



**RÉPUBLIQUE  
FRANÇAISE**

*Liberté  
Égalité  
Fraternité*



Sujet :

Conception et réalisation d'une application

Web

De planning poker

Réaliser par :

Mr.ABBAOUI MOHAMMED

Mr.HAMAD ALI ABDOURAZAK

Encadré par :

Professeur VALENTIN LACHAND pascal

Année universitaire : 2023-2024

## Remerciements

Nous tenons à adresser nos sincères remerciements à M. Valentin Lachand Pascal, notre professeur de cours et de travaux pratiques, ainsi que l'encadrant de ce projet. Son expertise, sa disponibilité et ses précieux conseils ont été essentiels tout au long de la réalisation de ce travail. Grâce à son accompagnement attentif, nous avons pu surmonter les défis rencontrés et enrichir nos connaissances dans le développement d'applications et l'intégration continue.

## Table des matières

<b>Remerciements</b> .....	2
Introduction .....	4
1.Présentation Globale du Projet .....	5
2. Motivation des Choix Techniques .....	5
3. Mise en Place de l'Intégration Continue (CI) .....	6
3.1 Outil Utilisé : GitHub Actions .....	6
3.2 Étapes de Configuration .....	6
3.3 Tests Automatisés .....	7
3.4 Résultats et Avantages .....	7
4. Fonctionnalités et Utilisation .....	8
5. Structure du Projet .....	9
6. Diagrammes Explicatifs .....	10
7.Conclusion Générale .....	13
8.Bibliographie et Références .....	13

## Introduction

Le planning poker est un outil essentiel des méthodologies agiles, utilisé pour estimer la complexité des tâches et assurer une planification efficace des projets. Cet outil collaboratif favorise la communication au sein des équipes et garantit une répartition équilibrée du travail.

Dans ce projet, nous avons développé une application numérique de planning poker, conçue pour répondre aux besoins d'équipes agiles travaillant dans des environnements modernes. L'objectif principal est de fournir une plateforme intuitive et interactive permettant d'estimer les tâches de manière collaborative, tout en intégrant des fonctionnalités avancées comme la gestion des modes de jeu, la sauvegarde et la reprise des sessions, et un chronomètre pour encadrer les votes.

En exploitant des technologies web modernes telles que HTML, CSS, et JavaScript, notre application garantit une interface utilisateur fluide et responsive. De plus, la mise en œuvre de l'intégration continue avec GitHub Actions assure une qualité de développement optimale et une maintenance simplifiée.

Ce rapport présente les étapes de conception et de réalisation de cette application, ainsi que les choix techniques et méthodologiques qui ont guidé son développement.

## 1.Présentation Globale du Projet

Ce projet consiste à développer une application numérique de planning poker, un outil essentiel des méthodologies agiles. L'objectif principal est de fournir une plateforme interactive et intuitive permettant d'estimer la complexité des tâches au sein d'une équipe, tout en respectant les règles classiques de cet outil. Nous avons intégré des fonctionnalités comme la gestion des modes de jeu, la sauvegarde et reprise des sessions, ainsi qu'une interface utilisateur responsive.

## 2. Motivation des Choix Techniques

Nous avons choisi **HTML, CSS, et JavaScript** pour leur complémentarité et leur pertinence dans le développement d'applications web modernes.

- **HTML :**  
Constitue la base de toute application web. Il permet de structurer les pages et de définir les éléments interactifs comme les boutons, les formulaires, et les tableaux.
- **CSS :**  
Indispensable pour styliser et mettre en forme les éléments définis par HTML. CSS offre une interface utilisateur visuellement attrayante et cohérente. En le combinant avec des frameworks comme **Bootstrap**, nous avons pu créer une interface **responsive**, adaptée à différentes tailles d'écran, tout en optimisant le temps de développement.
- **JavaScript :**  
Apporte la dynamique nécessaire à l'application. Il gère les interactions utilisateur, les mises à jour en temps réel, et la logique métier, comme le traitement des votes et la validation des données.

Ce trio, largement documenté, universellement compatible et soutenu par une communauté active, garantit un développement efficace, une maintenance facilitée, et une expérience utilisateur optimale.

### 3. Mise en Place de l'Intégration Continue (CI)

#### 3.1 Outil Utilisé : GitHub Actions

GitHub Actions est intégré à GitHub, ce qui facilite la configuration et l'automatisation des workflows de développement. Il ne nécessite aucun serveur externe, ce qui le rend adapté à un projet académique ou de taille modeste.

#### 3.2 Étapes de Configuration

##### **Création d'un Fichier Workflow :**

- Un fichier de configuration nommé `.ci.yml` a été créé dans le dossier `.github/workflows` du projet.
- Ce fichier contient les instructions pour exécuter les étapes d'intégration continue.

##### **Définition des Étapes du Workflow :**

- Installation des Dépendances : Téléchargement des outils nécessaires pour exécuter les tests (comme Node.js).
- Exécution des Tests : Automatisation des tests unitaires pour vérifier la validité des fonctionnalités principales.
- Linting et Vérification de Syntaxe : Utilisation d'outils comme ESLint pour s'assurer que le code respecte les conventions.
- Génération de la Documentation : Automatisation de la génération de documentation (par exemple, à l'aide de JSDoc).

Notre fichier ci.yml :

```
// filepath: /c:/Users/bilar/Projet-Poker/.github/workflows/ci.yml
name: Node.js CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  build:
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [14, 15]

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Setup Node.js
        uses: actions/setup-node@v2
        with:
          node-version: ${ matrix.node-version }

      - name: Install dependencies
        run: npm install

      - name: Run tests
```

### 3.3 Tests Automatisés

Les tests ont été écrits pour vérifier les fonctionnalités principales de l'application, telles que :

- La validation des votes selon les règles choisies.
- La sauvegarde et la reprise des sessions à partir de fichiers JSON.
- Le rendu correct de l'interface utilisateur.

### 3.4 Résultats et Avantages

**Automatisation des Vérifications:** À chaque "push" ou "pull request", GitHub Actions exécute automatiquement les tests définis dans le workflow.

**Gain de Temps:** Les erreurs sont détectées rapidement, avant même le déploiement.  
**Qualité du Code:** Grâce au linting et aux tests automatisés, le code est maintenu propre et fonctionnel.

#### 4. Fonctionnalités et Utilisation

Notre application de planning poker offre une variété de fonctionnalités qui rendent l'expérience collaborative fluide et efficace. Nous permettons aux utilisateurs de s'inscrire et de se connecter pour garantir une gestion personnalisée de leur session. Ils peuvent estimer les tâches en équipe en utilisant différentes règles de vote adaptées aux besoins du projet. Une fonctionnalité de reprise de partie permet de charger des fichiers JSON pour poursuivre une session interrompue sans perte de données.

Avec une discussion en temps réel, nous favorisons la collaboration entre les participants avant chaque vote. Un chronomètre de 60 secondes limite le temps de réflexion, maintenant un rythme dynamique et évitant les blocages. De plus, nous proposons la sauvegarde de l'état actuel de la partie, permettant de reprendre une session à tout moment.

Pour enrichir l'expérience, nous avons inclus des outils supplémentaires comme l'affichage en temps réel des résultats, une gestion simplifiée des tâches dans le backlog, et un système de notifications pour informer les joueurs des actions importantes et des délais.

##### **Fonctionnalités Clés :**

**Inscription et Connexion :** Les utilisateurs peuvent s'inscrire et se connecter pour accéder à l'application.

**Estimation des Tâches :** Les utilisateurs peuvent estimer les tâches en équipe en utilisant différentes règles de vote.

**Reprise de Partie :** Reprenez une partie en chargeant un fichier JSON contenant les tâches et les noms des utilisateurs.

**Discussion en Temps Réel :** Discutez en temps réel pendant l'estimation des tâches.

**Chronomètre de Vote :** Un chronomètre de 60 secondes est inclus pour limiter le temps de réflexion et de vote des utilisateurs. Cela permet de maintenir un rythme fluide pour la session, en évitant les blocages. Si un utilisateur ne vote pas dans le temps imparti, un vote par défaut peut être attribué ou l'utilisateur est invité à voter rapidement.



**Sauvegarde de l'État de la Partie:** Sauvegardez l'état actuel de la partie pour la reprendre plus tard.

## 5. Structure du Projet

L'organisation du projet suit une structure modulaire pour simplifier le développement, la maintenance et la compréhension de l'application. Chaque dossier et fichier joue un rôle spécifique dans la réalisation des fonctionnalités.

- **Pages HTML :**

Les fichiers HTML définissent la structure des pages web de l'application. Ils contiennent les éléments nécessaires pour interagir avec les utilisateurs, comme les formulaires, les boutons, et les tableaux.

- `index.html` : Page d'accueil de l'application, permettant aux utilisateurs de s'inscrire, se connecter, ou charger une partie existante.
- `setup.html` : Page utilisée pour configurer les paramètres de la partie, comme le nombre de joueurs, leurs noms, et les règles du jeu.
- `game.html` : Page principale où les utilisateurs votent, visualisent les tâches, et participent activement à l'estimation.

- **Dossier js :**

Ce dossier regroupe tous les scripts JavaScript qui gèrent la logique métier de l'application. Les fichiers JavaScript permettent de rendre les pages interactives, de gérer les données, et d'assurer la communication entre les différents composants. Par exemple, ils implémentent les règles de vote, la gestion des chronomètres, et la mise à jour des données.

- **Dossier css :**

Ce dossier contient les feuilles de style CSS qui contrôlent l'apparence de l'application. Il inclut également les règles nécessaires pour rendre les pages responsives, en s'adaptant à différents types d'écrans (ordinateurs, tablettes, téléphones). L'utilisation de Bootstrap dans ce dossier accélère la création de designs cohérents et modernes.

- **Dossier json :**

Ce dossier est dédié au stockage et à la gestion des données structurées. Les fichiers JSON permettent de sauvegarder et de charger l'état de l'application.

- `json_progress.json` : Contient l'état actuel de la partie, y compris les tâches en cours, les votes, et les configurations.

- json\_results.json : Stocke les résultats finaux, notamment les estimations validées par les utilisateurs.

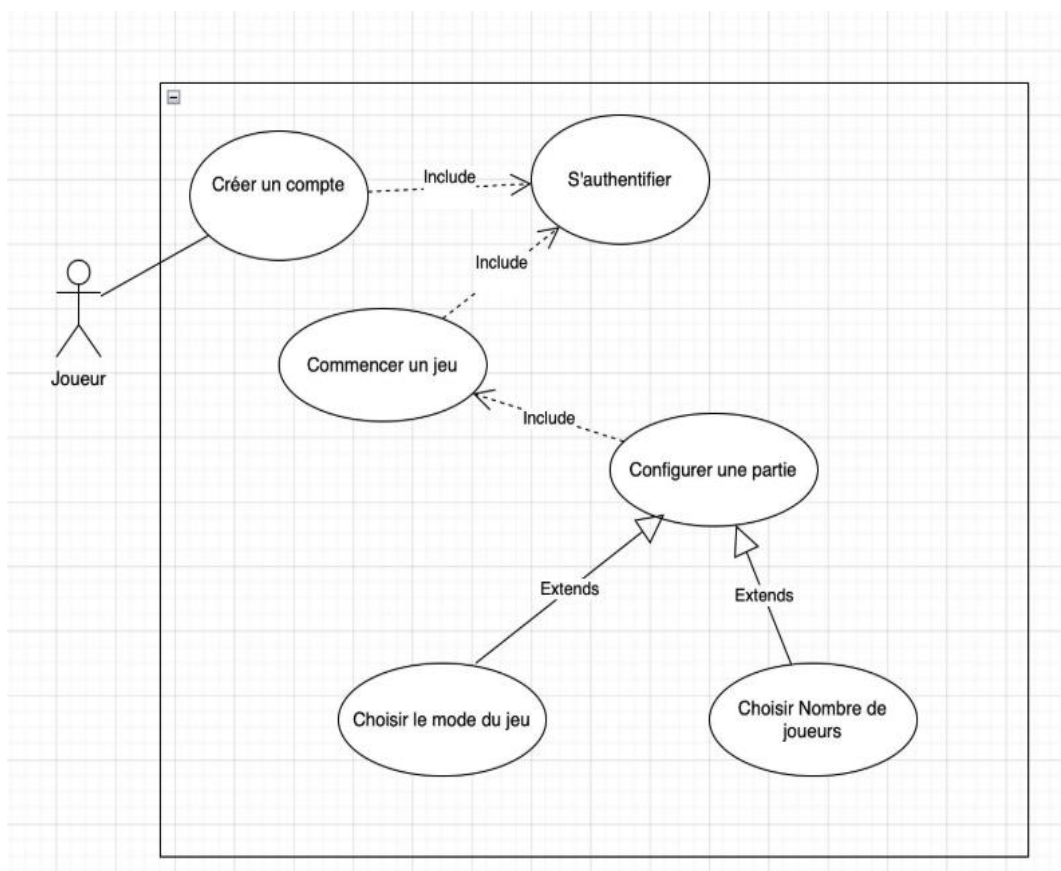
Cette structure organisée permet une séparation claire des responsabilités, facilitant le travail collaboratif et la lisibilité du code. Chaque dossier et fichier a été conçu pour jouer un rôle spécifique, garantissant ainsi une application modulaire et évolutive.

## 6. Diagrammes Explicatifs

Les diagrammes suivants illustrent les différents aspects du fonctionnement de l'application de planning poker. Ils permettent de comprendre la structure, les interactions entre les utilisateurs et le système, ainsi que le traitement des données :

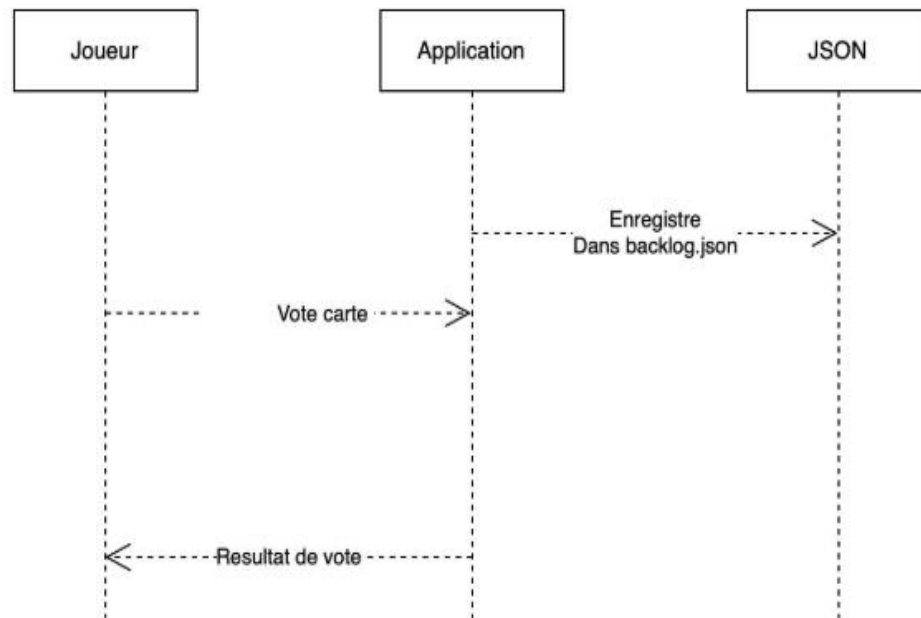
### 1. Diagramme de Cas d'Utilisation (Use Case) :

Représente les actions principales possibles pour les utilisateurs (joueurs) et leur interaction avec l'application, comme l'inscription, la configuration d'une partie, le vote, ou la sauvegarde.



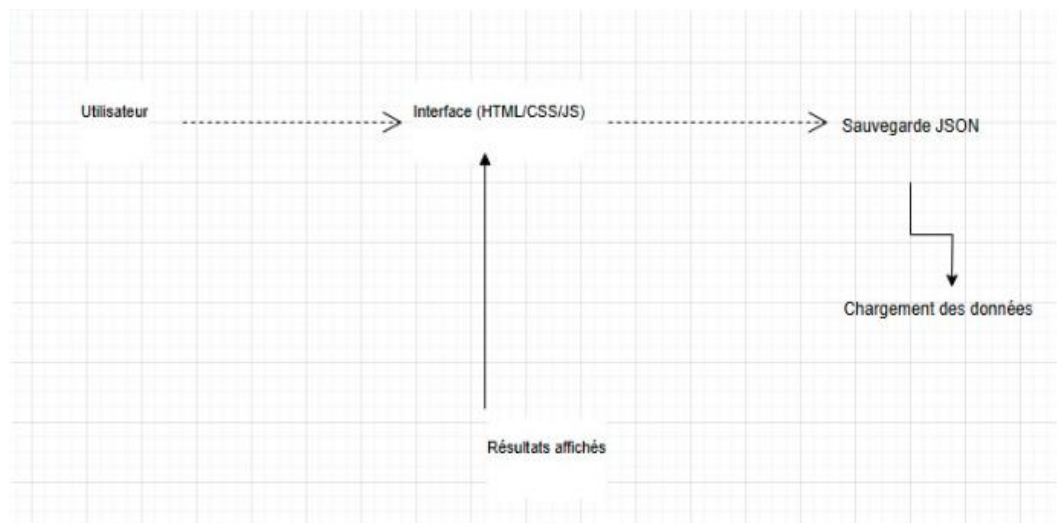
### 2. Diagramme de Séquence :

Détaille les étapes successives d'une session de vote, en montrant comment les tâches sont présentées, votées, et validées par les joueurs.



### 3. Diagramme de Flux de Données :

Illustre le parcours des données dans l'application, notamment depuis l'entrée des utilisateurs (votes) jusqu'à leur sauvegarde et reprise.



### 4. Diagramme d'Architecture :

Offre une vue d'ensemble des composants principaux de l'application (interface utilisateur, logique métier, gestion des données) et leur communication.



## 5. Diagramme de Classe :

Décrit les entités principales de l'application, telles que Joueur, Partie, et Tâche, avec leurs attributs et relations, afin de modéliser la logique du système.



## 7. Conclusion Générale

Le développement de notre application de planning poker a été une opportunité enrichissante de mettre en pratique les concepts des méthodologies agiles tout en explorant les technologies web modernes. Ce projet a permis de concevoir un outil collaboratif, intuitif et fonctionnel, répondant aux besoins d'estimation de tâches en équipe, un aspect essentiel des environnements de travail agiles.

Nous avons réussi à intégrer des fonctionnalités clés, telles que l'inscription et la gestion des utilisateurs, la sauvegarde et la reprise des parties, ainsi que l'estimation des tâches en respectant différentes règles de vote. L'ajout de fonctionnalités supplémentaires, comme un chronomètre de vote et un espace de discussion en temps réel, a permis d'améliorer l'expérience utilisateur. De plus, l'utilisation de GitHub Actions pour l'intégration continue a renforcé la qualité et la fiabilité du code.

Le choix des technologies, comme HTML, CSS, JavaScript, et JSON, s'est révélé judicieux pour atteindre nos objectifs de manière efficace. La structure modulaire de l'application garantit une flexibilité et une évolutivité pour d'éventuelles améliorations futures.

Bien que le projet réponde globalement aux exigences définies, certaines perspectives d'amélioration restent possibles, comme l'ajout de statistiques de vote ou la création d'une version mobile dédiée. Ces extensions permettraient de rendre l'application encore plus performante et adaptée à divers contextes.

En conclusion, ce projet a non seulement permis de développer une application fonctionnelle, mais également d'approfondir nos connaissances techniques et de renforcer notre capacité à travailler de manière collaborative dans un cadre agile.

## 8. Bibliographie et Références

- **NOTRE COURS** : *La Conception Agile de Projets Informatiques*.
- **MDN Web Docs** : Documentation pour HTML, CSS, et JavaScript.
- **Bootstrap** Documentation : Guide officiel pour l'utilisation de Bootstrap.
- **JSON.org** : Documentation officielle sur le format JSON.
- **GitHub Guides** : Guide pour l'utilisation de GitHub.
- **W3Schools** : Tutoriels pour HTML, CSS, et JavaScript.
- **Draw.io** : Outil pour créer des diagrammes UML et de flux.