
Équipe 102

Fais-moi un dessin
Document d'architecture logicielle

Version 1.8

Historique des révisions

Date	Version	Description	Auteur
2021-01-26	1.0	Première ébauche des sections 1, 2 et 7	Vlad Dreliciuc
2021-02-04	1.1	Première partie des vues de cas d'utilisation	Laura Beaudoin
2021-02-08	1.2	Vue de déploiement	Vlad Dreliciuc
2021-02-10	1.3	Vue de processus	Ming Xiao Yuan
2021-02-11	1.4	Vue logique diagramme de paquetages client lourd	Vlad Dreliciuc
2021-02-13	1.5	Vue de processus révision	Andi Podgorica
2021-02-15	1.6	Vue logique diagramme de paquetages client léger	Andi Podgorica
2021-02-18	1.7	Révision finale	Ming Xiao Yuan
2021-04-18	1.8	Corrections suite aux commentaires de l'appel d'offre	Vlad Dreliciuc

Table des matières

1. Introduction	4
2. Objectifs et contraintes architecturales	4
3. Vue des cas d'utilisation	5
4. Vue logique	8
4.1 Client lourd	8
4.2 Client léger	12
5. Vue des processus	17
6. Vue de déploiement	22
7. Taille et performance	23

Document d'architecture logicielle

1. Introduction

Ce document présente une modélisation de l'architecture de notre application. D'abord, on y définit les objectifs et contraintes architecturales de notre application. Ensuite, nous y présentons différentes vues architecturales, telles que la vue des cas d'utilisation, la vue logique, la vue des processus et la vue de déploiement. Enfin, nous aborderons les caractéristiques de taille et de performance ayant un impact sur le design de notre logiciel.

2. Objectifs et contraintes architecturales

Une des principales contraintes techniques de notre application est la plateforme sur laquelle celle-ci devra rouler. En effet, notre application doit être conçue de façon à pouvoir être exécutée sur une tablette roulant sous Android 9.0 ("client léger"), de même que sur un ordinateur Windows 10 ("client lourd").

Une autre contrainte est que les deux clients doivent pouvoir communiquer ensemble à travers un protocole robuste et léger, sans avoir à se soucier de la technologie du serveur qui relie le tout et qui s'occupe du stockage.

Pour ce qui est des contraintes de développement et de déploiement, le client léger sera développé en Kotlin sur Android Studio, alors que le client lourd sera développé en Angular/TypeScript sur Visual Studio Code. Lors du déploiement, le client lourd sera enveloppé ("wrappé") par le cadriciel Electron, qui lui permettra de rouler sur un ordinateur sans nécessiter un fureteur web. Enfin, le serveur sera développé sur une combinaison de Google Cloud Functions et de Google Firestore.

Nous visons également à ce que notre code soit conçu de manière à favoriser sa réutilisabilité. Pour ce faire, nous séparerons du mieux possible chacune des fonctionnalités du code, nous le commenterons lorsque c'est pertinent et nous développerons en anglais afin que le code soit accessible à une plus grande audience.

Enfin, toutes les données sensibles de nos utilisateurs (notamment les mots de passe) seront "hachées" sur le client avant d'être envoyées au serveur et ce dernier les comparera à une version déjà hachée pour éviter la transmission et le stockage des données sensibles, par souci de confidentialité et de sécurité.

3. Vue des cas d'utilisation

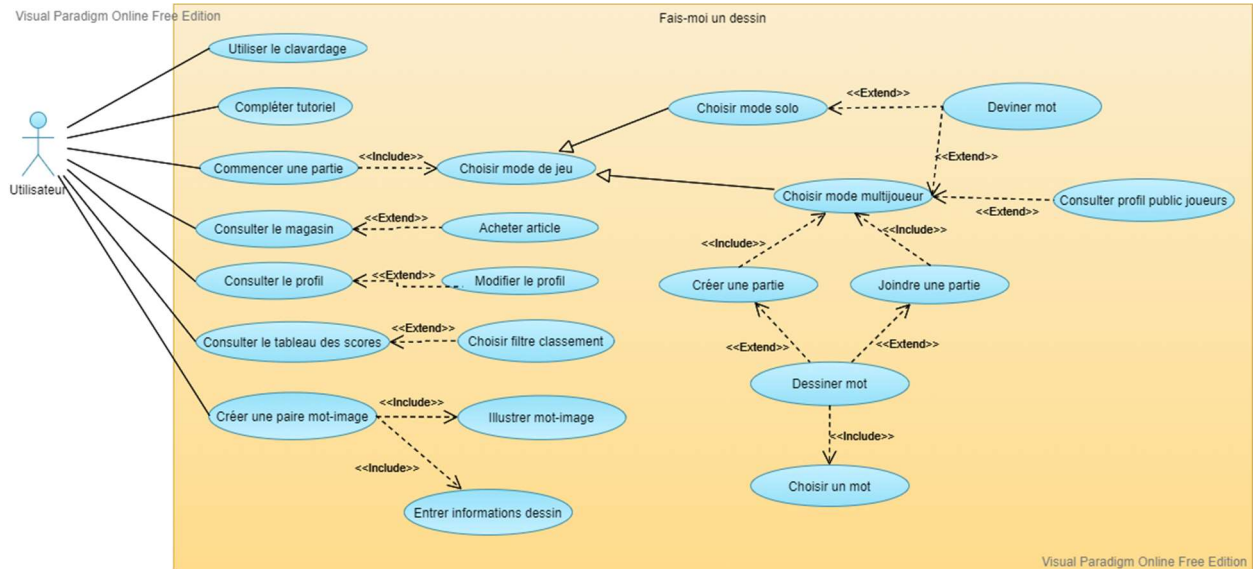


Figure 1. Diagramme de cas d'utilisation - Application générale

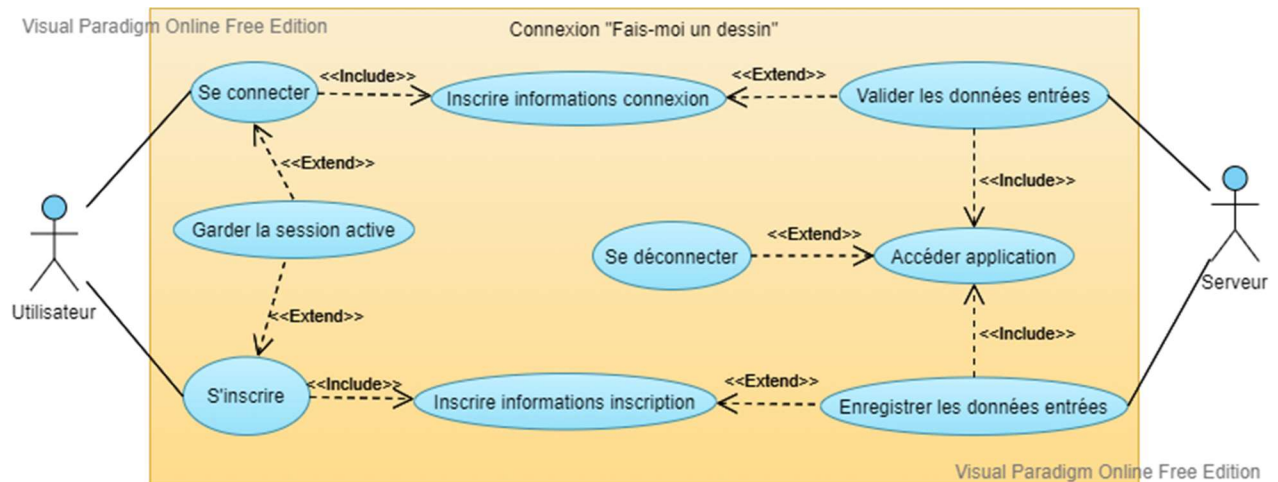


Figure 2. Diagramme de cas d'utilisation - Connexion

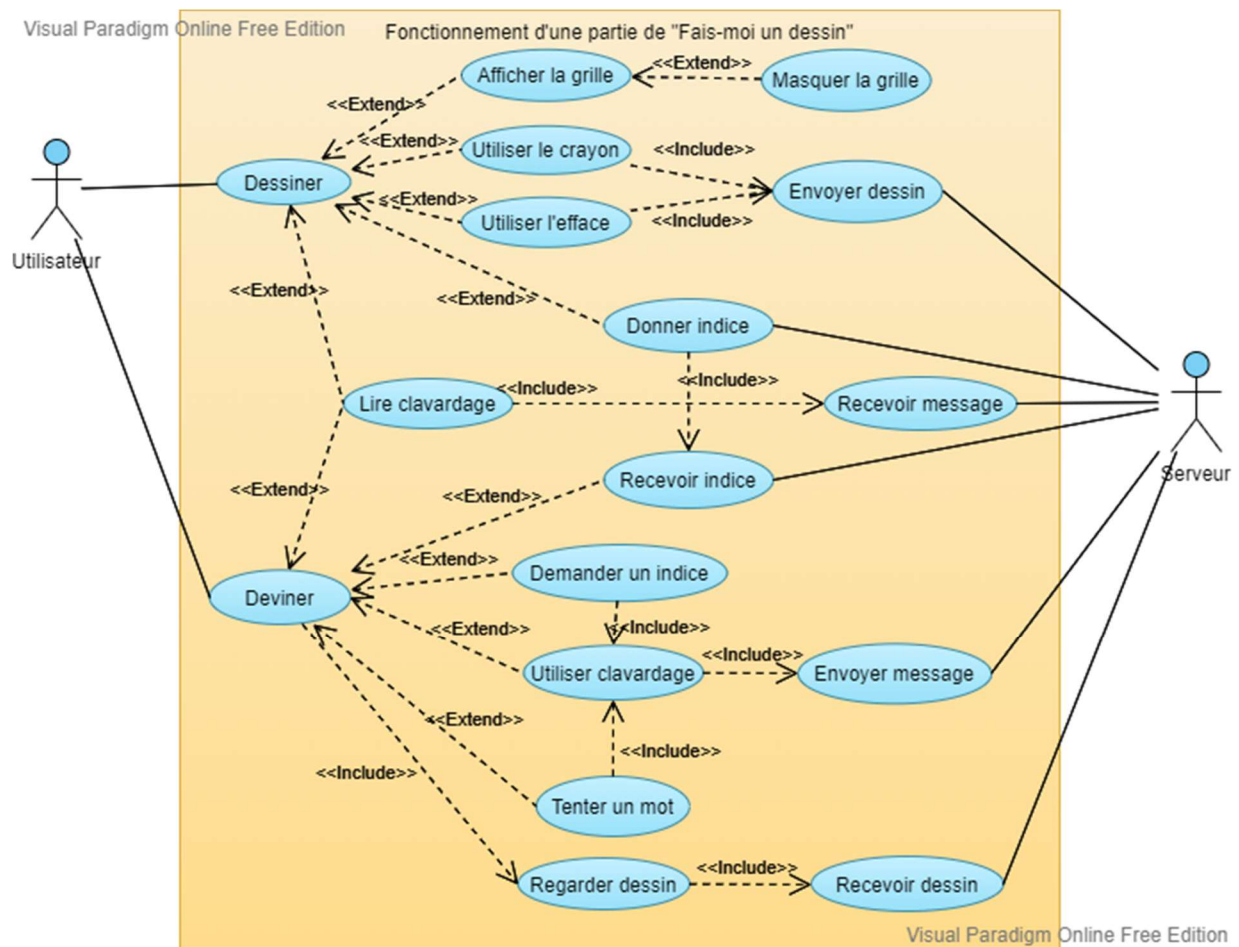


Figure 3. Diagramme de cas d'utilisation - Application lors d'une partie multijoueur

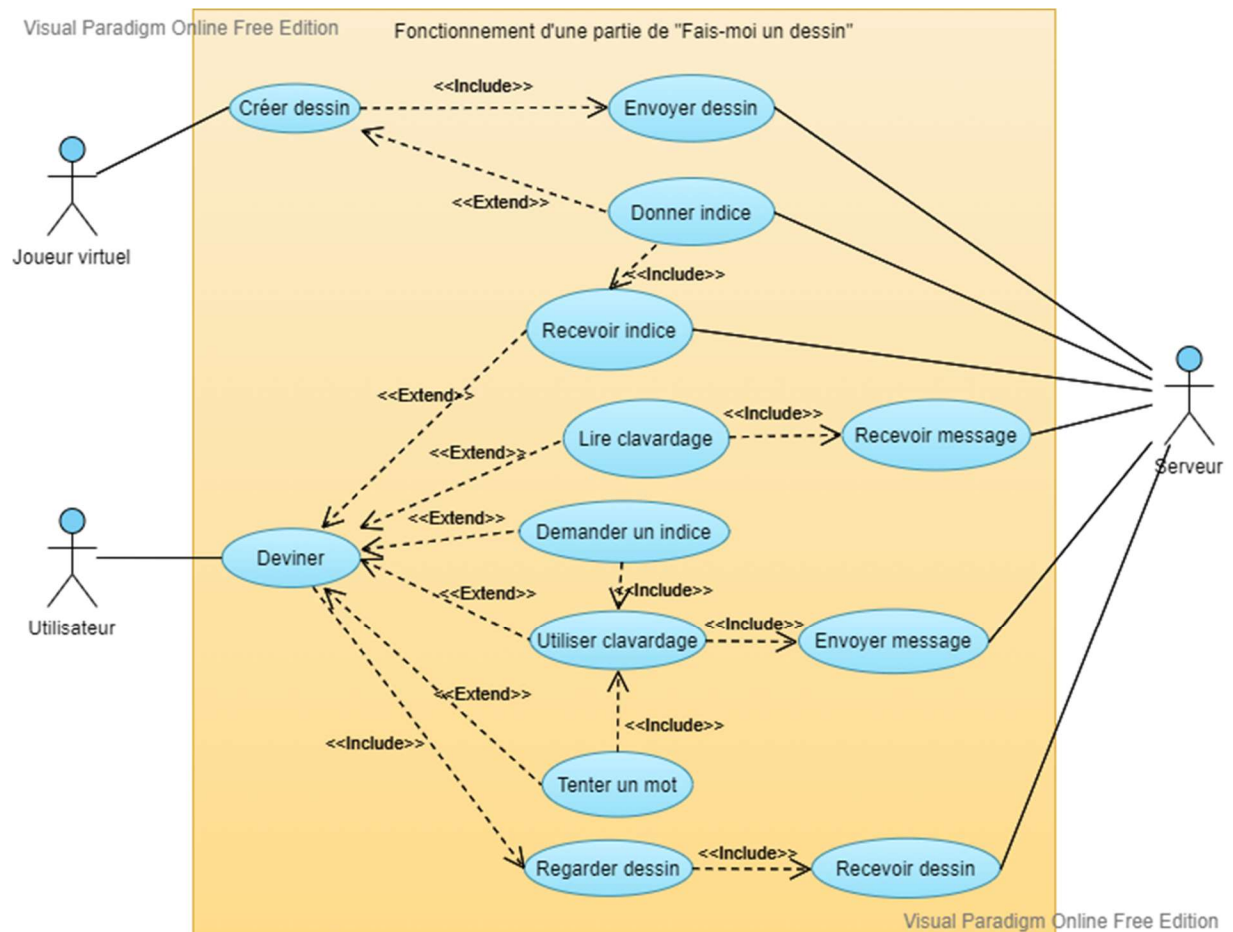


Figure 4. Diagramme de cas d'utilisation - Partie Solo Sprint

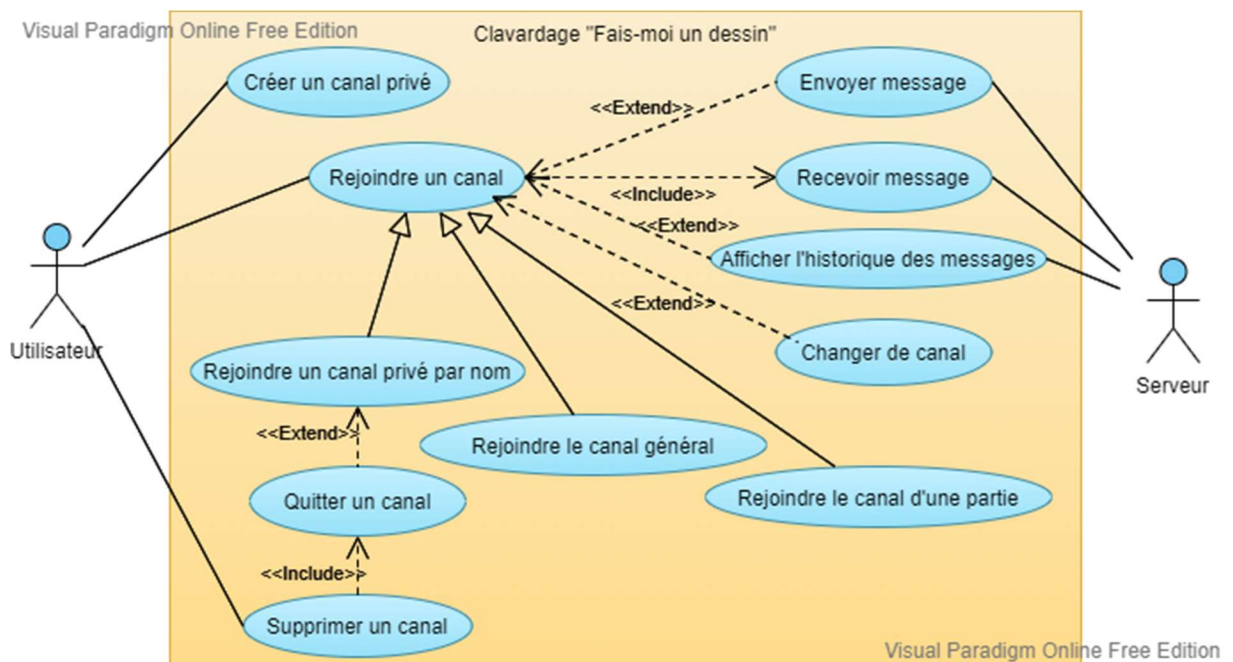


Figure 5. Diagramme de cas d'utilisation - Clavardage

4. Vue logique

4.1 Client lourd

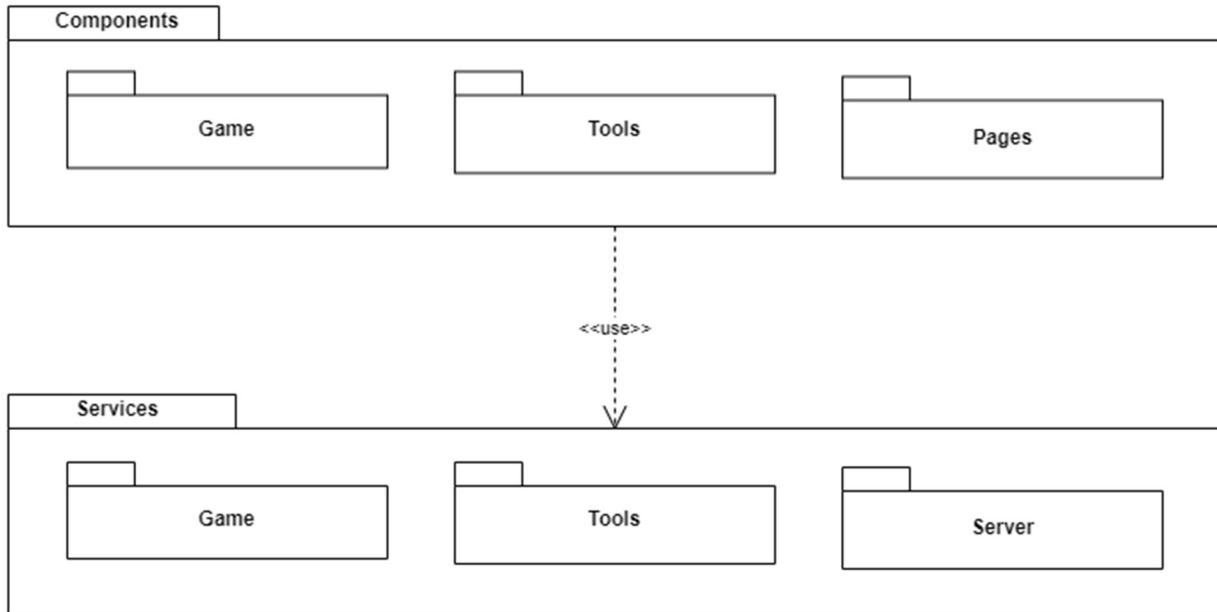


Figure 6. Diagramme de paquets (Lourd)- Fais-moi un dessin

Components

Ce paquetage de haut niveau contient toutes les composantes liées à la vue de l'application.

Components.Game

Ce paquetage contient toutes les composantes liées au fonctionnement d'une partie (espace de dessin, infos sur la partie, chat, etc.).

Components.Tools

Ce paquetage contient toutes les composantes liées aux outils (crayon, efface, grille, couleurs, etc.).

Components.Pages

Ce paquetage contient toutes les composantes liées aux pages (menu principal, connexion, magasin, etc.).

Services

Ce paquetage de haut niveau contient tous les services liés au fonctionnement de l'application.

Services.Game

Ce paquetage contient tous les services liés au fonctionnement d'une partie

(manipulation de SVG, traitement des événements, undo-redo, etc.).

Services.Tools

Ce paquetage contient tous les services liés au fonctionnement des outils (sélecteur d'outils, propriété des outils, etc.).

Services.Server

Ce paquetage contient tous les services liés à la communication avec le serveur (authentification, game engine, statistiques des joueurs, etc.)

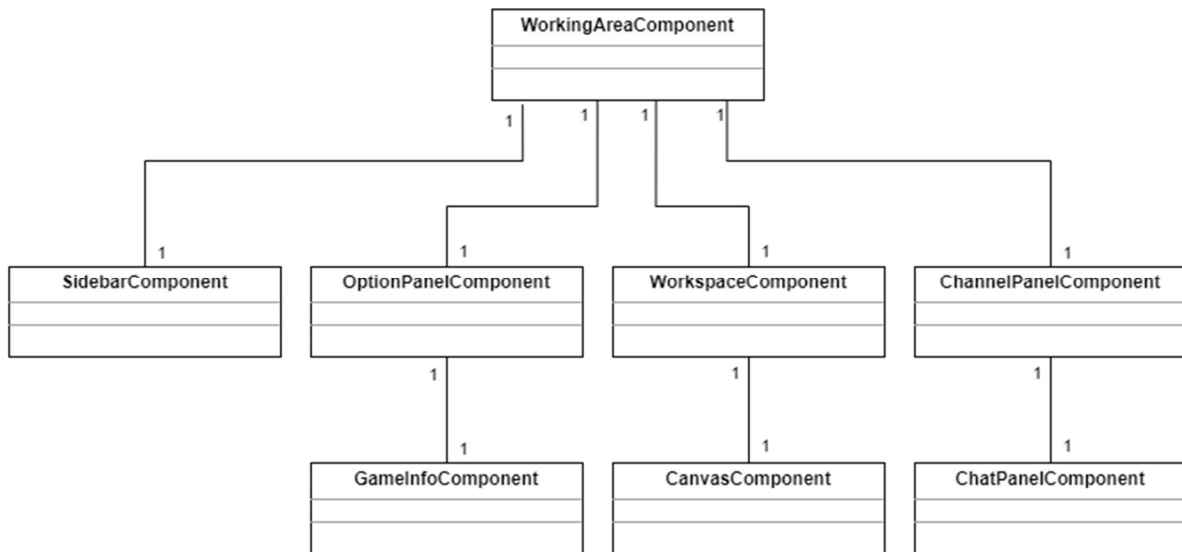


Figure 7. Diagramme de classes (Lourd) - Classes de Components.Game

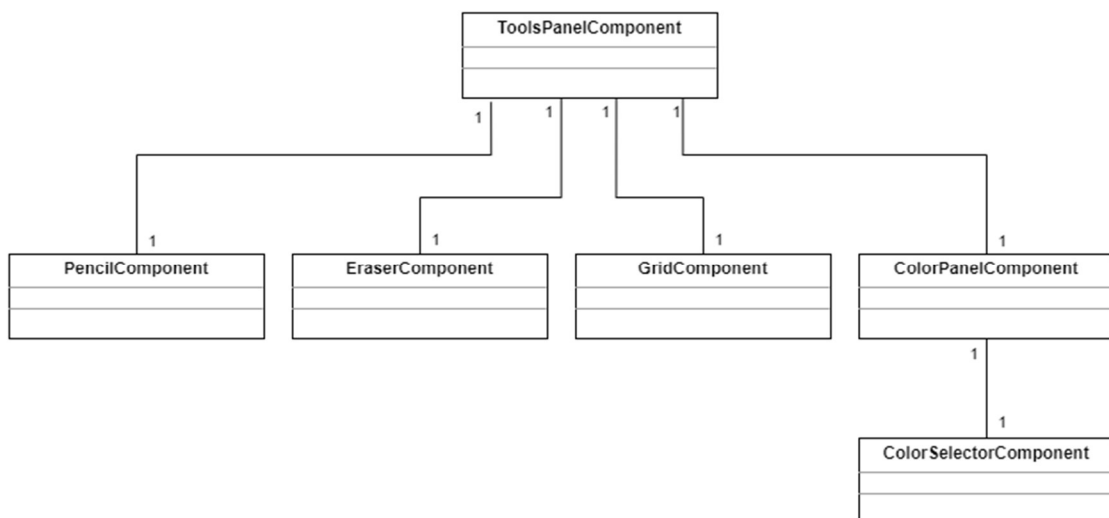


Figure 8. Diagramme de classes (Lourd) - Classes de Components.Tools

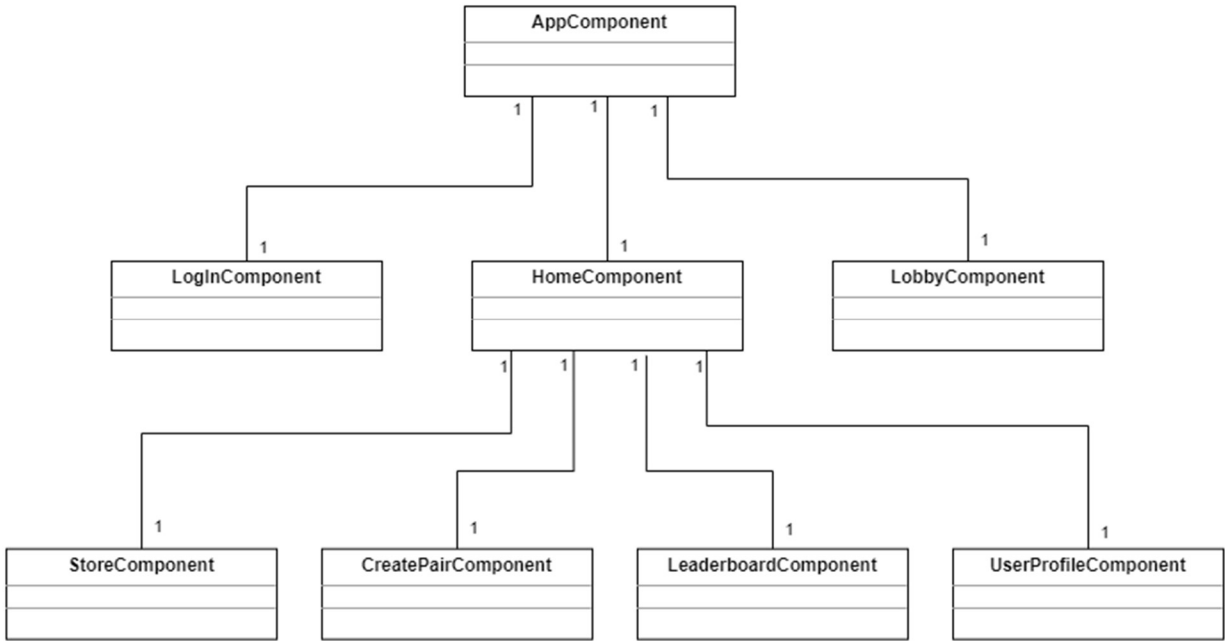


Figure 9. Diagramme de classes (Lourd) - Classes de Components.Pages

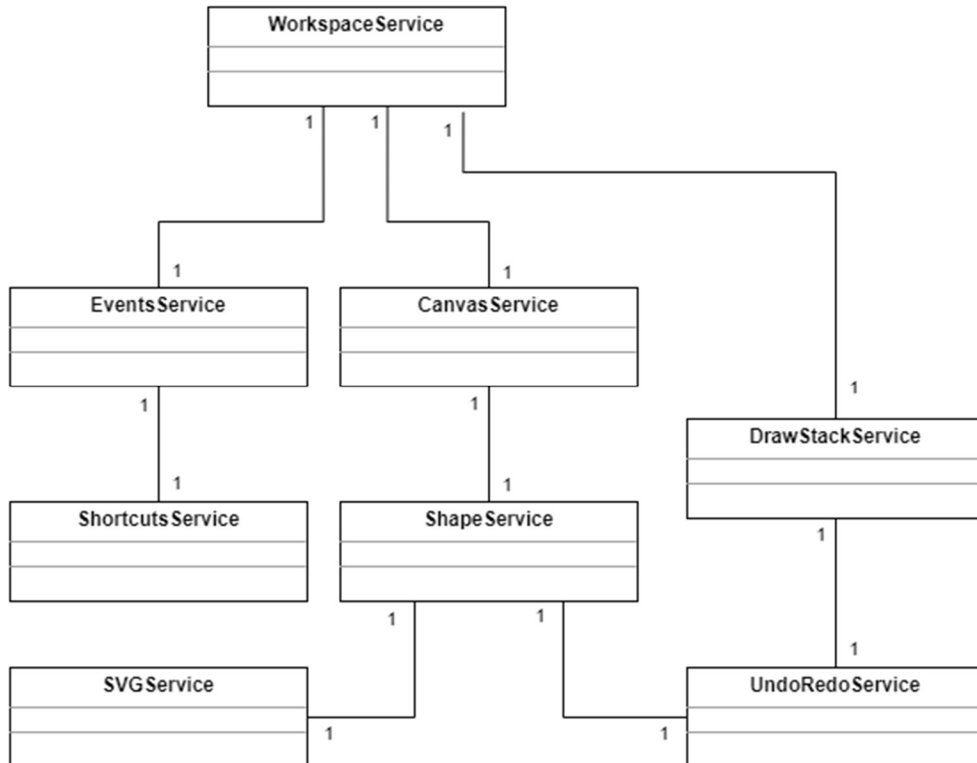


Figure 10. Diagramme de classes (Lourd) - Classes de Services.Game

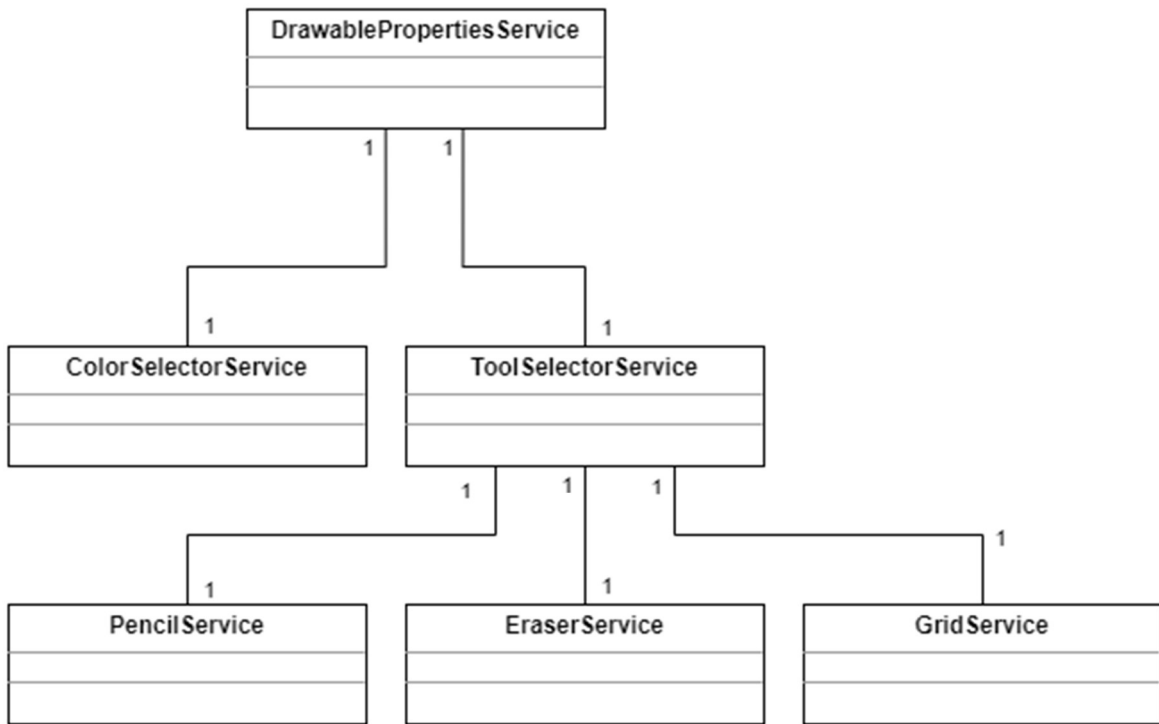


Figure 11. Diagramme de classes (Lourd) - Classes de Services.Tools

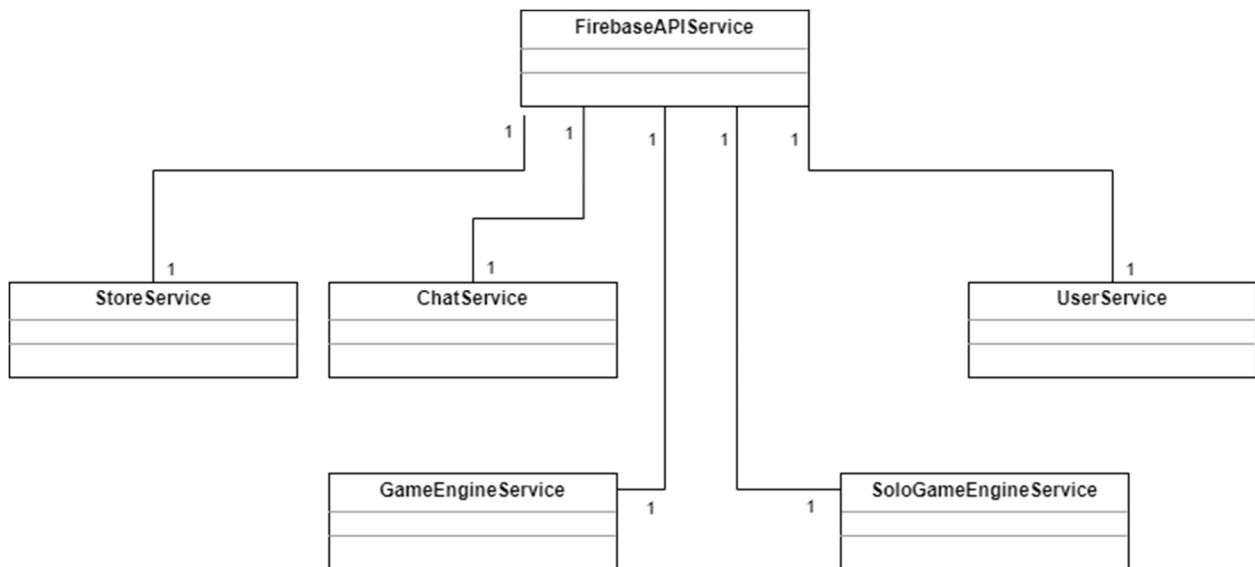


Figure 12. Diagramme de classes (Lourd) - Classes de Services.Server

4.2 Client léger

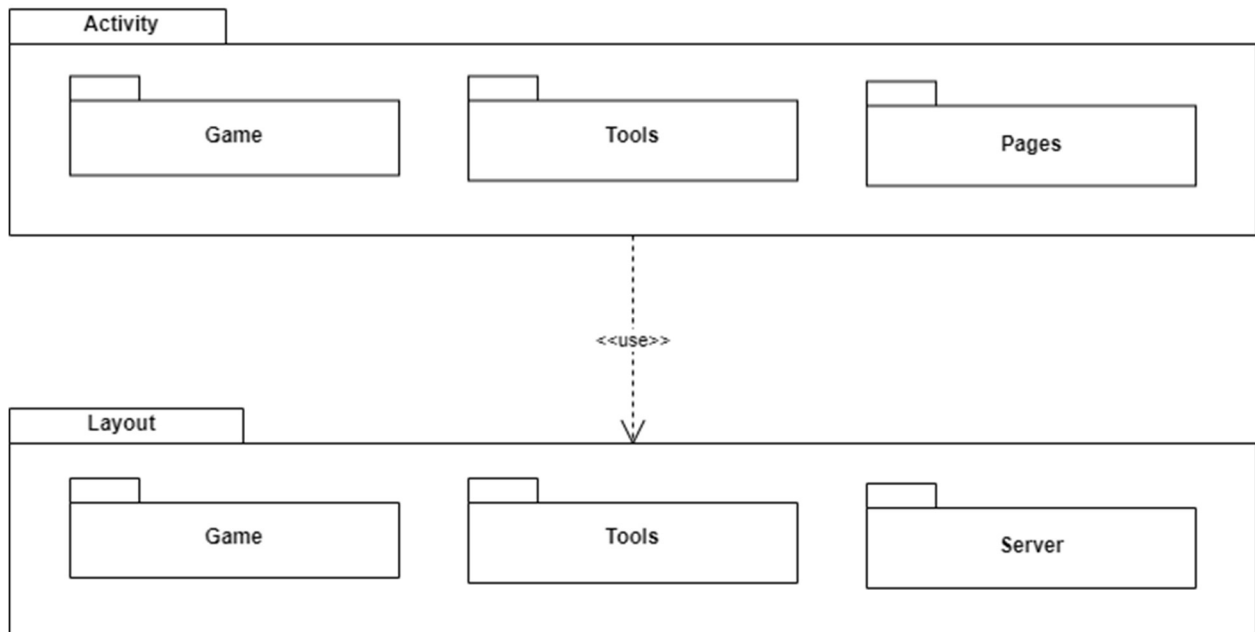


Figure 13. Diagramme de paquetages (Léger) - Fais-moi un dessin

Activity

Ce paquetage de haut niveau contient toutes les propriétés liées au fonctionnement de l'application.

Activity.Game

Ce paquetage contient toutes les propriétés liées au fonctionnement d'une partie (traitement des événements, annuler-refaire, informations sur le jeu etc.).

Activity.Tools

Ce paquetage contient toutes les propriétés liées au fonctionnement des outils (sélecteur d'outils, propriété des outils, etc.).

Activity.Server

Ce paquetage contient toutes les propriétés liées à la communication avec le serveur (authentification, game engine, statistiques des joueurs, etc.)

Layout

Ce paquetage de haut niveau contient toutes les composantes liées à la vue de l'application.

Layout.Game

Ce paquetage contient toutes les composantes liées au fonctionnement d'une partie (espace de dessin, infos sur la partie, chat, etc.).

Layout.Tools

Ce paquetage contient toutes les composantes liées aux outils (crayon, efface, grille, couleurs, etc.).

Layout.Views

Ce paquetage contient toutes les composantes liées aux pages (menu principal, connexion, classement des joueurs, etc.).

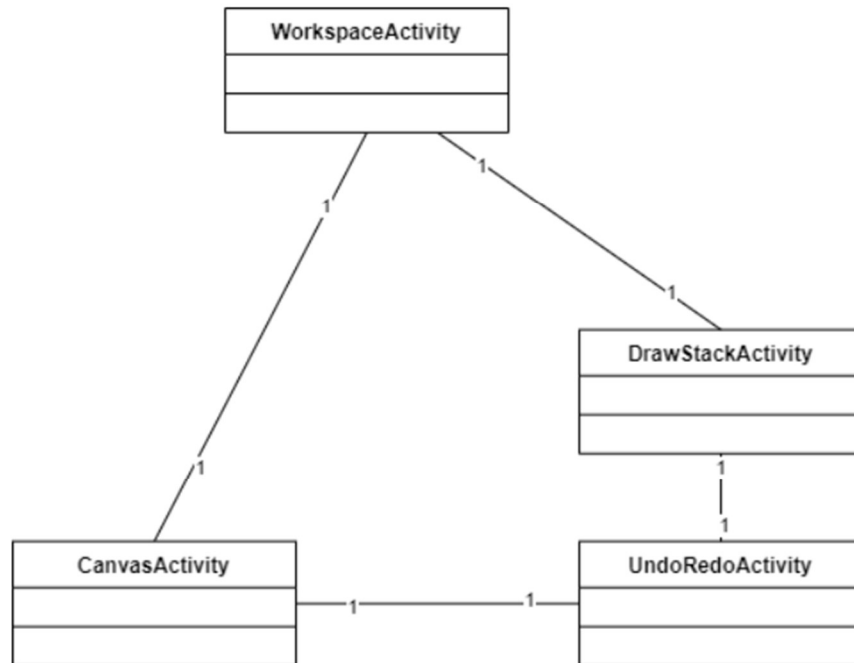


Figure 14. Diagramme de classes (Léger) - Classes de Activity.Game

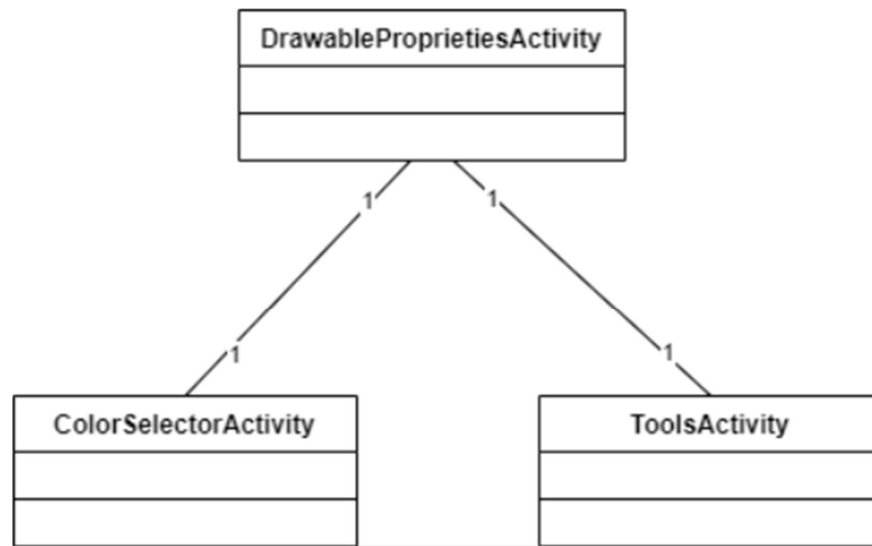


Figure 15. Diagramme de classes (Léger) - Classes de Activity.Tools

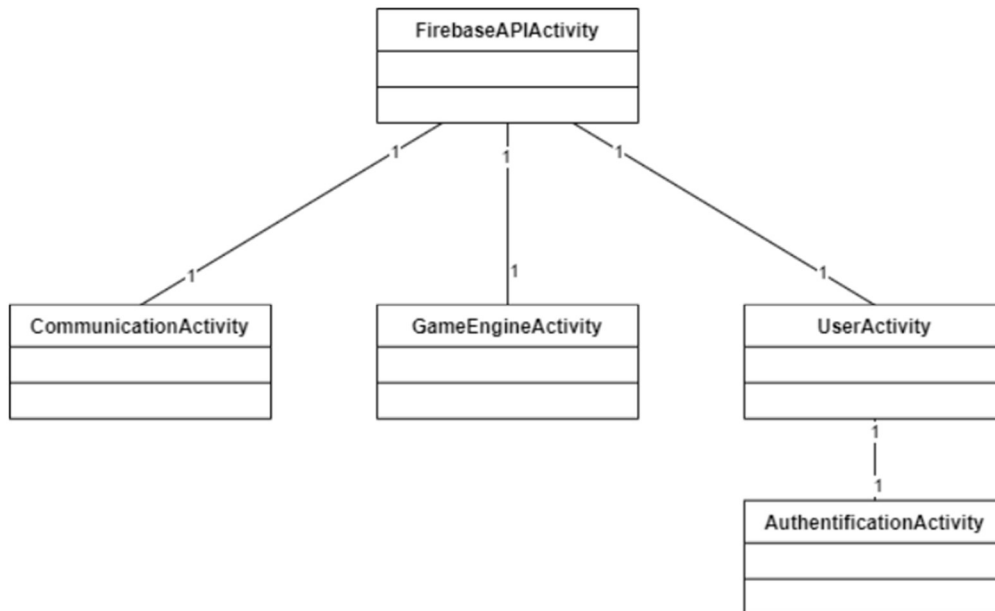


Figure 16. Diagramme de classes (Léger) - Classes de Activity.Server

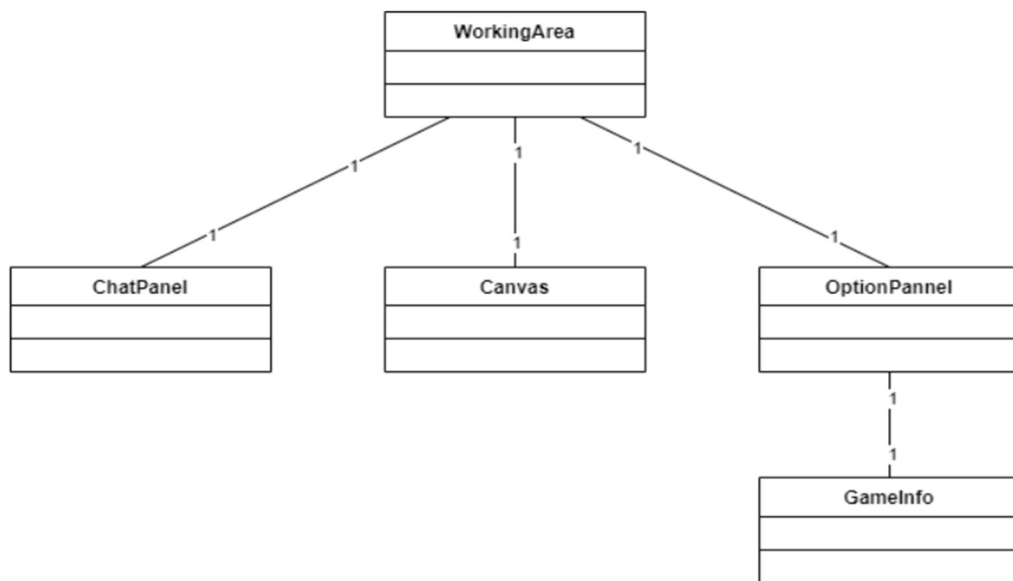


Figure 17. Diagramme de classes (Léger) - Classes de Layout.Game

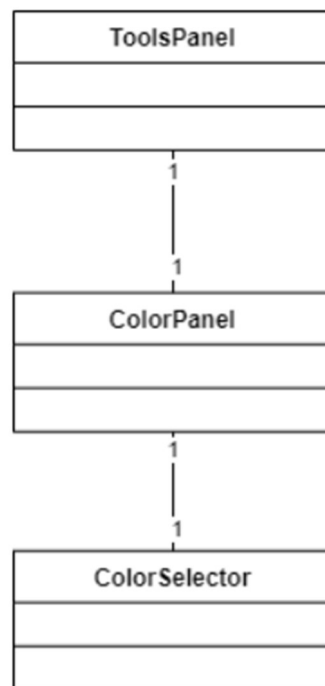


Figure 18. Diagramme de classes (Léger) - Classes de Layout.Tools

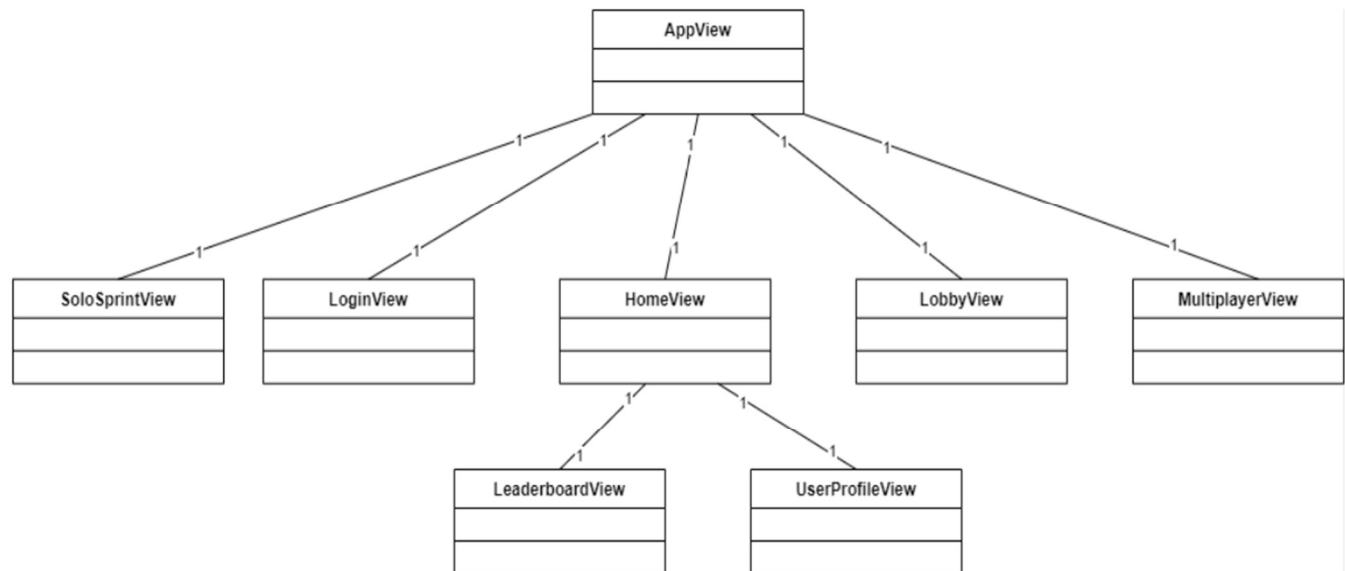


Figure 19. Diagramme de classes (Léger) - Classes de Layout.Views

5. Vue des processus

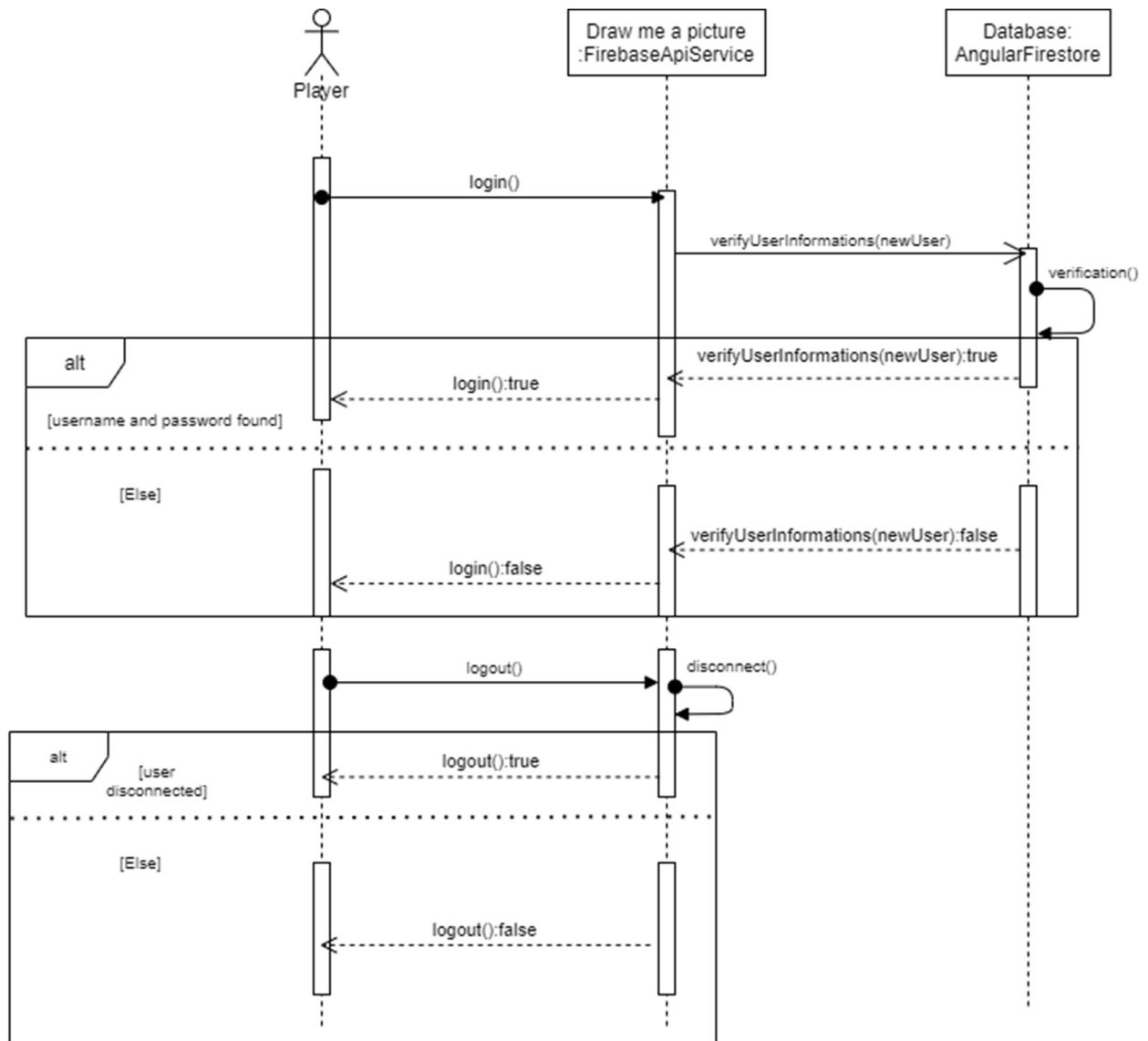


Figure 20. Diagramme de séquence - Connexion/Déconnexion

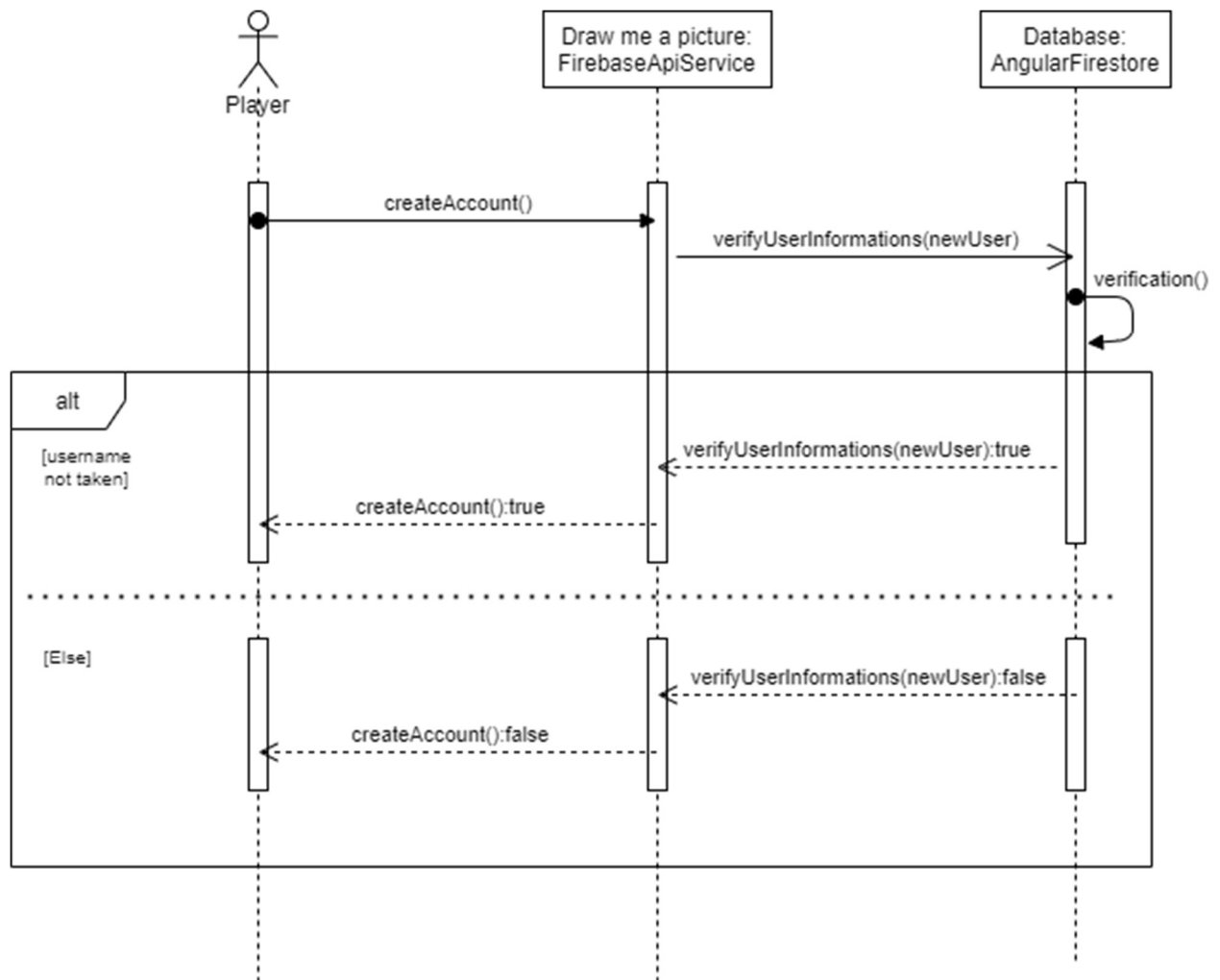


Figure 21. Diagramme de séquence - Création de compte

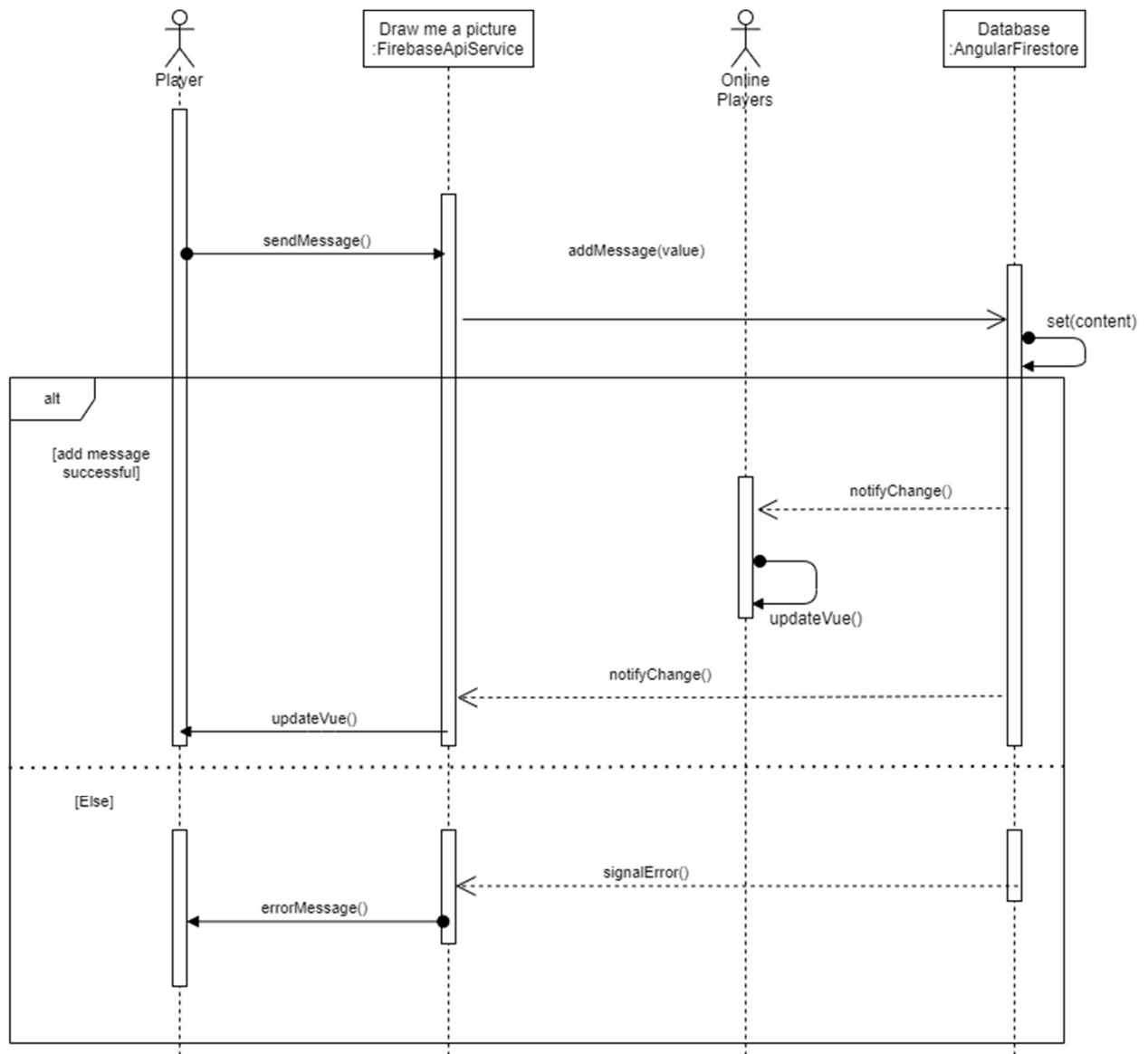


Figure 22. Diagramme de séquence - Clavardage

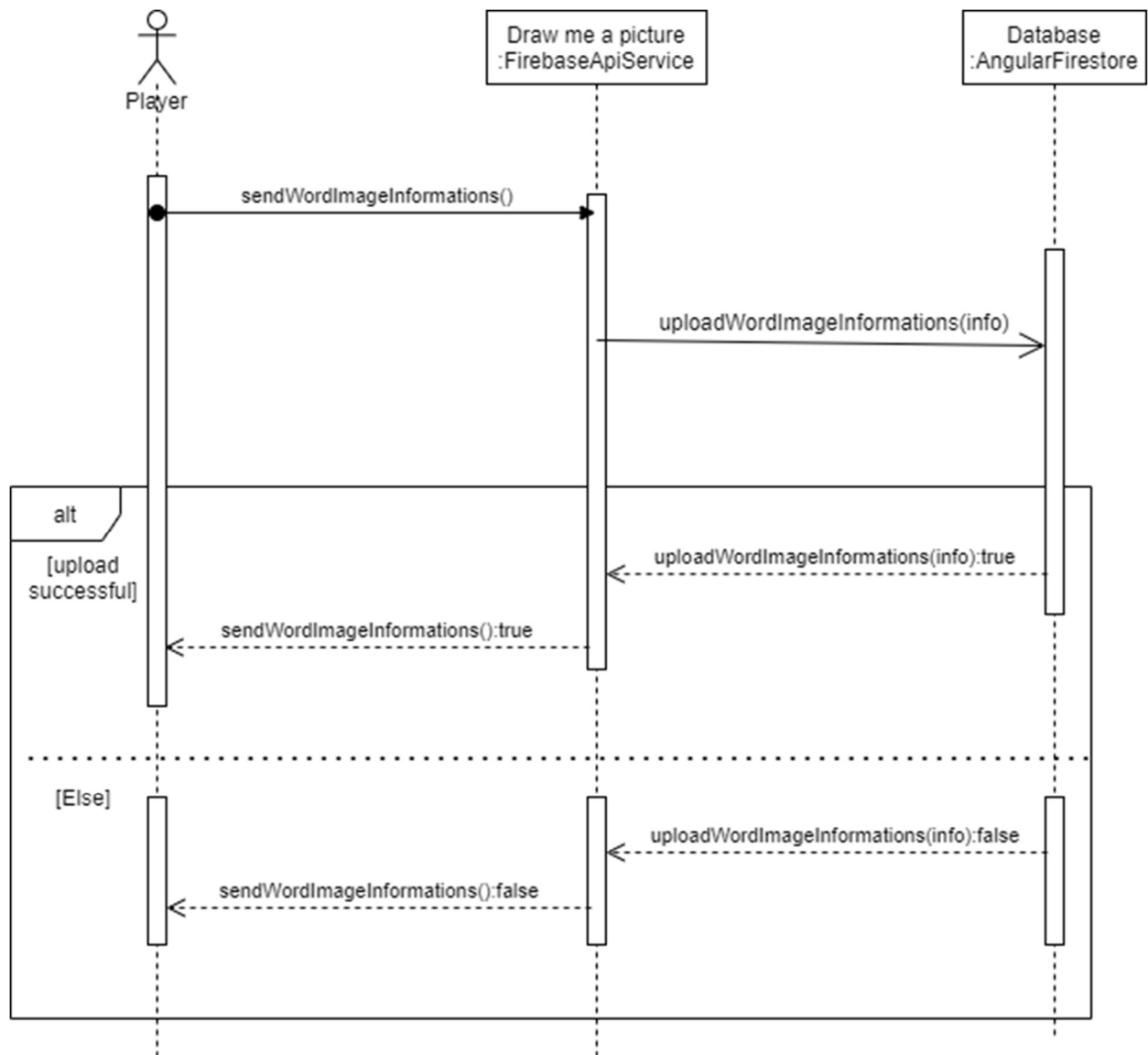


Figure 23. Diagramme de séquence - Création d'une paire mot-image (Manuelle I)

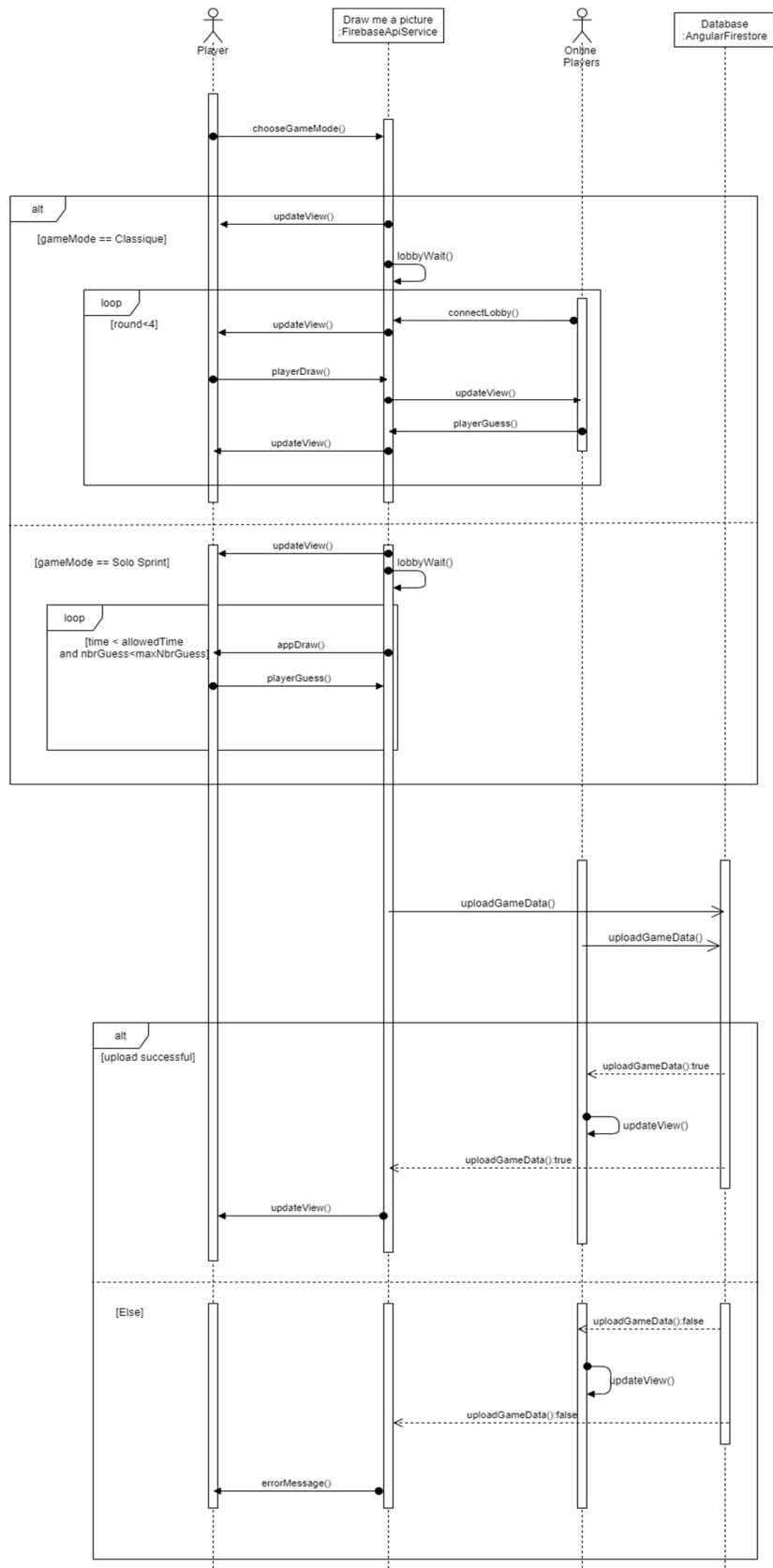


Figure 24. Diagramme de séquence - Mode de jeu (Classique et Sprint Solo)

6. Vue de déploiement

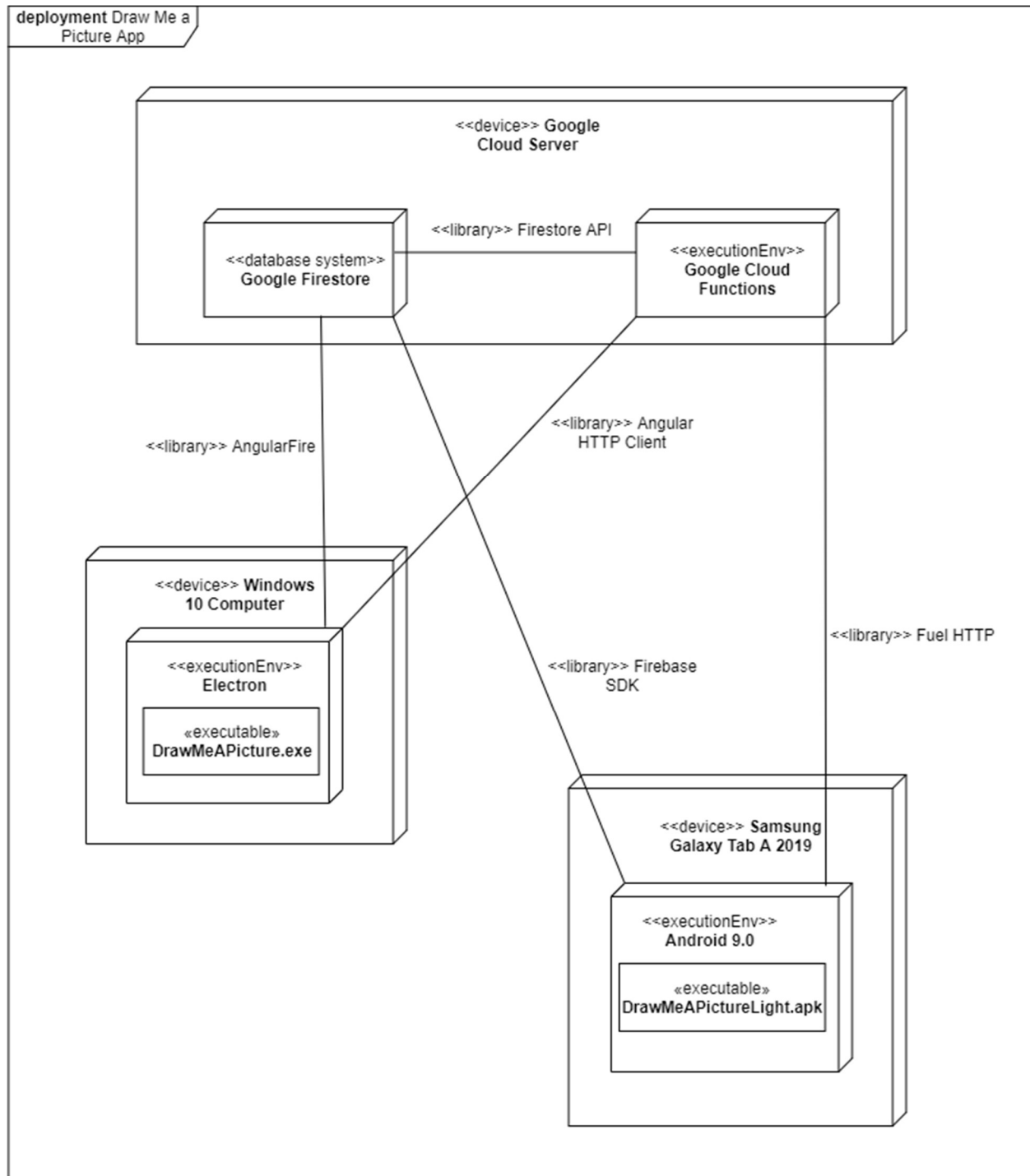


Figure 25. Diagramme de déploiement - Fais-moi un dessin

7. Taille et performance

Notre principale contrainte en termes de performance est de supporter plusieurs (minimalement 3) parties simultanément tout en visant une latence client-serveur-client minimale entre les joueurs (maximum 500 ms). Pour ce faire, nous avons opté pour un découpage en plus petites requêtes (décrites dans notre protocole de communication). Nous privilégions aussi les requêtes en temps réel directement du client vers la base de données Firestore lorsqu'aucune manipulation complexe n'est nécessaire. De plus, tel que décrit dans notre SRS, l'application doit prendre moins de 100 Mo de stockage sur le client léger et nécessiter un maximum de 1 Go de RAM. Nous n'avons pas défini de limites pour le client lourd étant donné les ressources plus abondantes sur un ordinateur moderne. Nous veillerons également à réduire nos dépendances aux bibliothèques externes, particulièrement sur le client léger, toujours dans l'objectif d'économiser les ressources limitées de stockage et de mémoire de la tablette.