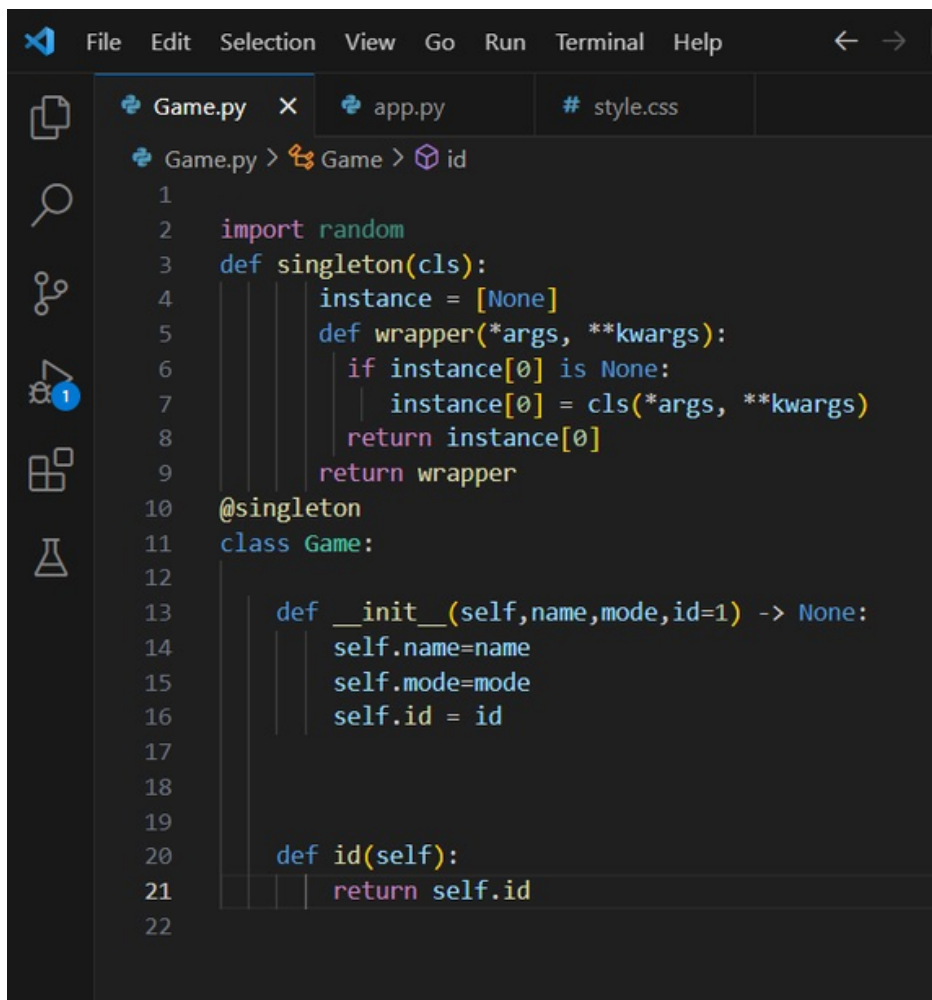
4. Architecture Client-Serveur :

L'adoption de l'architecture client-serveur pour notre application Planning Poker découle de son alignement avec les principes agiles. Cette architecture offre une séparation claire des responsabilités, facilitant la maintenance et l'évolutivité de l'application. En divisant la logique métier côté serveur de l'interface utilisateur côté client, nous favorisons une conception modulaire et adaptable, essentielle dans un environnement agile en constante évolution. De plus, cette approche permet une meilleure gestion des données et renforce la sécurité en limitant l'accès aux informations sensibles du côté serveur.



5. Pattern Singleton

L'intégration du motif de conception Singleton dans notre application Planning Poker est motivée par le besoin de garantir qu'une seule instance de la classe Game existe à tout moment. Ce choix architectural est particulièrement pertinent dans le contexte de notre application, où la gestion des parties de jeu doit rester cohérente et où plusieurs instances de la classe Game pourraient introduire des incohérences dans les données. En adoptant le Singleton, nous nous assurons que l'accès à la partie de jeu en cours est centralisé et qu'une seule instance de la classe Game est partagée de manière globale. Cela simplifie la gestion des données liées à une partie de jeu unique, évite les conflits potentiels entre instances et contribue à maintenir la cohérence de l'application Planning Poker dans un environnement agile où les changements peuvent être fréquents.

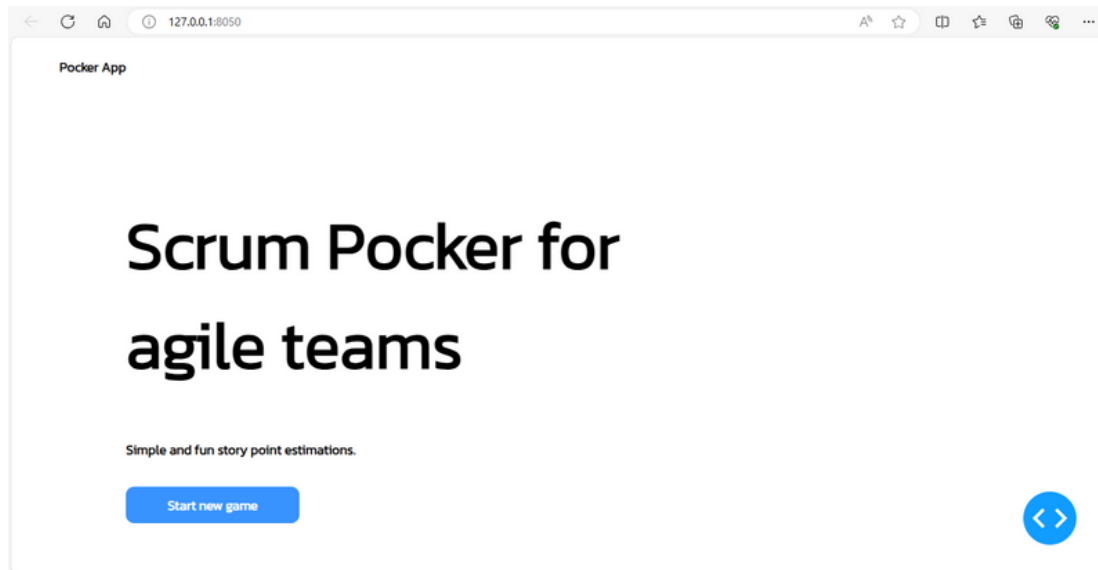


```
File Edit Selection View Go Run Terminal Help
Game.py X app.py # style.css
Game.py > Game > id
1
2 import random
3 def singleton(cls):
4     instance = [None]
5     def wrapper(*args, **kwargs):
6         if instance[0] is None:
7             instance[0] = cls(*args, **kwargs)
8         return instance[0]
9     return wrapper
10 @singleton
11 class Game:
12
13     def __init__(self, name, mode, id=1) -> None:
14         self.name=name
15         self.mode=mode
16         self.id = id
17
18
19
20     def id(self):
21         return self.id
22
```

PRÉSENTATION DES PAGES DE L'APPLICATION

1. Page d'Accueil :

- Titre de l'Application.
- Bouton "Start New Game" pour créer une nouvelle partie.



2. Page de Configuration de la Partie :

- Champs pour entrer le nom de la partie et choisir le mode de jeu.
- Bouton "Create Game" pour afficher un pop-up de gestion des joueurs.
- Pop-up pour ajouter ou supprimer des joueurs.
- Bouton "Go to the Game" pour accéder à l'interface de jeu.

Start a game

Game's name

Entrez le type

Create game



127.0.0.1:8050/Start

Start a game

Test Game

Average

Strict

Average

Median

Absolute Majority

Relative majority



Start a game

Manar

Kanther

+ Add Player

- Delete Player

Go to the game



3. Page de l'Interface de Jeu :

- Cartes de votes pour chaque joueur.
- Espace pour entrer les tâches à estimer.
- Fonctionnalités pour soumettre les estimations.
- Affichage des résultats et des discussions.
- Bouton pour clore la session.

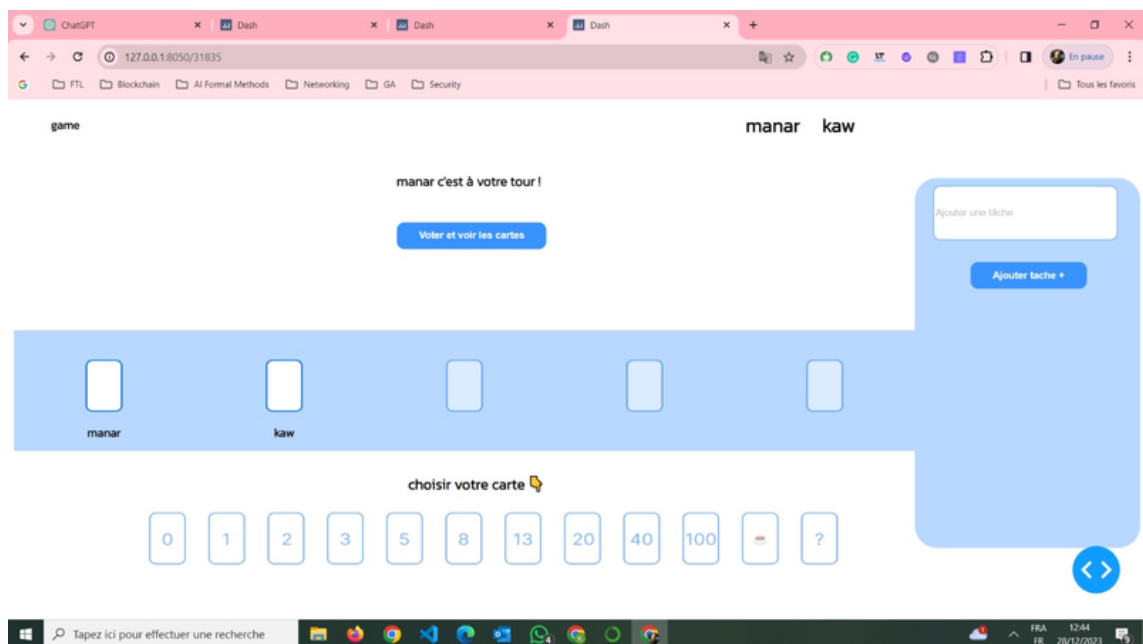
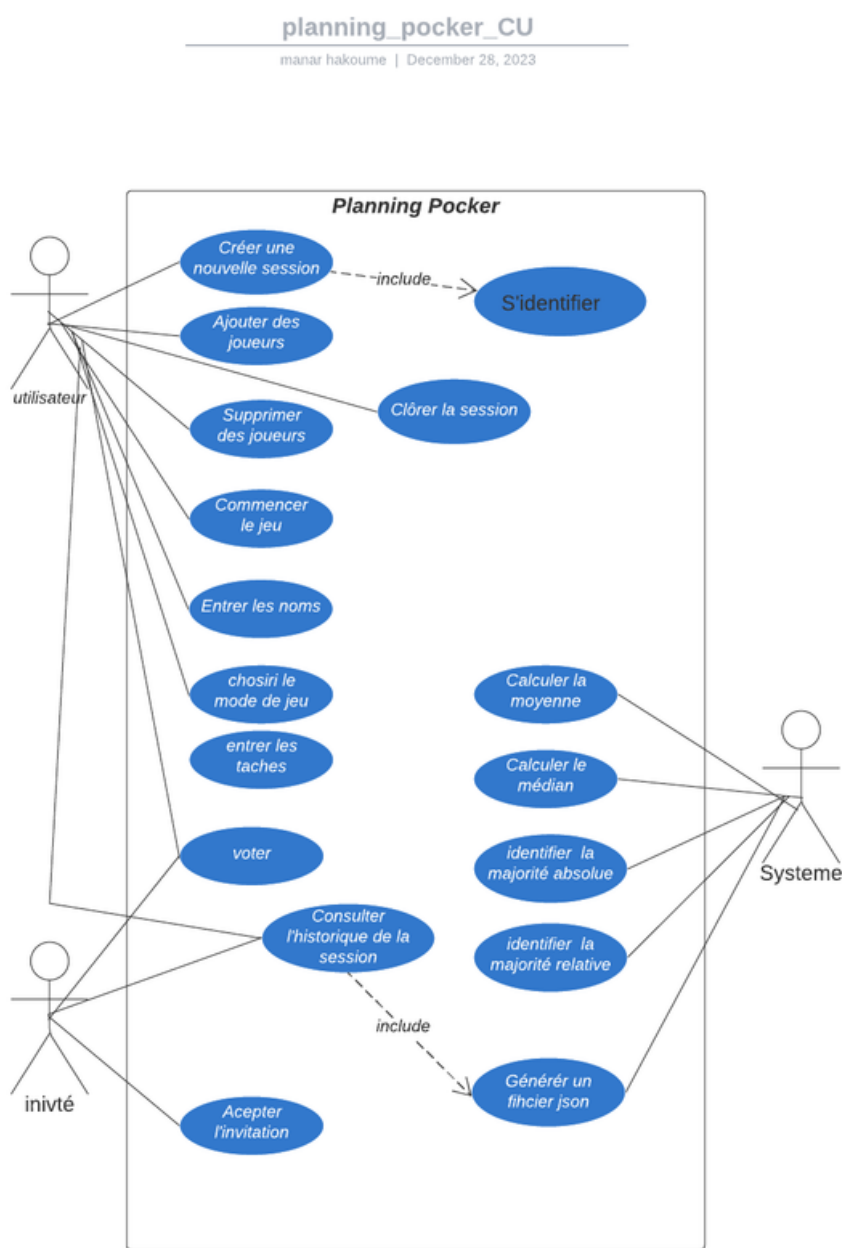


Diagramme de Cas d'utilisation

Dans cette section, nous présenterons visuellement le cas d'utilisation principal de notre application Planning Poker à travers un diagramme. Ce schéma illustrera de manière concise les interactions entre les utilisateurs et le système, mettant en lumière les étapes clés du processus d'estimation des tâches au sein d'une équipe agile. L'objectif est de fournir une représentation visuelle claire des fonctionnalités interactives de l'application sans recourir à une explication textuelle détaillée.



manarhik / Conception_Agile

Q Type to search

>

+

🔄

🔍

📧

👤

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Conception_AgilePublic

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to file

Add file

<> Code

About

Unknown2024 Update README.md8 minutes ago24 Commits

assets	Add files via upload	14 minutes ago
cartes	Commit1	last month
code	Commit1	last month
Game.py	Add files via upload	13 minutes ago
README.md	Update README.md	8 minutes ago
app.py	Add files via upload	13 minutes ago
planningpocker-f092a-firebase-adminsdk-9rqj...	Add files via upload	13 minutes ago

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages