Fais-moi un dessin

Document d'architecture logicielle

Version 1.1

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2021-02-17 | 1.0 | Première rédaction | Équipe 207 |
| 2021-02-18 | 1.1 | Révision de la structure du document | Marc-Olivier Riopel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

[**1. Introduction**](#_heading=h.buuydmdaj0r2) **5**

[**2. Objectifs et contraintes architecturaux**](#_heading=h.30j0zll) **5**

[2.1 Serveur](#_heading=h.sr2j76wbky80) 5

[2.1.1 Objectif](#_heading=h.nqfpnia0daof) 5

[2.1.2 Disponibilité](#_heading=h.1gl2emhwd0rg) 5

[2.1.3 Fiabilité](#_heading=h.xcibeuc90por) 6

[2.1.3 Intégrité des données](#_heading=h.kcllphigzo6p) 6

[2.2 Client léger](#_heading=h.vvrzje6qlq75) 6

[2.2.1 Objectif](#_heading=h.23cxa6kedw4m) 6

[2.2.2 Portabilité](#_heading=h.3vocleb1cazp) 6

[2.2.3 Cadriciel de développement](#_heading=h.39elcarbiaie) 6

[2.3 Client lourd](#_heading=h.oklswq5u4qgf) 6

[2.3.1 Objectif](#_heading=h.m4v2u6tt7yss) 6

[2.3.2 Portabilité](#_heading=h.30wec9eora8n) 6

[2.3.3 Cadriciel de développement](#_heading=h.e46qc4b217do) 6

[**3. Vue des cas d’utilisation**](#_heading=h.sh95jekm50ay) **7**

[3.1 Diagramme de cas d’utilisation du clavardage](#_heading=h.pwjk7tg300bs) 7

[3.2 Cas d’utilisation pour le profil utilisateur](#_heading=h.xmaem77zsafh) 7

[3.3 Cas d’utilisation de la sélection d’un mode de jeu](#_heading=h.beygngf3k0ul) 8

[3.4 Cas d’utilisation du mode de jeu classique](#_heading=h.wbxp6qobg0xw) 8

[3.5 Cas d’utilisation du mode de jeu sprint solo](#_heading=h.qevhbdnctmy7) 9

[3.6 Cas d’utilisation du mode de jeu sprint coopératif](#_heading=h.2bg3vbvojd4s) 9

[3.7 Cas d’utilisation de la création paire mot-image](#_heading=h.pfqr4i7o5ksg) 10

[3.8 Cas d’utilisation du tutoriel](#_heading=h.cp1zretpzlv) 10

[3.9 Cas d’utilisation du tableau de classement](#_heading=h.rn0ejnjwsxcz) 11

[3.10 Cas d’utilisation de l’envoi du dessin après une partie classique](#_heading=h.n92b26s1m94c) 11

[3.11 Cas d’utilisation des parties privées](#_heading=h.hhp4sys2hcro) 12

[3.12 Cas d’utilisation des liens d'amitié](#_heading=h.20mhwrtqlkk9) 12

[3.13 Cas d’utilisation du fil d’image des parties jouées](#_heading=h.7kc03jehj9ra) 13

[3.14 Cas d’utilisation du vote des dessin du joueur virtuel](#_heading=h.xlb9rxssghcj) 13

[**4. Vue logique**](#_heading=h.v81n5os7lm8n) **14**

[4.1 Serveur](#_heading=h.ykzj03w6t065) 14

[4.1.1 Diagramme de paquetages](#_heading=h.xi7p90xm0ild) 14

[4.1.1.1 App](#_heading=h.f58fpbmkyn5x) 15

[4.1.1.2 Contrôleur](#_heading=h.pvrfp8bzjshb) 15

[4.1.1.3 Service](#_heading=h.xeip1ki57o8h) 16

[4.1.1.4 Modèle](#_heading=h.vdprchpazsrn) 18

[4.1.2 Diagramme de classes](#_heading=h.r6ww2d47be4i) 20

[4.1.2.1 Jeux](#_heading=h.m6ky1rb5vvs0) 20

[4.1.2.2 Clavardage](#_heading=h.7o0h5o5hrjvj) 21

[4.1.2.3 Authentification](#_heading=h.e6hnbsfc2juq) 21

[4.1.2.4 Statistiques](#_heading=h.8djrqc4gknue) 22

[4.1.2.5 Utilisateur](#_heading=h.lvv44o7m4y3j) 22

[4.1.2.6 Joueur virtuel](#_heading=h.loj15dm58hpz) 23

[4.2 Client Léger](#_heading=h.d99l1wes59xm) 24

[4.2.1 Vue](#_heading=h.nkt8yku9g7qz) 25

[4.2.3 Vue modèle](#_heading=h.l3ue3u42rsqw) 27

[4.2.3 Modèle](#_heading=h.om450wima1v0) 29

[4.2.4 Diagrammes de classes](#_heading=h.1ucsratrnpa1) 31

[4.3 Client lourd](#_heading=h.z29xlm5of062) 33

[4.3.1 Component](#_heading=h.zg155gkfew0) 33

[4.3.2 Services](#_heading=h.avt27wu637vg) 35

[4.3.3 Diagramme de classes des principales fonctionnalités du client lourd](#_heading=h.6i5t7tpz1yw) 37

[4.3.3.1 Authentification](#_heading=h.u5eqyrahoa3p) 37

[4.3.3.2 Dessiner](#_heading=h.yksafkps0htw) 37

[4.3.3.3 Clavardage](#_heading=h.xlr2ou6yvgzn) 38

[4.3.3.4 Mécanisme de jeu](#_heading=h.xf99c89587oo) 38

[**5. Vue des processus**](#_heading=h.m8zz2qveyx3z) **39**

[5.1 Connexion](#_heading=h.vbrc0qp1c3l0) 39

[5.2 Création d’une partie](#_heading=h.o66lyfkafbqb) 40

[5.3 Partie sprint solo et coopérative](#_heading=h.irgsvtc01int) 41

[5.4 Partie mode classique](#_heading=h.je3iucwsnc0w) 42

[5.5 Fin de partie](#_heading=h.ht16femo0lyl) 43

[5.6 Clavardage](#_heading=h.ne6rdwl9ynn6) 44

[5.7 Tutoriel](#_heading=h.4r910ddyit2u) 45

[5.8 Requêtes au serveur](#_heading=h.7tbch2w7b38p) 46

[**6. Vue de déploiement**](#_heading=h.u7q1yaubq60h) **47**

[**7. Taille et performance**](#_heading=h.1t3h5sf) **47**

Document d'architecture logicielle

# 1. Introduction

Le document d’architecture logicielle qui suit présente en détail le plan d’architecture pour la soumission de l’équipe 207. Les différentes sections présentent l’ensemble du système sous différentes vues (cas d’utilisation, logique, processus et déploiement) ainsi que les objectifs, les contraintes, les caractéristiques de tailles et de performance du projet ayant un impact sur l'architecture du logiciel. La section 2 traite d’abord des objectifs et des contraintes du logiciel qui influence l’architecture. Ensuite, les sections 3 à 6 présentent les différentes vues du système dans l’ordre suivant: vue des cas d’utilisation, vue logique, vue des processus et vue de déploiement. Puis, la section 7 traite des caractéristiques de taille et de performance du logiciel qui influence l’architecture.

# 2. Objectifs et contraintes architecturaux

Les objectifs et contraintes du projet qui auront un impact sur l’architecture du logiciel sont nombreux. Il faut d’abord que l’application puisse être utilisée sur l’ordinateur et sur une tablette. Pour optimiser l’expérience de l’utilisateur, chacune de ces applications devra être codée de façon native ce qui nécessitera certaines adaptations quant aux interactions possibles entre l’utilisateur et l’application. Toutes les technologies utilisées, soit le langage Typescript avec le cadriciel Angular et Électron pour le client lourd ainsi que le langage Kotlin pour le client léger apportent également leurs contraintes respectives. De plus, puisqu’il s'agit d’un jeu en réseau, il faudra que le client lourd et le client léger puissent communiquer entre eux en temps réel pour s’envoyer de l’information. Il faudra donc mettre en place un serveur qui sert d’intermédiaire et qui renvoie l’information en temps réel à l’aide de web sockets.

## 2.1 Serveur

### 2.1.1 Objectif

Le serveur sera responsable de la communication entre les différents clients, de la logique derrière plusieurs mécanismes clés de système (jeux, ami(es), authentification, etc.) ainsi que de la persistance des données importantes.

### 2.1.2 Disponibilité

Pour assurer le fonctionnement de notre application en production, nous devrons rendre le serveur accessible aux clients à distance. Pour ce faire, le service EC2 d’AWS sera utilisé. Le code du serveur sera déployé sur une instance t2.micro et disponible en tout temps. Cette instance faisant partie de l’offre gratuite d’AWS, nous n’aurons pas besoin de débourser pour son exploitation pour la durée du projet.

### 

### 2.1.3 Fiabilité

Le serveur doit être en mesure de supporter un minimum de deux parties simultanément (8 joueurs). En dessous de cette limite, le serveur doit offrir une grande fiabilité. Il ne doit pas avoir de problèmes de latence, de *crashs* ou toutes autres situations similaires.

### 2.1.3 Intégrité des données

Le serveur doit assurer la persistance des données importantes. Pour ce faire, une base de données MongoDB sera utilisée pour conserver ces données. Également, le serveur devra être développé de manière à ce que les clients ne puissent pas manipuler de données importantes de leur côté. Il faudra donc implémenter la validation côté serveur et éviter le traitement des données importantes du côté client.

## 2.2 Client léger

### 2.2.1 Objectif

L’objectif du client léger est de permettre aux utilisateurs d’accéder à l'application à partir d’une tablette mobile pour utiliser certaines fonctions du logiciel.

### 2.2.2 Portabilité

Le client léger doit être disponible pour la tablette Samsung Galaxy Tab A 2019.

### 2.2.3 Cadriciel de développement

Le client léger sera développé avec le langage Kotlin avec Android Studio.

## 2.3 Client lourd

### 2.3.1 Objectif

L’objectif du client lourd est de permettre aux utilisateurs d’accéder à l’application à partir d’un ordinateur de bureau pour utiliser toutes les fonctions du logiciel.

### 2.3.2 Portabilité

Le client lourd doit être disponible pour les systèmes d’exploitation Windows, Mac et Linux.

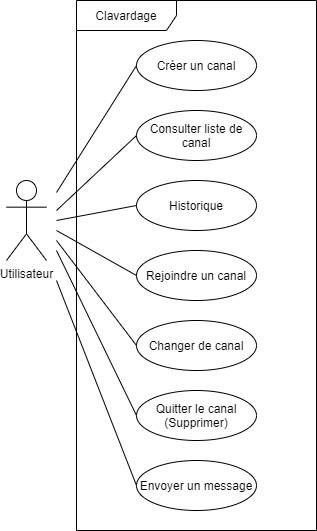
### 2.3.3 Cadriciel de développement

Le client lourd sera développé avec le cadriciel Angular et utilisera Electron pour rendre l’application disponible pour ordinateur de bureau.

# 

# 3. Vue des cas d’utilisation

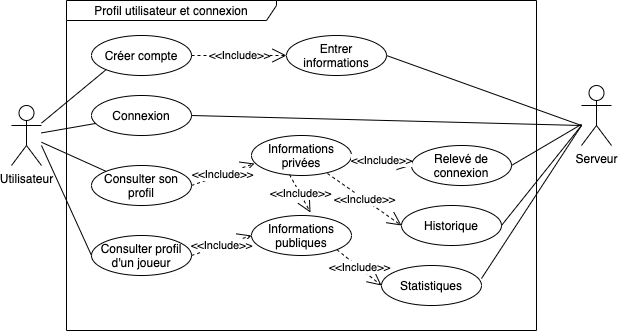
## 3.1 Diagramme de cas d’utilisation du clavardage



*Figure 1. Diagramme de cas d’utilisation du clavardage*

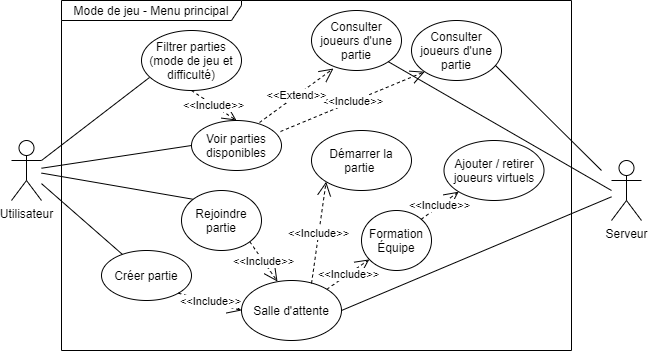
## 

## 3.2 Cas d’utilisation pour le profil utilisateur



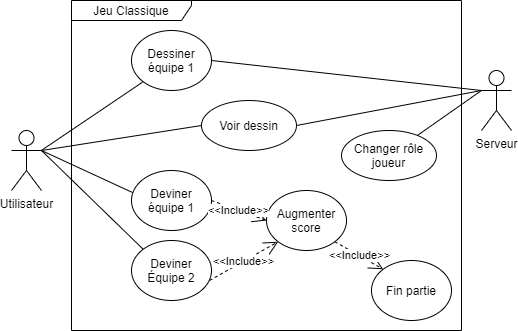
*Figure 2. Diagramme de cas d’utilisation du profil utilisateur*

## 3.3 Cas d’utilisation de la sélection d’un mode de jeu



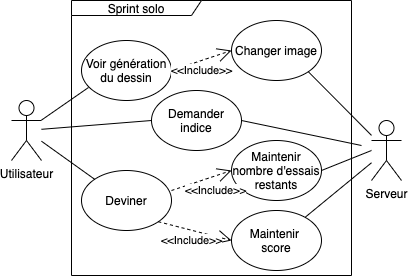
*Figure 3. Diagramme de cas d’utilisation pour la sélection de modes de jeu*

## 3.4 Cas d’utilisation du mode de jeu classique



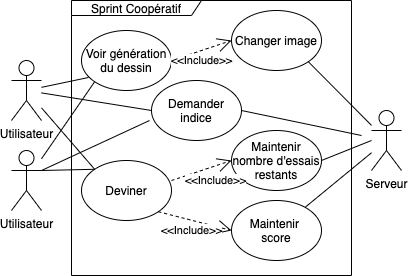
*Figure 4. Diagramme de cas d’utilisation pour le mode de jeu classique*

## 3.5 Cas d’utilisation du mode de jeu sprint solo



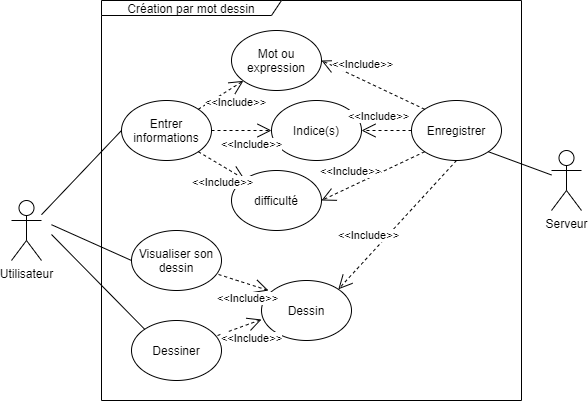
*Figure 5. Diagramme de cas d’utilisation du mode de jeu classique*

## 3.6 Cas d’utilisation du mode de jeu sprint coopératif



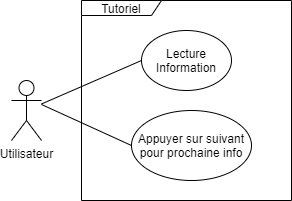
*Figure 6. Diagramme de cas d’utilisation du mode de jeu classique*

## 3.7 Cas d’utilisation de la création paire mot-image



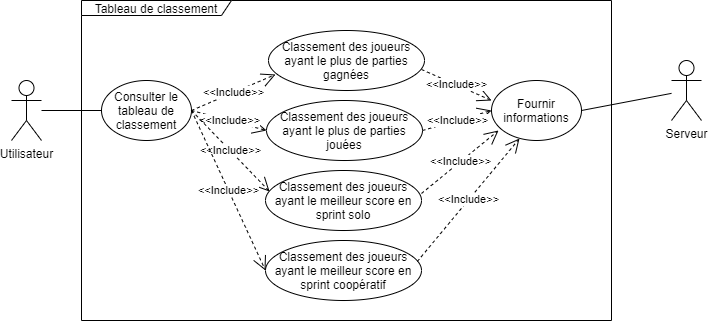
*Figure 7. Diagramme de cas d’utilisation pour la création d’une paire mot-image*

## 3.8 Cas d’utilisation du tutoriel



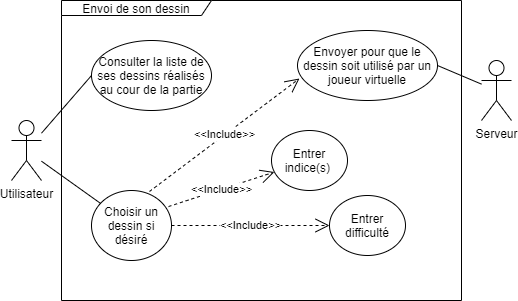
*Figure 8. Diagramme de cas d’utilisation pour le tutoriel*

## 3.9 Cas d’utilisation du tableau de classement



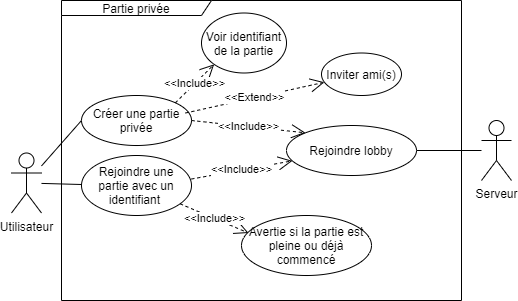
*Figure 9. Diagramme de cas d’utilisation pour la consultation du tableau de classement*

## 3.10 Cas d’utilisation de l’envoi du dessin après une partie classique



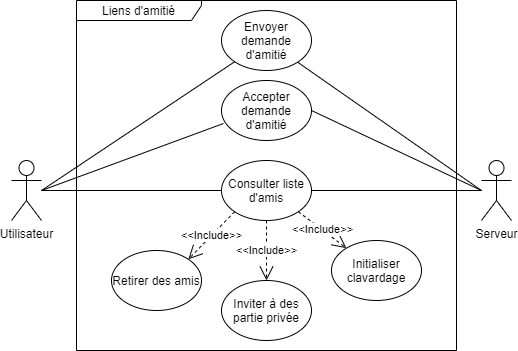
*Figure 10. Diagramme de cas d’utilisation pour l’envoi de dessins après une partie*

## 3.11 Cas d’utilisation des parties privées



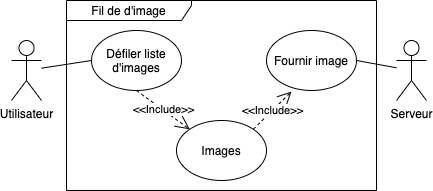
*Figure 11. Diagramme de cas d’utilisation pour les parties privées*

## 3.12 Cas d’utilisation des liens d'amitié



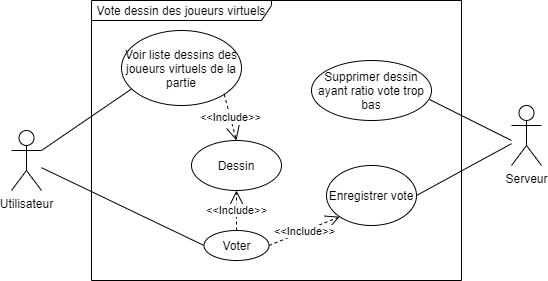
*Figure 12. Diagramme de cas d’utilisation pour les liens d’amitié*

## 3.13 Cas d’utilisation du fil d’image des parties jouées



*Figure 13. Diagramme de cas d’utilisation pour le fil d’images*

## 3.14 Cas d’utilisation du vote des dessin du joueur virtuel

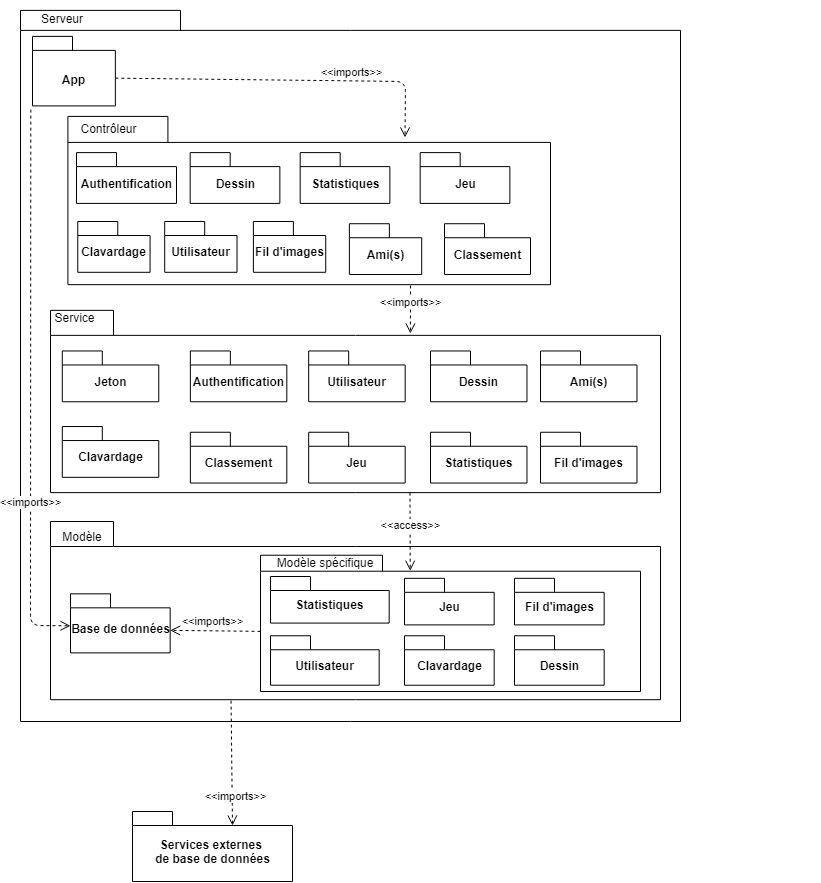


*Figure 14. Diagramme de cas d’utilisation pour les votes de dessins*

# 4. Vue logique

## 4.1 Serveur

### 4.1.1 Diagramme de paquetages



*Figure 15. Diagramme de paquetages pour le serveur*

#### 

#### 4.1.1.1 App

Il s’agit du point d’entrée de l’application serveur. Ce paquetage contient les fichiers tels que *www.ts*, server.ts et app.ts. C'est à ce niveau que nous initialisons la communication par sockets et que nous redirigeons les requêtes HTTP vers les bons contrôleurs. C’est aussi à ce niveau que nous initialisons la communication avec la base de données.

#### 4.1.1.2 Contrôleur

Les contrôleurs sont chargés de gérer les routes du REST API et des Sockets provenant du client lourd et du mobile. Il n’y a que très peu de logique dans les paquetages des contrôleurs puisque celle-ci est déléguée au paquetage de Service. Les contrôleurs ont donc la responsabilité d'appeler les bons services pour faire le traitement des données qu’ils reçoivent des clients. Les tableaux ci-dessous décrivent les paquetages principaux pour les contrôleurs:

*Tableau 1. Description du paquetage d’authentification*

|  |
| --- |
| **Authentification** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec l’authentification des utilisateurs. |

*Tableau 2. Description du paquetage de dessin*

|  |
| --- |
| **Dessin** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les dessins des utilisateurs. |

*Tableau 3. Description du paquetage de jeu*

|  |
| --- |
| **Jeu** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets pour les parties jouées par les utilisateurs. |

*Tableau 4. Description du paquetage d’authentification*

|  |
| --- |
| **Statistiques** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les statistiques des utilisateurs. |

*Tableau 5. Description du paquetage de clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les canaux de discussion. |

*Tableau 6. Description du paquetage d’utilisateur*

|  |
| --- |
| **Utilisateur** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les informations relatives aux utilisateurs |

*Tableau 7. Description du paquetage de fil d’images*

|  |
| --- |
| **Fil d’images** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec le fil d’actualité des utilisateurs. |

*Tableau 8. Description du paquetage d’amis*

|  |
| --- |
| **Ami(s)** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les liens d’amitié des utilisateurs. |

*Tableau 9. Description du paquetage de classement*

|  |
| --- |
| **Classement** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec les tableaux de classement. |

#### 4.1.1.3 Service

Ce paquetage contient toute la logique de l'application serveur. Les modules de ce paquetage ont la responsabilité de faire le traitement des données reçu de l’application client et de la base de données avant de retourner une réponse au client. Ces modules peuvent donc utiliser les modules du paquetage de niveau Modèle pour interagir avec la base de données. Les tableaux ci-dessous décrivent les paquetages principaux pour les services:

*Tableau 10. Description du paquetage de jeton*

|  |
| --- |
| **Jeton** |
| Ce paquetage regroupe les services responsables de la logique relative au jeton jsonwebtoken . |

*Tableau 11. Description du paquetage d’authentification*

|  |
| --- |
| **Authentification** |
| Ce paquetage regroupe les services responsables de la logique relative à l'authentification des utilisateurs. |

*Tableau 12. Description du paquetage d’utilisateur*

|  |
| --- |
| **Utilisateur** |
| Ce paquetage regroupe les services responsables de la logique relative à la gestion des informations des utilisateurs. |

*Tableau 13. Description du paquetage de dessin*

|  |
| --- |
| **Dessin** |
| Ce paquetage regroupe les services responsables de la logique relative aux dessins. |

*Tableau 14. Description du paquetage d’amis*

|  |
| --- |
| **Ami(s)** |
| Ce paquetage regroupe les services responsables de la logique relative aux liens d’amitié entre les utilisateurs. |

*Tableau 15. Description du paquetage de clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les services responsables de la logique relative aux canaux de discussions des utilisateurs |

*Tableau 16. Description du paquetage de classement*

|  |
| --- |
| **Classement** |
| Ce paquetage regroupe les services responsables de la logique relative aux tableaux de classement. |

*Tableau 17. Description du paquetage de jeu*

|  |
| --- |
| **Jeu** |
| Ce paquetage regroupe les services responsables de la logique relative aux parties de l’application. |

*Tableau 18. Description du paquetage de statistiques*

|  |
| --- |
| **Statistiques** |
| Ce paquetage regroupe les services responsables de la logique relative aux statistiques des utilisateurs. |

*Tableau 19. Description du paquetage de fil d’images*

|  |
| --- |
| **Fil d’images** |
| Ce paquetage regroupe les services responsables de la logique relative au jeton jsonwebtoken |

#### 

#### 4.1.1.4 Modèle

Ce paquetage contient tous les modules qui devront interagir avec les différentes collections (tables) de notre base de données. Ces modules ne contiennent que la logique nécessaire pour interagir avec la base de données. Le traitement des informations reçues devra se faire au niveau des services qui utilisent les modèles pour obtenir ces informations. Les tableaux ci-dessous décrivent les paquetages principaux pour les modèles:

*Tableau 20. Description du paquetage de base de données*

|  |
| --- |
| **Base de données** |
| Ce paquetage regroupe les modèles responsables de la gestion de la base de données comme telle. Par exemple, la connexion du serveur avec celle-ci. Tous les autres modèles utilisent ce paquetage pour avoir accès à la BD. |

*Tableau 21. Description du paquetage des informations des utilisateurs*

|  |
| --- |
| **Informations utilisateurs** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif aux identifiants de connexion des utilisateurs |

*Tableau 22. Description du paquetage d’utilisateur*

|  |
| --- |
| **Utilisateur** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif aux informations propres aux utilisateurs. |

*Tableau 23. Description du paquetage de jeu*

|  |
| --- |
| **Jeu** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif aux informations propres aux parties. |

*Tableau 24. Description du paquetage de clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif aux canaux de discussion des utilisateurs. |

*Tableau 25. Description du paquetage de dessin*

|  |
| --- |
| **Dessin** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif aux informations des dessins. |

*Tableau 24. Description du paquetage de fil d’images*

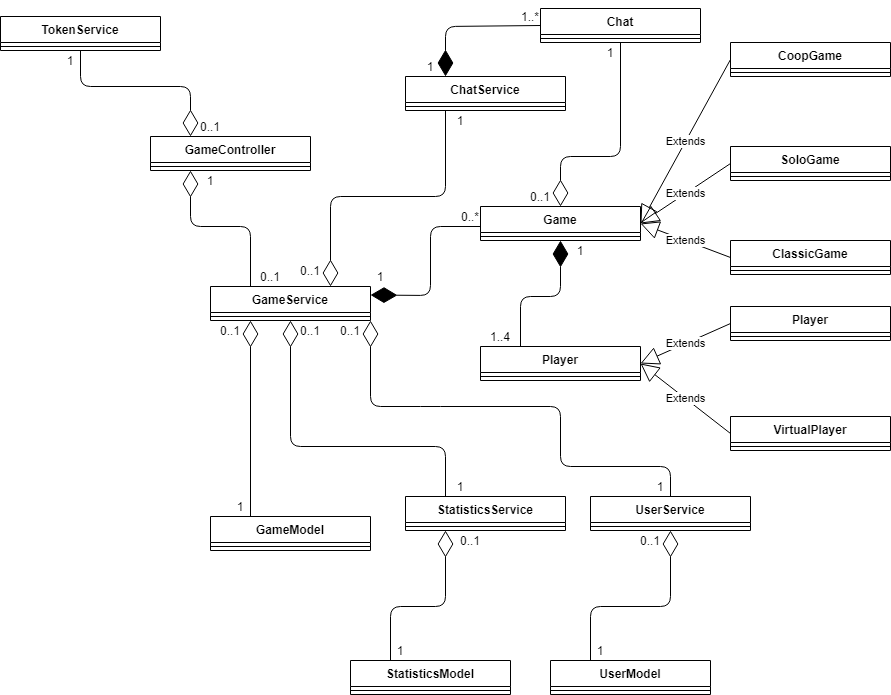
|  |
| --- |
| **Fil d’images** |
| Ce paquetage regroupe les modèles responsables de la communication avec la BD pour ce qui est relatif au fil d’actualité. |

### 

### 4.1.2 Diagramme de classes

#### 4.1.2.1 Jeux

Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion des parties en cours.

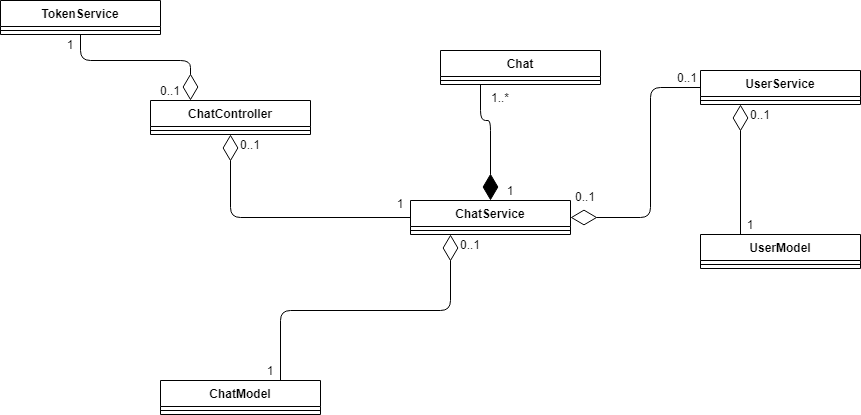


*Figure 16. Diagramme de classes pour les jeux*

#### 

#### 4.1.2.2 Clavardage

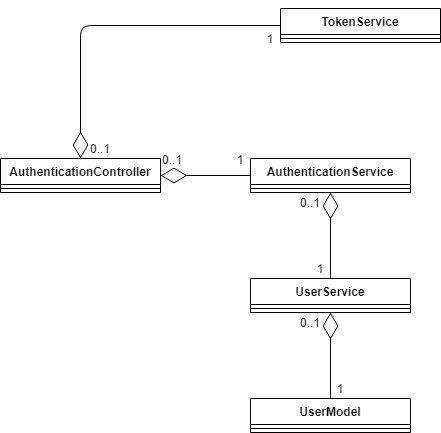
Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion de clavardage.



*Figure 17. Diagramme de classes pour le clavardage*

#### 4.1.2.3 Authentification

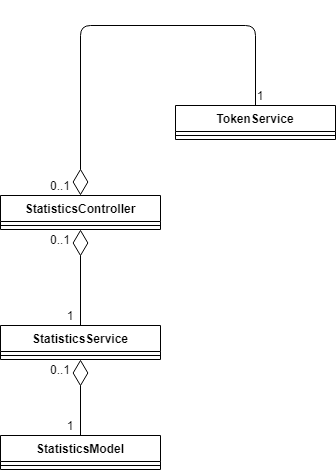
Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion de l’authentification des utilisateurs.



*Figure 18. Diagramme de classes pour l’authentification*

#### 4.1.2.4 Statistiques

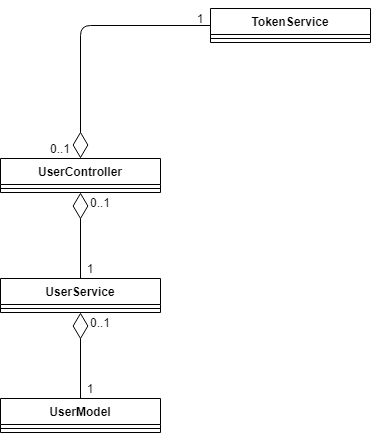
Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion des statistiques relatives aux utilisateurs.



*Figure 19. Diagramme de classes pour les statistiques*

#### 4.1.2.5 Utilisateur

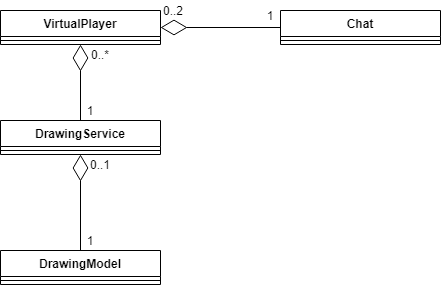
Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion des utilisateurs.



*Figure 20. Diagramme de classes pour l’utilisateur*

#### 4.1.2.6 Joueur virtuel

Le diagramme ci-dessous montre les interactions entre les classes du côté serveur pour le système de gestion des joueurs virtuels dans les parties.

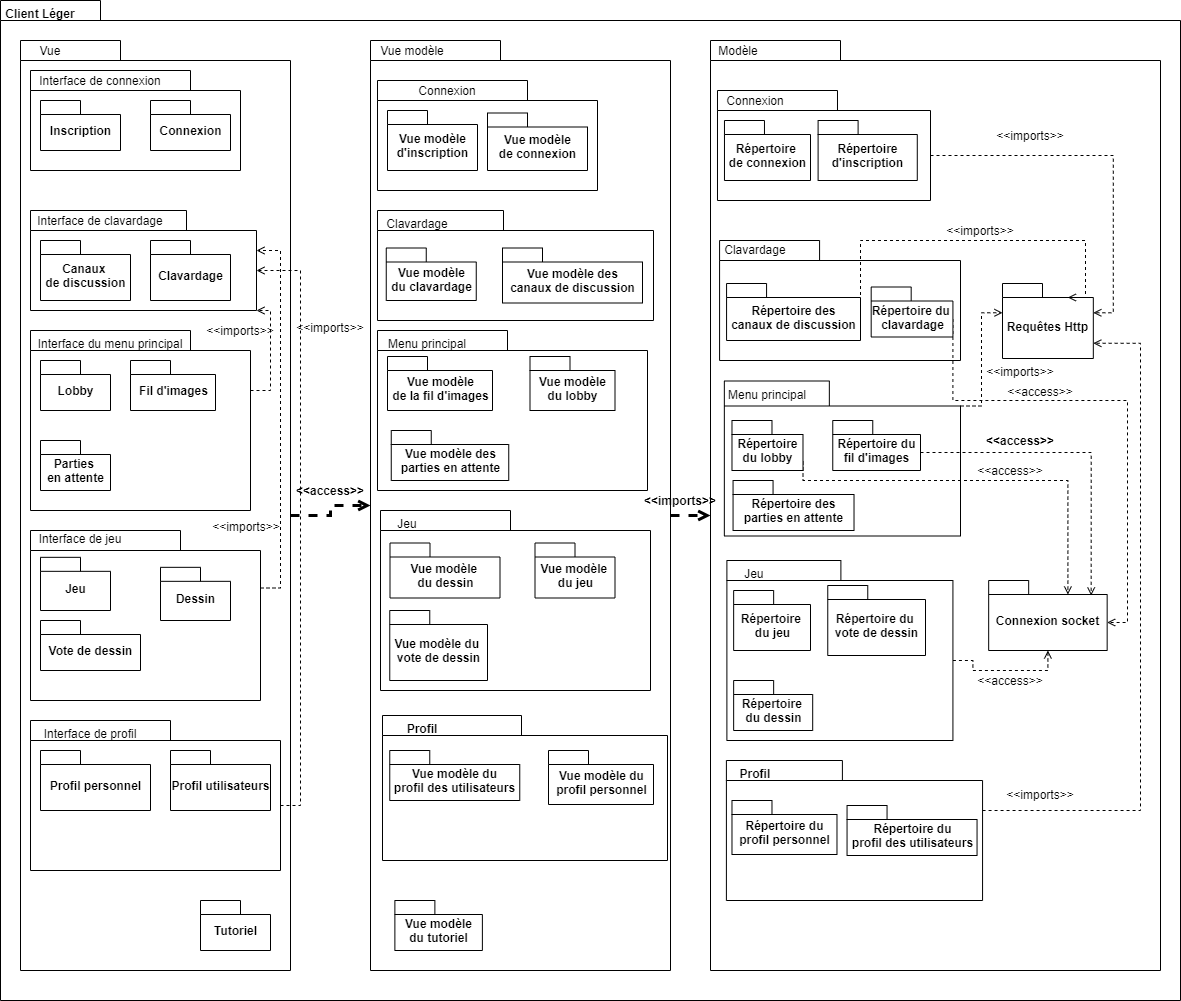


*Figure 21. Diagramme de classes pour le joueur virtuel*

## 

## 4.2 Client Léger

Le client léger est bâti selon une architecture MVVM. Cette architecture est composée de trois couches: vue, vue modèle et modèle. La vue s’occupe de regrouper les classes d’interface utilisateur. La couche vue modèle permet d’adapter les données du modèle pour la vue. La vue observe les changements de la couche vue modèle pour adapter les données qu’elle affiche. La couche vue modèle s’occupe de communiquer au modèle les changements à apporter aux données. Le modèle avertit la couche vue modèle si des données ont changé. Le diagramme à la figure 22 indique les principaux paquetages que l’on peut retrouver dans les différentes couches.

**

*Figure 22. Diagramme de paquetages pour le client léger*

### 4.2.1 Vue

*Tableau 25. Description du paquetage de l’interface de connexion*

|  |
| --- |
| **Interface de connexion** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de l’authentification et de l’inscription de l’utilisateur. |

*Tableau 26. Description du paquetage de la connexion*

|  |
| --- |
| **Connexion** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de l’authentification de l’utilisateur. |

*Tableau 27. Description du paquetage de l’inscription*

|  |
| --- |
| **Inscription** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de la création de comptes. |

*Tableau 28. Description du paquetage de l’interface de clavardage*

|  |
| --- |
| **Interface de clavardage** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent des canaux de discussion et de clavardage. |

*Tableau 29. Description du paquetage de clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de la section pour envoyer et lire les messages. |

*Tableau 30. Description du paquetage de canaux de discussion*

|  |
| --- |
| **Canaux de discussion** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter les différents canaux de discussion. |

*Tableau 31. Description du paquetage de l’interface du menu principal*

|  |
| --- |
| **Interface du menu principal** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter le menu principal. |

*Tableau 32. Description du paquetage de lobby*

|  |
| --- |
| **Lobby** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter le lobby que l’utilisateur a rejoint. |

*Tableau 33. Description du paquetage de fil d’images*

|  |
| --- |
| **Fil d’images** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter les images du fil d’images. |

*Tableau 34. Description du paquetage de parties en attente*

|  |
| --- |
| **Parties en attente** |
| Ce paquetage regroupe les classes d’interface utilisateur qui affichent les parties pouvant être rejointes par l’utilisateur. |

*Tableau 35. Description du paquetage de l’interface de jeu*

|  |
| --- |
| **Interface de jeu** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter l’interface de jeu. |

*Tableau 36. Description du paquetage de jeu*

|  |
| --- |
| **Jeu** |
| Ce paquetage regroupe les classes d’interface utilisateur qui présentent les éléments d’interface du jeu, tels que le score, le temps restant et la boîte de soumission de mot. |

*Tableau 37. Description du paquetage de dessin*

|  |
| --- |
| **Dessin** |
| Ce paquetage regroupe les classes d’interface utilisateur qui présente le dessin, ainsi que les outils pour dessiner lorsque cela est nécessaire. |

*Tableau 38. Description du paquetage de vote de dessin*

|  |
| --- |
| **Vote de dessin** |
| Ce paquetage regroupe les classes d’interface utilisateur qui affichent les dessins du joueur virtuel. |

*Tableau 39. Description du paquetage de l’interface de profil*

|  |
| --- |
| **Interface de profil** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter le profil (partie publique et partie privée) ainsi que le profil des autres utilisateurs. |

*Tableau 40. Description du paquetage de profil personnel*

|  |
| --- |
| **Profil personnel** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de permettre à l’utilisateur de voir son profil et de le modifier. |

*Tableau 41. Description du paquetage de profil d’utilisateurs*

|  |
| --- |
| **Profil utilisateur** |
| Ce paquetage regroupe les classes d’interface utilisateur qui permet à l’utilisateur de visualiser le profil d’autres utilisateurs, identifier ses amis et envoyer des demandes d’amitié. |

*Tableau 42. Description du paquetage de l’interface de tutoriel*

|  |
| --- |
| **Interface de tutoriel** |
| Ce paquetage regroupe les classes d’interface utilisateur qui s’occupent de présenter le tutoriel. |

### 4.2.3 Vue modèle

*Tableau 43. Description du paquetage de vue modèle d’inscription*

|  |
| --- |
| **Vue modèle d’inscription** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle pour l’interface d’inscription. |

*Tableau 44. Description du paquetage de vue modèle de connexion*

|  |
| --- |
| **Vue modèle de connexion** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle pour l’interface de connexion. |

*Tableau 45. Description du paquetage de vue modèle du clavardage*

|  |
| --- |
| **Vue modèle du clavardage** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle pour le clavardage. |

*Tableau 46. Description du paquetage de vue modèle des canaux de discussion*

|  |
| --- |
| **Vue modèle des canaux de discussion** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle des canaux de discussion |

*Tableau 47. Description du paquetage de vue modèle du fil d’images*

|  |
| --- |
| **Vue modèle du fil d’images** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du fil d’images |

*Tableau 48. Description du paquetage de vue modèle du lobby*

|  |
| --- |
| **Vue modèle du lobby** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du lobby |

*Tableau 49. Description du paquetage de vue modèle des parties en attente*

|  |
| --- |
| **Vue modèle des parties en attente** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle des parties en attentes |

*Tableau 50. Description du paquetage de vue modèle du dessin*

|  |
| --- |
| **Vue modèle du dessin** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du dessin |

*Tableau 51. Description du paquetage de vue modèle du jeu*

|  |
| --- |
| **Vue modèle du jeu** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du jeu |

*Tableau 52. Description du paquetage de vue modèle du vote*

|  |
| --- |
| **Vue modèle du vote de dessin** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle vote du dessin |

*Tableau 53. Description du paquetage de vue modèle du profil des utilisateurs*

|  |
| --- |
| **Vue modèle du profil des utilisateurs** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle profil des utilisateurs |

*Tableau 54. Description du paquetage de vue modèle du profil personnel*

|  |
| --- |
| **Vue modèle du profil personnel** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du profil personnel |

*Tableau 55. Description du paquetage de vue modèle du tutoriel*

|  |
| --- |
| **Vue modèle du tutoriel** |
| Ce paquetage regroupe les classes qui préparent les données provenant du modèle du tutoriel |

### 4.2.3 Modèle

*Tableau 56. Description du paquetage de modèle de répertoire de connexion*

|  |
| --- |
| **Répertoire de connexion** |
| Ce paquetage regroupe les classes qui s’occupent de conserver les données et d’obtenir les données provenant du serveur pour l’authentification de l’utilisateur. |

*Tableau 57. Description du paquetage de modèle de répertoire d’inscription*

|  |
| --- |
| **Répertoire d’inscription** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour l’inscription de l’utilisateur. |

*Tableau 58. Description du paquetage de modèle de répertoire des canaux de discussion*

|  |
| --- |
| **Répertoire des canaux de discussion** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour les canaux de discussion. |

*Tableau 59. Description du paquetage de modèle de répertoire du clavardage*

|  |
| --- |
| **Répertoire du clavardage** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le clavardage. |

*Tableau 60. Description du paquetage de modèle de répertoire du lobby*

|  |
| --- |
| **Répertoire du lobby** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le lobby. |

*Tableau 61. Description du paquetage de modèle de répertoire du fil d’images*

|  |
| --- |
| **Répertoire du fil d’images** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le fil d’images. |

*Tableau 62. Description du paquetage de modèle de répertoire des parties en attente*

|  |
| --- |
| **Répertoire des parties en attente** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour les parties en attente. |

*Tableau 63. Description du paquetage de modèle de répertoire du jeu*

|  |
| --- |
| **Répertoire du jeu** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le jeu. |

*Tableau 64. Description du paquetage de modèle de répertoire du vote*

|  |
| --- |
| **Répertoire du vote de dessin** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le vote de dessin. |

*Tableau 65. Description du paquetage de modèle de répertoire du dessin*

|  |
| --- |
| **Répertoire du dessin** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le dessin. |

*Tableau 66. Description du paquetage de modèle de répertoire du profil personnel*

|  |
| --- |
| **Répertoire du profil personnel** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le profil personnel d’un utilisateur. |

*Tableau 67. Description du paquetage de modèle de répertoire du profil des utilisateurs*

|  |
| --- |
| **Répertoire du profil des utilisateurs** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour le profil des utilisateurs. |

*Tableau 68. Description du paquetage de modèle de répertoire des requêtes Http*

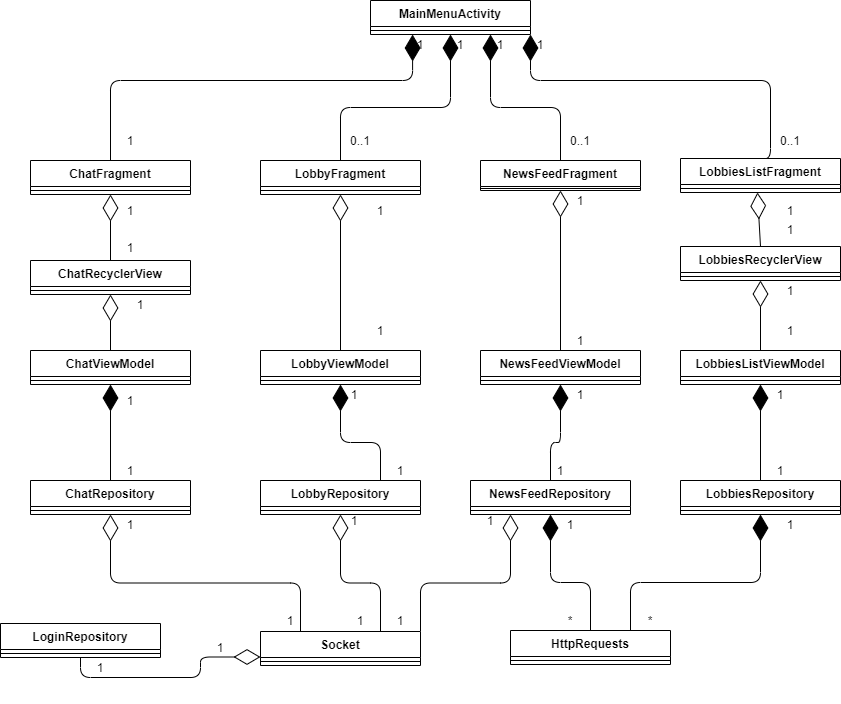
|  |
| --- |
| **Requêtes Http** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour |

*Tableau 69. Description du paquetage de modèle de connexion socket*

|  |
| --- |
| **Connexion socket** |
| Ce paquetage regroupe les classes qui s'occupent de conserver les données et d’obtenir les données provenant du serveur pour |

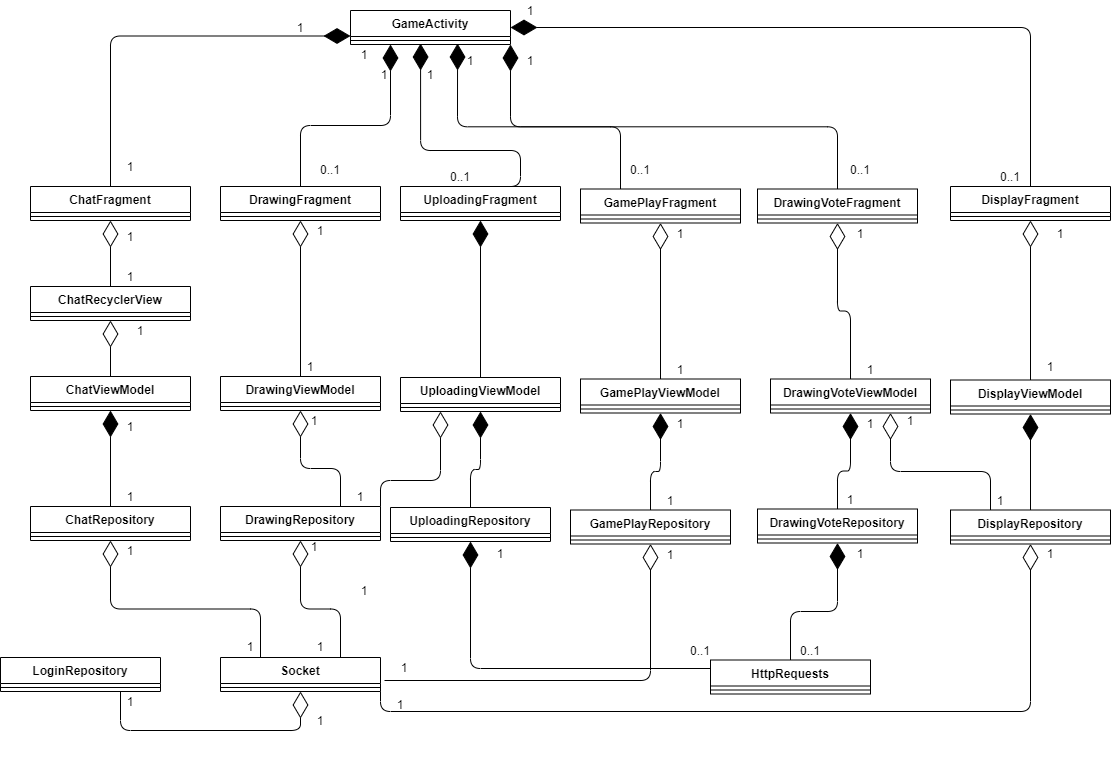
### 4.2.4 Diagrammes de classes

Le diagramme de la figure 23 présente les différentes classes en lien avec les fonctionnalités du menu principal. On retrouve les fragments pour le clavardage, la salle d’attente, le fil d’images ainsi que la liste des parties en attente de débuter.



*Figure 23. Diagramme de classes pour les différentes composantes du menu principal du client léger*

Le diagramme de la figure 24 présente les différentes classes en lien avec les fonctionnalités de jeu. On voit que le clavardage se retrouve également dans ce diagramme puisque le clavardage doit être disponible dans l’activité du menu principal ainsi que dans l’activité de jeu.



*Figure 24. Diagramme de classes pour les différentes composantes de la portion jeu pour le client léger*

## 

## 

## 

## 

## 

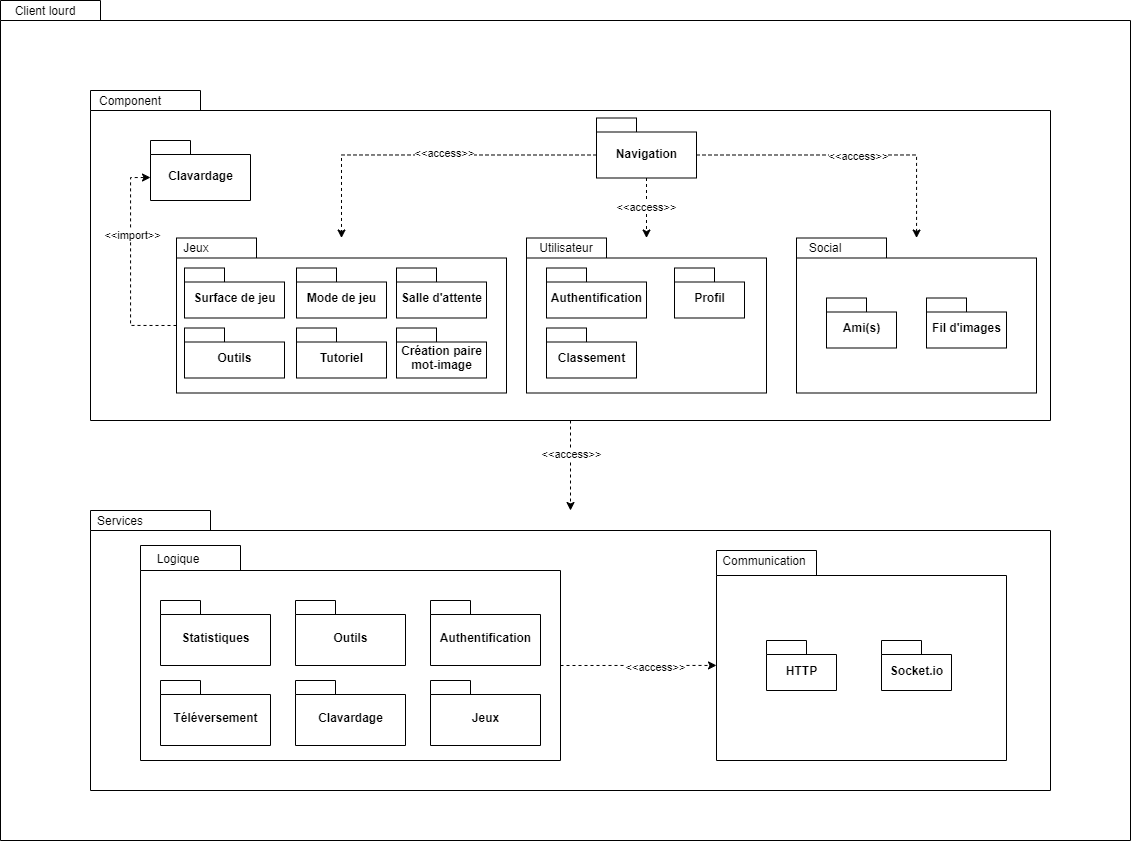
## 

## 

## 

## 4.3 Client lourd

Le client lourd est composé de “components” et de “services”. Ces termes sont propres au cadriciel de développement Angular. Les “components” contiennent les éléments d’interfaces (HTML et CSS) et un fichier Typescript servant à fournir les données au “template”. Les services offrent une logique de persistance de données qui n’est pas possible seulement avec les “components”. Le diagramme suivant représente la séparation de la logique du client lourd en “components” et en services.



*Figure 25. Diagramme de paquetages du client lourd*

### 4.3.1 Component

*Tableau 70. Description du paquetage de component clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les components responsables de la logique du clavardage utilisé dans toute l’application. |

*Tableau 71. Description du paquetage de component navigation*

|  |
| --- |
| **Navigation** |
| Ce paquetage regroupe les components responsables de la navigation entre les différentes composantes de l’application. |

*Tableau 72. Description du paquetage de component surface de jeu*

|  |
| --- |
| **Surface de jeu** |
| Ce paquetage regroupe les components englobants les différentes fonctionnalités utilisées dans le cadre du jeu. |

*Tableau 73. Description du paquetage de component de mode de jeu*

|  |
| --- |
| **Mode de jeu** |
| Ce paquetage regroupe les components responsables des différents modes de jeu dans leur entièreté. |

*Tableau 74. Description du paquetage de component de salle d’attente*

|  |
| --- |
| **Salle d’attente** |
| Ce paquetage regroupe les components responsables de la salle d’attente d’une partie, incluant l’affichage des parties en cours visible du menu principal. |

*Tableau 75. Description du paquetage de component d’outils*

|  |
| --- |
| **Outils** |
| Ce paquetage regroupe les components responsables de l’interaction avec le canevas de dessin. |

*Tableau 76. Description du paquetage de component de tutoriel*

|  |
| --- |
| **Tutoriel** |
| Ce paquetage regroupe les components responsables du tutoriel. |

*Tableau 77. Description du paquetage de component d’authentification*

|  |
| --- |
| **Authentification** |
| Ce paquetage regroupe les components responsables du mécanisme d’authentification (enregistrement, connexion, déconnexion et protection des parties inaccessibles de l’application sans authentification). |

*Tableau 78. Description du paquetage de component de profil*

|  |
| --- |
| **Profil** |
| Ce paquetage regroupe les components responsables de la gestion du profil d’un utilisateur. |

*Tableau 79. Description du paquetage de component de classement*

|  |
| --- |
| **Classement** |
| Ce paquetage regroupe les contrôleurs responsables de la gestion des routes du REST API et des Sockets en lien avec l’authentification des utilisateurs. |

*Tableau 80. Description du paquetage de component d’amis*

|  |
| --- |
| **Ami(s)** |
| Ce paquetage regroupe les components responsables du mécanisme d’amitié de l’application |

*Tableau 81. Description du paquetage de component de fil d’images*

|  |
| --- |
| **Fil d’images** |
| Ce paquetage regroupe les components responsables du fil d’images de la page principale. |

*Tableau 82. Description du paquetage de component de création paire mot-image*

|  |
| --- |
| **Création paire mot-image** |
| Ce paquetage regroupe les components responsables du mécanisme d’ajout d’une paire mot-image à la banque utilisée par les joueurs virtuels. |

### 4.3.2 Services

*Tableau 83. Description du paquetage de service de statistiques*

|  |
| --- |
| **Statistiques** |
| Ce paquetage regroupe les services responsables de la gestion des informations utilisées pour le profil utilisateur et le système de classement. |

*Tableau 84. Description du paquetage de service d’outils*

|  |
| --- |
| **Outils** |
| Ce paquetage regroupe les services responsables du fonctionnement des outils utilisés pour dessiner sur le canevas. |

*Tableau 85. Description du paquetage de service d’authentification*

|  |
| --- |
| **Authentification** |
| Ce paquetage regroupe les services chargés de traiter les requêtes en lien avec l’authentification. |

*Tableau 69. Description du paquetage de modèle de connexion socket*

|  |
| --- |
| **Téléversement** |
| Ce paquetage regroupe les services responsables du téléversement d’un dessin vers le serveur. |

*Tableau 86. Description du paquetage de service de clavardage*

|  |
| --- |
| **Clavardage** |
| Ce paquetage regroupe les services chargés de gérer les données utilisées par le système de clavardage. |

*Tableau 87. Description du paquetage de service de jeux*

|  |
| --- |
| **Jeux** |
| Ce paquetage regroupe les services responsables des fonctionnalités pour les modes de jeu qui ne seront pas gérées dans les components. |

*Tableau 88. Description du paquetage de service Http*

|  |
| --- |
| **HTTP** |
| Ce paquetage regroupe les services responsables des requêtes http. |

*Tableau 89. Description du paquetage de service de socket*

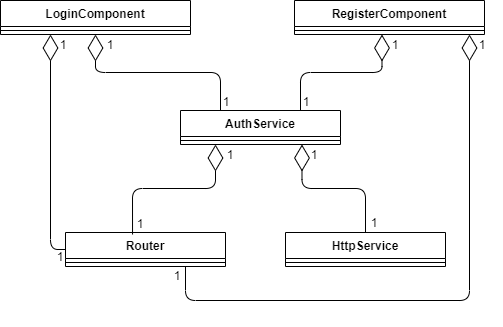
|  |
| --- |
| **Socket.io** |
| Ce paquetage regroupe les services responsables des communications utilisant socket.io. |

### 

### 4.3.3 Diagramme de classes des principales fonctionnalités du client lourd

#### 4.3.3.1 Authentification

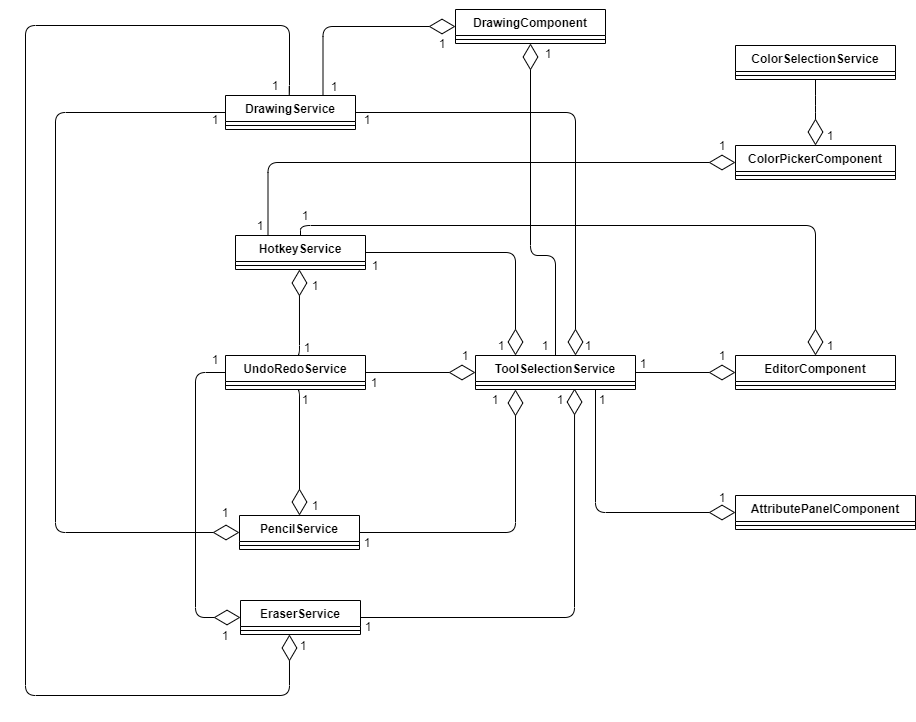
Le diagramme ci-dessous montre les interactions entre les classes du système d’authentification pour le client lourd.



*Figure 26. Diagramme de classes du système d’authentification*

#### 4.3.3.2 Dessiner

Le diagramme ci-dessous montre les interactions entre les classes du système de dessin pour le client lourd.



*Figure 27. Diagramme de classes pour le système de dessin*

#### 4.3.3.3 Clavardage

Le diagramme ci-dessous montre les interactions entre les classes du système de clavardage pour le client lourd.

#### 

*Figure 28. Diagramme de classes pour le système de clavardage*

#### 4.3.3.4 Mécanisme de jeu

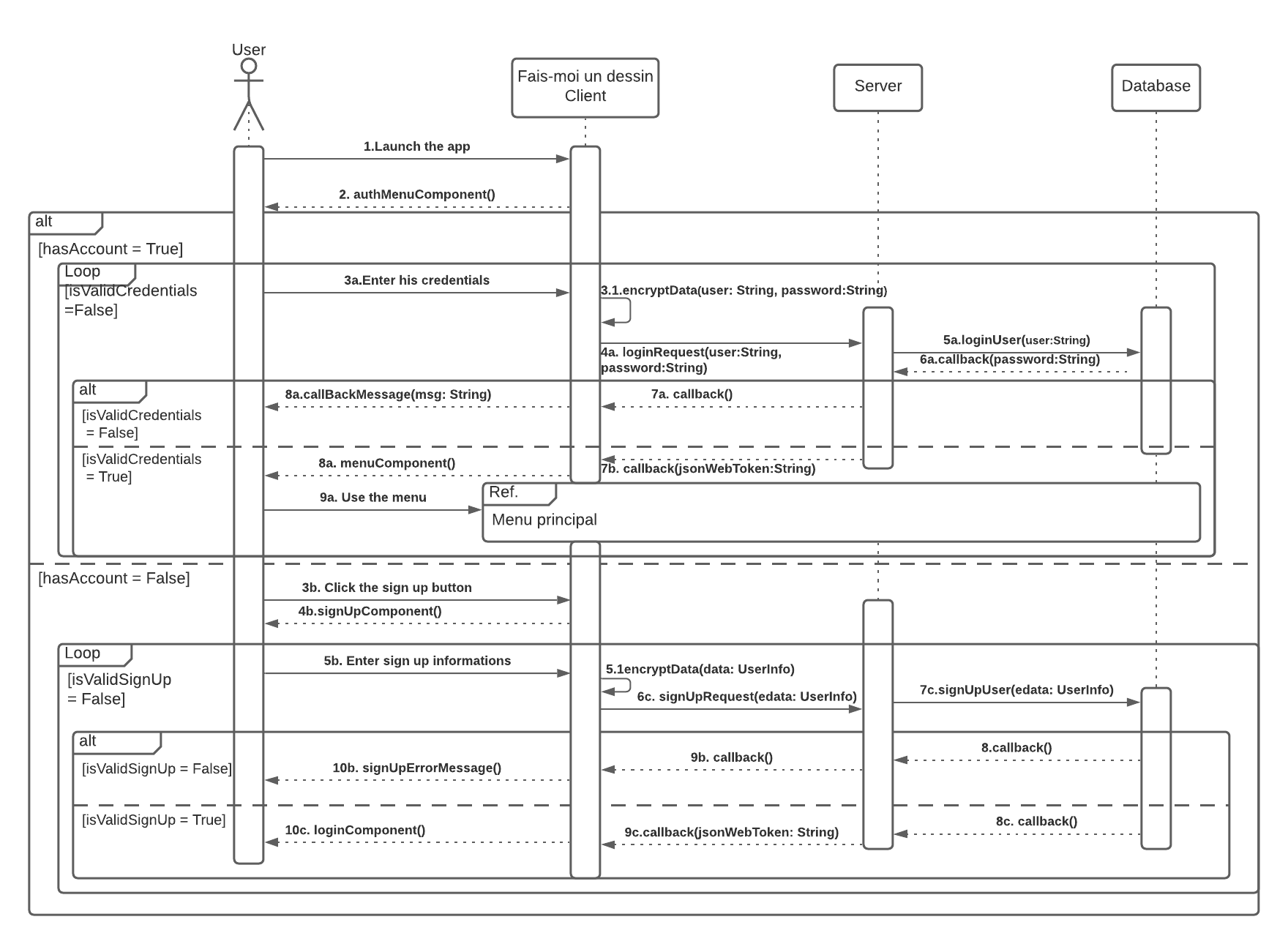
Le diagramme ci-dessous montre les interactions entre les classes du composant le mécanisme de jeu pour le client lourd.

#### 

*Figure 29. Diagramme de classes pour le système de jeu*

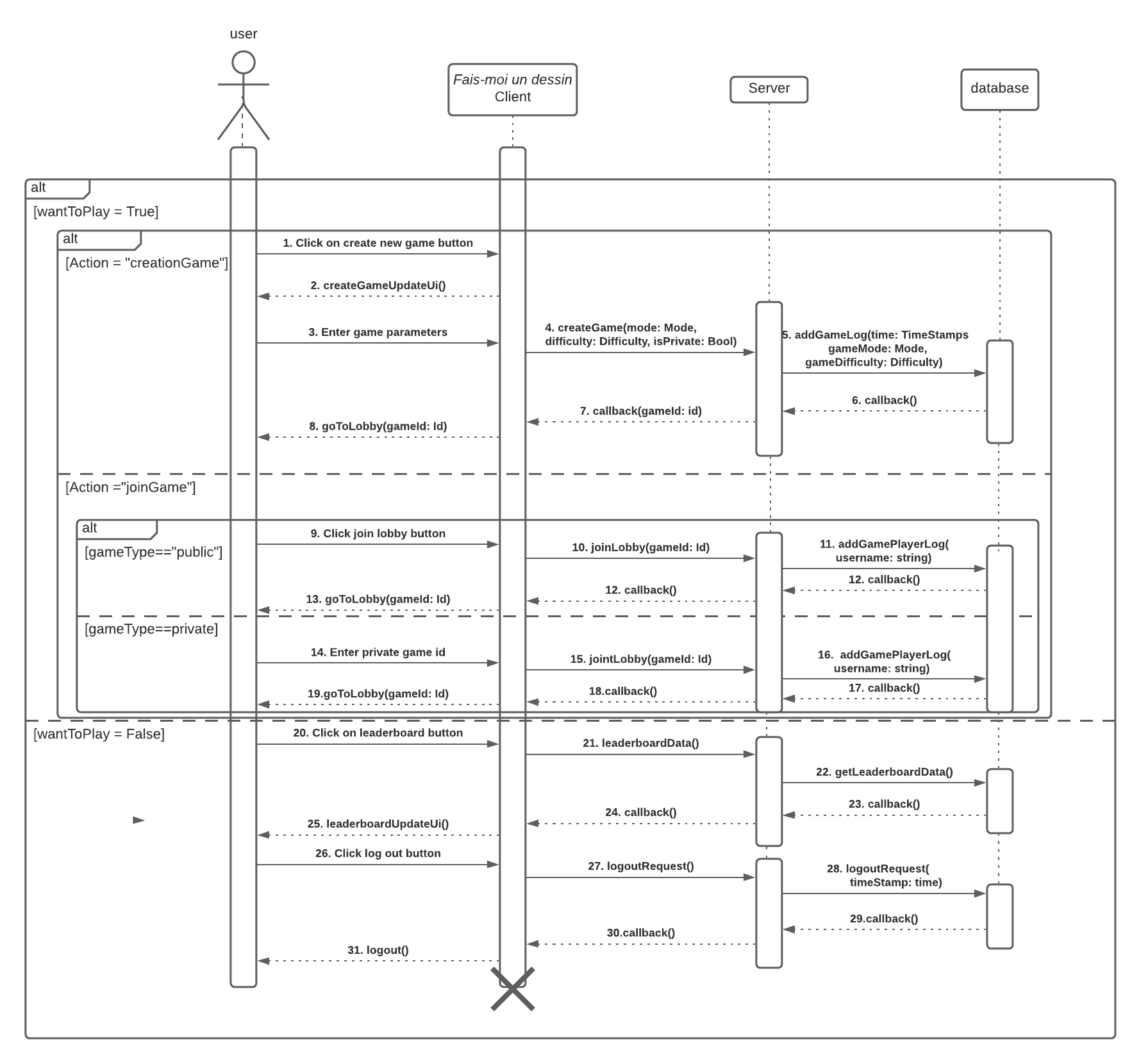
# 5. Vue des processus

## 5.1 Connexion

**

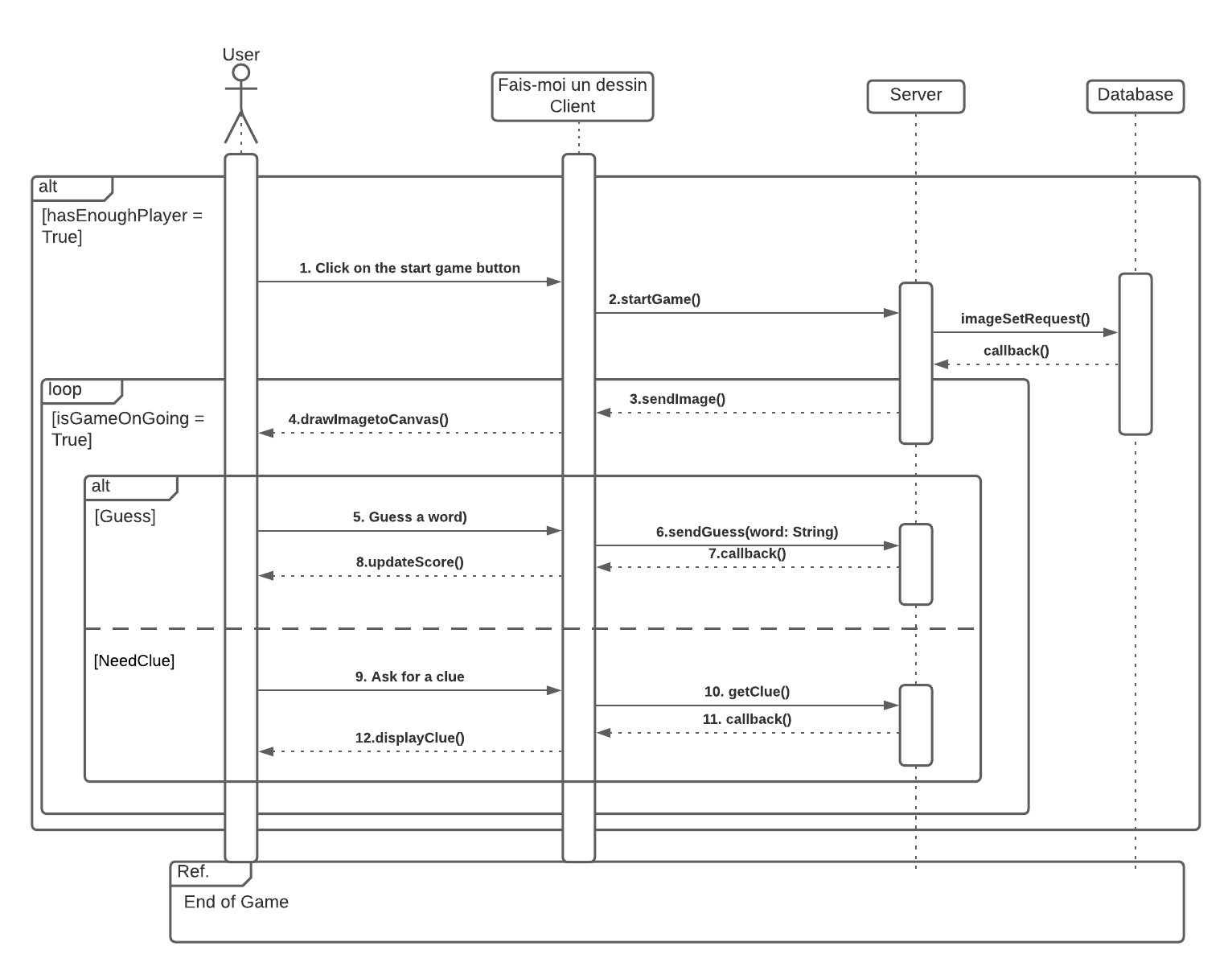
*Figure 30. Diagramme de séquence du système de connexion*

## 5.2 Création d’une partie

**

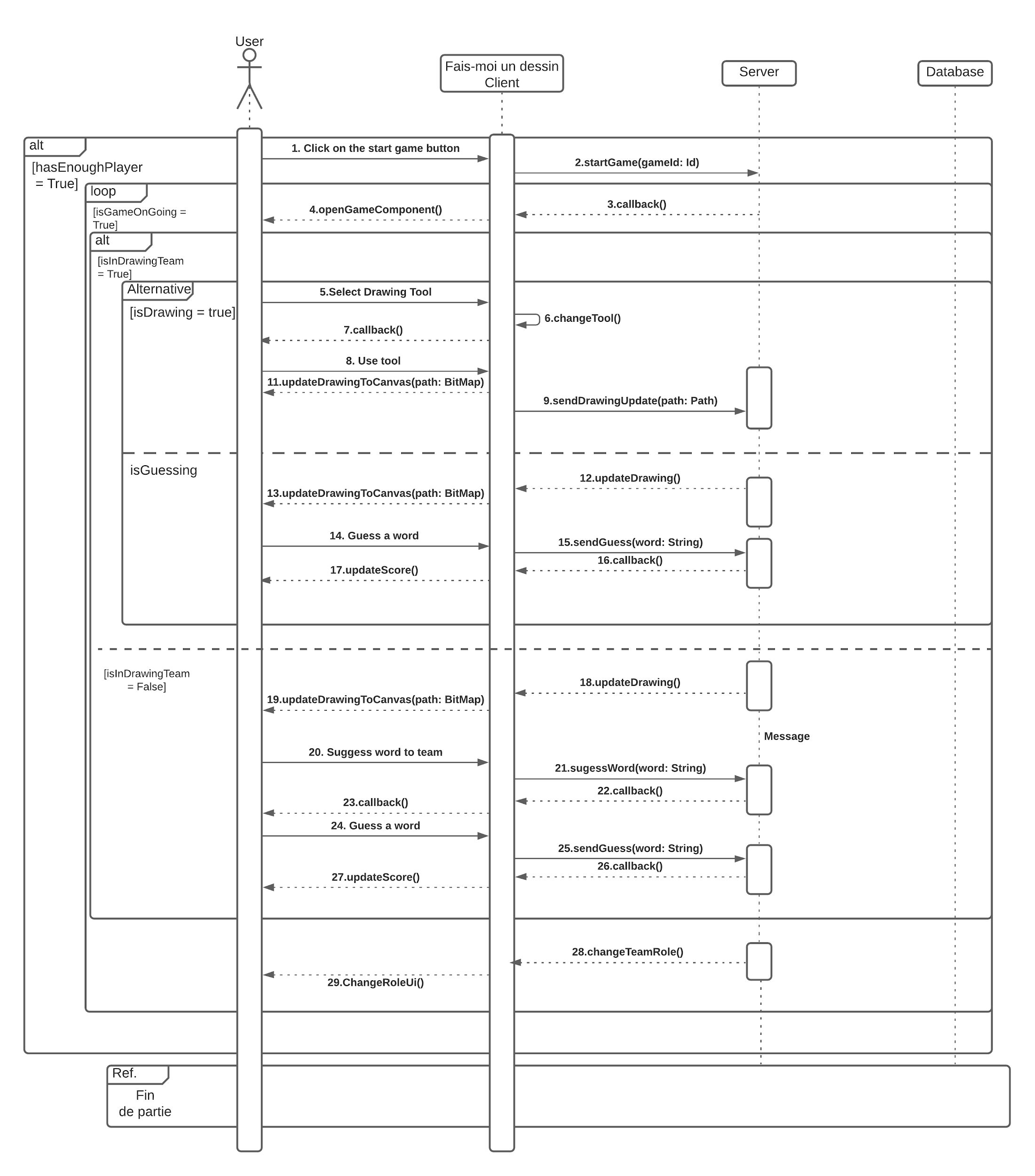
*Figure 31. Diagramme de séquence pour la création d’une partie*

## 5.3 Partie sprint solo et coopérative

**

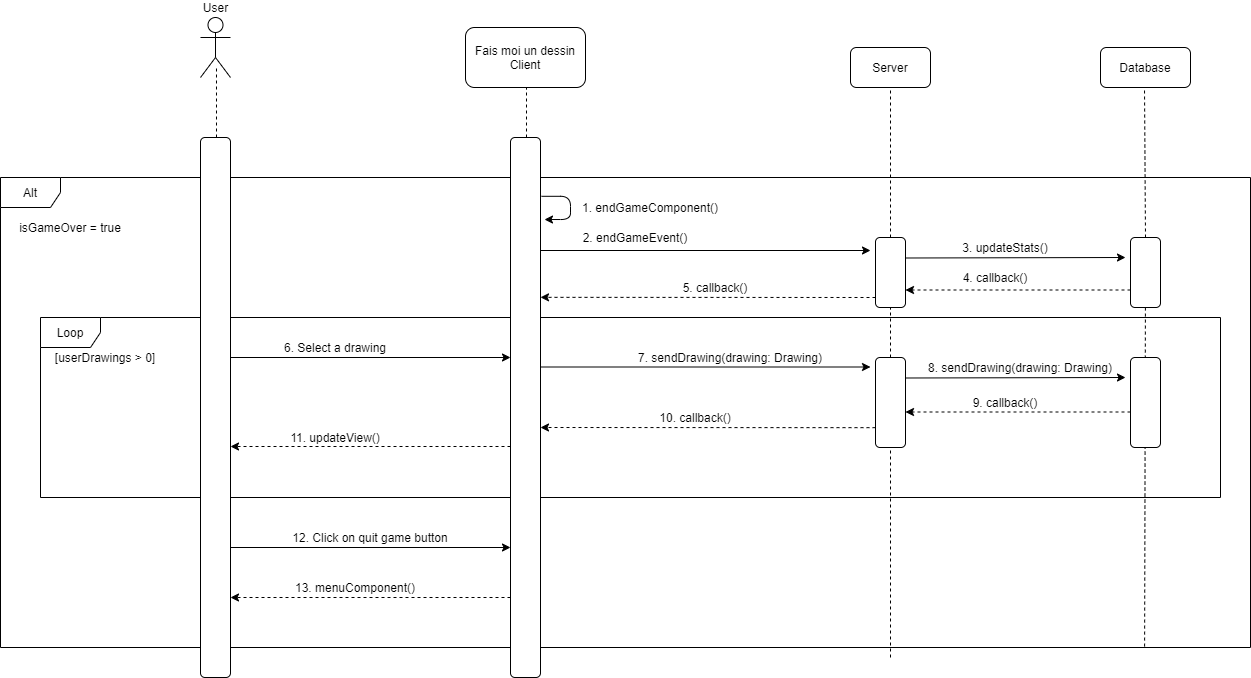
*Figure 32. Diagramme de séquence pour les modes de jeu sprint solo et coopératif*

## 5.4 Partie mode classique

****

*Figure 33. Diagramme de séquences pour le mode de jeu classique*

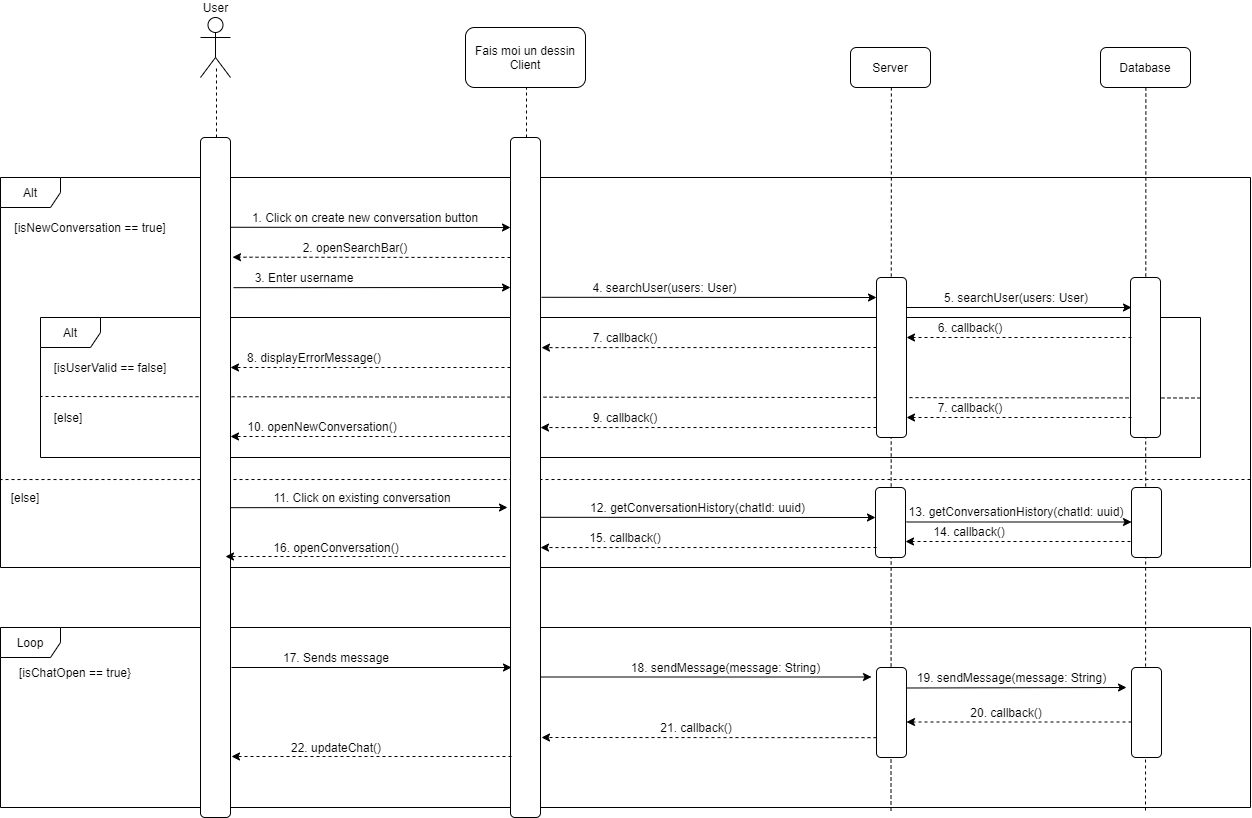
## 5.5 Fin de partie

****

*Figure 34. Diagramme de séquence pour la fin d’une partie*

# 

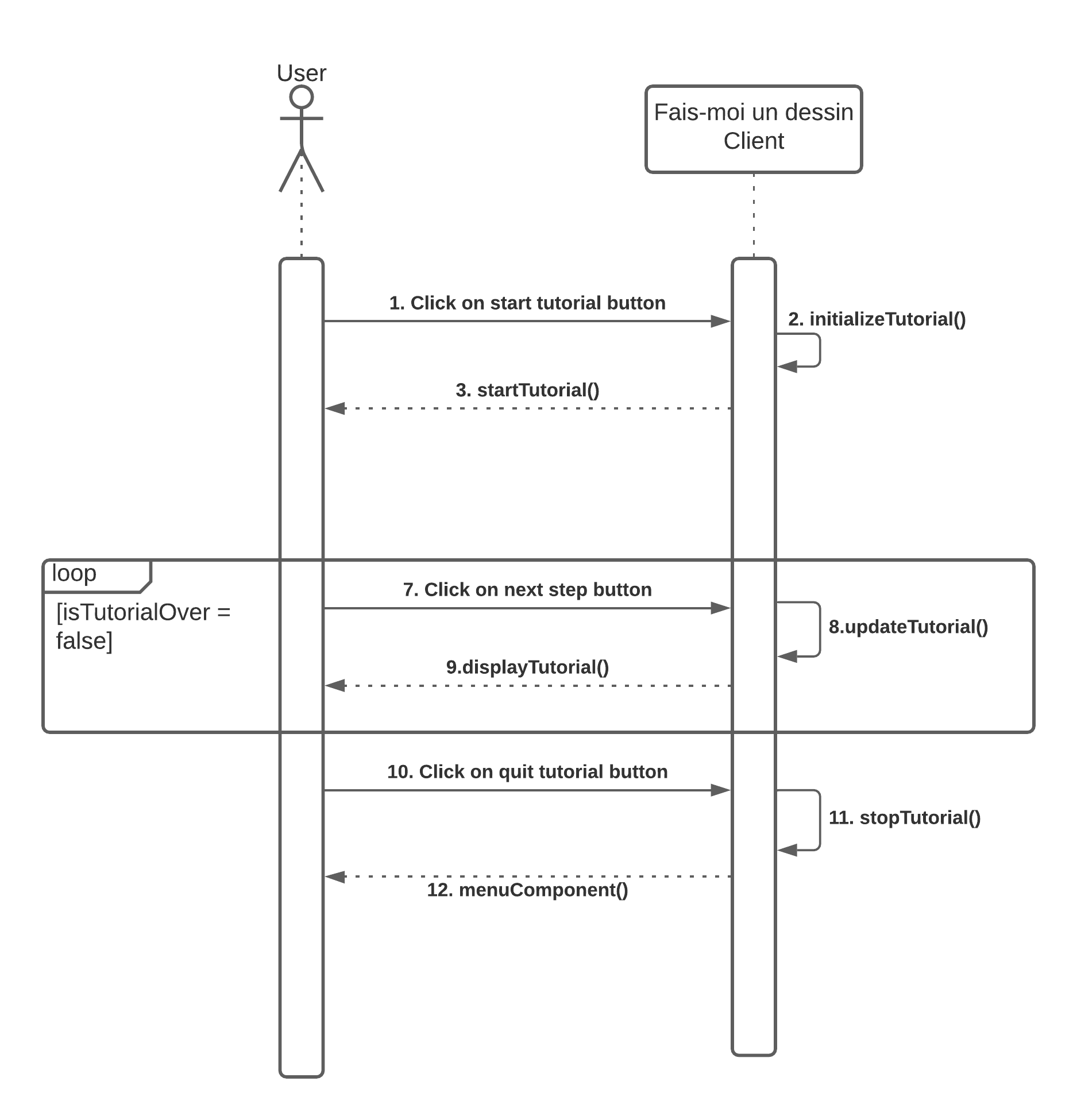
## 5.6 Clavardage

****

*Figure 35. Diagramme de séquence pour le système de clavardage*

# 

## 5.7 Tutoriel

**

*Figure 36. Diagramme de séquence pour le tutoriel*

# 

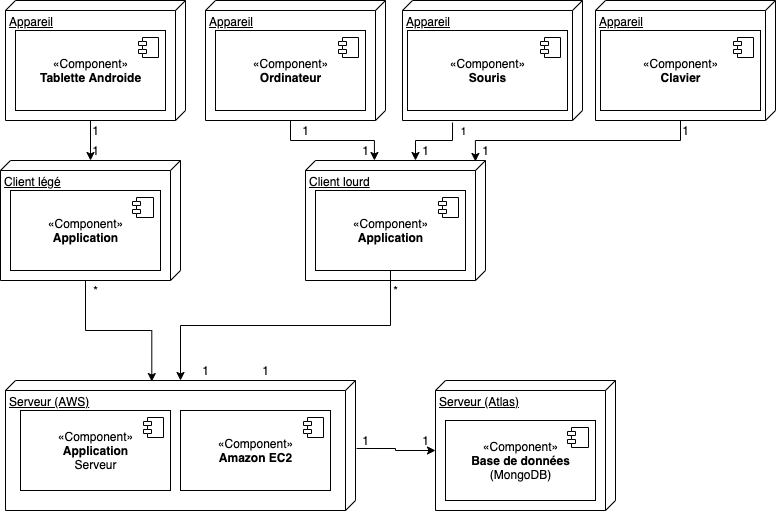
## 5.8 Requêtes au serveur

# 

*Figure 37. Diagramme de séquence pour le mécanisme de requêtes au serveur*

# 

# 6. Vue de déploiement



*Figure 38. Diagramme de déploiement du système*

# 7. Taille et performance

Pour la taille et la performance de l’application, plusieurs éléments sont à prendre en considération. Le serveur, le client lourd ainsi que le client léger fonctionnent sur des appareils ayant des capacités très différentes. Le serveur, construit à l’aide du cadriciel Node.js, sera hébergé par Amazon Web Service. En utilisant Amazon Web Service, nous n’avons pas à ajuster le serveur en fonction de la mémoire requise ou de l’intensité des calculs, car il est possible d’obtenir ces éléments à la demande. Par contre, utiliser beaucoup de mémoire ou utiliser des algorithmes intensifs en calcul à un coût monétaire. Il faudra donc tenter de les limiter.

Pour le client lourd, l’application doit fonctionner sur un ordinateur avec 8 Go de mémoire vive et un processeur i5. Nous souhaitons limiter l’utilisation de la RAM à 300 Mo et que l’utilisation du processeur ne dépasse pas 10% pour s’assurer que l’expérience utilisateur soit plaisante et que l’utilisateur puisse continuer d’utiliser d’autres applications en même temps. De plus, l’application ne devra pas dépasser 500 Mo d’espace disque.

Pour le client léger, l’application doit fonctionner sur une tablette Android possédant 2 Go de mémoire vive, 32 Go de mémoire interne ainsi qu’un processeur Exynos 7904A. Le client léger doit donc fonctionner sur un appareil moins performant que le client lourd. La mémoire vive est particulièrement limitée sur l’appareil et il sera important de limiter son utilisation à 100Mo.