



UFR Sciences et Techniques

Département Informatique

Master Génie de l'Informatique Logicielle

Rapport De Projet



Développement d'une Plateforme de
Mots-Croisés (Cruci■Web)



Année Universitaire 2024-2025

Table des matières

Introduction	1
I. Objectif de l'application.....	1
II. Architecture de l'application.....	2
III. La sécurité	6
IV. Ergonomie	6

Introduction

Le projet présenté dans ce rapport consiste en la création d'une application web interactive dédiée à la gestion et à la résolution de grilles de mots croisés. Ce projet s'inscrit dans le cadre de nos apprentissages académiques, avec pour objectif de mettre en pratique les concepts clés abordés en cours. Il intègre notamment l'utilisation de HTML/CSS et JavaScript pour la manipulation du DOM, les requêtes AJAX, ainsi que PHP pour la gestion des sessions, la manipulation des bases de données et l'application de l'architecture MVC.

I. Objectif de l'application

L'objectif principal de l'application est de permettre aux utilisateurs de consulter, résoudre et gérer des grilles de mots croisés de manière intuitive et dynamique. L'application a trois (03) types d'utilisateurs :

1. Utilisateurs anonymes :

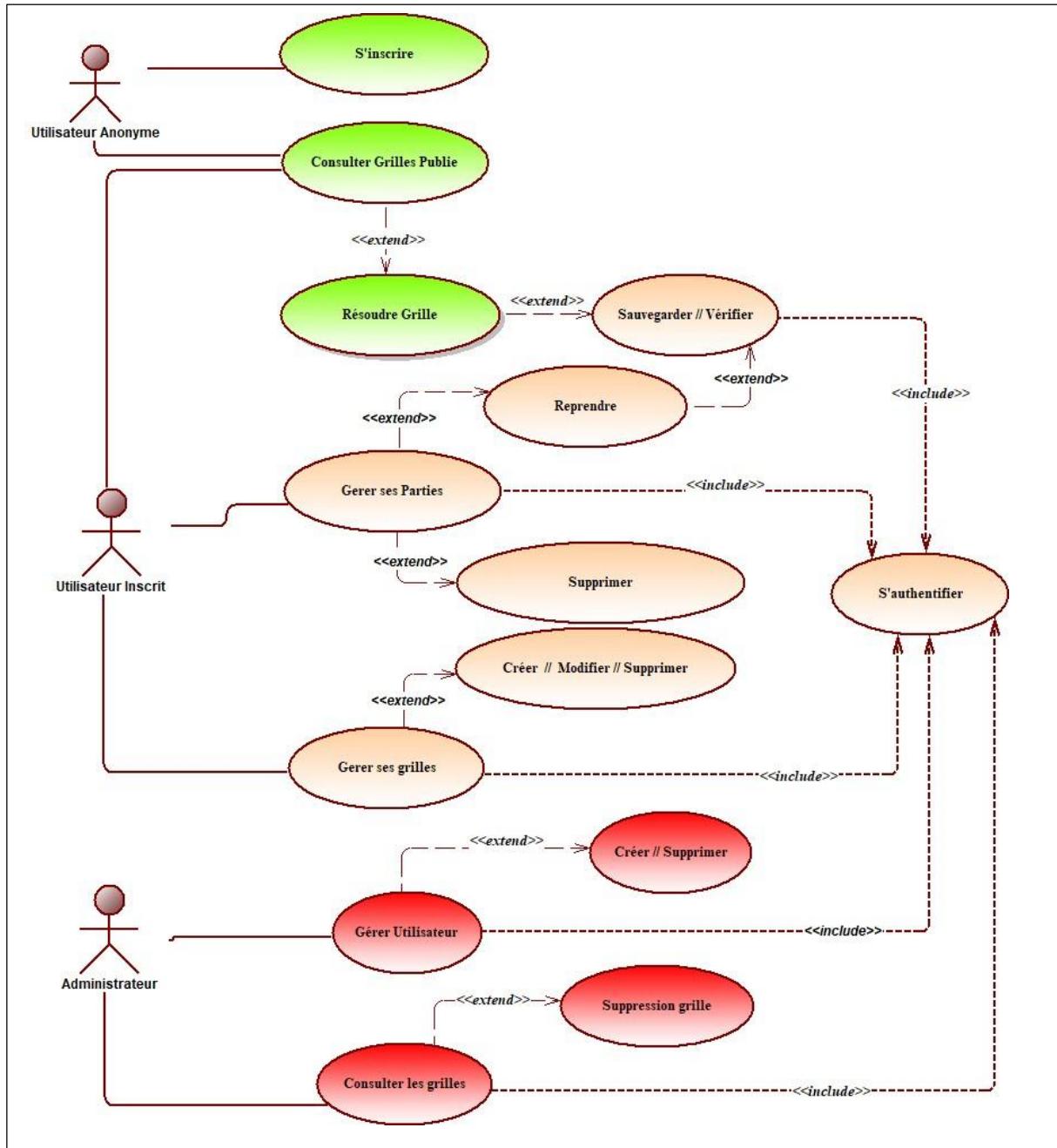
- ✓ Ils peuvent consulter l'ensemble des grilles publiées et tenter de les résoudre.
- ✓ Ils ne peuvent pas sauvegarder leurs grilles en cours de résolution.
- ✓ Un utilisateur anonyme peut s'inscrire sur la plate-forme pour accéder à des fonctionnalités supplémentaires.

2. Utilisateurs inscrits :

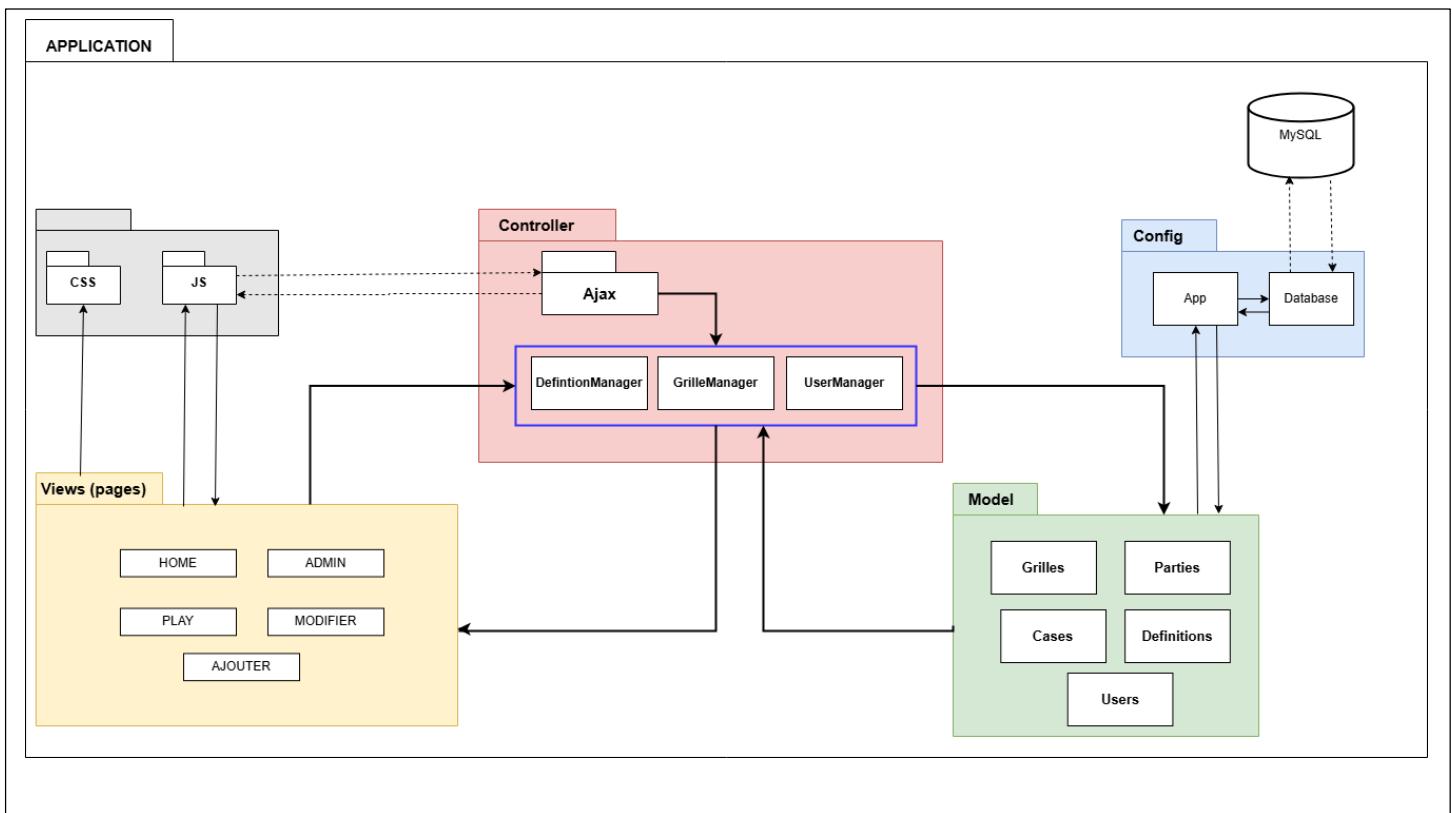
- ✓ Ils peuvent créer et publier de nouvelles grilles de mots croisés.
- ✓ Ils peuvent résoudre des grilles de mots croisés et, contrairement aux utilisateurs anonymes, ils peuvent sauvegarder leurs grilles en cours de résolution pour les reprendre ultérieurement.

3. Administrateur de la plate-forme :

- ✓ Il peut créer et supprimer des utilisateurs.
- ✓ Il peut également supprimer des grilles.



II. Architecture de l'application



L'architecture repose sur quatre couches principales : le Modèle, la Vue, le Contrôleur, et une couche de configuration et base de données. Chaque couche joue un rôle précis dans le fonctionnement global de l'application.

Controller

Les contrôleurs sont responsables de la gestion des requêtes utilisateur et de la communication entre la vue et le modèle. Dans ce projet nous avons trois contrôleurs :

1. **GrilleManager** : Cette classe est responsable de la gestion des grilles ainsi que des parties. Elle intervient à plusieurs niveaux de l'application :

- ✓ **Interaction avec la vue** : La classe peut être appelée directement par les vues ou via une requête AJAX, par exemple lorsqu'il s'agit de dessiner une grille ou de charger une table contenant la liste des grilles (publiques ou privées) ainsi que la liste parties associées à un utilisateur.
- ✓ **Interaction avec le modèle** : La classe joue un rôle d'intermédiaire essentiel entre la vue et le modèle, elle est en relation avec trois (03) classes du model (Grilles, Partie et Cases). Elle effectue les opérations suivantes :

- **Création, mise à jour, suppression et sauvegarde:** Cette classe interagit avec le modèle pour créer de nouvelles grilles, mettre à jour les grilles existantes, sauvegarder une grille ou encore les supprimer de la base de données.
- **Récupération des informations :** Elle est également responsable de la récupération des données concernant les grilles afin de proposer un contenu aux vues.
- ✓ **Vérification des grilles :** Cette classe intègre également la logique de vérification d'une grille avant une demande de sauvegarde dans la base de données, s'assurant que les solutions proposées sont conformes à la grille.

2. **DefinitionManager** est chargée de la gestion des définitions associées à une grille.

- ✓ **Interaction avec la vue**
 - Affichage des définitions : Lorsqu'une vue souhaite afficher les définitions d'une grille, cette classe est appelée. Par exemple, sur la page "play", les définitions sont chargées pour être présentées aux utilisateurs.
 - Chargement pour modification : Lorsqu'une vue a besoin de charger une définition pour modification, comme sur la page "modifier", *DefinitionManager* est responsable de fournir les données nécessaires.
- ✓ **Interaction avec le modèle**
 - Gestion des définitions : *DefinitionManager* interagit avec la classe *Definition* du modèle pour ajouter modifié ou supprimer les définitions associé à une grille existantes dans la base de données.

3. **UserManager** est chargé de la gestion des utilisateurs associés au système

- ✓ **Interaction avec la vue**
 - La classe est appelée pour fournir la liste des utilisateurs, par exemple lors de la gestion des comptes dans l'interface administrateur.
 - La vue peut également solliciter cette classe de manière indirecte via une requête AJAX. Par exemple, lors du chargement du formulaire d'inscription ou d'authentification, *UserManager* est utilisé pour valider les données saisies ou fournir les informations nécessaires.
- ✓ **Interaction avec le modèle**
 - Gestion des utilisateurs : Cette classe interagit avec la classe *Users* du modèle pour effectuer diverses opérations,

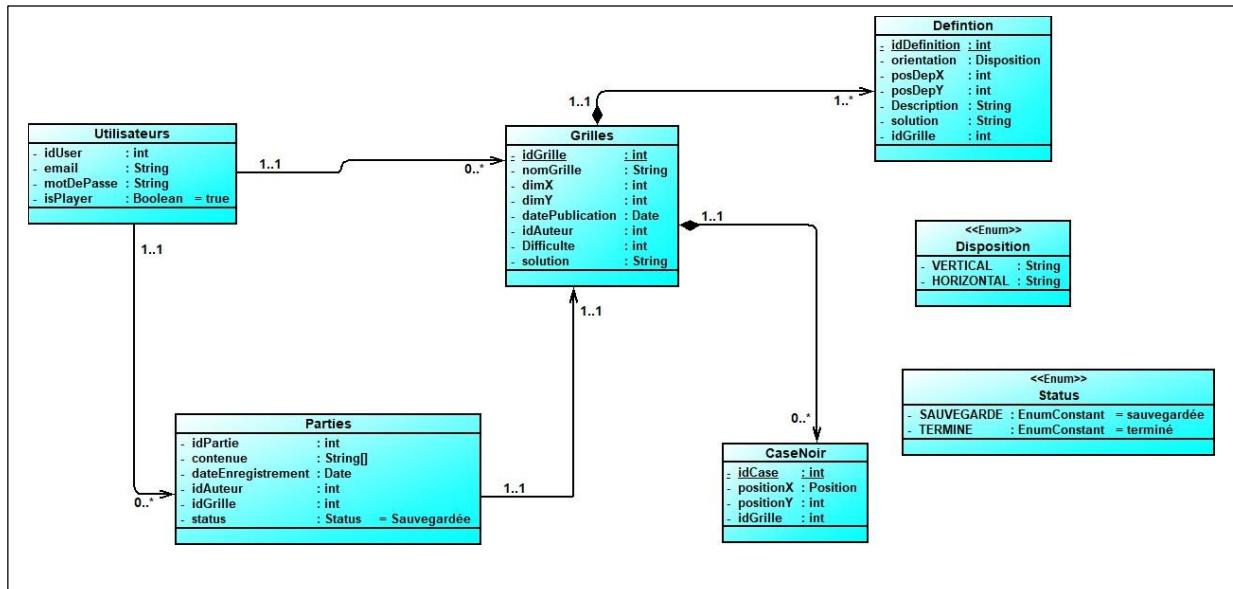
telles que l'authentification des utilisateurs, l'ajout d'un nouvel utilisateur, ou encore la suppression d'un utilisateur.

4. **Ajax** : Contient les API permettant d'envoyer et de recevoir des données entre le client (navigateur) et le serveur sans recharger la page. Comme lors du chargement de la grille, le chargement du Dashboard mais aussi le chargement des tableaux pour afficher la liste grilles, parties ou users.

Model

Le modèle constitue la couche qui gère les données et leur logique métier. Il est responsable de l'interaction avec la base de données et des opérations sur les données. Les classes du modèle dans ce projet sont les suivantes

Classes	Descriptions
Grilles	Gère les données relatives aux grilles : ajout, modification, suppression, récupération des grilles publiques/privées.
Parties	Gère les parties des utilisateurs : sauvegarde, chargement, suppression des parties.
Cases	Représente et gère l'état des cases Noires d'une grille : ajout, modification et suppression
Definitions	Gère les définitions des grilles : ajout, modification, suppression, récupération des définitions associées à une grille.
Users	Gère les utilisateurs : ajout, authentification, suppression des comptes.



Vue

La vue est responsable de l'affichage et de l'interaction avec les utilisateurs. Elle présente les données provenant des contrôleurs sous forme compréhensible et interactive.

III. La sécurité

La sécurité est un aspect crucial de l'application, garantissant la protection des données des utilisateurs et l'intégrité du système. Plusieurs mesures ont été mises en place pour assurer cette sécurité :

- ✓ **Hashage des Mots de Passe** : Les mots de passe des utilisateurs sont stockés dans la base de données sous forme de hash sécurisé utilisant des fonctions telles que `password_hash()` en PHP, empêchant ainsi leur récupération en clair en cas de fuite de données.
- ✓ **Validation et Filtrage des Entrées** : Toutes les données entrées par les utilisateurs sont validées et filtrées pour éviter les injections SQL et les attaques XSS. L'utilisation de requêtes préparées (PDO avec des prepared statements) empêche les injections SQL.
- ✓ **Protection contre les Attaques XSS** : Les données affichées dans les vues sont échappées correctement en utilisant `htmlspecialchars()`, empêchant l'exécution de scripts malveillants insérés par les utilisateurs.

IV. Ergonomie

L'ergonomie de l'application est pensée pour offrir une expérience utilisateur fluide, intuitive et agréable.

1. Interface Utilisateur

- ✓ **Navigation Intuitive** : Les menus et les boutons sont organisés de manière logique, permettant aux utilisateurs de naviguer facilement entre les différentes sections (accueil, jeu, création de grilles, liste des grilles ou partie ...etc).
- ✓ **Feedback Utilisateur** : Des messages de confirmation et des notifications sont affichés lors des actions importantes (sauvegarde d'une grille, création d'une grille, connexion), offrant un retour immédiat sur les actions effectuées.

2. Interactivité avec AJAX

- ✓ **Chargements Dynamiques** : L'utilisation d'AJAX permet de charger et de mettre à jour des éléments de la page sans recharger complètement la vue, rendant l'application plus réactive.

- ✓ Formulaires Interactifs : Les formulaires d'inscription et de connexion utilisent des validations en temps réel via AJAX, améliorant ainsi l'expérience utilisateur en fournissant des retours instantanés sur les données saisies.