CAHIER DES CHARGES

Table des matières

Introduction

Le projet consiste à créer une plateforme en ligne dédiée au jeu de Go.
Le but est de proposer une expérience complète pour les amateurs et passionnés de Go, notamment avec des fonctionnalités comme la résolution de problèmes de Go (Tsumego), la soumission de problèmes, l'affichage de parties de Go, et des statistiques.

Cette plateforme vise à être bien plus qu'un simple site de jeu en ligne. Elle aspire à devenir un véritable hub communautaire pour les passionnés de Go, offrant un espace d'apprentissage, d'échange, et de partage autour de ce jeu fascinant.

En combinant des fonctionnalités interactives, éducatives et sociales, elle aspire à enrichir l'expérience de jeu de chacun de ses utilisateurs, tout en contribuant à l'essor de la communauté Go à travers le monde.

En résumé, la plateforme sera un outil indispensable pour les joueurs souhaitant progresser dans leur pratique du Go, qu'ils soient débutants désireux d'apprendre les bases ou joueurs avancés cherchant à perfectionner leur jeu. Grâce à ses fonctionnalités diversifiées et son approche centrée sur la communauté, elle vise à devenir un point de référence incontournable dans l'univers du Go en ligne.

Analyse client

I. Analyse du Besoin du Client

1.1. Contexte :

Le client, un club de jeu dédié principalement au jeu de Go, envisage de développer une plateforme en ligne pour offrir une expérience enrichissante aux amateurs et passionnés de ce jeu millénaire. L'objectif est de créer un environnement interactif où les joueurs peuvent non seulement jouer, mais aussi apprendre, s'entraîner, et interagir avec d'autres membres de la communauté.

II. Objectifs du Projet

2.1. Objectif Principal

Le client vise à créer une plateforme en ligne dédiée au jeu de Go, offrant une gamme complète de fonctionnalités pour les joueurs de tous niveaux. Cette plateforme doit être conviviale, éducative et centrée sur la communauté.

2.2. Objectifs Spécifiques

- Création d'une plateforme flexible et modulaire, permettant l'intégration de nouvelles fonctionnalités facilement.
- Offrir une expérience utilisateur intuitive et engageante, favorisant l'apprentissage et la pratique du jeu de Go.
- Fournir des outils d'apprentissage interactifs, tels que la résolution de problèmes (Tsumego) et la consultation de parties de Go.
- Permettre aux joueurs de soumettre leurs propres problèmes et de contribuer à l'enrichissement de la bibliothèque de contenu.
- Intégration de fonctionnalités sociales pour favoriser l'interaction entre les membres de la communauté, notamment des forums de discussion, des classements, et des outils de partage.
- Développement de statistiques détaillées pour permettre aux joueurs de suivre leur progression et de comparer leurs performances avec celles des autres membres.

2.3. Objectifs Fonctionnels

- Mise en place d'une interface conviviale pour jouer au Go en ligne, avec des fonctionnalités telles que la recherche de parties, le jeu contre l'ordinateur ou d'autres joueurs, et la sauvegarde des parties jouées.

- Intégration d'une bibliothèque de problèmes de Go de différentes difficultés, avec des solutions interactives pour aider les joueurs à progresser.

- Mise en œuvre d'un système de soumission de problèmes, permettant aux joueurs de partager leurs propres créations et de les voir ajoutées à la bibliothèque principale après validation par les administrateurs.

- Création de statistiques détaillées sur les parties jouées, les problèmes résolus, et la progression des joueurs au fil du temps.

2.4. Objectifs de release v1.0

- Mise en place de l'interface de jeu de base, permettant aux joueurs de jouer en ligne contre d'autres membres ou contre un ordinateur.

- Intégration d'une première série de problèmes de Go dans la bibliothèque, avec des solutions interactives.

- Développement des fonctionnalités de base pour la soumission de problèmes par les utilisateurs, avec un processus de validation par les administrateurs.

- Création des premières statistiques de jeu, telles que le nombre de parties jouées, le taux de victoire, et les problèmes résolus par les utilisateurs.

Choix des technologies :

I. Python

1.1. Description :

Python est un langage de programmation polyvalent et très populaire, idéal pour le développement de notre plateforme de jeu de Go. Il offre une syntaxe claire, une grande flexibilité, et une vaste gamme de bibliothèques, ce qui en fait un choix adapté pour ce projet.

1.2. Rôle dans le Projet :

Langage principal : Python sera le langage principal utilisé pour le développement de l'ensemble de la plateforme, y compris le back-end, les scripts de gestion de données, et les outils d'administration.

Flexibilité : La polyvalence de Python nous permettra d'implémenter efficacement toutes les fonctionnalités requises, des fonctionnalités de jeu aux outils de gestion des données et de l'interface utilisateur.

II. Django

2.1. Description :

Django est un framework web Python hautement productif, qui facilite le développement rapide d'applications web complexes.
Il offre une architecture MVC (Modèle-Vue-Contrôleur) robuste, un ORM (Object-Relational Mapping) intégré, et de nombreux modules prêts à l'emploi pour la sécurité, l'authentification, etc.

2.2. Rôle dans le Projet :

Développement du back-end : Django sera utilisé pour développer le back-end de notre plateforme de jeu de Go, en gérant les requêtes HTTP, l'authentification des utilisateurs, la logique métier, et l'accès à la base de données.

Productivité : Grâce à sa structure MVC et à ses fonctionnalités intégrées, Django nous permettra de développer rapidement des fonctionnalités complexes tout en maintenant un code propre et organisé.

III. SQL

3.1. Description :

SQL est un langage de programmation standard utilisé pour gérer et manipuler des bases de données relationnelles.
Il permet de créer, lire, mettre à jour et supprimer des données dans une base de données, ainsi que de définir et gérer la structure de la base de données elle-même.
SQL est largement utilisé dans l'industrie en raison de sa robustesse, de sa flexibilité et de sa capacité à gérer des volumes de données importants avec efficacité.

3.2. Rôle dans le Projet :

Stockage des données : SQL sera utilisé pour gérer la base de données relationnelle de la plateforme de Go, assurant le stockage des informations des utilisateurs, des parties jouées, des problèmes de Go soumis et des statistiques.

Intégration avec Django : L'ORM (Object-Relational Mapping) intégré de Django permettra une interaction fluide avec la base de données SQL, simplifiant les opérations CRUD (Create, Read, Update, Delete).

Gestion des requêtes complexes : SQL sera utilisé pour effectuer des requêtes complexes nécessaires à l'affichage des statistiques détaillées, à la gestion des utilisateurs et à l'extraction des données pour diverses fonctionnalités de la plateforme.

Performance et scalabilité : SQL, grâce à sa capacité à optimiser les requêtes et à gérer des transactions complexes, assurera que la plateforme puisse évoluer et maintenir de bonnes performances même avec une augmentation du nombre d'utilisateurs et de données.

IV. BeautfullSoup

4.1. Description :

BeautifulSoup est une bibliothèque Python utilisée pour extraire des données de fichiers HTML et XML. Elle crée un arbre d'analyse des documents analysés, ce qui permet une navigation et une modification faciles du contenu. BeautifulSoup est particulièrement utile pour le web scraping, où elle facilite l'extraction de données structurées à partir de pages web.

4.2. Rôle dans le Projet :

Scraping de contenu : BeautifulSoup sera utilisé pour extraire des données pertinentes de sites web externes, tels que des problèmes de Go ou des parties historiques, afin de les intégrer dans la base de données de la plateforme.

Automatisation des tâches : La bibliothèque simplifiera l'automatisation des tâches liées à la collecte de données, réduisant ainsi le besoin d'entrées manuelles et augmentant l'efficacité.

Enrichissement de la base de données : Les données collectées via BeautifulSoup enrichiront la bibliothèque de problèmes de Go et les archives de parties, offrant aux utilisateurs un contenu diversifié et de haute qualité.

Mise à jour régulière des données : BeautifulSoup facilitera la mise à jour régulière du contenu en extrayant automatiquement les nouveaux problèmes de Go et parties de sources fiables, assurant que la plateforme reste à jour et pertinente pour les utilisateurs.

V. React

5.1. Description :

React est une bibliothèque JavaScript populaire pour la création d'interfaces utilisateur dynamiques et réactives.
Développée par Facebook, elle permet de construire des composants réutilisables qui facilitent la gestion de l'état et des interactions de l'application.
React se distingue par son utilisation du Virtual DOM, qui optimise les mises à jour de l'interface pour une performance améliorée.

5.2. Rôle dans le Projet :

Développement de l'interface utilisateur : React sera utilisé pour développer l'interface utilisateur de la plateforme de jeu de Go.
Les composants réutilisables de React permettront de créer des fonctionnalités spécifiques au jeu, comme le rendu du plateau de jeu, l'affichage des pierres et la mise à jour des scores en temps réel.
La gestion efficace de l'état par React facilitera la synchronisation des mouvements des joueurs et les mises à jour instantanées de l'interface utilisateur.

Gestion de projet :

I. Spécifications :

Durée : 10 semaines
Taille de l'Équipe : 4 personnes
Évaluation du temps de travail : 10 semaines

II. Déroulement du projet :

1. Constitution de l'équipe
2. Lecture et compréhension du brief
3. Recherches sur le fonctionnement des plateformes de jeu en ligne et des fonctionnalités spécifiques au jeu de Go
4. Définition des tâches préliminaires au développement (mise en place et veille)
5. Création d'un environnement de travail sur un outil de gestion de projet (comme Trello)
6. Mise en place des documents de support/livrables (cahier des charges, documentation)
7. Définition de l'architecture et du fonctionnement de la plateforme
8. Choix des technologies et des outils nécessaires au développement
9. Découpage en tâches selon les fonctionnalités spécifiques du jeu de Go
10. Écriture du code en suivant une approche itérative et agile
11. Tests unitaires et fonctionnels pour chaque fonctionnalité développée
12. Documentation complète du code source

III. Liste fonctionnelle :

1. Interface Utilisateur Conviviale :

    - Conception d'une interface utilisateur intuitive et conviviale pour permettre aux utilisateurs de naviguer facilement sur la plateforme.

    - Utilisation de principes de conception UX/UI pour garantir une expérience utilisateur optimale.

2. Résolution de Problèmes de Go (Tsumego) :

    - Mise en place d'une bibliothèque de problèmes de difficultés variées, comprenant des problèmes déjà enregistrés dans la base de données de l'application.

    - Affichage clair de chaque problème avec une représentation visuelle de la disposition des pierres sur le plateau.

    - Possibilité pour les membres de cliquer sur un bouton pour voir la solution du problème.

    - Transformation du format JSON des problèmes en SGF standard si nécessaire pour une intégration harmonieuse.

3. Soumission de Problèmes :

    - Permettre aux membres inscrits de soumettre de nouveaux problèmes avec leurs solutions via un formulaire structuré.

    - Ajout d'une note explicative pour décrire le problème lors de la soumission.

    - Validation des problèmes soumis par les administrateurs avant leur ajout à la bibliothèque principale.

    - Attribution de différents rôles aux membres pour faciliter la gestion des contributions : joueur, éditeur, administrateur.

4. Affichage de Parties Complètes :

    - Affichage de parties complètes de Go avec des statistiques relatives à chaque partie.

    - Possibilité pour les utilisateurs de rechercher des parties en fonction des noms des joueurs, de l'année de la partie, ou de l'ouverture (séquence des premiers coups).

5. Statistiques :

    - Développement d'une fonctionnalité pour afficher des statistiques sur les parties de Go et l'utilisation de la plateforme.

    - Collecte de données telles que la fréquence des ouvertures, les séquences de mouvements courantes, les joueurs les plus référencés, etc.

    - Affichage des statistiques pertinentes pour les utilisateurs inscrits sur la plateforme et les administrateurs.

6. Authentification et Gestion des Utilisateurs :

    - Mise en place d'un système d'authentification sécurisé pour les utilisateurs avec différents niveaux d'accès : joueur, éditeur, administrateur.

    - Gestion des comptes utilisateurs, y compris la création, la modification et la suppression de comptes.

    - Contrôle d'accès pour limiter les actions en fonction du rôle de l'utilisateur.

7. Stockage des Données dans une Base SQL :

    - Utilisation de SQL pour stocker les données de manière structurée et relationnelle : Les bases de données relationnelles SQL sont bien adaptées pour gérer des relations complexes entre les données et assurent l'intégrité référentielle.

8. Modélisation des données adaptée aux besoins de l'application :

    - Utilisation de schémas relationnels pour représenter les informations, avec des tables, des colonnes et des relations définies pour structurer les données de manière optimale.Utilisation de l'Environnement Django

    - Création de l'application web en utilisant Django, un framework web Python robuste et évolutif.

    - Intégration des fonctionnalités de Django telles que l'ORM (Object-Relational Mapping) pour simplifier l'interaction avec la base de données.

IV. Outils :

Figma (Conception UI/UX)

Description :

Figma est un outil de conception collaboratif en ligne qui permet aux équipes de créer des interfaces utilisateur et des expériences utilisateur de manière interactive et en temps réel. Il offre des fonctionnalités de prototypage, de création de wireframes, et de design collaboratif, permettant à plusieurs utilisateurs de travailler simultanément sur le même projet.

Rôle dans le Projet :

Création et itération des designs d'interface utilisateur.
Collaboration en temps réel avec les membres de l'équipe de conception et les parties prenantes.
Partage de prototypes interactifs pour recueillir des feedbacks et effectuer des tests utilisateurs.

Trello (Gestion des tâches)

Description :

Trello est une plateforme de gestion de projet qui utilise des tableaux, des listes et des cartes pour organiser les tâches et collaborer en équipe. Chaque carte représente une tâche, et vous pouvez les organiser dans des listes selon leur statut (à faire, en cours, terminé).

Rôle dans le Projet :

Gestion efficace des tâches et des étapes du projet.
Suivi visuel des avancements individuels et de l'équipe.
Facilitation de la collaboration et de la communication entre les membres de l'équipe.

Github (Partage de code en mise en commun)

Description :

Github est une plateforme de développement collaboratif qui permet aux équipes de partager et de collaborer sur des projets logiciels. Il offre des fonctionnalités telles que le contrôle de version avec Git, le suivi des problèmes, la gestion des branches, et la fusion de code.

Rôle dans le projet :

Hébergement centralisé du code source du projet.
Suivi détaillé des modifications apportées au code via les commits.

Google Doc :

Description :

Google Docs est un service de traitement de texte en ligne qui permet la création et la modification de documents collaboratifs en temps réel. Il offre des fonctionnalités telles que le partage de documents, les commentaires en direct et la gestion des versions.

Rôle dans le projet :

Google Docs sera utilisé pour la collaboration sur la documentation du projet, y compris la rédaction du cahier de spécifications, la documentation des décisions prises lors des réunions, et la rédaction de tout autre document nécessaire au projet. En permettant à plusieurs membres de l'équipe de travailler simultanément sur les documents, Google Docs facilite la coordination et la communication, tout en assurant la cohérence et la traçabilité des informations.

V. Évaluation du temps de travail :

Mise en place du projet Django : 2 jours
- Installation de l'environnement de développement Django.
- Configuration initiale du projet, y compris la création de l'arborescence des répertoires, des fichiers de configuration et des fichiers statiques.
- Création et configuration de la base de données.

Conception de l'interface utilisateur conviviale : 5 jours
- Conception et développement d'une interface utilisateur intuitive et conviviale pour la plateforme de jeu de Go.
- Réalisation de wireframes et de maquettes.
- Création des modèles Django pour représenter les données à afficher dans l'interface.
- Intégration des modèles avec les vues Django.
- Création de templates HTML/CSS/JavaScript pour rendre l'interface utilisateur.

Mise en place de la bibliothèque de problèmes de Go (Tsumego) : 4 jours
- Création des modèles Django pour représenter les problèmes de Go.
- Mise en place des vues Django pour afficher les problèmes.
- Développement des fonctionnalités permettant d'ajouter, de supprimer et de modifier des problèmes dans la bibliothèque.
- Intégration des tests unitaires pour garantir le bon fonctionnement de la bibliothèque.

Affichage visuel des problèmes de Go : 2 jours
- Développement d'une représentation visuelle des problèmes de Go sur le plateau.
- Intégration des bibliothèques JavaScript (comme D3.js) pour créer des diagrammes interactifs des problèmes de Go.
- Liaison des données des problèmes avec l'interface utilisateur Django.

Fonctionnalité de visualisation de la solution des problèmes : 3 jours
- Ajout d'une fonctionnalité permettant aux utilisateurs de voir la solution d'un problème après avoir tenté de le résoudre.
- Création de vues Django pour afficher la solution du problème.
- Intégration de boutons interactifs dans l'interface utilisateur pour permettre aux utilisateurs de demander la solution.

Implémentation de la soumission de problèmes : 4 jours
- Développement d'un formulaire de soumission pour permettre aux utilisateurs d'ajouter de nouveaux problèmes de Go avec leurs solutions.
- Création des vues Django pour gérer la soumission des problèmes.
- Intégration des fonctionnalités de validation des données soumises par les utilisateurs.

Gestion des soumissions de problèmes : 3 jours

- Création d'un processus de validation des problèmes soumis par les utilisateurs par les administrateurs de la plateforme.
- Développement de vues Django pour permettre aux administrateurs de valider ou de rejeter les problèmes soumis.
- Intégration des notifications par e-mail pour informer les utilisateurs de l'état de leur soumission.

Mise en place de la fonctionnalité de recherche de parties de Go : 4 jours

- Développement de vues Django pour gérer les recherches de parties de Go.
- Intégration de fonctionnalités de recherche avancées, y compris la recherche par joueur, année, ouverture, etc.
- Création d'index pour améliorer les performances de recherche dans la base de données.

Affichage détaillé des parties de Go : 3 jours

- Création de vues Django pour afficher les détails des parties de Go.
- Intégration des modèles Django pour récupérer les données des parties de Go à afficher.
- Développement de templates HTML/CSS pour présenter les détails des parties de manière conviviale et intuitive.

Collecte et affichage de statistiques sur les parties de Go : 6 jours

- Développement d'algorithmes pour collecter et afficher des statistiques sur les parties de Go.
- Création de vues Django pour afficher les statistiques agrégées.
- Intégration de graphiques et de tableaux pour représenter visuellement les statistiques.

Implémentation de l'authentification utilisateur : 4 jours

- Mise en place d'un système d'authentification sécurisé pour permettre aux utilisateurs de se connecter et d'accéder à leurs fonctionnalités personnalisées.
- Création de vues Django pour gérer l'inscription, la connexion et la déconnexion des utilisateurs.

Gestion des comptes utilisateurs : 3 jours

- Développement de vues Django pour permettre aux utilisateurs de gérer leurs comptes (modification de mot de passe, mise à jour des informations, etc.).
- Intégration de fonctionnalités de validation des données utilisateur.

Configuration et utilisation d’une base de données SQL pour le stockage des données : 4 jours

- Installation et configuration d’une base de donées SQL comme base de données pour stocker les données de manière flexible et évolutive.

Développement de l'application web avec Django : 6 jours

- Création de vues Django pour gérer le flux de données et les interactions utilisateur.
- Intégration des modèles Django avec les vues pour récupérer et manipuler les données de la base de données.
- Développement de templates HTML/CSS/JavaScript pour rendre l'interface utilisateur.
- Ajout de fonctionnalités pour gérer les sessions utilisateur.

Tests unitaires et fonctionnels : 2 jours

- Écriture de tests unitaires pour chaque fonctionnalité développée pour garantir le bon fonctionnement de l'application.
- Mise en place de tests fonctionnels pour simuler les interactions utilisateur et vérifier la réponse attendue de l'application.
- Création de jeux de données de test pour couvrir différents scénarios d'utilisation.

Documentation : 2 jours

- Rédaction de documentation détaillée pour chaque composant de l'application, y compris les modèles de données, les vues, les formulaires et les fonctionnalités.
- Création de guides d'utilisation pour les utilisateurs finaux.
- Documentation des API et des endpoints pour les développeurs tiers.

VI. Recettage :

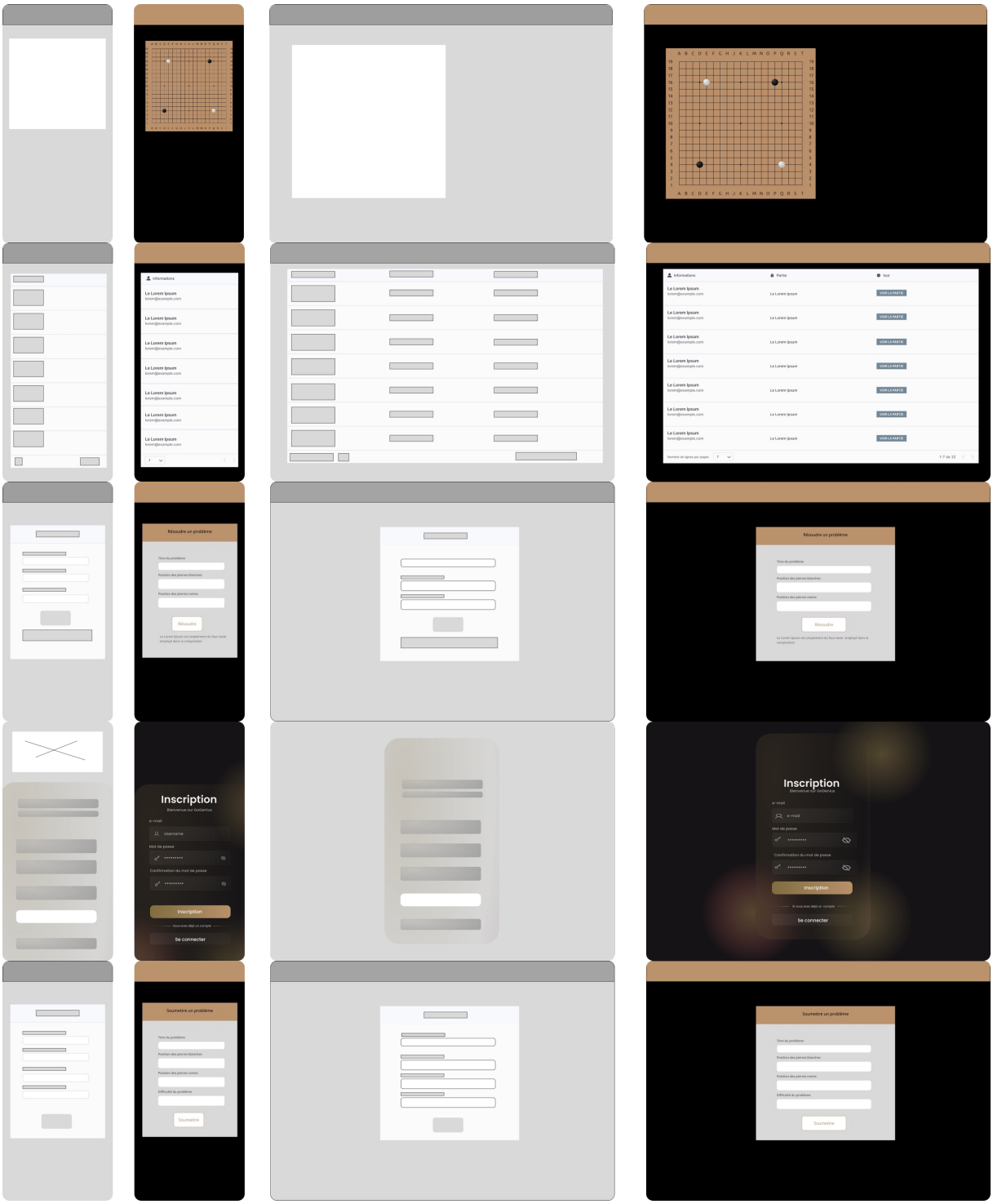
Tâches	Status (opérationnelle / non opérationnelle)
Conception d'une interface utilisateur conviviale	Opérationnelle
Mise en place d'une bibliothèque de problèmes de Go (Tsumego)	Opérationnelle
Affichage visuel des problèmes de Go	Opérationnelle
Fonctionnalité de visualisation de la solution des problèmes	Opérationnelle
Implémentation de la soumission de problèmes	Opérationnelle
Gestion des soumissions de problèmes	Opérationnelle
Mise en place de la fonctionnalité de recherche de parties de Go	Opérationnelle
Affichage détaillé des parties de Go	Opérationnelle
Implémentation de l'authentification utilisateur	Opérationnelle
Gestion des comptes utilisateurs	Opérationnelle
Configuration et utilisation d'une base de données SQL pour le stockage des données	Opérationnelle
Développement de l'application web avec Django	Opérationnelle
Tests unitaires et fonctionnels	Opérationnelle

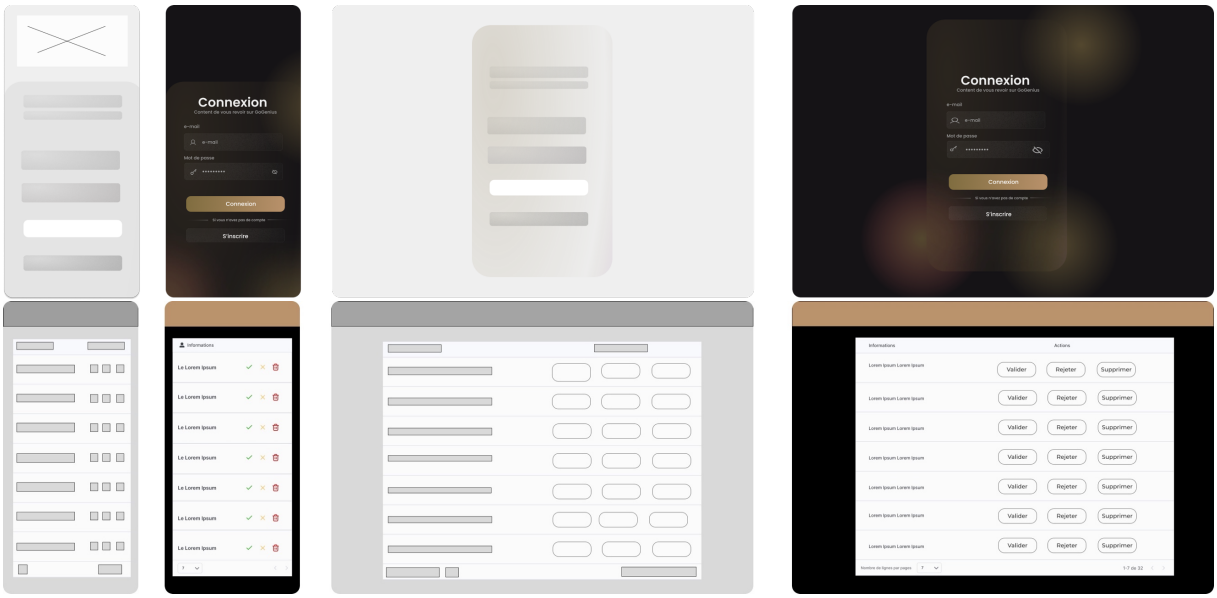
VII. Améliorations pour une version ultérieure :

- Développement des fonctionnalités non développées du MVP
- Sécurisation de la partie authentification
- Ajout d'un chat sur les parties
- Ajout d'une interface pour les évènements locaux autour du jeu de GO

Annexes :

Maquettes :




Connexion
Connexion
Connexion
Connexion
S'inscrire
Informations
Le Lorem Ipsum
Informations
Actions
Valider
Rejeter
Supprimer