

***Fais-moi un dessin***  
**Document d'architecture logicielle**

**Version 2.3**

# Historique des révisions

Date	Version	Description	Auteur
2020-01-28	1.0	Première rédaction	Florence Cimon-Paquet, Geneviève Laroche, Annie Rochette
2020-02-04	1.1	Ajout des diagrammes des cas d'utilisation, des diagrammes de séquence et du diagramme de déploiement	Florence Cimon-Paquet, Geneviève Laroche, Annie Rochette
2020-02-06	1.2	Ajout de la section 7	Florence Cimon-Paquet
2020-02-07	1.3	Ajout des diagrammes de paquetages	Maxime Bernier
2020-02-07	1.4	Ajout de la section 2	Geneviève Laroche
2020-02-07	1.5	Révision du document d'architecture	Florence Cimon-Paquet
2020-04-09	2.0	Modifications de la section 5 selon les commentaires de la correction de l'appel d'offre	Geneviève Laroche
2020-04-09	2.1	Modifications du diagramme de déploiement selon la correction de l'appel d'offres	Florence Cimon-Paquet
2020-04-12	2.2	Modifications de la section 7 selon la correction de l'appel d'offres	Florence Cimon-Paquet
2020-04-13	2.3	Modification des diagrammes de paquetages	Maxime Bernier

# Table des matières

<b>1. Introduction</b>	4
<b>2. Objectifs et contraintes architecturaux</b>	4
<b>3. Vue des cas d'utilisation</b>	5
<b>4. Vue logique</b>	7
4.1 Vue de haut niveau	8
4.1.1 Client lourd	<b>Error! Bookmark not defined.</b>
4.1.2 Serveur	<b>Error! Bookmark not defined.</b>
4.2 Vue détaillée	10
4.2.1 Client lourd	10
4.2.2 Serveur	16
4.2.3 Client léger	21
<b>5. Vue des processus</b>	25
<b>6. Vue de déploiement</b>	29
<b>7. Taille et performance</b>	29

# Document d'architecture logicielle

## 1. Introduction

Ce document présent décrit l'architecture logicielle du logiciel *Fais-moi un dessin*. Tous les diagrammes d'architecture vont devoir respecter le langage UML.

Dans cet artefact, il sera question d'expliquer les objectifs et les contraintes architecturales, de présenter les cas d'utilisation ainsi que la vue logique en expliquant le rôle de chaque paquetage, la vue des processus en démontrant leurs interactions, la vue de déploiement en montrant la configuration des différentes composantes matérielles physiques et, finalement, les caractéristiques de la taille et de la performance qui pourraient avoir un impact sur l'architecture du logiciel.

## 2. Objectifs et contraintes architecturaux

### 2.1 Performance

Le serveur devra être en mesure de supporter plusieurs parties simultanément.

### 2.2 Sécurité

Afin de garder confidentielles les informations personnelles de nos utilisateurs, les mots de passe devront être encryptés.

### 2.3 Réutilisation

Pour le client lourd, il ne sera pas nécessaire de développer des outils pour dessiner. En effet, l'application réutilisera certains des outils de l'application *PolyPaint* dont le crayon et pinceau, l'efface de trait, l'efface par segment, le sélecteur de pointe, le sélecteur de taille de pointe et le sélecteur de couleur .

### 2.4 Langage de développement

Le client lourd sera développé en C#/WPF et le client léger en java. Pour le serveur et la base de donnée, afin de respecter l'échéancier, les développeurs utiliseront des technologies qu'ils connaissent déjà, soit Node.js et MongoDB.

### 3. Vue des cas d'utilisation

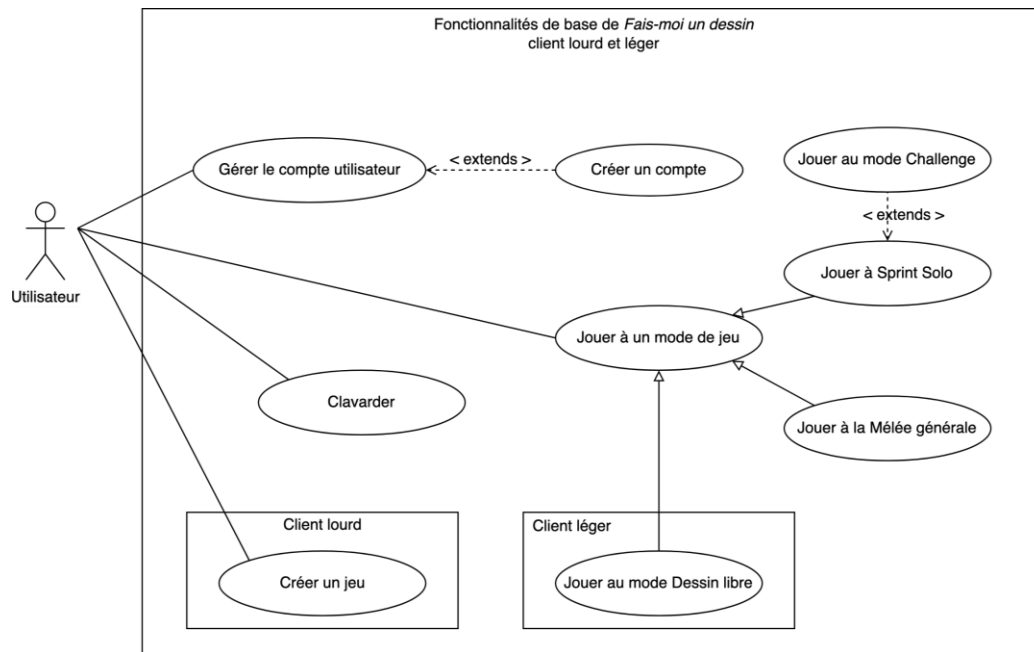


Figure 3.1 Diagramme de cas d'utilisation pour le comportement général du logiciel

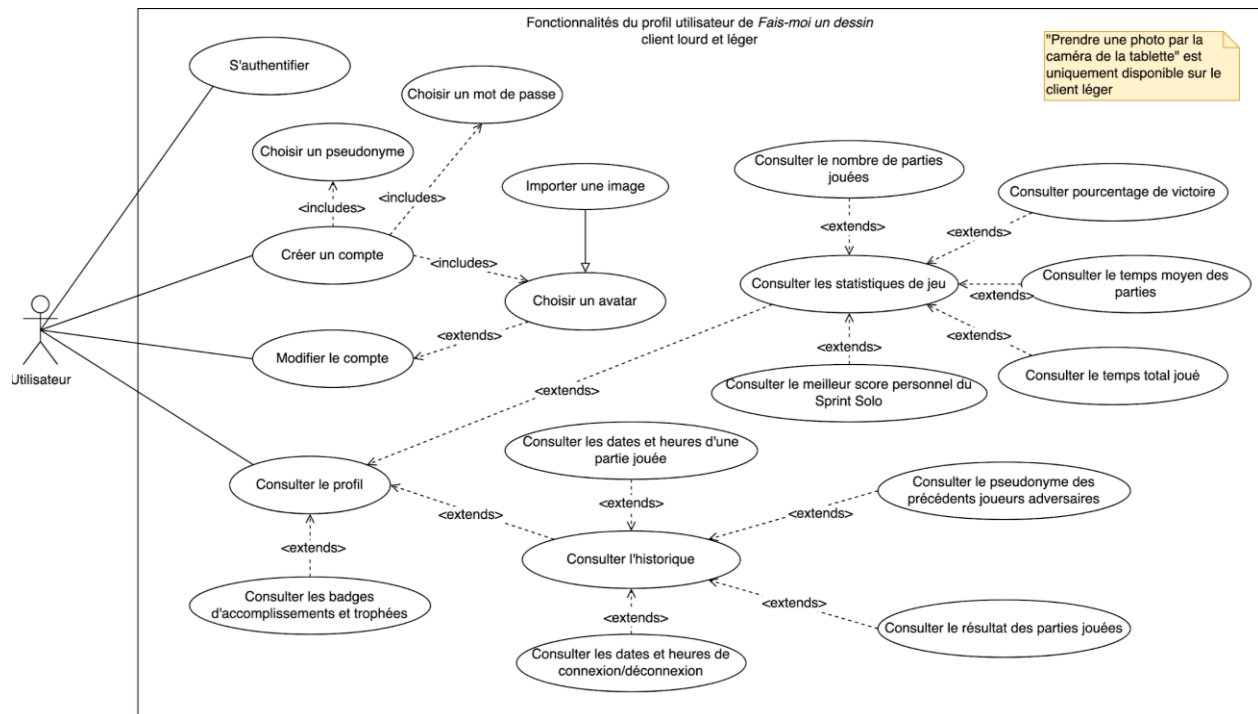


Figure 3.2 Diagramme de cas d'utilisation pour le comportement du profil utilisateur du logiciel

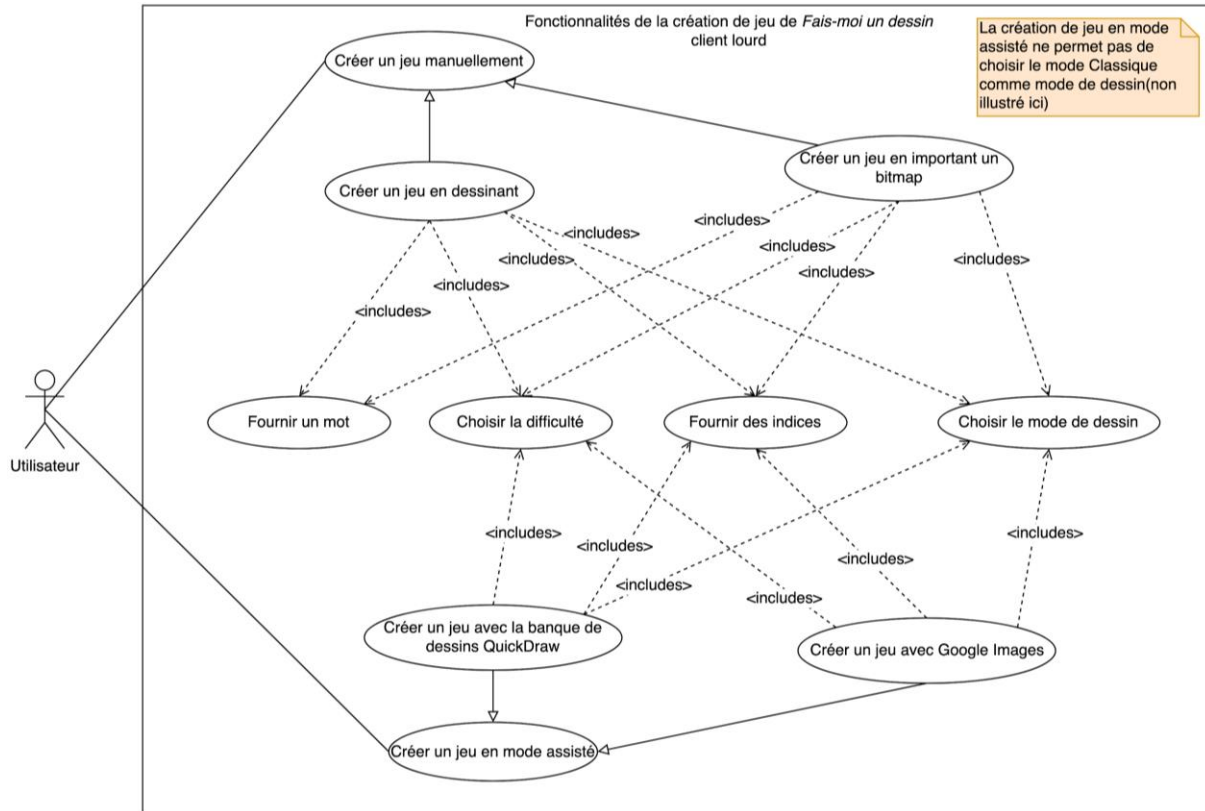


Figure 3.3 Diagramme de cas d'utilisation pour le comportement de création d'un jeu du logiciel

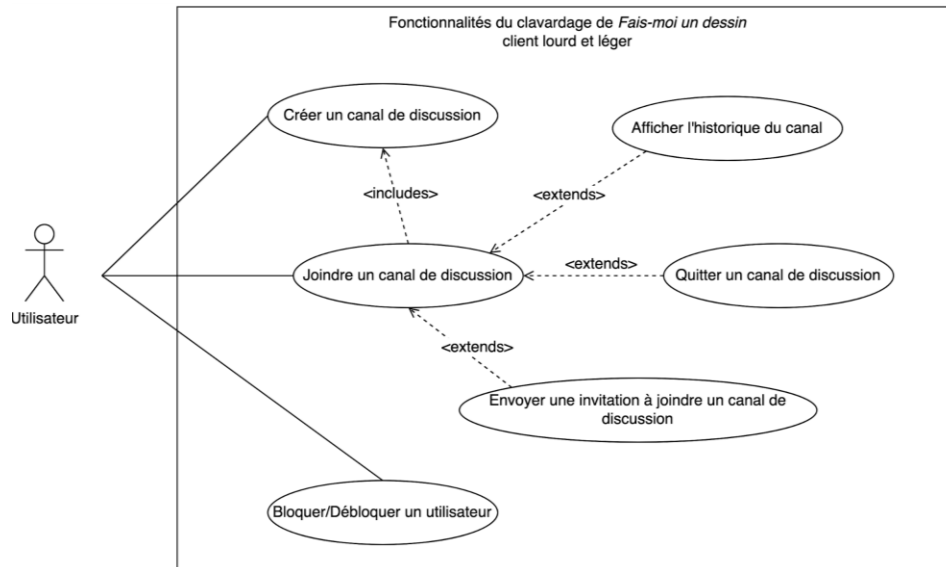


Figure 3.4 Diagramme de cas d'utilisation pour le comportement du clavardage du logiciel

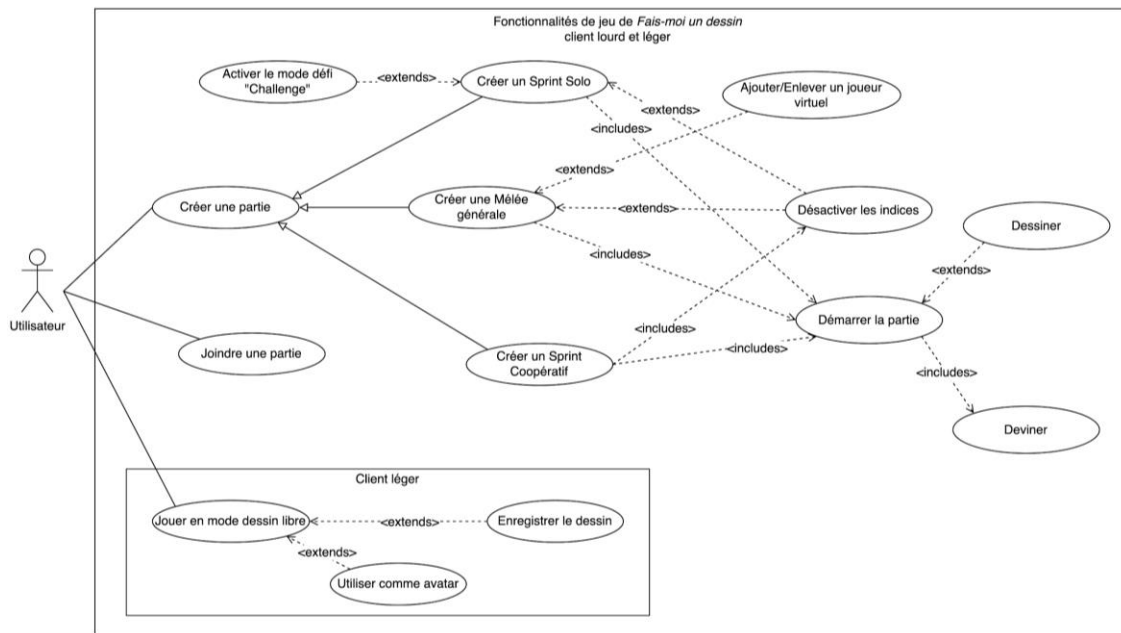


Figure 3.6 Diagramme de cas d'utilisation pour le comportement d'une partie ou d'un jeu du logiciel

## 4. Vue logique

#### 4.1 Vue de haut niveau

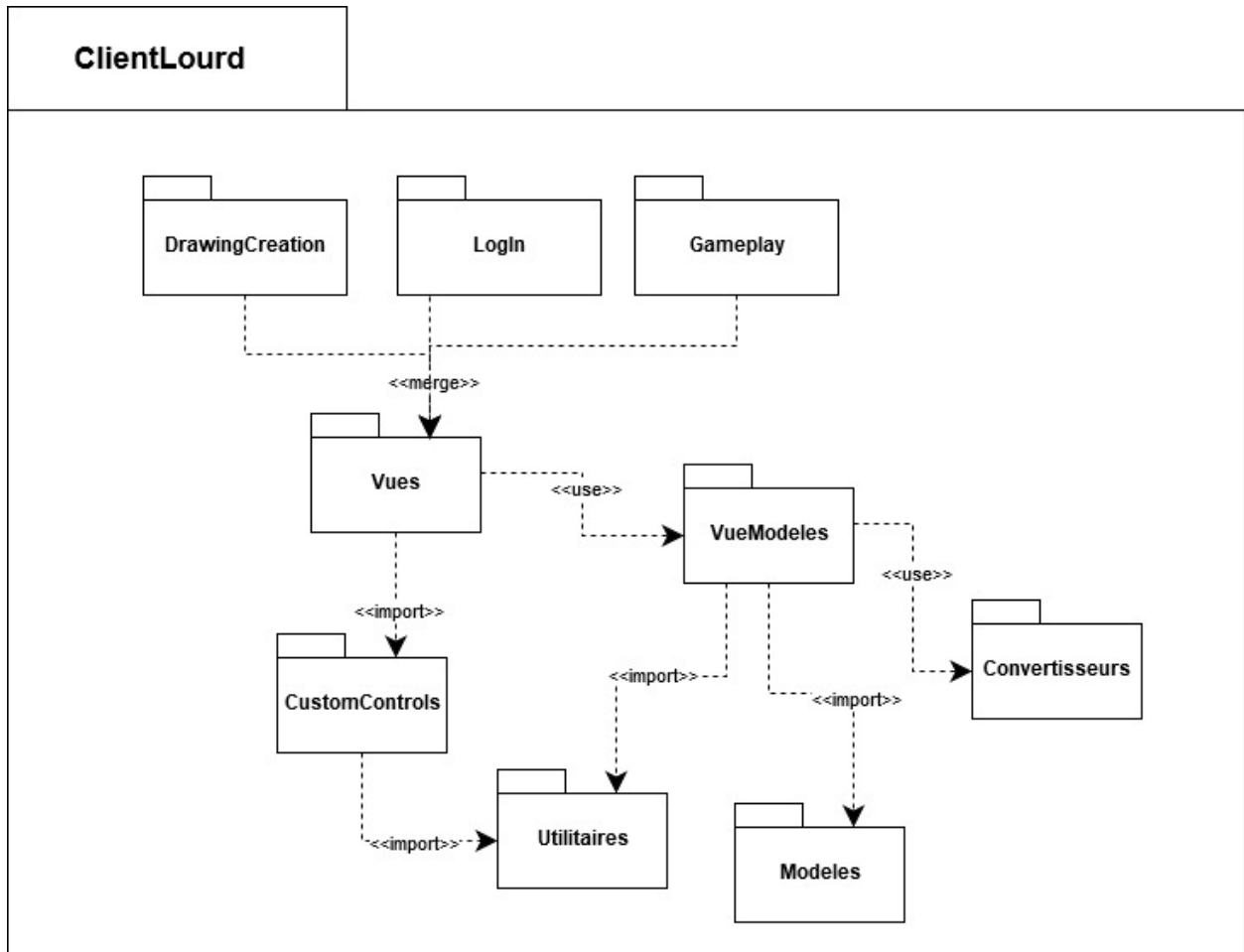


Figure 4.1.1: Diagramme de paquetage de haut niveau du client lourd



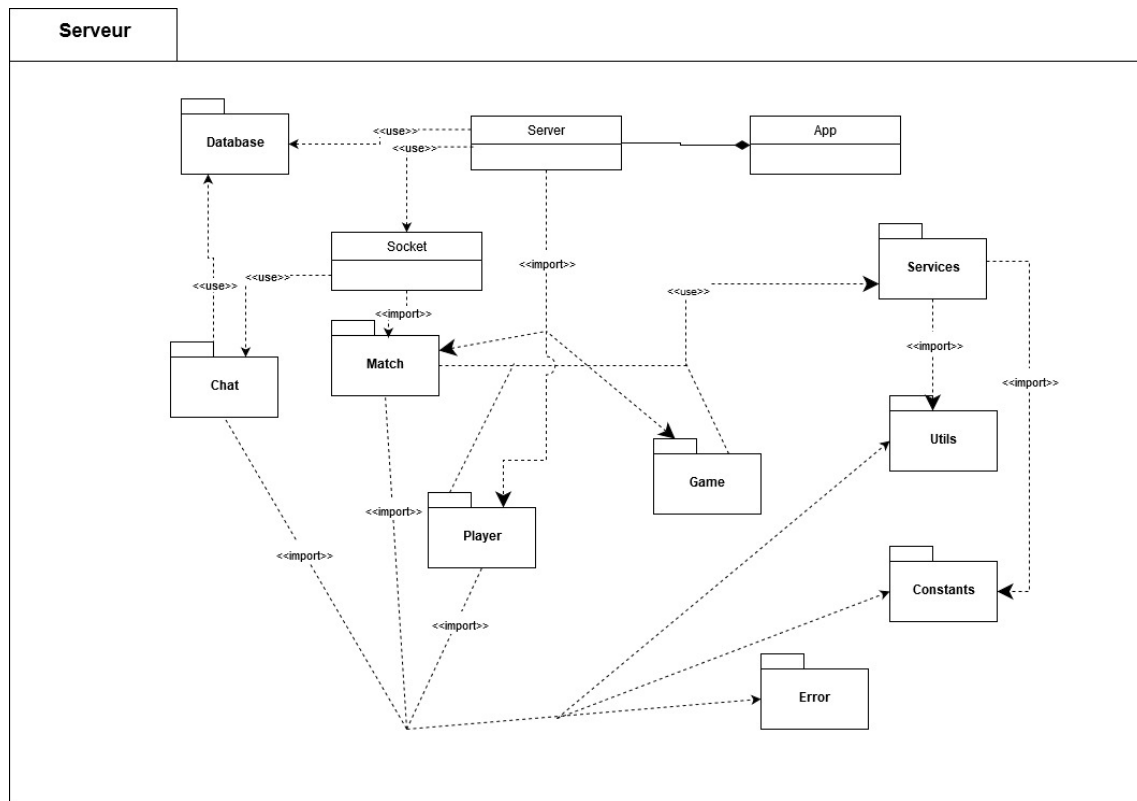


Figure 4.1.2: Diagramme de paquetage de haut niveau du serveur

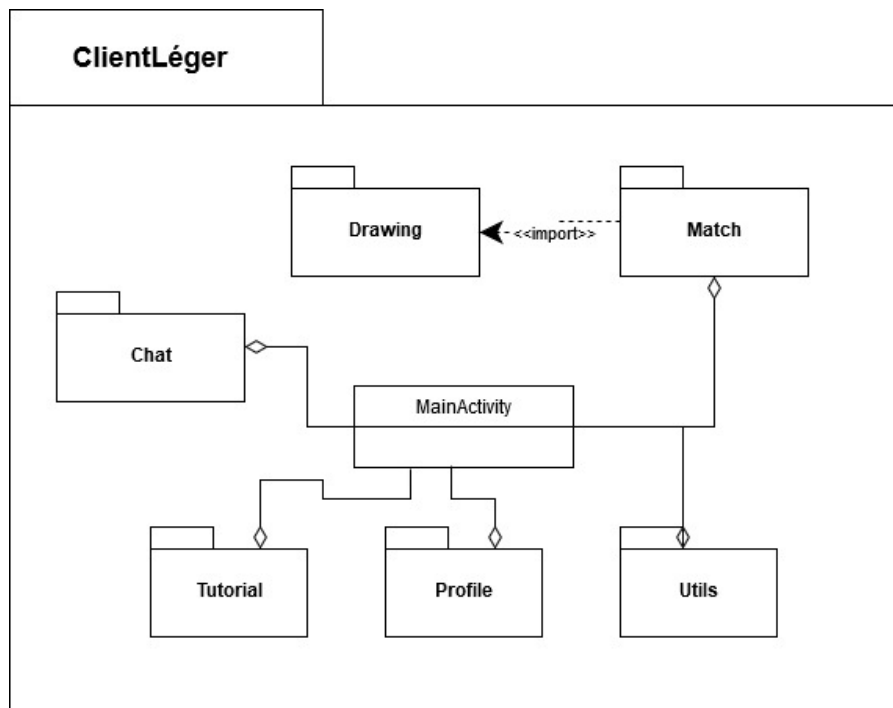


Figure 4.1.3: Diagramme de paquetage de haut niveau du client léger

## 4.2 Vue détaillée

### 4.2.1 Client lourd

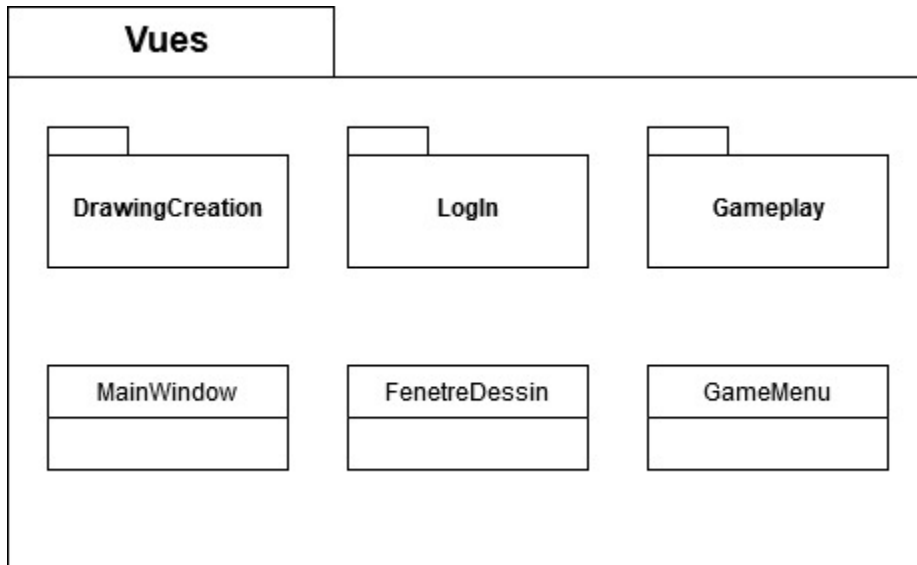


Figure 4.2.1: Diagramme de paquetage Vues du client lourd

Tableau 4.2.1 : Description de la responsabilité du paquetage Vues

Vues
L'objectif de ce paquetage est d'implémenter toutes les vues que l'utilisateur verra lors de l'utilisation de l'application.

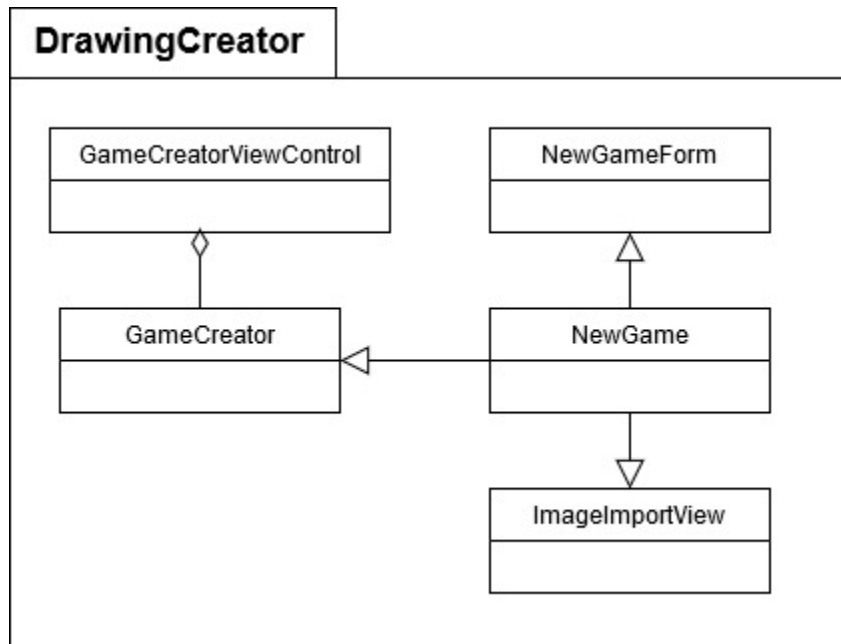


Figure 4.2.2: Diagramme de classes du paquetage DrawingCreator du client lourd

Tableau 4.2.2 : Description de la responsabilité du paquetage DrawingCreator

DrawingCreator
L'objectif de ce paquetage est d'implémenter une vue pour que l'utilisateur puisse créer un jeu.

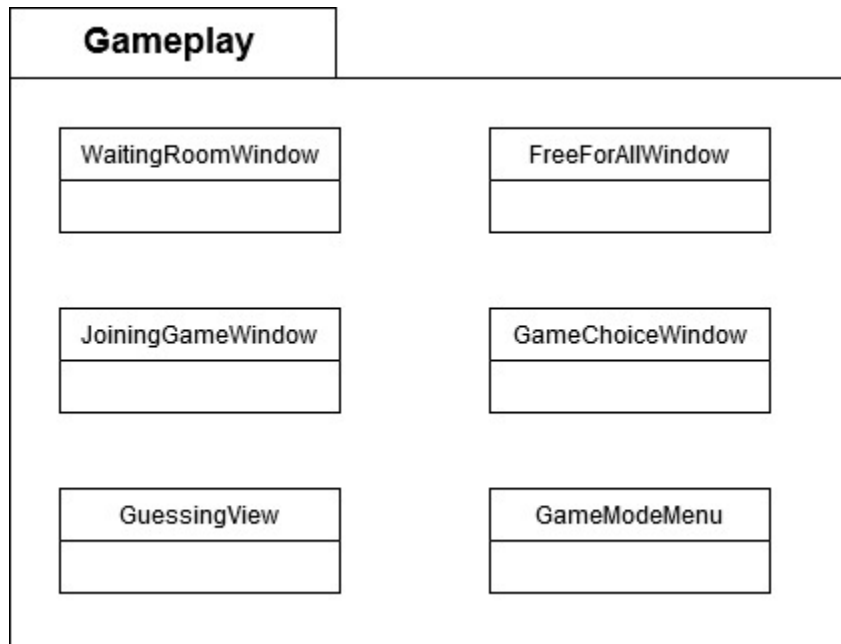


Figure 4.2.3: Diagramme de classes du paquetage Gameplay du client lourd

Tableau 4.2.3 : Description de la responsabilité du paquetage Gameplay

Gameplay
L'objectif de ce paquetage est d'implémenter des vues pour assurer le bon déroulement d'une partie.

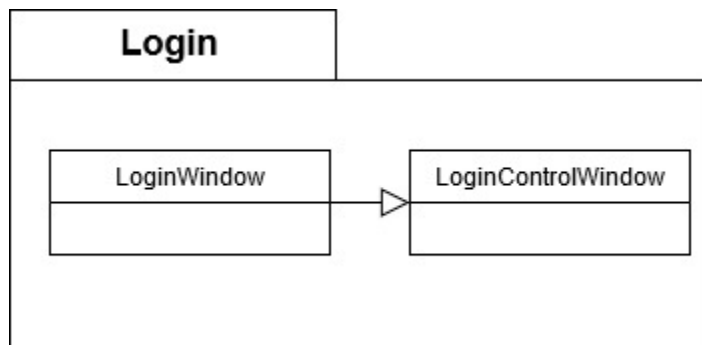


Figure 4.2.4: Diagramme de classes du paquetage Login du client lourd

Tableau 4.2.4 : Description de la responsabilité du paquetage Login

Login
L'objectif de ce paquetage est d'implémenter une vue pour que l'utilisateur puisse se connecter.

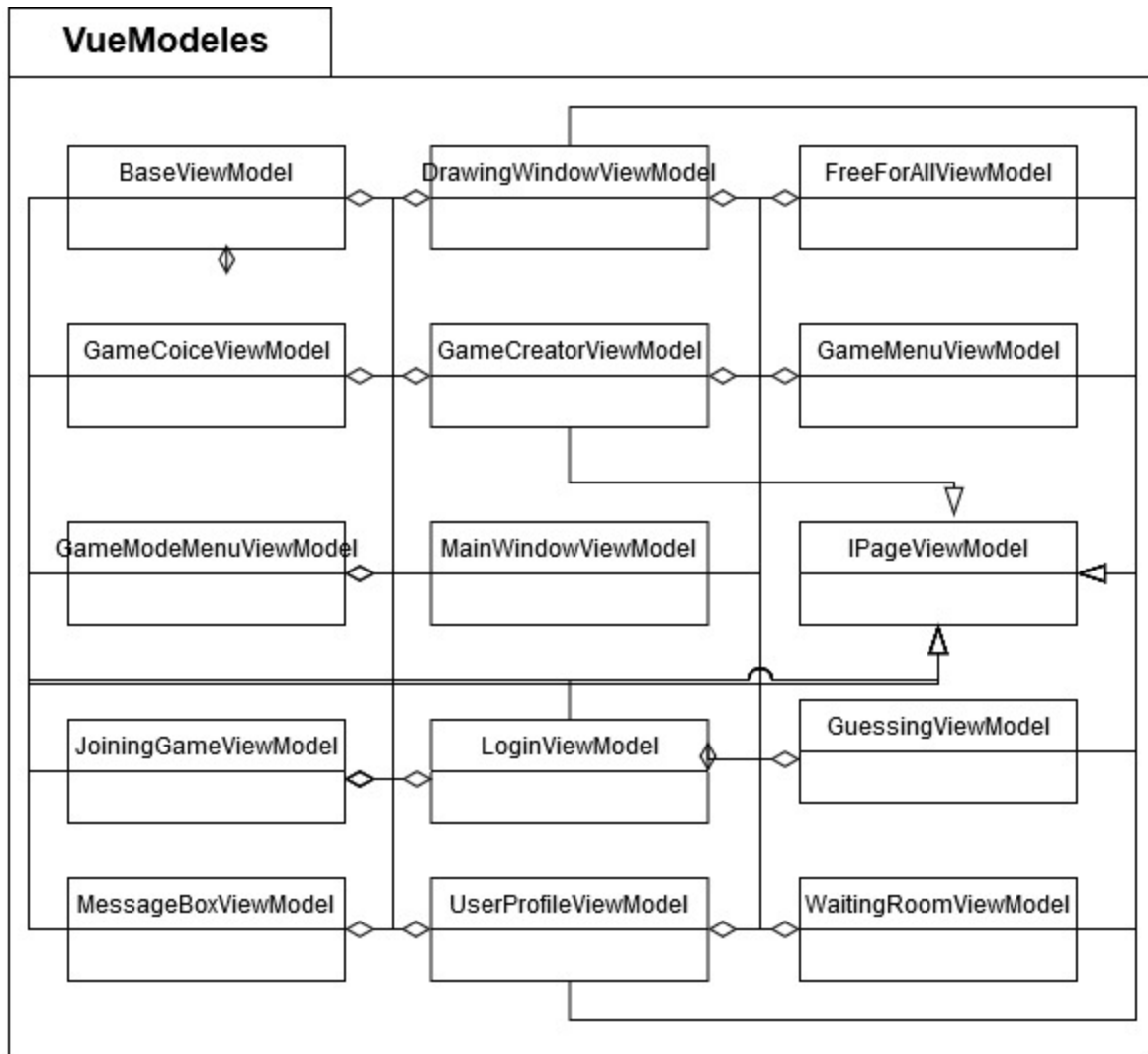


Figure 4.2.5: Diagramme de classes du paquetage VueModeles du client lourd

Tableau 4.2.5 : Description de la responsabilité du paquetage VueModeles

VueModeles
L'objectif de ce paquetage est de gérer les événements perçus dans la vue et d'accomplir le comportement désiré..

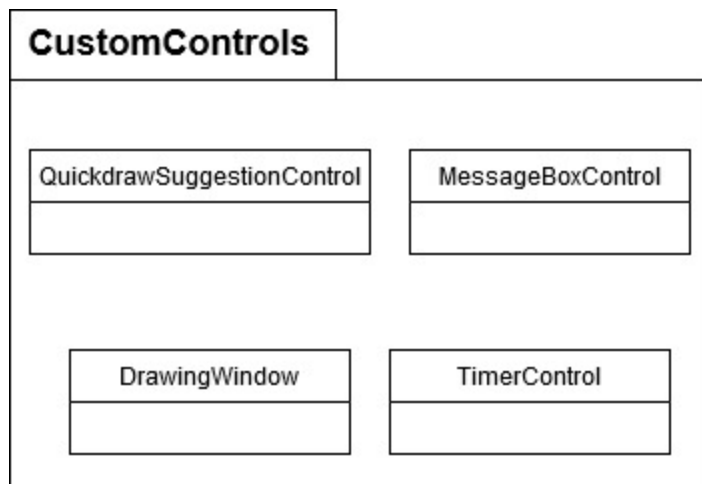


Figure 4.2.6: Diagramme de classes du paquetage CustomControls du client lourd

Tableau 4.2.6 : Description de la responsabilité du paquetage CustomControls

CustomControls
L'objectif de ce paquetage est de développer des composantes graphiques qui peuvent utilisées dans différentes vues.

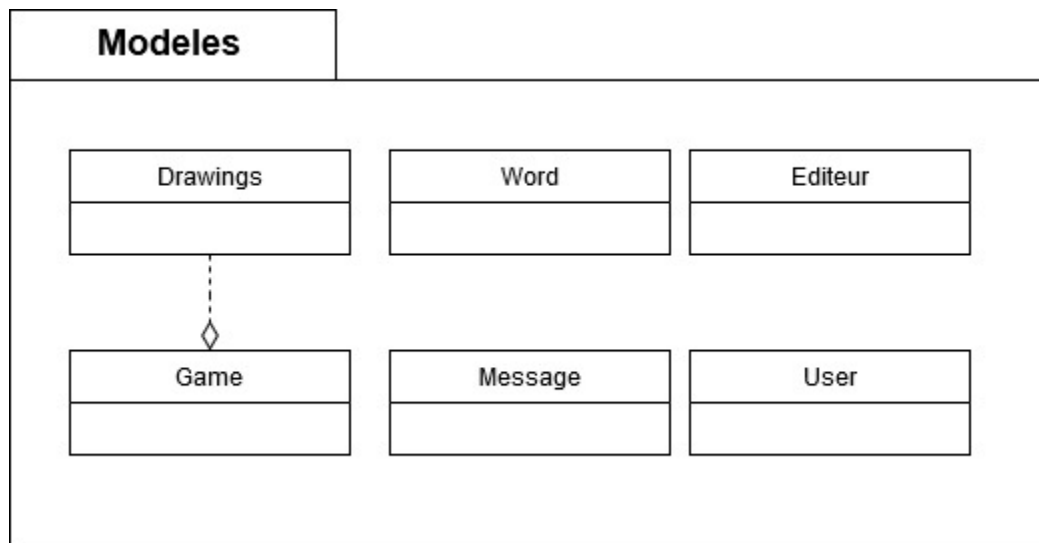


Figure 4.2.7: Diagramme de classes du paquetage Modeles du client lourd

Tableau 4.2.7 : Description de la responsabilité du paquetage Modeles

Modeles
L'objectif de ce paquetage est d'offrir une structure pour contenir les données obtenues de différentes sources.

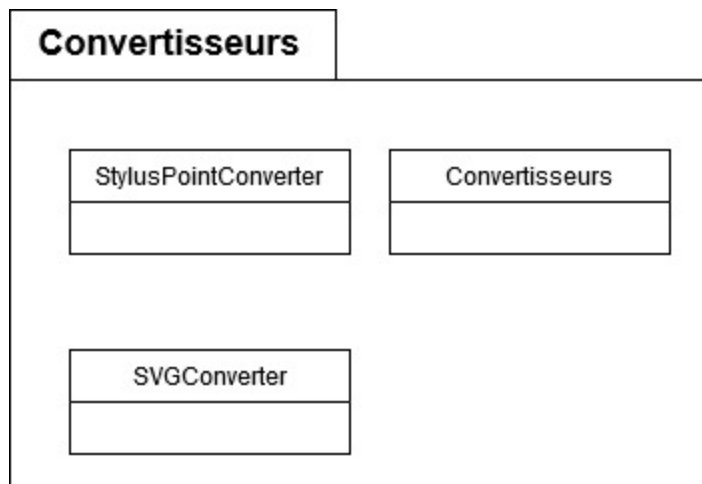


Figure 4.2.8: Diagramme de classes du paquetage Convertisseurs du client lourd

Tableau 4.2.8 : Description de la responsabilité du paquetage Convertisseurs

Convertisseurs
L'objectif de ce paquetage est de convertir des données reçues afin de faciliter la communication avec le serveur.

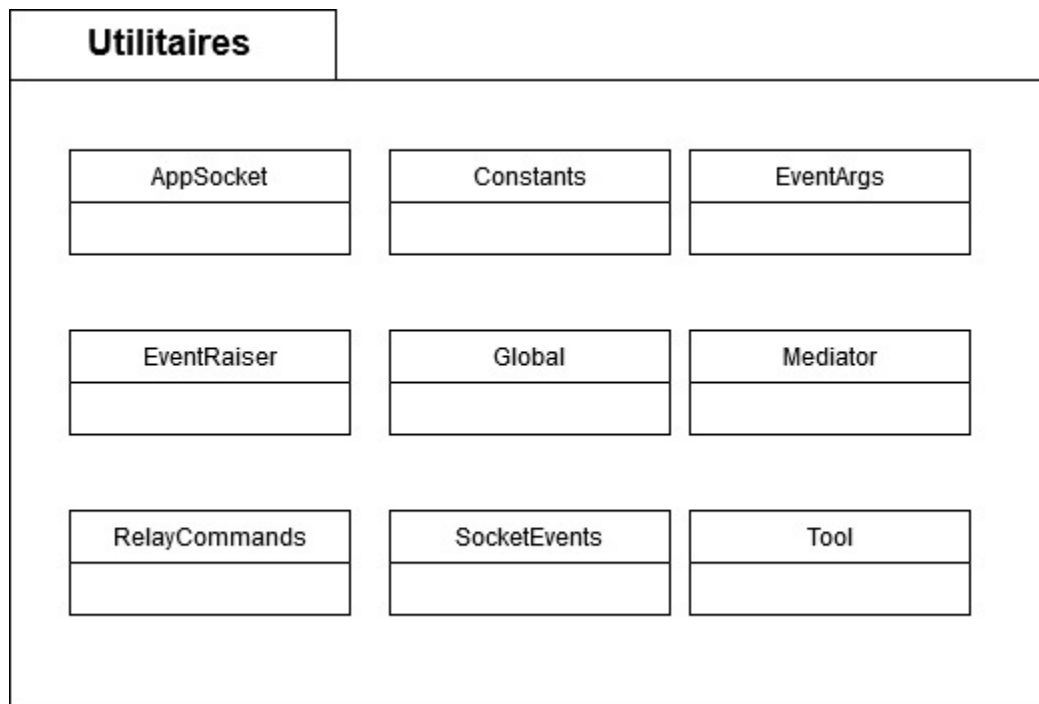


Figure 4.2.9: Diagramme de classes du paquetage Utilitaires du client lourd

Tableau 4.2.9 : Description de la responsabilité du paquetage Utilitaires

Utilitaires
L'objectif de ce paquetage est d'ouvrir des outils aux développeurs pour faciliter le développement.

#### 4.2.2 Serveur

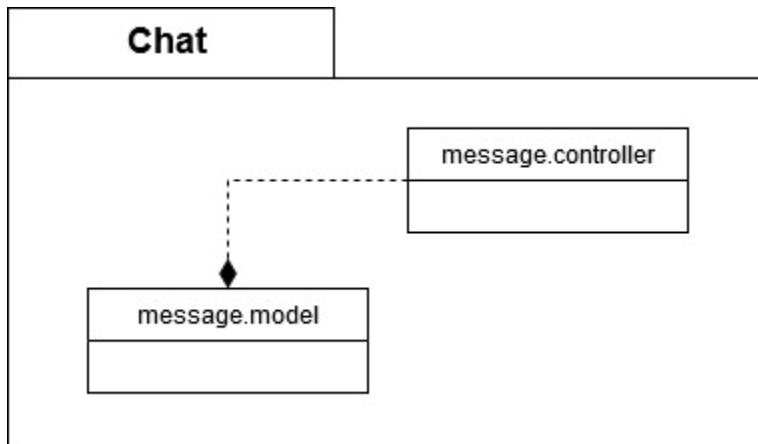


Figure 4.2.10: Diagramme de classes du paquetage Chat du serveur

Tableau 4.2.10 : Description de la responsabilité du paquetage Chat

Chat
L'objectif de ce paquetage est de permettre la communication entre plusieurs personnes avec un service de messagerie instantané.

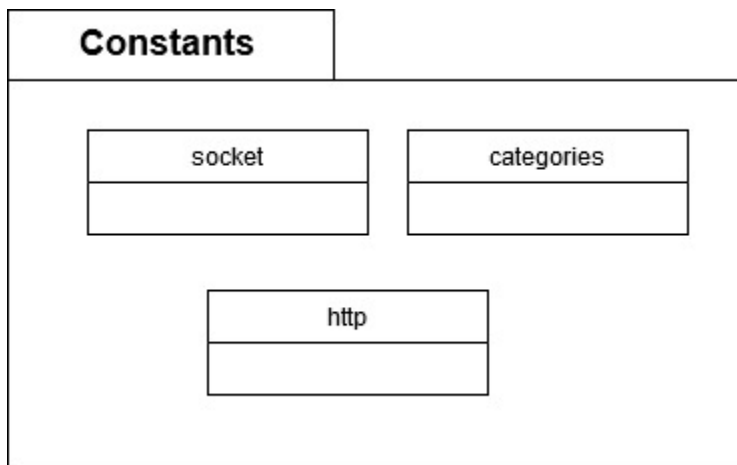


Figure 4.2.11: Diagramme de classes du paquetage Constants du serveur

Tableau 4.2.11 : Description de la responsabilité du paquetage Constants

Constants
L'objectif de ce paquetage est de rendre le code plus lisible et faciliter le maintien de l'application.



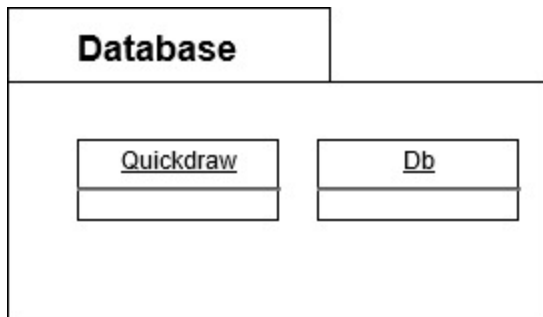


Figure 4.2.12: Diagramme de classes du paquetage Database du serveur

Tableau 4.2.12 : Description de la responsabilité du paquetage Database

Database
L'objectif de ce paquetage est d'établir la connection avec une base de données.

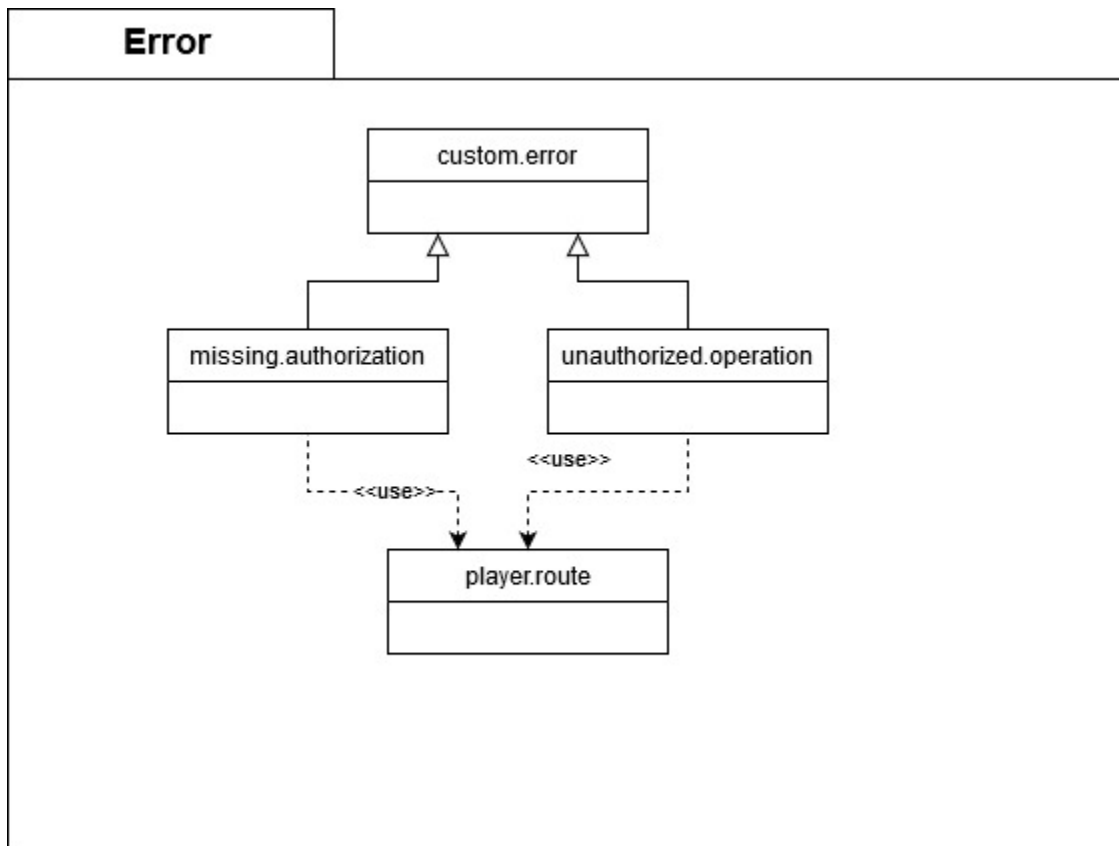


Figure 4.2.13: Diagramme de classes du paquetage Error du serveur

Tableau 4.2.13 : Description de la responsabilité du paquetage Error

Error
L'objectif de ce paquetage est de rendre le code plus maintenable. En ayant des erreurs personnalisées, il est plus aisé de trouver la source de l'erreur.

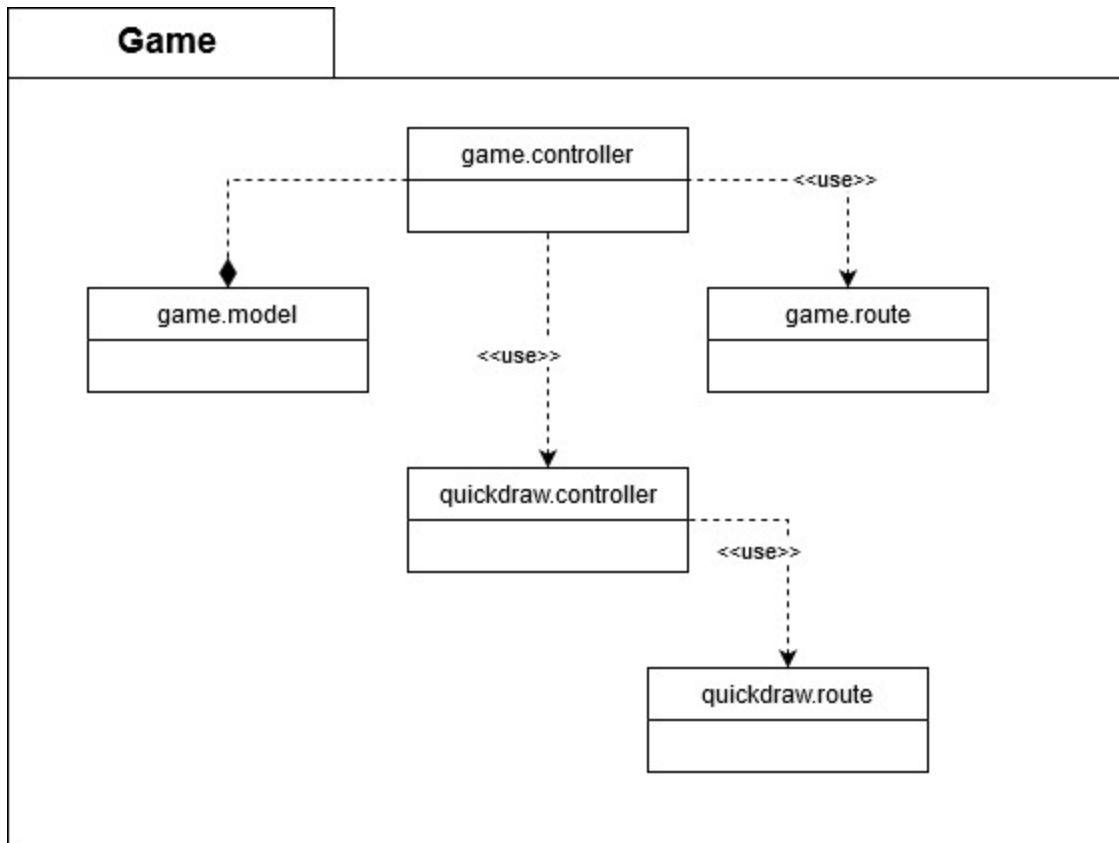


Figure 4.2.14: Diagramme de classes du paquetage Game du serveur

Tableau 4.2.14 : Description de la responsabilité du paquetage Game

Game
L'objectif de ce paquetage est de gérer tout ce qui concerne un jeu. Il supporte l'obtention d'image provenant d'un fichier local ou de Quickdraw.

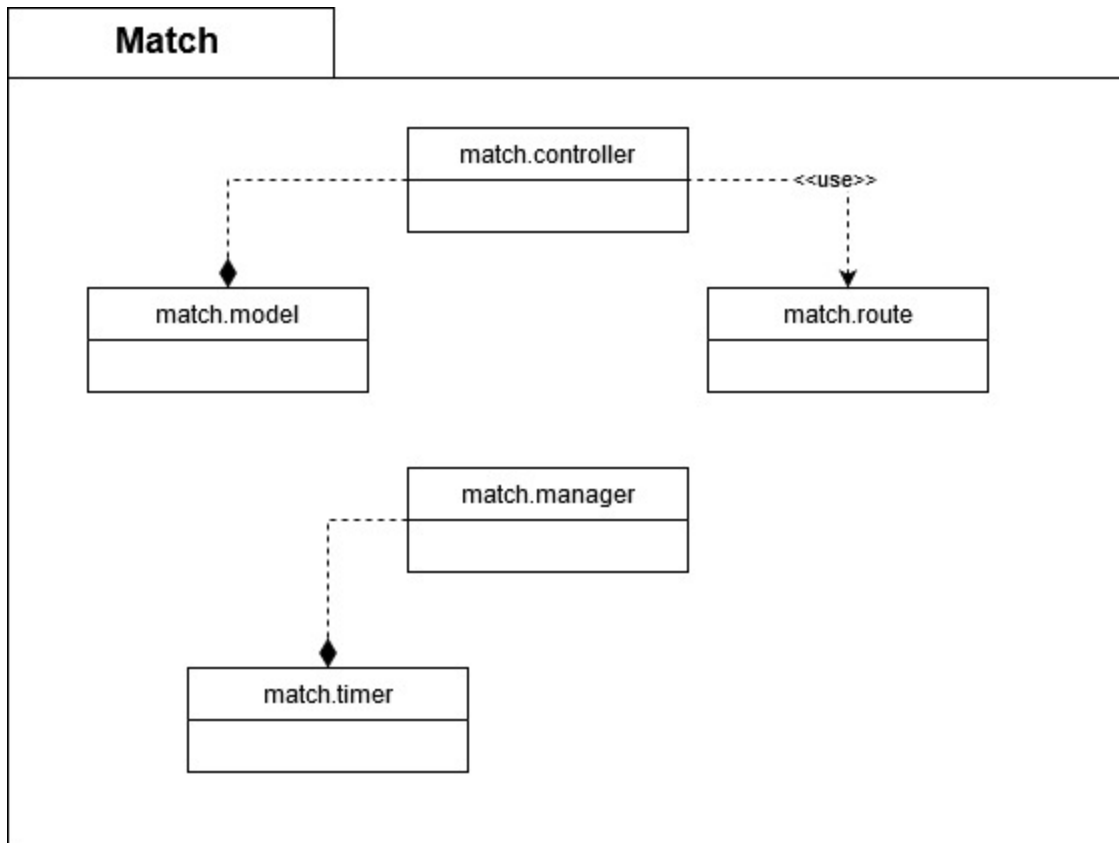


Figure 4.2.15: Diagramme de classes du paquetage Match du serveur

Tableau 4.2.15 : Description de la responsabilité du paquetage Match

Match
L'objectif de paquetage est de gérer tout ce qui concerne une partie. Cela inclut le déroulement d'une partie et les appels à la base de données.

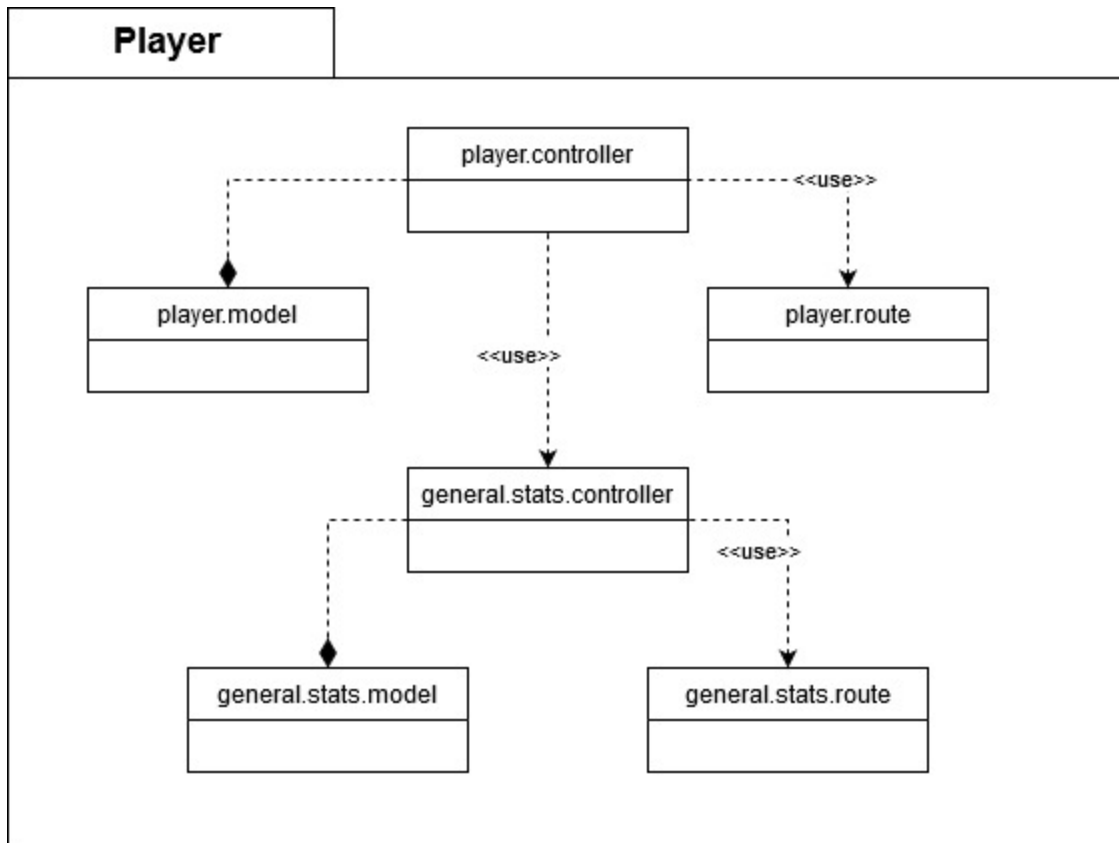


Figure 4.2.16: Diagramme de classes du paquetage Player du serveur

Tableau 4.2.16 : Description de la responsabilité du paquetage Player

Player
L'objectif de paquetage est de gérer tout ce qui concerne l'utilisateur. Cela inclut ses statistiques personnelles ainsi que ses informations associées à son compte.

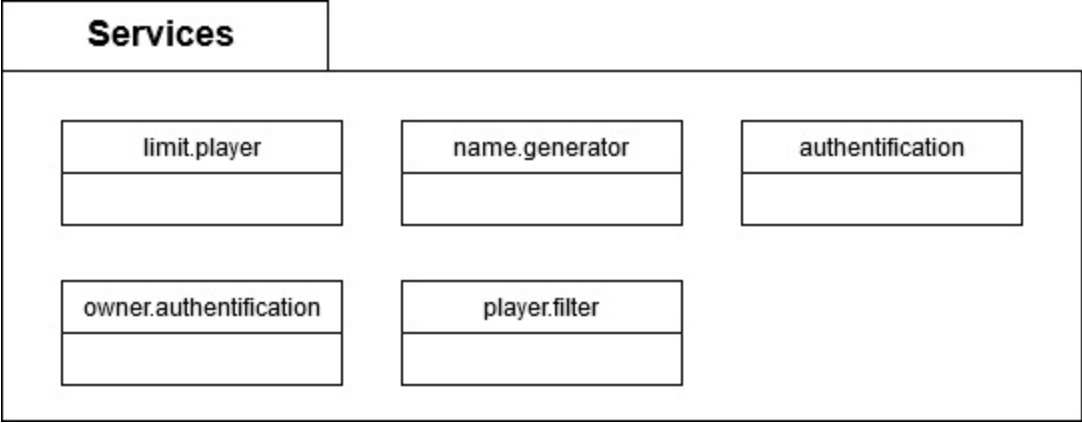


Figure 4.2.17: Diagramme de classes du paquetage Services du serveur

Tableau 4.2.17 : Description de la responsabilité du paquetage Services

Services
L’objectif de paquetage est de servir d’intermédiaire avant que la requête ne soit acheminée à l’objectif initial désiré. Les différentes classes vérifient la validité de la requête, ajoutent des éléments dans le corps de la requête ou établissent des critères pour que la requête soit acceptée.

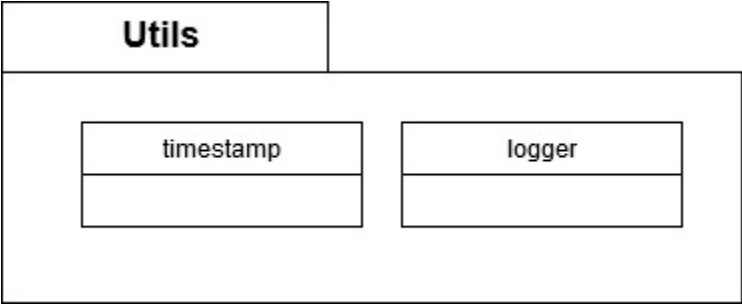


Figure 4.2.18: Diagramme de classes du paquetage Utils du serveur

Tableau 4.2.18 : Description de la responsabilité du paquetage Utils

Utils
L’objectif de paquetage est d’établir des outils pour rendre le code plus clair.

#### 4.2.3 Client léger

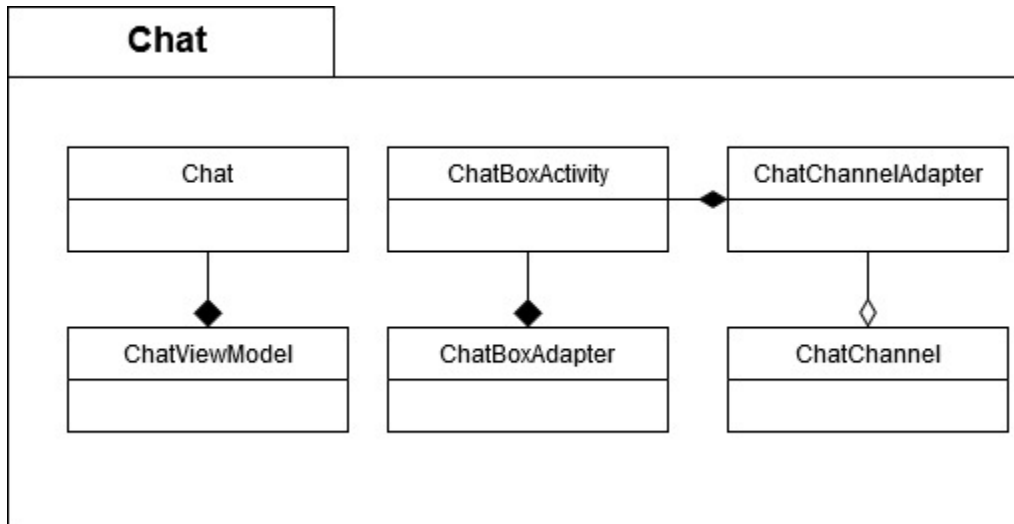


Figure 4.2.19: Diagramme de classes du paquetage Chat

Tableau 4.2.19 : Description de la responsabilité du paquetage Chat

Chat
L'objectif de paquetage est de permettre à l'utilisateur de communiquer avec d'autres utilisateurs avec un service de messagerie en temps réel.

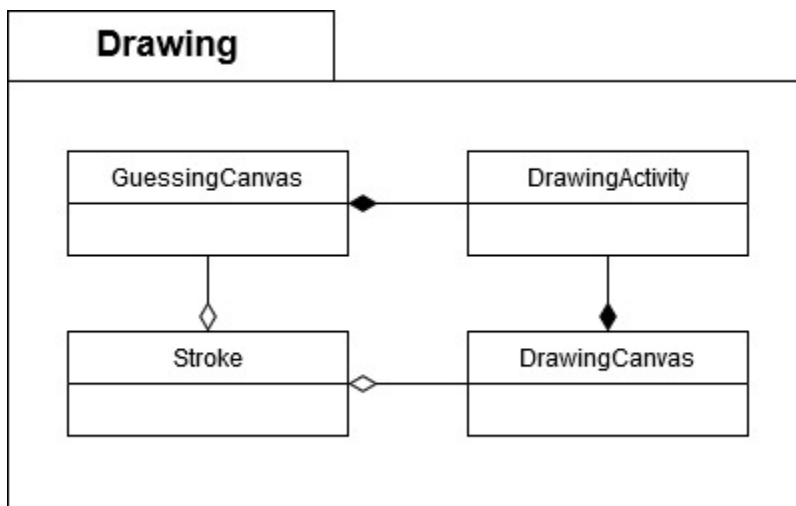


Figure 4.2.20: Diagramme de classes du paquetage Drawing

Tableau 4.2.20 : Description de la responsabilité du paquetage Drawing

Drawing
L'objectif de paquetage est de permettre à l'utilisateur de dessiner et d'afficher un dessin.

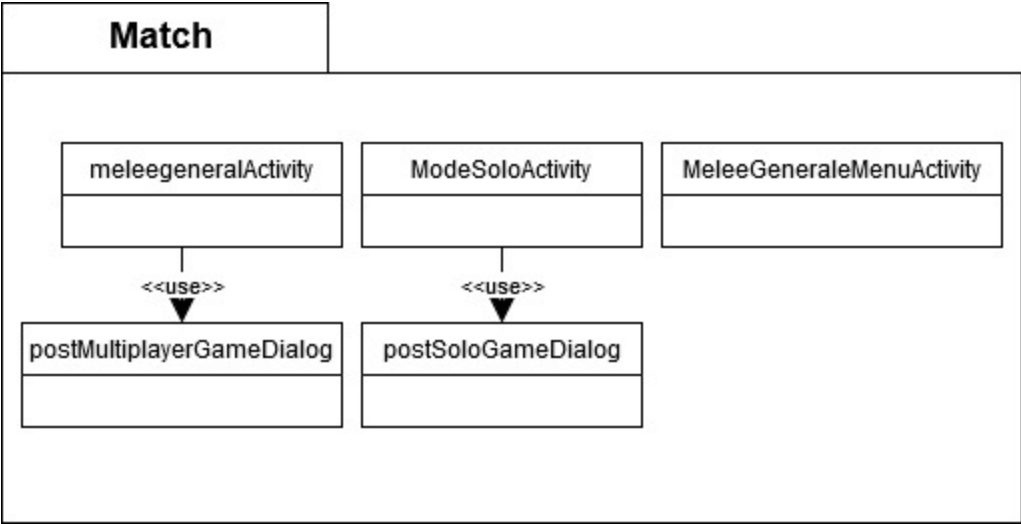


Figure 4.2.21: Diagramme de classes du paquetage Match

Tableau 4.2.21 : Description de la responsabilité du paquetage Match

Match
L’objectif de paquetage est de permettre au joueur de jouer une partie.

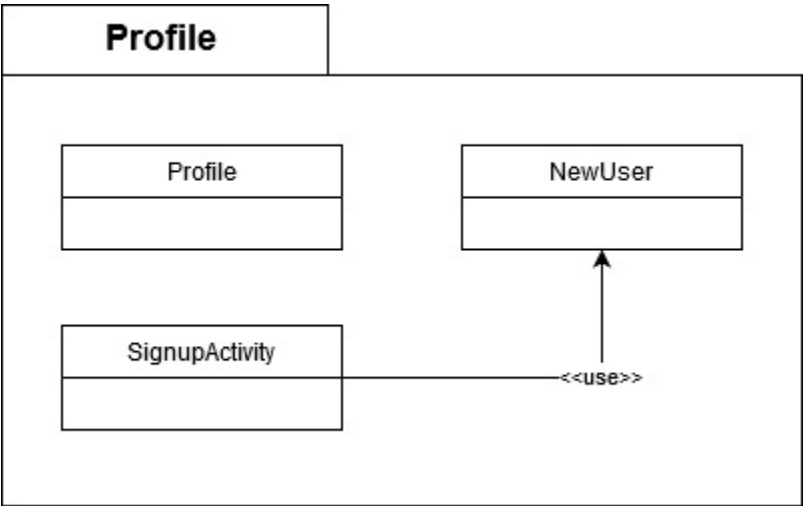


Figure 4.2.22: Diagramme de classes du paquetage Profile

Tableau 4.2.2 : Description de la responsabilité du paquetage Profile

Profile
L’objectif de paquetage est de gérer la création et l’accès à un compte pour accéder à l’application.

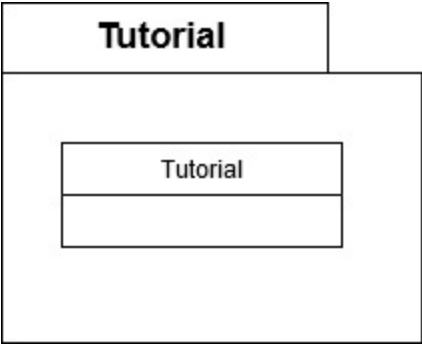


Figure 4.2.23: Diagramme de classes du paquetage Tutorial

Tableau 4.2.23 : Description de la responsabilité du paquetage Tutorial

Tutorial
L’objectif de paquetage est de permettre à l’utilisateur de comprendre comment naviguer à travers l’application.

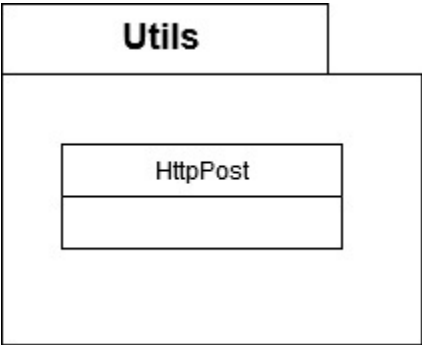


Figure 4.2.24: Diagramme de classes du paquetage Utils

Tableau 4.2.24 : Description de la responsabilité du paquetage Utils

Utils
L’objectif de paquetage est d’établir des outils pour rendre le code plus clair.



## 5. Vue des processus

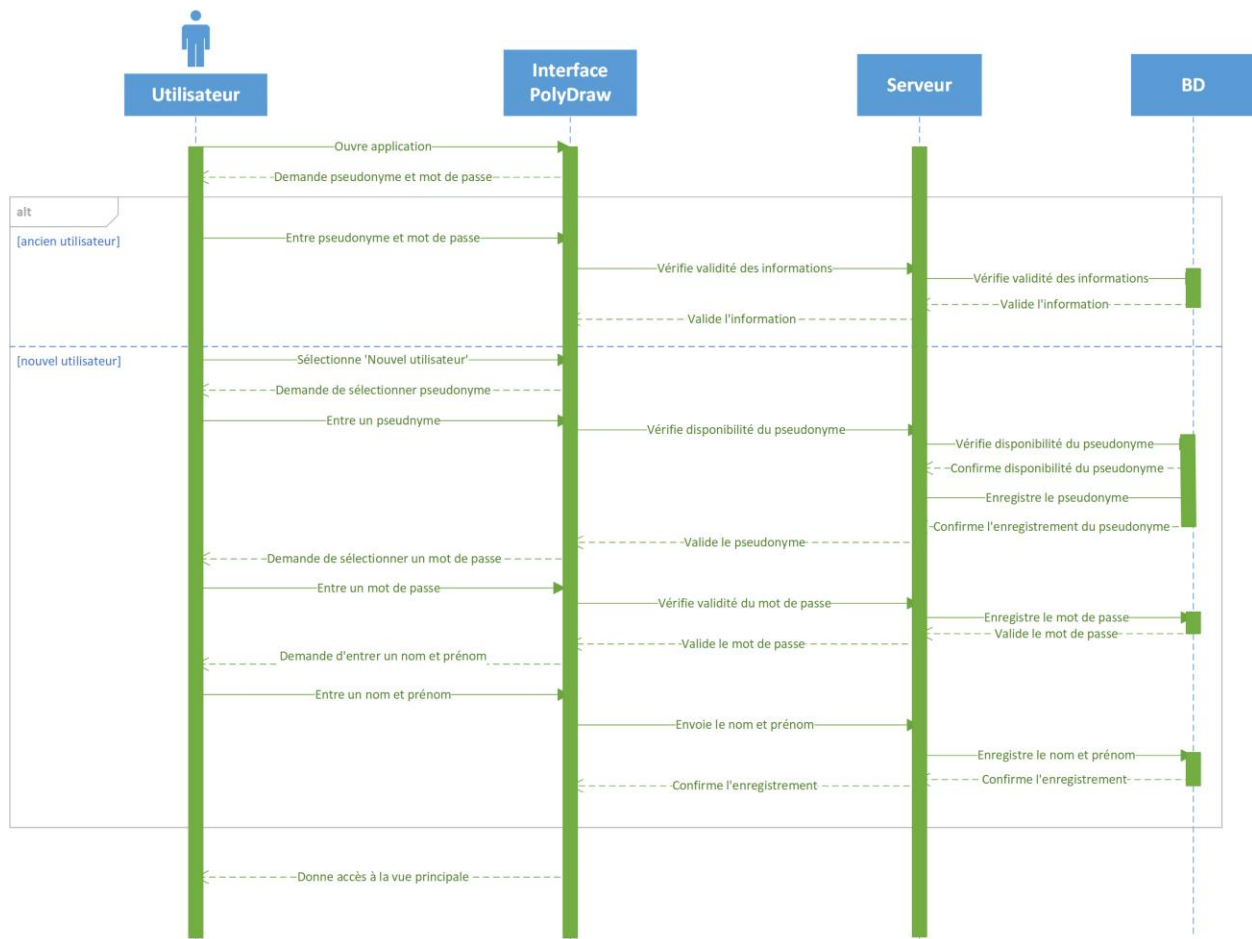


Figure 5.1 Diagramme de séquence pour l'authentification au système

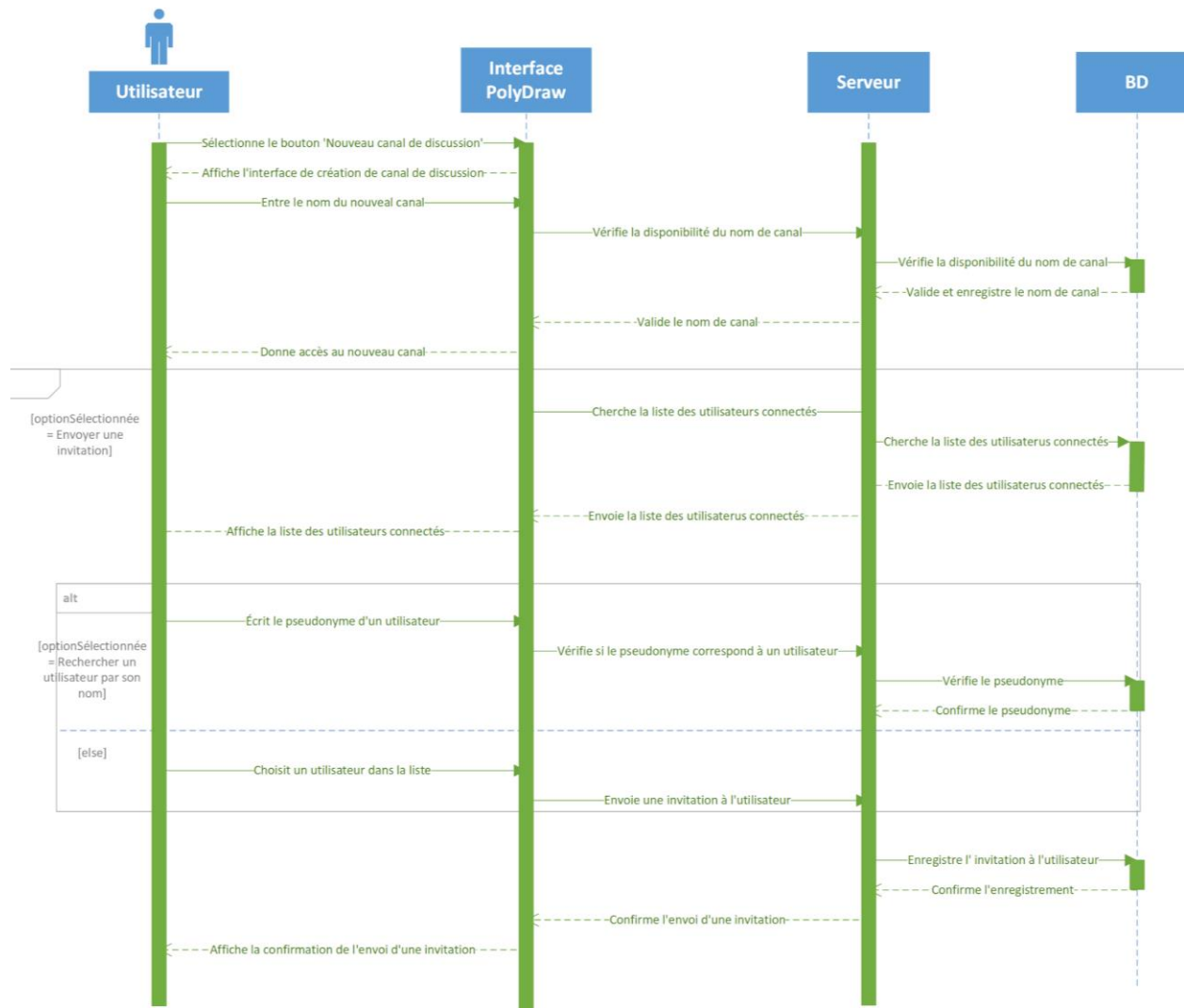


Figure 5.2 Diagramme de séquence pour la création d'un canal de discussion

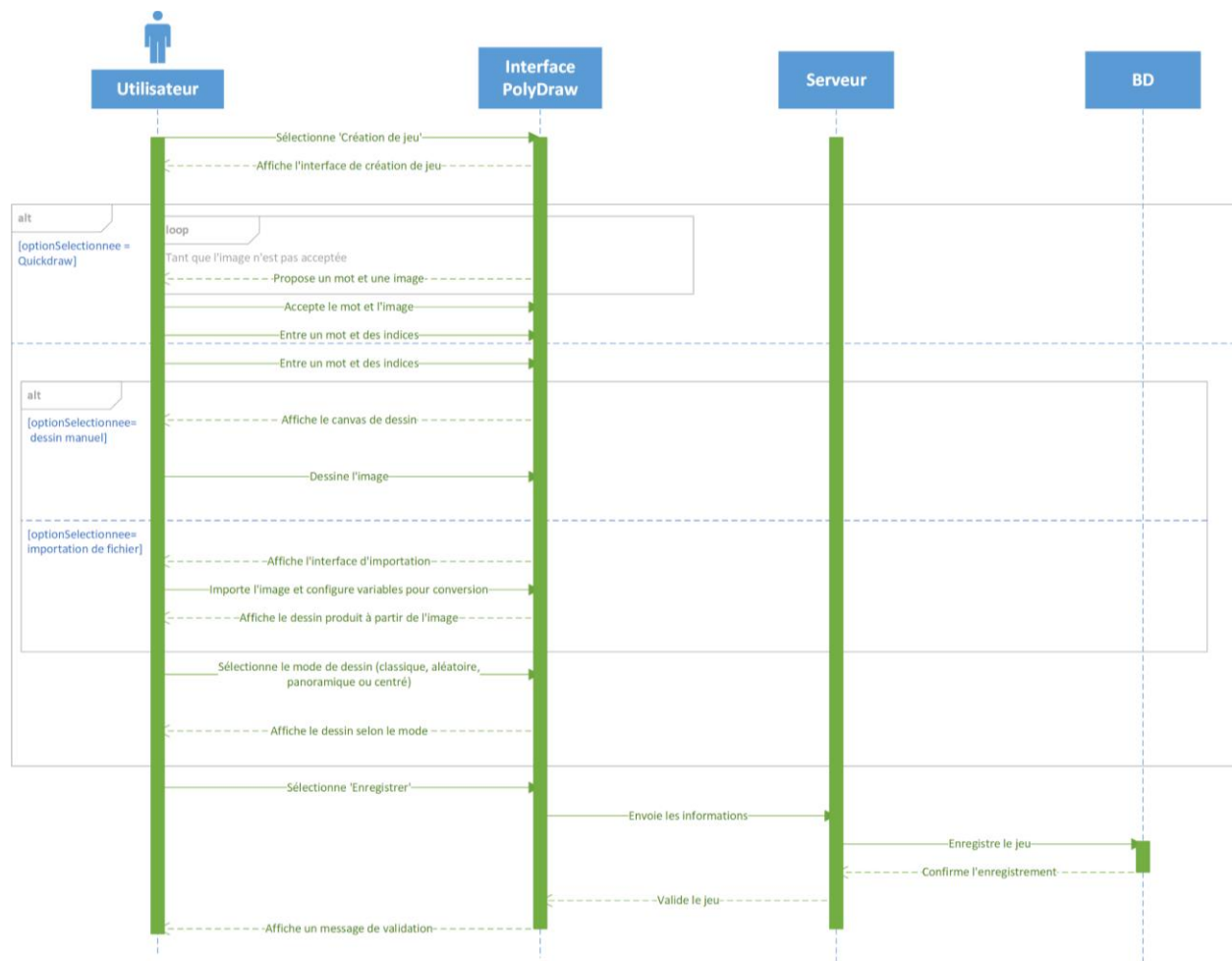


Figure 5.3 Diagramme de séquence pour la création d'un jeu

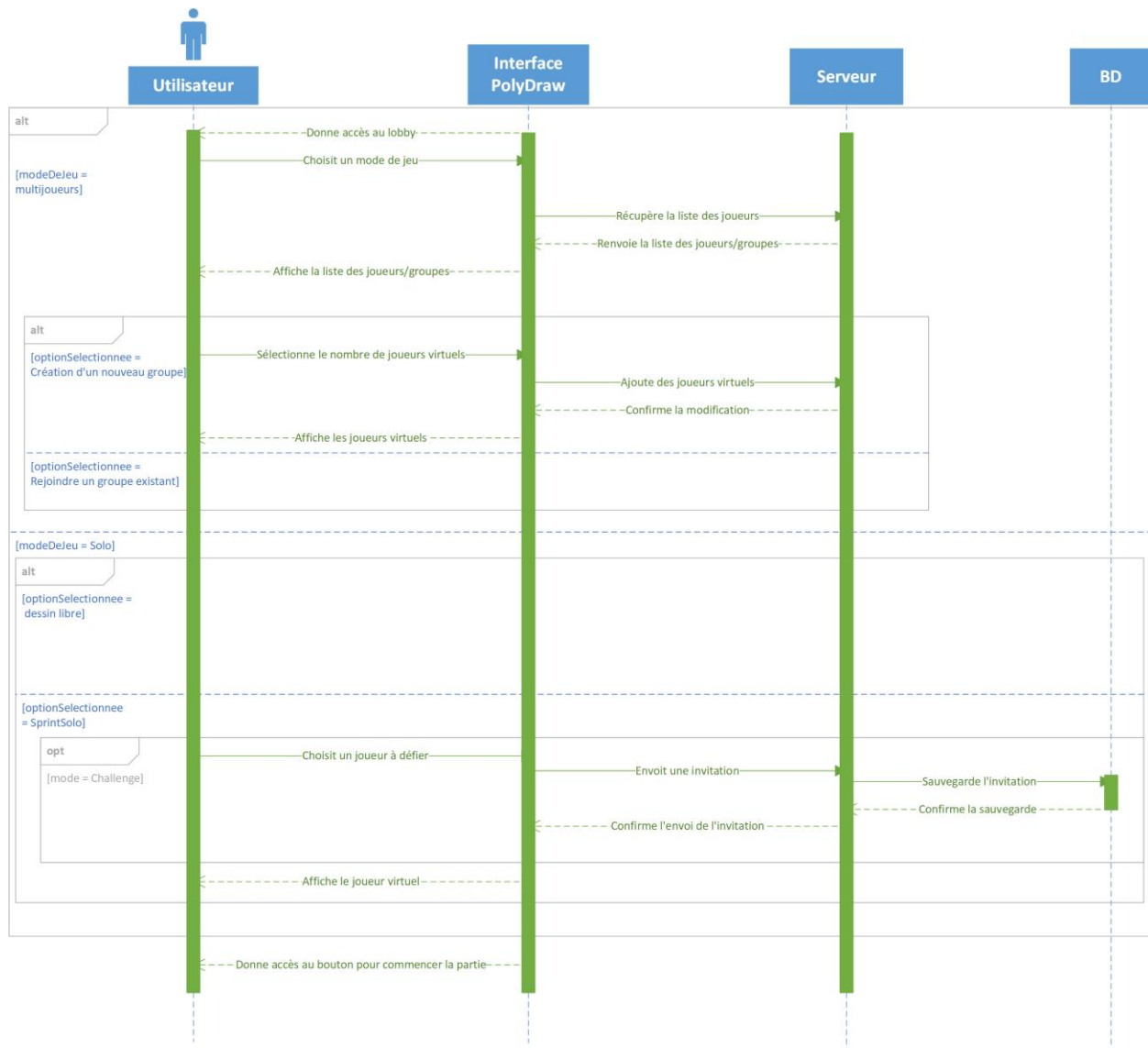


Figure 5.4 Diagramme de séquence pour le lancement d'une nouvelle partie

## 6. Vue de déploiement

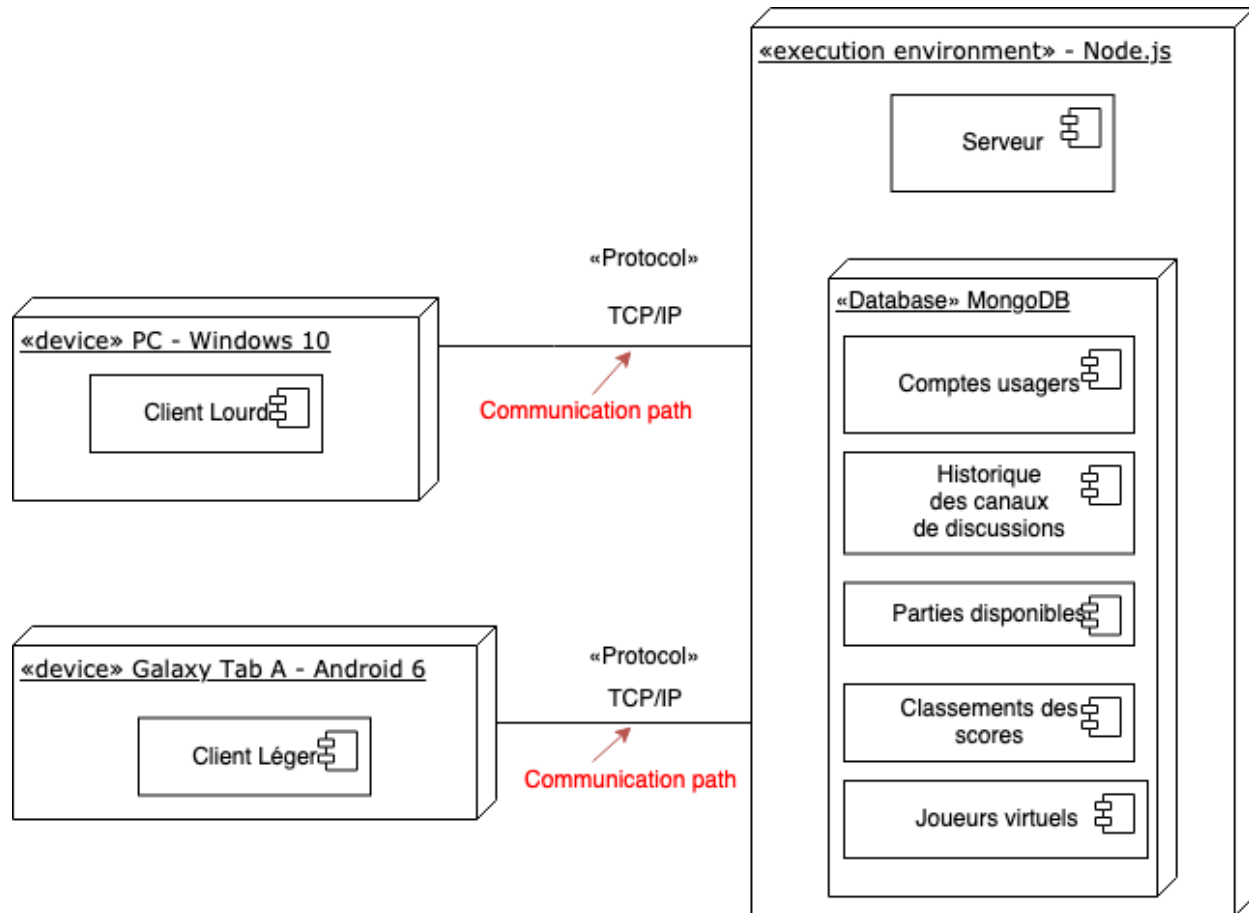


Figure 6.1 Diagramme de déploiement

## 7. Taille et performance

Plusieurs caractéristiques de taille et de performance peuvent impacter l'architecture et le design du logiciel. En effet, les deux types de clients posséderont de légères différences puisque le client lourd doit être en mesure de créer un jeu. Cette fonctionnalité n'est pas présente sur le client léger. Ceci pourrait donc avoir un impact sur la performance lorsque l'utilisateur va téléverser une image au serveur. Cependant, cela ne devrait pas avoir un impact sur la performance du logiciel lorsque l'utilisateur joue une partie. L'expérience utilisateur ne devrait pas être affectée lorsque l'utilisateur joue une partie soit sur le client lourd soit sur le client léger. Aucun délai notable devrait être constaté lors de la partie jouée. Plus précisément, le délai devrait être sous les 200 millisecondes. Aucun délai notable devrait être constaté dans la messagerie lorsque l'utilisateur envoie un message. Le délai devrait être sous les 500 millisecondes. Le temps d'authentification devrait être au maximum de 2 secondes.

De plus, les informations stockées sur la base de données seront appelées que lorsque l'utilisateur désire voir son profil. L'application ne devrait pas utiliser plus de 200 Mo pour le client léger une fois installée sur la tablette. Aussi, sur le client léger, il y aura la possibilité de télécharger un dessin dans le mode "dessin libre". Le fichier ne devrait pas dépasser la taille de 2 Mo. L'utilisation maximale du CPU devrait être de 80%. La mémoire utilisée devrait être au plus 6 Mo. Pour le client lourd, l'utilisation maximale du CPU devrait être de 80%. La mémoire devra atteindre au plus 250 Mo.