

Rapport de projet

M1 informatique – Conception agile

Université Lumière Lyon 2

Sous la direction de

M. Valentin Lachand-Pascal

**Application Web – Planning Poker Scrum**

**Rapporteurs :**

Inas HAKKOU **N°étudiant** :2235375

Hind KHAYAT **N°étudiant** :5231307

**Remerciement :**

Cher Professeur,

Au terme de ce projet d'application de Planning Poker, nous tenions à exprimer notre sincère gratitude pour votre encadrement, vos conseils précieux et votre soutien tout au long de cette aventure. Votre expertise et vos enseignements nous ont été d'une aide précieuse, nous permettant de mieux appréhender les concepts, de relever les défis et de progresser dans nos compétences techniques.

Votre engagement et votre dévouement ont été une source d'inspiration pour nous, nous poussant à exceller et à dépasser nos limites dans la réalisation de ce projet. Votre implication a été déterminante pour notre apprentissage et notre réussite.

Nous vous remercions chaleureusement pour l'opportunité que vous nous avez offerte de travailler sur ce projet passionnant. Votre guidance a été cruciale pour notre développement professionnel et nous vous en sommes profondément reconnaissants.

**Table des matières :**

1. Introduction
2. Contexte du projet
3. Choix de Langage et Architecture
4. Langage de programmation choisi
5. Architecture de l'application
6. Implémentation des Fonctionnalités
7. Menu Principal
8. Mécanismes de Vote
9. Design Patterns Utilisés
10. Modèle-Vue-Contrôleur (MVC)
11. Singleton
12. Factory
13. Intégration Continue et Suivi de Projet
14. Mécanismes d'Intégration Continue
15. Tests Automatisés
16. Conclusion
17. Annexes
18. Captures d'écran pour l’application
19. **Introduction :**
20. **Contexte du projet :**

Le projet d'application de Planning Poker est né de la nécessité de fournir une solution numérique pour faciliter le processus de planification au sein d'équipes de développement logiciel. L'idée émane des défis rencontrés lors des sessions de planification traditionnelles, où il était souvent difficile d'atteindre un consensus sur l'estimation des tâches.

Dans un environnement agile, où la collaboration et la communication sont cruciales, le Planning Poker est une technique populaire pour estimer le travail à réaliser. Cependant, la nécessité de faire ces sessions de manière efficace a motivé la création d'une application dédiée.

L'objectif principal de cette application est de permettre aux équipes de réaliser des sessions de Planning Poker de manière fluide en présentiel tout en respectant les différentes règles de planification disponibles (mode strict, moyenne).

En mettant l'accent sur la simplicité d'utilisation, la flexibilité et la prise en charge des différents modes de jeu, cette application vise à fournir un outil convivial pour l'estimation des tâches dans un cadre agile, facilitant ainsi la prise de décision collective et la planification des sprints de développement.

1. **Choix de langage et architecture :**
2. **Langage de programmation choisi :**

Le choix des langages **HTML, CSS** et **JavaScript** pour le développement de l'application de Planning Poker a été fait en tenant compte de la nature web de l'outil. ***HTML (HyperText Markup Language)*** est utilisé pour la structure et la création des éléments de base de l'interface utilisateur, tandis que ***CSS (Cascading Style Sheets)*** est employé pour la mise en forme, le style et l'esthétique de l'application. ***JavaScript***, quant à lui, a été sélectionné pour la logique de programmation, les interactions dynamiques et les fonctionnalités interactives de l'interface. Cette combinaison de langages offre une base solide pour le développement d'une application web conviviale, permettant une expérience utilisateur fluide et réactive tout en assurant une compatibilité multiplateforme essentielle pour une utilisation à distance et sur différents dispositifs.

1. **Architecture de l’application :**

L'architecture de l'application de Planning Poker est basée sur une approche front-end, exploitant les langages HTML, CSS et JavaScript pour créer une interface utilisateur interactive et dynamique. Voici une description de l'architecture générale :

**HTML** : Utilisé pour la structure de base de l'application, définissant les éléments et leur agencement dans la page web.

**CSS** : Responsable de la présentation visuelle de l'interface, permettant le stylisme des éléments HTML pour une expérience utilisateur attrayante et cohérente.

**JavaScript** : Gère la logique de l'application, les interactions utilisateur et la communication avec le backend (si applicable). Il permet également de rendre l'interface interactive, réactive et dynamique, en répondant aux actions des utilisateurs.

1. **Implémentations des fonctionnalités :**
2. **Menu principale :**

Le menu principal a été développé pour permettre à l'utilisateur de créer son jeu de gestion de projets, il permet :

* Choisir le type de Vote, Le choix des différentes règles de planification (Strict, Moyenne).
* Importer le fichier Json dans lequel on a stocker les tâches à réaliser dans notre projet.
* Saisir le nombre des joueur (membre d’équipe).
* Attribuer les noms des joueurs.
* Sauvegarder les données pour commencer la partie du jeu.

1. **Mécanisme de vote :**

Les mécanismes de vote sont conçus pour permettre à chaque joueur de participer à l'estimation des fonctionnalités du backlog. Les joueurs ont la possibilité de voter en utilisant des cartes numériques ou symboliques pour estimer la difficulté des tâches.

L'application applique les règles de planification sélectionnées pour valider les estimations, que ce soit en mode strict ou en utilisant d'autres règles comme la moyenne.

Dans le cas du choix du mode ***Strict*** :

* Chaque participant doit choisir la même carte pour chaque tâche pendant la session de planification. Si des cartes différentes sont sélectionnées pour une tâche donnée, l'application invite les utilisateurs à recommencer le vote afin de parvenir à un consensus et attribuer une complexité identique à cette tâche.

Pour le choix du mode ***Moyenne*** :

* Les utilisateurs ont la liberté de choisir n'importe quelle carte pour chaque tâche. Par la suite, l'application calcule la moyenne des chiffres sélectionnés, offrant ainsi une estimation basée sur la valeur moyenne des estimations individuelles.

Pour les 2 modes :

* **La carte " ? "** : L'option de choisir la carte " ? " déclenche une demande de recommencer le vote pour la tâche en question, offrant ainsi la possibilité aux participants de revoir et de clarifier leurs estimations pour parvenir à un consensus.
* **La carte " coffee "** : est sélectionnée, l'application affiche une page de pause pour les utilisateurs, leur accordant le contrôle pour choisir la durée de la pause. Pendant ce temps, l'application conserve l'état précédent du jeu, prête à reprendre là où la session s'était interrompue après la pause, offrant une continuité fluide dans le processus de planification.

1. **Design Patterns Utilisés :**
2. **Modèle-Vue-Contrôleur (MVC) :**

Le modèle ***MVC*** a été largement adopté dans le développement de l'application de Planning Poker pour séparer les préoccupations liées à la logique métier, à l'interface utilisateur et au contrôle des actions. Le modèle représente la structure des données et la logique sous-jacente. La vue présente l'interface utilisateur permettant aux utilisateurs d'interagir avec l'application. Le contrôleur gère les entrées utilisateur, interagit avec le modèle et actualise la vue en conséquence, assurant ainsi une séparation claire des composants pour une maintenance et une évolutivité optimale.

1. **Singleton :**

Le design pattern ***Singleton*** a été implémenté dans certains modules de l'application de Planning Poker pour garantir qu'une seule instance d'une classe donnée est créée pendant le cycle de vie de l'application. Par exemple, cela peut être utilisé pour assurer qu'une unique instance de la classe de gestion du jeu est disponible pour coordonner les opérations de jeu.

1. **Factory :**

Le pattern ***Factory*** a été utilisé pour la création des différentes règles de planification disponibles dans l'application. Chaque type de règle de planification est considéré comme une "factory" capable de produire des instances d'objets de règles spécifiques en fonction du choix de l'utilisateur. Cela offre une flexibilité dans la gestion des différentes règles sans avoir à modifier directement le code principal de l'application.

1. **Intégration Continue et Suivi de Projet :**
2. **Mécanismes d'Intégration Continue :**

Pour garantir un développement fluide et continu, des mécanismes d'intégration continue ont été mis en place. L'utilisation de services comme ***GitHub*** permet d'automatiser divers processus tels que la vérification du code source, les tests automatiques et le déploiement continu. Ces outils assurent une collaboration transparente entre les membres de l'équipe de développement, facilitant ainsi la gestion efficace des versions du code et des changements apportés.

1. **Teste unitaire :**

L'absence de tests unitaires dans notre projet s'explique principalement par plusieurs facteurs, le plus important c’est ***la complexité de l'Environnement Front-End*** : JavaScript, en particulier dans un environnement front-end, peut parfois rendre la mise en place de tests unitaires plus complexe, surtout lorsqu'on travaille avec des interactions complexes, des dépendances externes ou des événements asynchrones. Cela peut rendre la création de tests unitaires plus laborieuse et nécessiter une expertise supplémentaire.

Malgré l'absence de tests unitaires formels, l'application a été développée en adoptant des pratiques de programmation robustes, en effectuant des tests manuels approfondis pour garantir la stabilité et le bon fonctionnement des fonctionnalités. Bien que les tests unitaires soient un élément important pour garantir la qualité du code, leur absence dans ce projet découle en grande partie des contraintes de temps, de ressources et de priorités liées à la nature spécifique du développement dans un contexte JavaScript pur.

1. **Conclusion :**

Ce projet d'application de Planning Poker a été une initiative visant à faciliter le processus de planification au sein des équipes agiles en offrant une plateforme conviviale pour l'estimation collaborative des tâches. Nous avons développé une application fonctionnelle permettant la participation en locale, tout en respectant différentes règles de planification telles que le mode strict et moyenne.

Les fonctionnalités clés, telles que le menu principal, la gestion du backlog, les mécanismes de vote et la sauvegarde/reprise de parties, ont été implémentées pour offrir une expérience utilisateur fluide et personnalisable.

En conclusion, ce projet a été une étape initiale réussie dans le développement d'une application de Planning Poker, offrant des fonctionnalités de base pour la planification agile. Les améliorations futures pourraient renforcer davantage l'utilité et l'efficacité de l'application, en répondant encore mieux aux besoins spécifiques des équipes travaillant dans un environnement agile de développement logiciel.

1. **Annexes :**
2. **Captures d’écrans pour l’application :**
3. **Page de départ :**

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

1. **Page de création de jeu :**

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

1. **Page de vote :**

**Une image contenant capture d’écran, texte, Logiciel multimédia, logiciel

Description générée automatiquement**

* Après appuyer sur le bouton Start voting et commencer le vote :

Une image contenant capture d’écran, texte, logiciel, Logiciel multimédia

Description générée automatiquement

1. **Page de pose coffee :**

Une image contenant capture d’écran, logiciel, Logiciel multimédia, Icône d’ordinateur

Description générée automatiquement

1. **Page d’affichage des résultats :**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement**