

Elaborato di Software Architecture Design

Anna Lama, M63/950

Cristian Mascia, M63/980

Manfredi Napolitano, M63/951

Link al repository del progetto: <https://github.com/Manfr-edi/angular-gameapp.git>

Documento di Visione

Il progetto prevede di realizzare una web app che riporti un catalogo di videogiochi e che metta a disposizione una raccolta personale per gli utenti registrati. Il catalogo di videogiochi raccoglierà tutti i giochi disponibili e in uscita. Oltre a creare una raccolta personale, gli utenti registrati potranno inoltre utilizzare un sistema di “amicizia” per comunicare con gli altri utenti e visualizzare le raccolte e le statistiche di questi. La raccolta si divide in “liste”, per raggruppare quali giochi sono stati completati, quali si stanno giocando in questo momento e quali si desidera giocare in futuro. Per ogni videogioco completato, l’utente può inserire dei dati personali, tra cui il tempo di completamento e un voto. Ogni videogioco nel catalogo presenterà una serie di dati e, in particolare, riporterà un tempo di completamento e un voto. Questi due dati vengono calcolati come media dei tempi di completamento e dei voti di tutti gli utenti che hanno completato quel videogioco. Il sistema mette inoltre a disposizione una serie di statistiche per l’utente registrato (tempi di completamento, spese medie, piattaforma preferita, etc).

Obiettivi delle parti interessate

Questa applicazione è dedicata a tutti gli utenti appassionati del settore videoludico.

Ad oggi, data la frammentazione del mercato delle piattaforme console, non esiste un sistema di gestione unificata dei propri dati tra le varie piattaforme. I videogiocatori utilizzeranno quest’app come “diario” personale per tenere traccia dei giochi in proprio possesso e dei dati relativi (tempi di completamento e spese). Inoltre, attraverso l’uso di questa raccolta, è possibile tenere traccia delle spese effettuate e valutare la spesa media per videogioco.

Requisiti Informali

Un videogioco sarà composto da titolo, data d’uscita, sviluppatore, editore, piattaforme disponibili, genere, prezzo di listino, immagine di copertina, tempo medio di completamento e voto medio (indicando anche quanti giocatori hanno segnato il tempo e il voto).

Un utente si può registrare sul sito. Nella registrazione si dovrà inserire Username ed e-mail.

Un utente registrato può scrivere una breve descrizione personale, indicare quali i suoi nickname sulle varie piattaforme, quali sono le console in suo possesso e quali sono i suoi generi preferiti.

Un utente può ricercare un videogioco nel catalogo.

Un utente registrato può aggiungere un videogioco in una delle sue liste.

Le liste a disposizione di un utente sono *completati/in corso/desiderati*.

L’utente, nell’aggiungere un videogioco ad una lista, deve indicare su quale piattaforma l’ha giocato o lo sta giocando.

Un utente registrato deve indicare, per ogni videogioco aggiunto alla lista *completati*, il tempo di completamento e, optionalmente, un voto da 1 a 5 per quest’ultimo.

Per ogni videogioco aggiunto alle liste *completati/in corso*, l’utente deve indicare la spesa e la piattaforma sulla quale si sta giocando o si ha giocato il titolo.

Un utente registrato, per ogni videogioco presente nelle sue liste, può indicare eventuali note. Un utente registrato può conoscere i suoi tempi di completamento medi sia in generale, sia per genere e sia per

piattaforma. Un utente registrato può conoscere le statistiche per genere. Un utente registrato può conoscere le statistiche riguardo le console (ad esempio giochi *completati* o *da giocare* su una data console). Un utente registrato può conoscere le spese medie e totali sia in generale, sia per genere/piattaforma. Un utente può effettuare la ricerca di altri utenti per Username.

La pagina di un utente riporterà l'username e una breve descrizione personale. Se l'utente è amico, verranno visualizzate le sue liste dei giochi ed eventuali statistiche.

Un utente registrato può aggiungere un altro utente come amico. Un utente registrato può inviare messaggi privati ad un amico. Un utente registrato può vedere quali sono i giochi in comune (anche se appartenenti a liste diverse).

Il gestore del catalogo potrà inserire un videogioco nella raccolta, indicando necessariamente tutti i dati relativi al videogioco. Il gestore del catalogo potrà cancellare e modificare un videogioco dalla raccolta.

Documento dell'Architettura Software

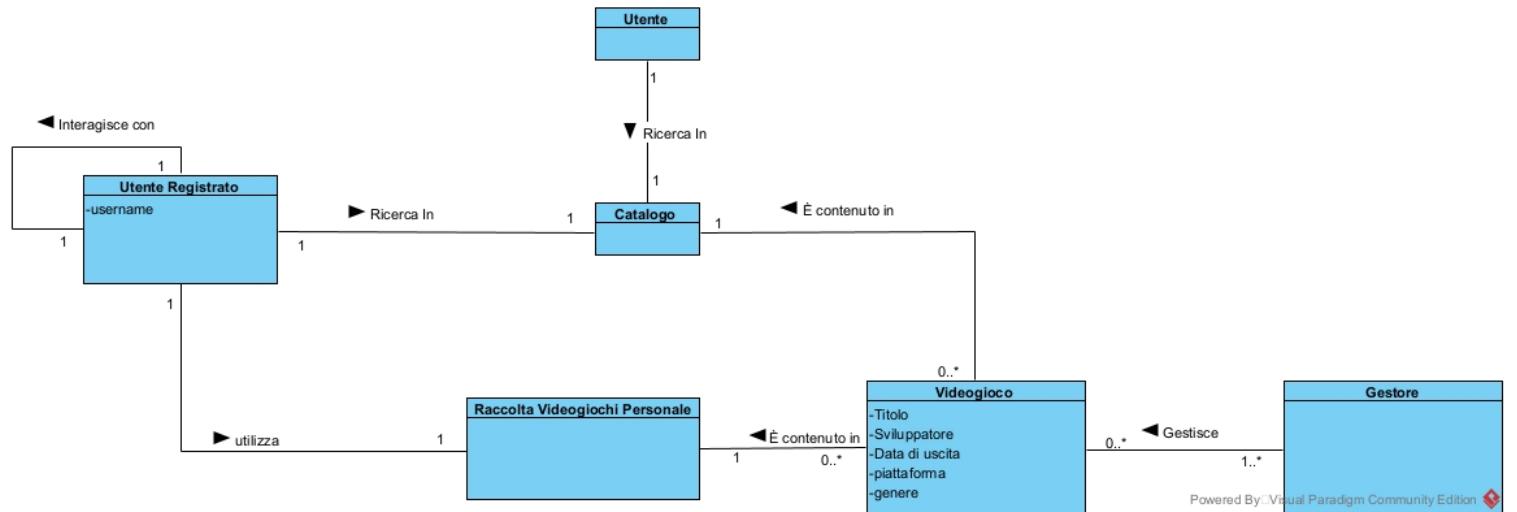
Per implementare la web app sono state prese diverse decisioni organizzative e architettoniche. Per il processo di sviluppo, è stato adottato l'**Unified Process (UP)**, andando quindi a seguire le fasi principali di questo framework. Nella fase di *Ideazione* è stato composto un **Documento di Visione** che andasse a riportare tutte le caratteristiche e le motivazioni del nostro progetto. Successivamente, è stata condotta un'analisi riguardo al pattern architettonicale più adatto per questo tipo di sistema. Il pattern scelto per il sistema è il **Model-View-Controller (MVC)** visto il suo largo utilizzo per le applicazioni web.

La web-app è stata implementata attraverso il framework *Angular 2+*, ed è stata testata attraverso l'uso di *Node.js*. La scelta è ricaduta su questo framework perché il binding tra la componente di interfaccia grafica e il controller logico di un modulo è completamente automatizzato, semplice da capire e da gestire. Ciò ha facilitato lo sviluppo di un progetto che seguisse il pattern MVC.

Per la parte dei dati, abbiamo fatto affidamento all'insieme dei servizi messi a disposizione da *Firebase* di Google. In particolare, per la creazione del DB ci siamo affidati al DB non relazionale *Firestore*, che permette una gestione dei dati rapida e chiara, oltre a fornire di default un pattern *Observer* per la lettura continua dei dati. Per la gestione degli account (registrazione, login, verifica email, reset password, ...) ci siamo affidati al servizio *Authentication* di Firebase: con questo è stato implementato il metodo di autenticazione tramite email e password. In ultimo, per la gestione delle immagini presenti sul sito, abbiamo fatto affidamento al servizio *Storage* di Firebase: con questo è stato possibile memorizzare tutte le copertine dei videogiochi del catalogo e le immagini di profilo degli utenti.

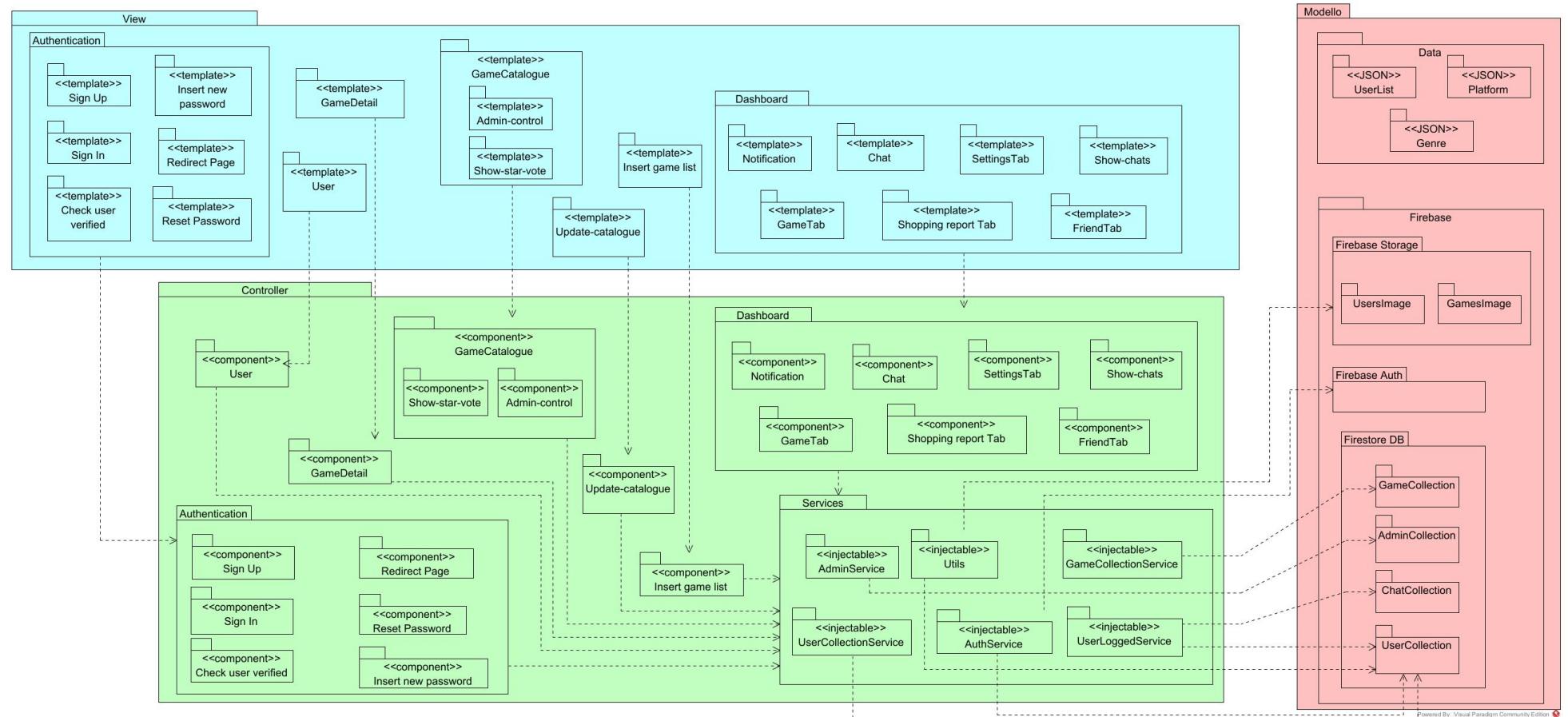
Modello di Dominio

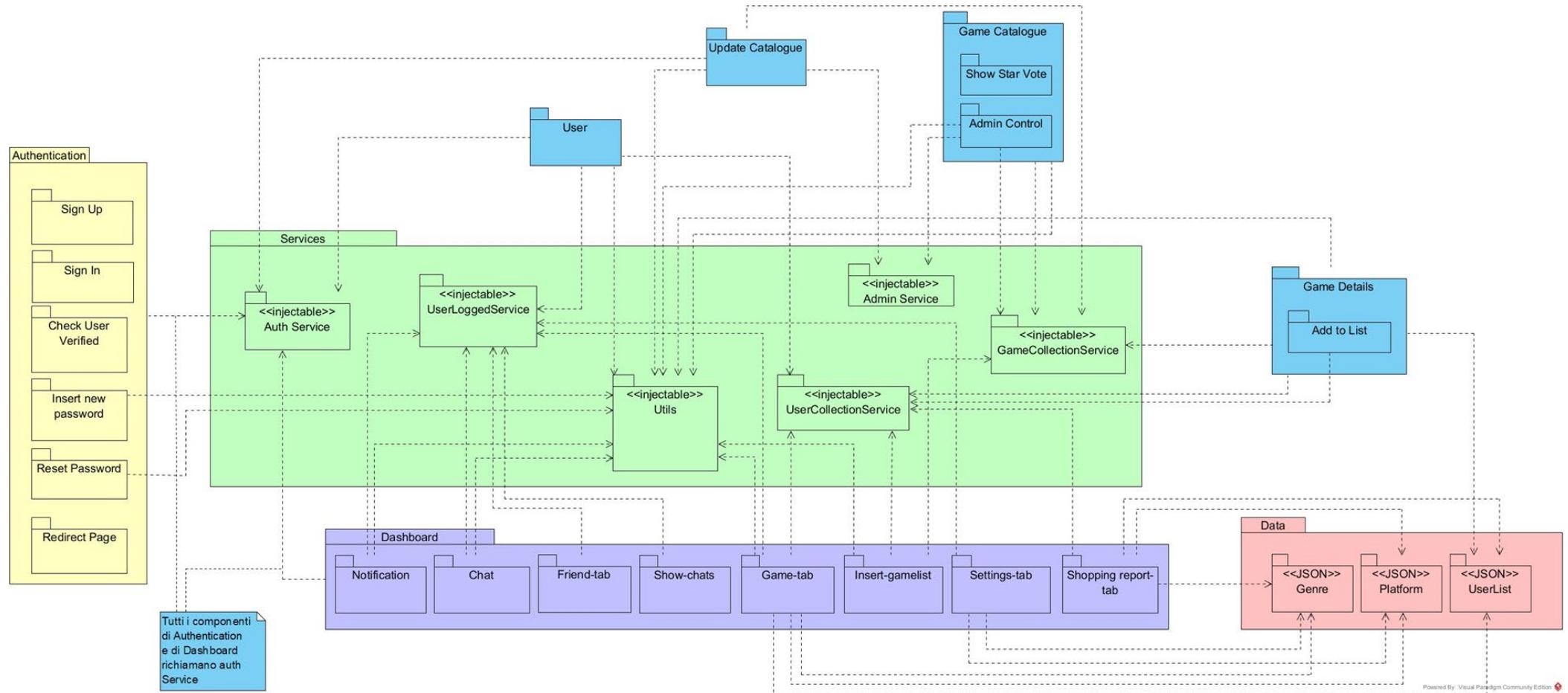
Di seguito si riporta il modello di dominio del progetto, che mostra quali sono le entità in gioco e come interagiscono tra di loro.



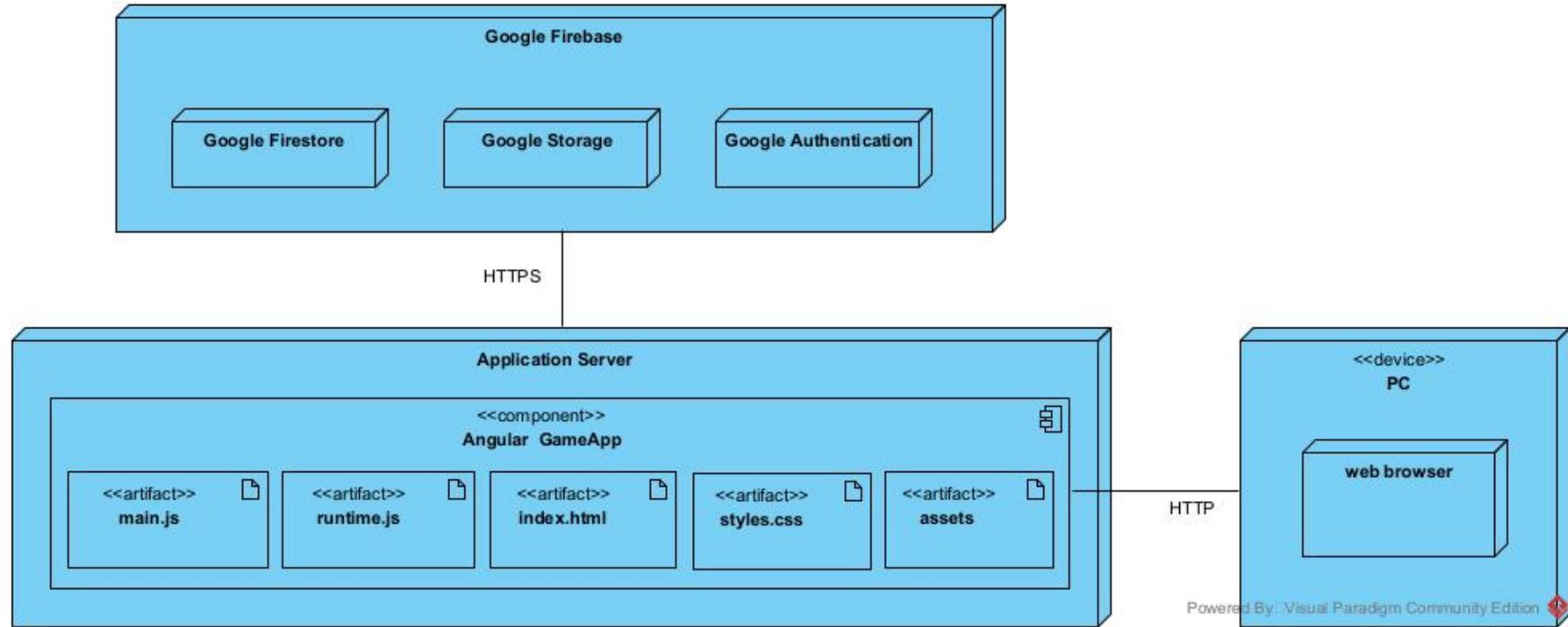
Modello di Progetto – Package Diagram

Di seguito si riportano due package diagram che rappresentano l'intera struttura dell'applicazione. Abbiamo scelto di riportare due diagrammi per dare una maggiore leggibilità e comprensione del progetto. Il primo diagramma, seguendo lo schema architetturale MVC, riporta i moduli della web app secondo lo stereotipo *template/component* e l'interconnessione della parte di controllo rispetto ai servizi offerti da Firebase e ai dati locali. Il secondo specifica le dipendenze di questi moduli con i servizi implementati, che rispettano lo stereotipo *injectable*: questo stereotipo si indica per i moduli che partecipano al processo di *Dependency Injection*. La *Dependency Injection* o DI è un design pattern nel quale una classe, attraverso il suo costruttore, richiede delle funzionalità presenti nei moduli *Injectable*. Viene utilizzato questo pattern per favorire la modularità e il riutilizzo del codice. Nel nostro progetto, tutti i moduli *injectable* hanno come suffisso il termine *Service* e sono gli unici componenti di controllo a comunicare con Firebase.



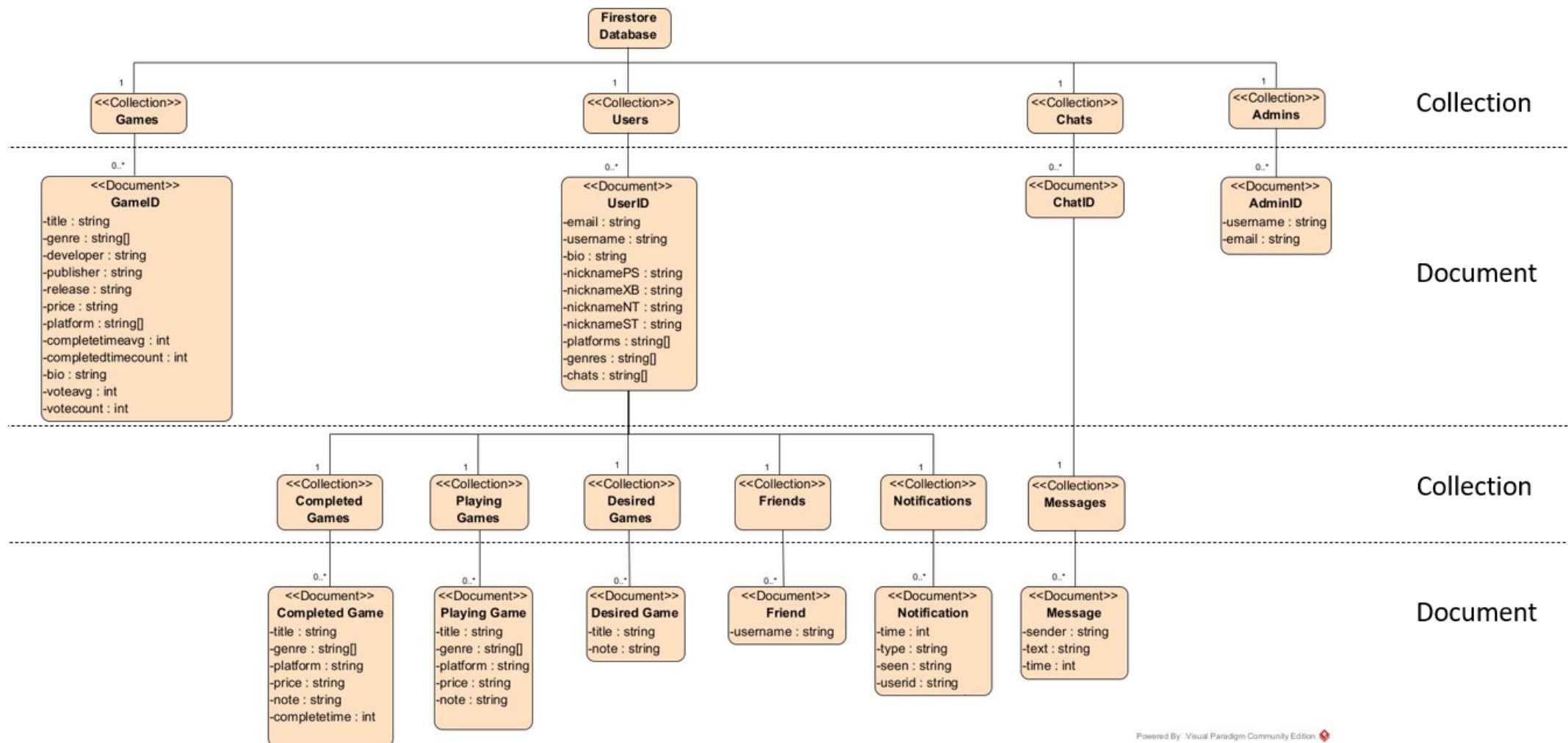


Modello di Progetto – Deployment Diagram



Modello dei Dati – Schema Strutturale Firestore

Per quanto riguarda il modello dei dati, avendo fatto affidamento al database non relazionale di Firebase, è stato implementato uno schema ad-hoc che rappresentasse correttamente il tipo di struttura del DB di Google. Il modo in cui Firestore memorizza i dati, infatti, segue lo schema *Collection/Document/Collection/...* dove una Collection può contenere più documenti e/o altre collection e a loro volta anche i documenti, oltre a contenere degli attributi tipizzati, possono contenere collection. Seguendo questa struttura, abbiamo costruito uno schema ad albero che ne evidenziasse il pattern.



Lista Attori

Attori	Casi d'uso
Utente	Ricerca Videogioco, Registrazione
Utente Registrato	Aggiungi Videogioco ad una Lista, Rimuovi Videogioco da una lista, Modifica Videogioco di una lista, Aggiorna Preferenze, Ricerca Utente, Aggiungi Amico, Invia messaggio privato
Gestore Catalogo	Inserisci Videogioco, Modifica Videogioco, Elimina Videogioco

Glossario

Termine	Descrizione	Sinonimi
Videogioco	Oggetto contenuto nel catalogo e che può essere aggiunto in una raccolta personale	Gioco
Catalogo	Lista completa di tutti i videogiochi disponibili e in uscita	
Raccolta Personale	Lista di videogiochi legata ad un dato utente registrato	Raccolta, Lista
Utente	Visitatore del sito	
Utente Registrato	Visitatore che attraverso la registrazione e il login, si è autenticato al sito	
Gestore Catalogo	Addetto alla gestione e modifica del Catalogo	Gestore, Admin

Lista dei casi d'uso

Di seguito si riporta una lista dei casi d'uso principali del nostro progetto.

N.B. per tutti i casi d'uso riportati si presuppone di trovarsi nella “sezione” giusta dell'app prima di poter avviare la funzionalità.

Caso d'uso:	Ricerca Videogioco
Attore primario	Utente o Utente Registrato
Attore secondario	
Descrizione	L'utente ricerca un videogioco tramite il titolo
Pre-Condizioni	
Scenario	<ol style="list-style-type: none"> 1. L'utente fornisce un titolo 2. L'utente invia il comando di ricerca 3. SE il videogioco è presente nel catalogo <ol style="list-style-type: none"> 3.1. Il sistema restituisce una lista di videogiochi con il titolo corrispondente 4. altrimenti <ol style="list-style-type: none"> 4.1. Il sistema restituisce un errore (campo vuoto o invalido)
Post-Condizioni	Viene riportato il videogioco con i dati relativi
Casi d'uso correlati	

Caso d'uso:	Aggiungi Videogioco ad una lista
Attore primario	Utente Registrato
Attore secondario	
Descrizione	L'utente aggiunge un videogioco ad una lista
Pre-Condizioni	L'utente ha effettuato la ricerca di un videogioco

Scenario	<ol style="list-style-type: none"> 1. L'utente invia il comando "aggiungi alla lista" 2. L'utente indica in quale lista inserire il videogioco 3. SE Il videogioco non è presente in nessuna lista <ol style="list-style-type: none"> 3.1. SE la lista selezionata è "Completati" <ol style="list-style-type: none"> 3.1.1. Il sistema chiede di fornire come dati obbligatori il tempo di completamento, il prezzo a cui è stato acquistato il gioco e la piattaforma su cui l'ha giocato e come dati facoltativi un voto e delle note 3.1.2. SE L'utente inserisce correttamente i dati <ol style="list-style-type: none"> 3.1.2.1. Il sistema memorizza nella lista il videogioco selezionato (con tutte le informazioni fornite) e presenta un messaggio di conferma 3.1.3. altrimenti <ol style="list-style-type: none"> 3.1.3.1. Il sistema non memorizza il videogioco e restituisce un errore sui campi non compilati correttamente 3.2. SE la lista selezionata è "In gioco" <ol style="list-style-type: none"> 3.2.1. Il sistema chiede di fornire il prezzo a cui è stato acquistato il gioco, la piattaforma e delle note facoltative 3.2.2. SE L'utente inserisce correttamente i dati <ol style="list-style-type: none"> 3.2.2.1. Il sistema memorizza nella lista "In gioco" il videogioco selezionato (con tutte le informazioni fornite) e presenta un messaggio di conferma 3.2.3. altrimenti <ol style="list-style-type: none"> 3.2.3.1. Il sistema non memorizza il videogioco e restituisce un errore sui campi non compilati correttamente 3.3. SE la lista selezionata è "Desiderati" <ol style="list-style-type: none"> 3.3.1. Il sistema chiede di aggiungere solo le note facoltative 4. altrimenti <ol style="list-style-type: none"> 4.1. Il sistema restituisce un messaggio "Videogioco già presente"
Post-Condizioni	Viene aggiunto il videogioco nella lista selezionata
Casi d'uso correlati	Ricerca Videogioco

Di seguito si riportano le form di aggiunta di un videogioco per ciascuna delle tre liste :

Inserisci i dati

Lista
Completati

Tempo di completamento *
0 h

Prezzo di acquisto *
0 €

Piattaforma
PC

Voto

Note Aggiuntive

* Dati obbligatori

Update List

Inserisci i dati

Lista
In gioco

Prezzo di acquisto *
0 €

Piattaforma
PC

Note Aggiuntive

* Dati obbligatori

Update List

Caso d'uso:	Registrazione
Attore primario	Utente
Attore secondario	
Descrizione	L'utente inserisce le informazioni per ottenere delle credenziali che gli permetteranno di accedere al sistema come Utente registrato.
Pre-Condizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente seleziona l'opzione "Registrazione" 2. L'utente inserisce un nickname univoco, e-mail e password 3. L'utente seleziona l'opzione di Conferma 4. SE i dati sono corretti <ol style="list-style-type: none"> 4.1. Il sistema registra l'utente memorizzandone i dati e invia un'e-mail di conferma 5. altrimenti <ol style="list-style-type: none"> 5.1. Il sistema restituisce un errore su uno o più campi non validi 6. SE l'utente conferma la registrazione tramite la mail <ol style="list-style-type: none"> 6.1. Il sistema attiva l'account 7. altrimenti <ol style="list-style-type: none"> 7.1. L'account resta in uno stato inattivo
Post-Condizioni	L'utente è considerato registrato nel sistema
Casi d'uso correlati	

Caso d'uso:	Rimuovi videogioco da una lista
Attore primario	Utente Registrato
Attore secondario	
Descrizione	L'utente registrato rimuove un videogioco dalla lista
Pre-Condizioni	Il gioco risulta inserito in una lista per l'utente
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente registrato seleziona il videogioco da eliminare all'interno della lista visualizzata 2. L'utente registrato clicca sul tasto "Elimina" 3. Il sistema rimuove il videogioco dalla lista dell'utente 4. Il sistema restituisce un messaggio di conferma dell'avvenuta rimozione
Post-Condizioni	Il videogioco è stato rimosso dalla lista
Casi d'uso correlati	

Caso d'uso:	Modifica videogioco di una lista
Attore primario	Utente Registrato
Attore secondario	
Descrizione	L'utente registrato modifica un videogioco presente nella lista
Pre-Condizioni	Il gioco risulta inserito in una lista per l'utente
Scenario Principale	<p>1. L'utente registrato seleziona il videogioco da modificare all'interno della lista visualizzata</p> <p>2. L'utente registrato clicca sul tasto "Modifica"</p> <p>3. L'utente indica in quale lista inserire il videogioco</p> <p>3.1. SE la lista selezionata è "Completati"</p> <p>3.1.1. Il sistema permette di modificare il tempo di completamento, il prezzo o la piattaforma, oppure di aggiungere il voto e le note</p> <p>3.1.2. SE L'utente inserisce correttamente i dati e clicca sul tasto di conferma</p> <p>3.1.2.1. Il sistema memorizza nella lista il videogioco selezionato, con tutte le informazioni fornite, e presenta un messaggio di conferma</p> <p>3.1.3. altrimenti</p> <p>3.1.3.1. Il sistema non permette la modifica dei dati e restituisce un errore sui campi non compilati correttamente</p> <p>3.2. SE la lista selezionata è "In gioco"</p> <p>3.2.1. Il sistema permette di modificare o aggiungere la piattaforma, il prezzo e le note</p> <p>3.2.2. SE L'utente inserisce correttamente i dati e clicca sul tasto di conferma</p> <p>3.2.2.1. Il sistema memorizza nella lista il videogioco selezionato, con tutte le informazioni fornite, e presenta un messaggio di conferma</p> <p>3.2.3. altrimenti</p> <p>3.2.3.1. Il sistema non permette la modifica dei dati e restituisce un errore sui campi non compilati correttamente</p> <p>3.3. SE la lista selezionata è "Desiderati"</p> <p>3.3.1. Il sistema permette di aggiungere o modificare solo le note</p> <p>3.3.2. SE L'utente clicca sul tasto di conferma</p> <p>3.3.2.1. Il sistema memorizza nella lista il videogioco selezionato, con le informazioni fornite, e presenta un messaggio di conferma</p>
Post-Condizioni	Il videogioco è stato modificato
Casi d'uso correlati	

Caso d'uso:	Aggiorna preferenze
Attore primario	Utente Registrato
Attore secondario	
Descrizione	L'utente ha la possibilità di indicare le console in suo possesso e i suoi generi preferiti.
Pre-Condizioni	L'utente ha effettuato il login
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente seleziona l'opzione Modifica Piattaforme (oppure Modifica generi preferiti) 2. L'utente seleziona tra quelle indicate, le piattaforme che possiede (oppure i generi che preferisce) e conferma la sua selezione. 3. Il sistema memorizza piattaforme e generi indicati e li mostra nella sezione Informazioni.
Post-Condizioni	L'utente ha una o più piattaforme (oppure generi) associate

Caso d'uso:	Ricerca Utente
Attore primario	Utente Registrato
Attore secondario	
Descrizione	L'utente ricerca un altro utente registrato al sistema mediante nickname
Pre-Condizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente invia il comando "Ricerca utente" 2. Il sistema chiede di inserire il nickname 3. L'utente fornisce un nickname 4. SE Il nickname corrisponde ad un utente presente nel sistema <ol style="list-style-type: none"> 4.1. Il sistema restituisce le informazioni dell'utente cercato 5. altrimenti <ol style="list-style-type: none"> 5.1. Il sistema restituisce un messaggio "Utente non trovato"
Post-Condizioni	

Caso d'uso:	Aggiungi Amico
Attore primario	Utente Registrato
Attore secondario	Utente Registrato
Descrizione	L'utente aggiunge un altro utente come amico
Pre-Condizioni	L'utente ha effettuato la ricerca di un altro utente
Scenario Principale	<ol style="list-style-type: none"> 1. SE l'utente ricercato non è nella lista amici <ol style="list-style-type: none"> 1.1 L'utente invia il comando "aggiungi amico" 1.2 Il sistema invia la richiesta di amicizia all'altro utente registrato 1.3 Il sistema notifica l'altro utente della richiesta effettuata 1.4 SE l'altro utente accetta la richiesta 1.5 Il sistema aggiunge reciprocamente gli utenti nelle proprie liste amici 1.6 Il sistema notifica l'utente che ha inviato la richiesta dell'avvenuta accettazione 1.7 Il sistema restituisce un messaggio di conferma 1.8 altrimenti <ol style="list-style-type: none"> 1.8.1 il sistema elimina la richiesta di amicizia 2. altrimenti <ol style="list-style-type: none"> 2.1 Il sistema presenta l'utente come amico
Post-Condizioni	Viene aggiunto l'utente nella lista amici
Casi d'uso correlati	Ricerca Amico

Caso d'uso:	Invia messaggio privato
Attore primario	Utente Registrato
Attore secondario	Utente Registrato
Descrizione	L'utente invia un messaggio privato ad un suo amico
Pre-Condizioni	L'utente ha nella lista amici l'utente a cui vuole inviare un messaggio
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente scrive un testo 2. L'utente esegue il comando "invia messaggio privato" 3. il sistema memorizza il messaggio inviato e lo riporta nella chat 4. Il sistema notifica il messaggio all'altro utente
Post-Condizioni	Il messaggio viene visualizzato dall'altro utente
Casi d'uso correlati	

Caso d'uso:	Inserisci Videogioco
Attore primario	Gestore del Catalogo
Attore secondario	
Descrizione	Il gestore del catalogo inserisce un videogioco nel catalogo
Pre-Condizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. Il gestore compila le informazioni riguardo il videogioco da inserire (Titolo, data d'uscita, ...) 2. Il gestore invia il comando "inserisci videogioco" 3. SE tutti i campi inseriti sono validi <ol style="list-style-type: none"> 1.1 Il sistema inserisce il videogioco nel catalogo generale 1.2 Il sistema restituisce un messaggio di conferma dell'avvenuto inserimento 2. altrimenti 3. Il sistema indica i campi di errore
Post-Condizioni	Viene aggiunto il videogioco al catalogo
Casi d'uso correlati	

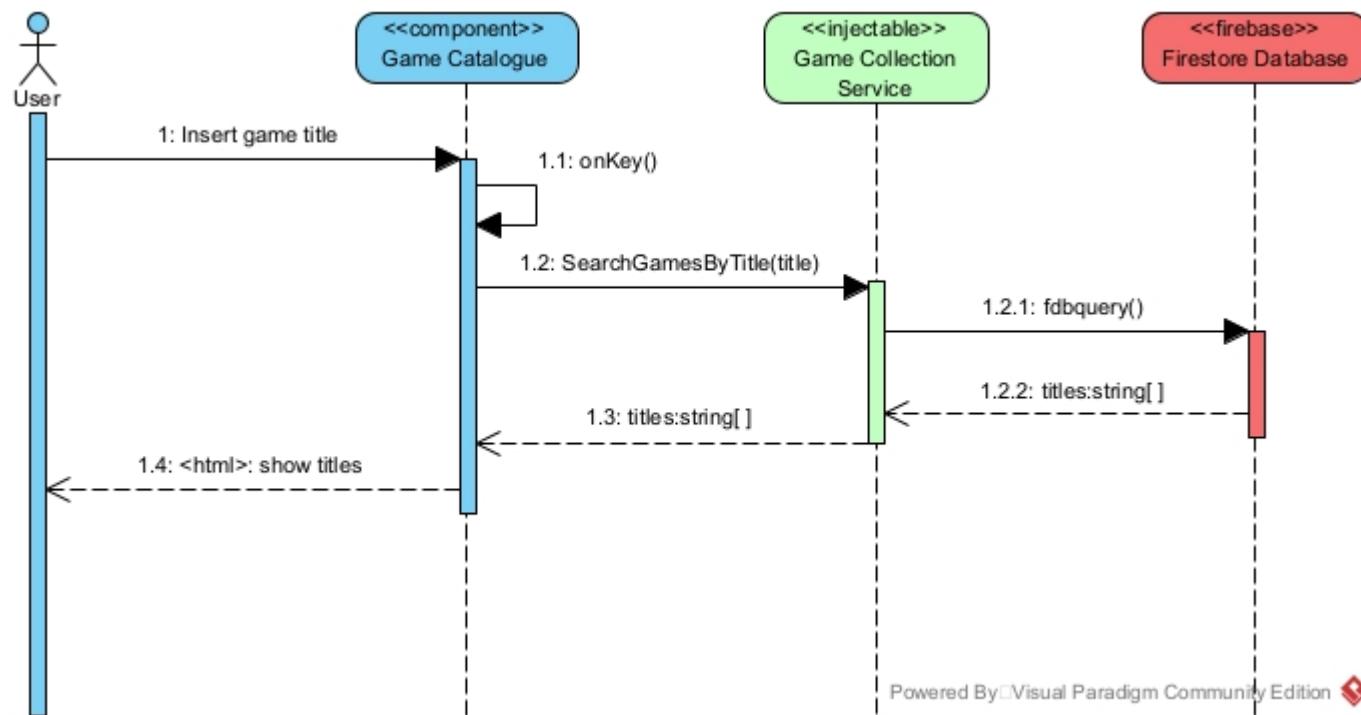
Caso d'uso:	Eliminare un videogioco
Attore primario	Gestore del Catalogo
Attore secondario	
Descrizione	Il gestore può scegliere di eliminare un gioco dal catalogo
Pre-Condizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. Il gestore seleziona dal catalogo il videogioco da eliminare 2. Il gestore clicca sul tasto "Elimina" 3. Il sistema rimuove il videogioco dal catalogo generale 4. Il sistema restituisce un messaggio di conferma dell'avvenuta rimozione
Post-Condizioni	Il videogioco è stato rimosso dal catalogo
Casi d'uso correlati	

Caso d'uso:	Modifica Videogioco
Attore primario	Gestore del Catalogo
Attore secondario	
Descrizione	Il gestore del catalogo modifica un videogioco nel catalogo
Pre-Condizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. Il gestore seleziona dal catalogo il videogioco da modifica 2. Il gestore modifica le informazioni riguardo il videogioco da inserire (Titolo, data d'uscita, ...) 3. Il gestore invia il comando "modifica videogioco" 4. SE tutti i campi inseriti sono validi <ol style="list-style-type: none"> a. Il sistema modifica il videogioco nel catalogo generale b. Il sistema restituisce un messaggio di conferma dell'avvenuta modifica 5. altrimenti 6. Il sistema indica i campi di errore
Post-Condizioni	Viene aggiunto il videogioco al catalogo
Casi d'uso correlati	

Lista dei Sequence Diagram

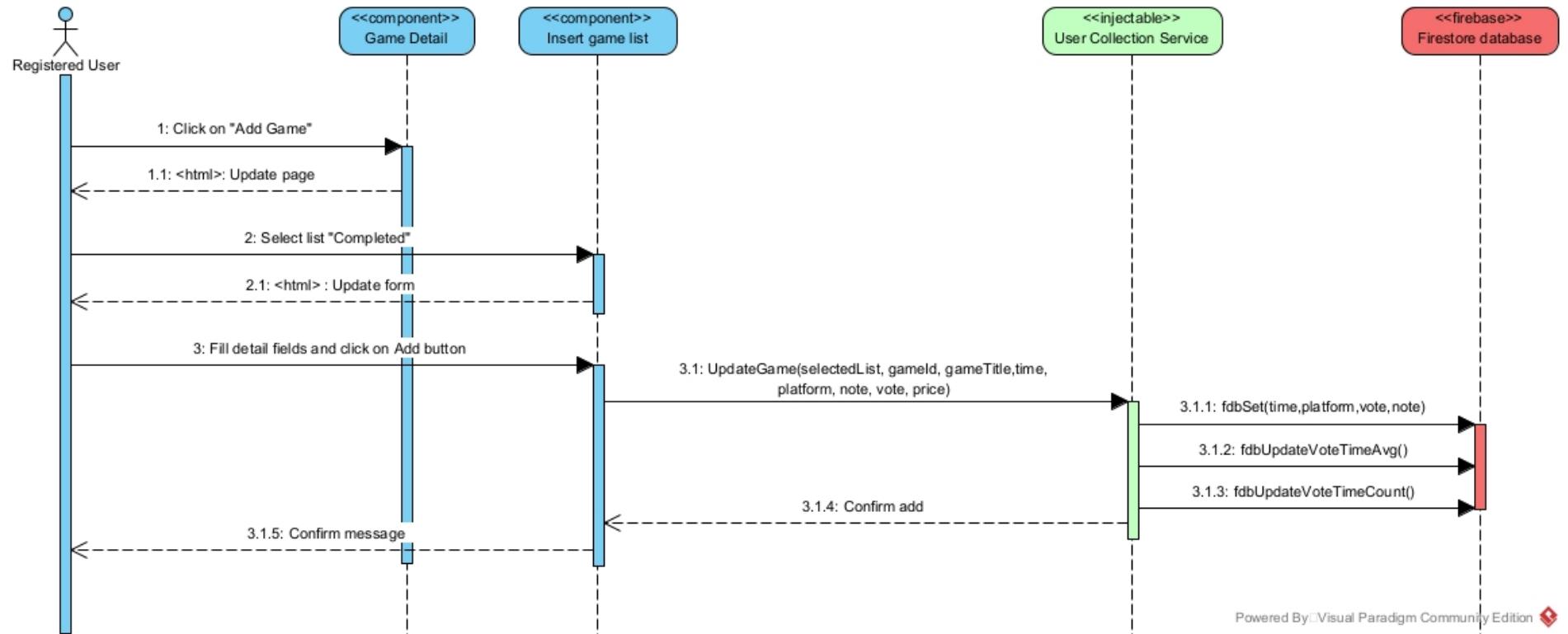
Di seguito si riportano i sequence diagram relativi ai casi d'uso di sopra. Le chiamate ai servizi di Firebase (principalmente *Firebase* e *Storage*) e alcuni argomenti di input sono stati semplificati per una maggiore comprensibilità del diagramma.

Ricerca Videogioco:



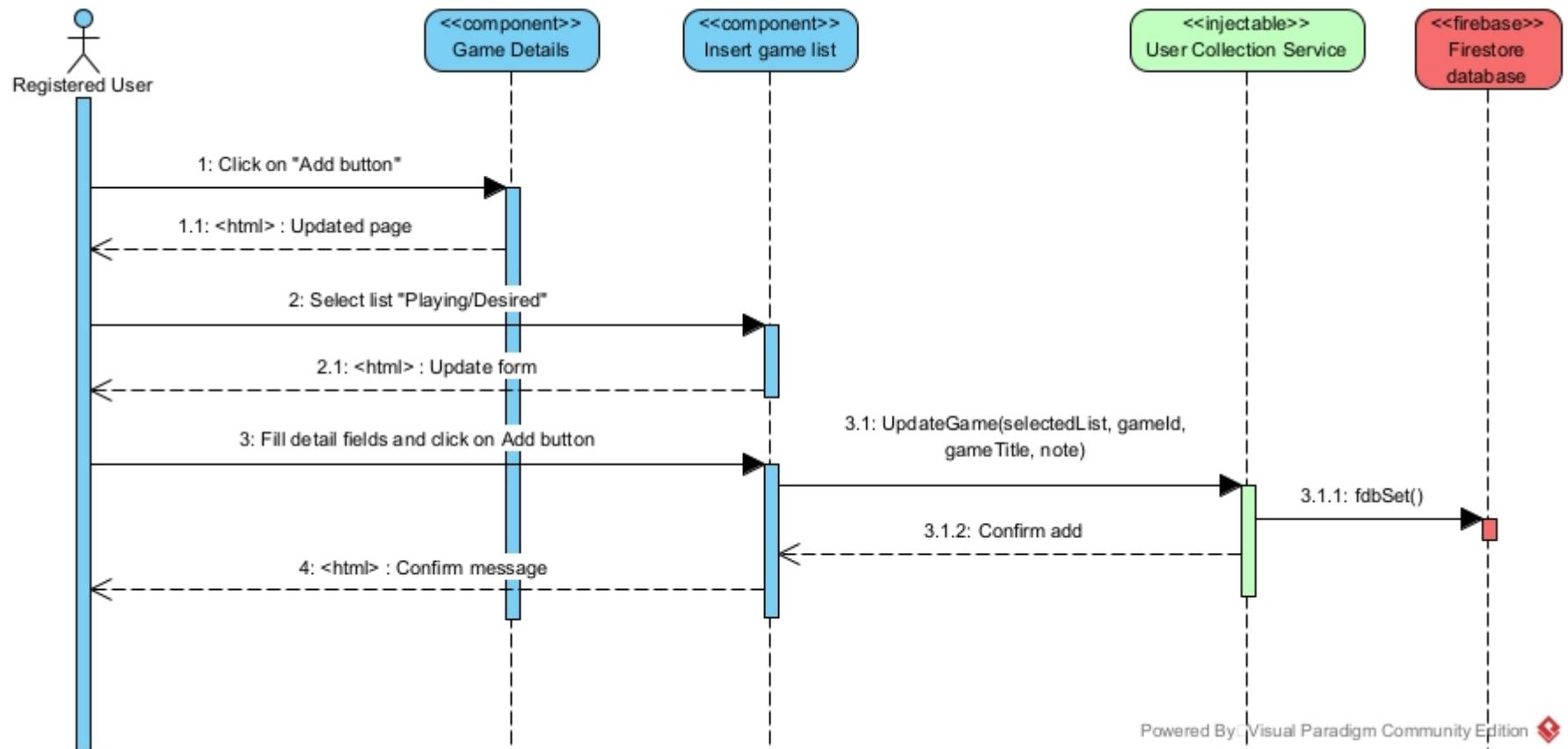
Powered By Visual Paradigm Community Edition

Aggiungi Videogioco alla Lista (Completati):



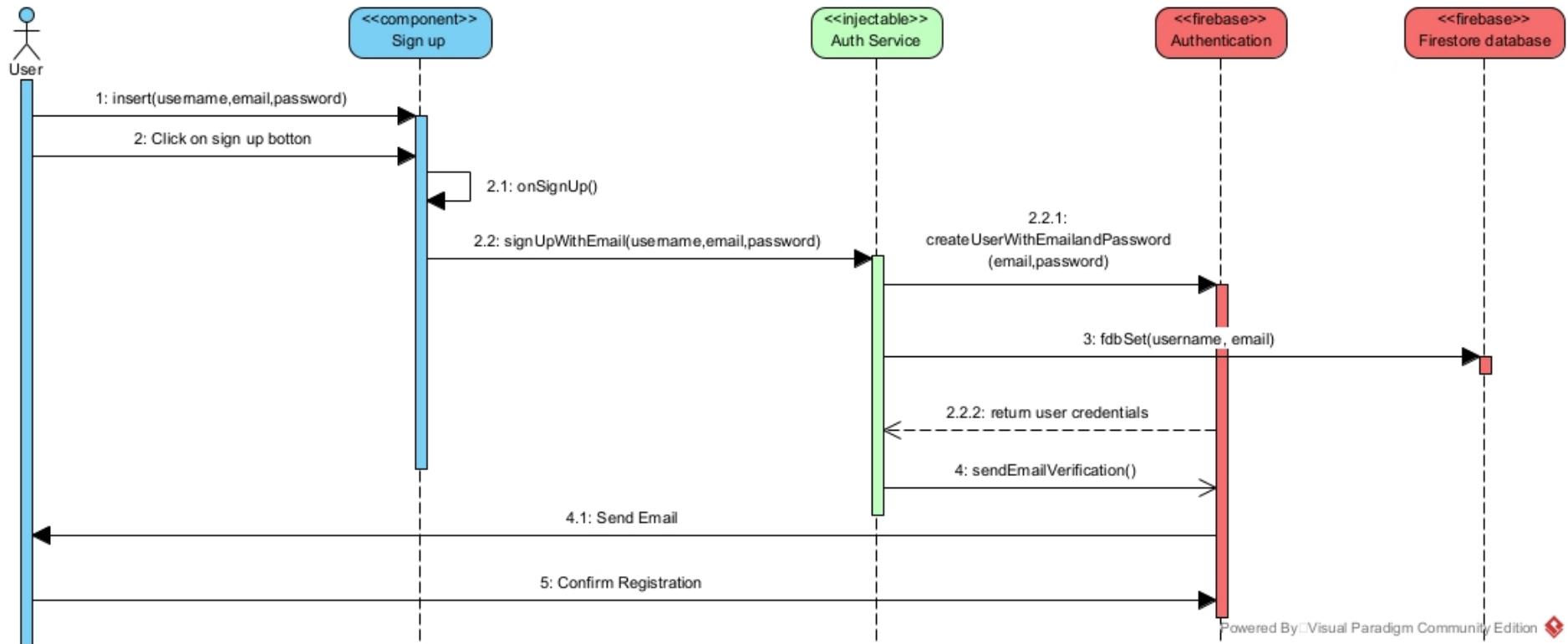
La chiamata di funzione fdbSet assume che si conosca la reference per accedere al documento del videogioco. In generale gli accessi al firestore nei sequence diagram ometteranno l'ottenimento della reference. (ad esempio la reference viene presa indicando l'utente e l'id del videogioco per l'accesso ad un videogioco della lista)

Aggiungi Videogioco alla Lista (In corso/Desiderati):

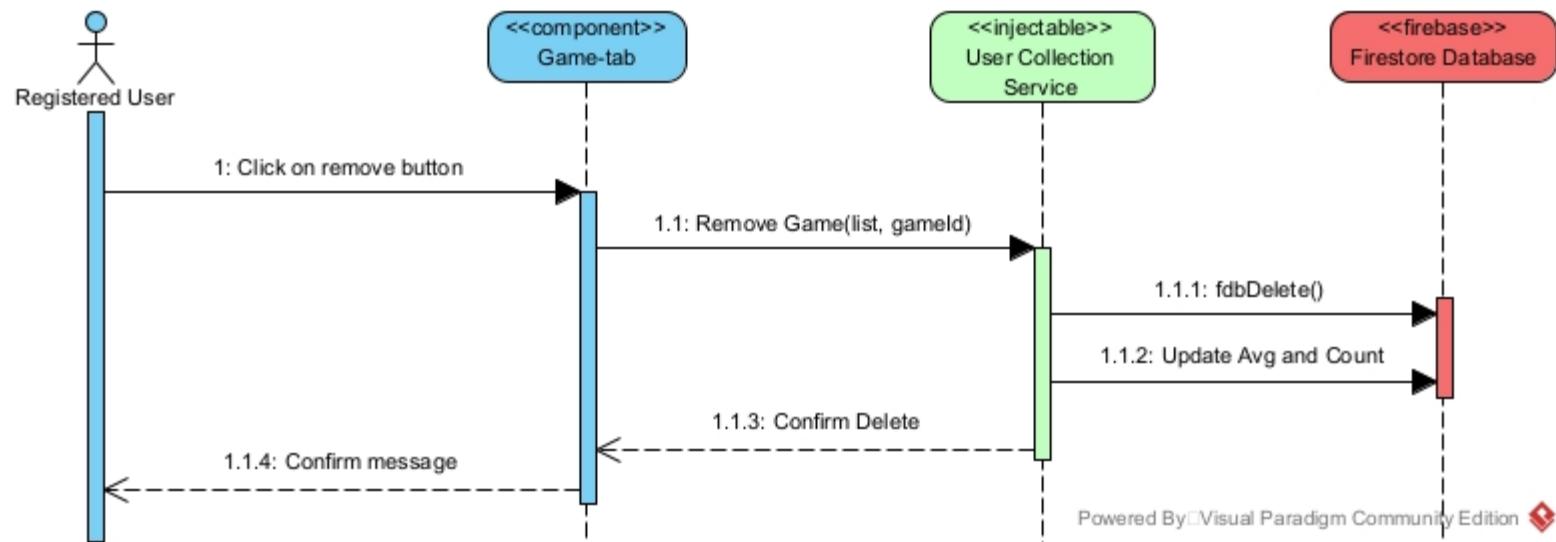


Powered By Visual Paradigm Community Edition

Registrazione:

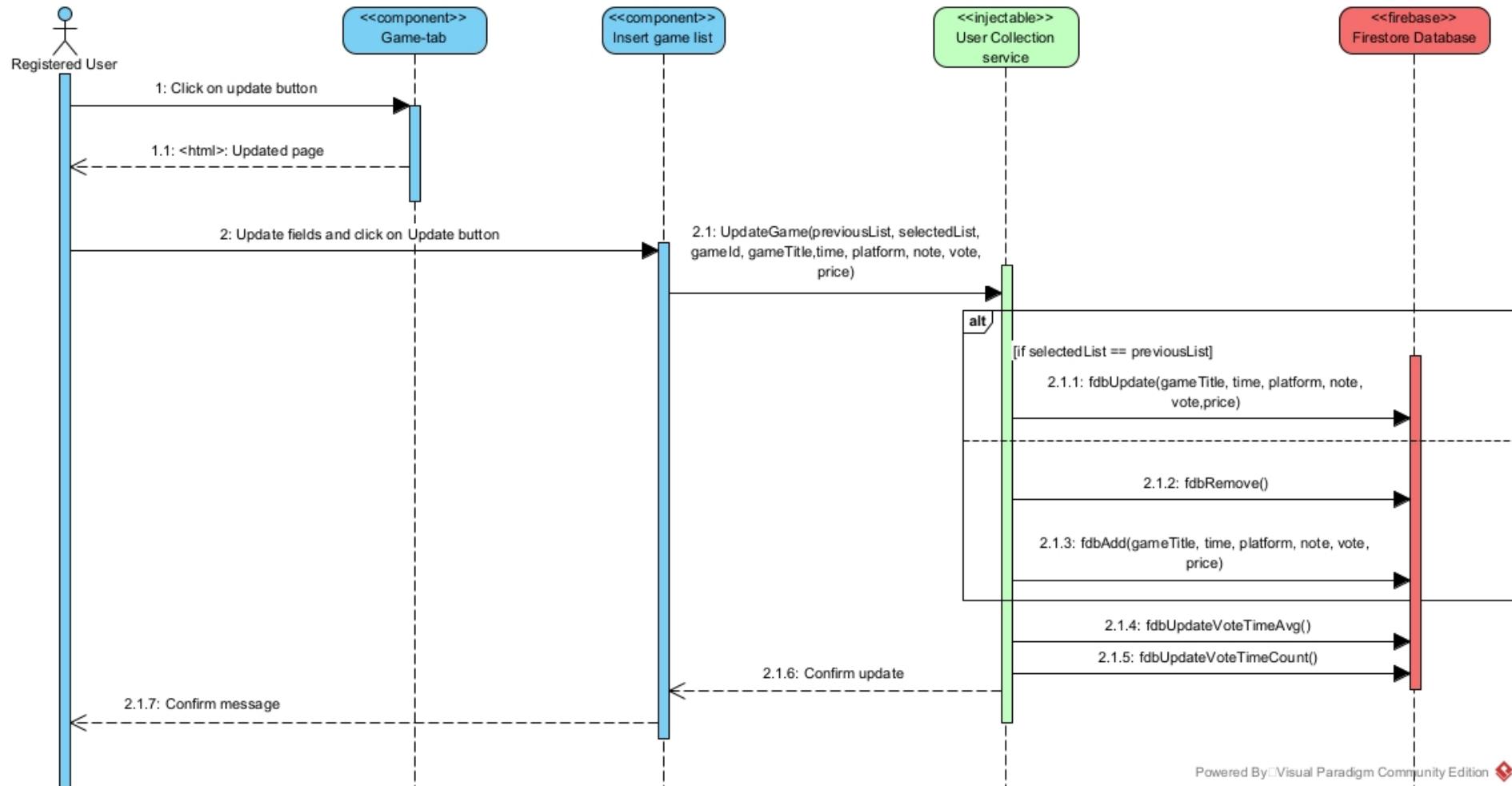


Rimuovi videogioco da una lista:



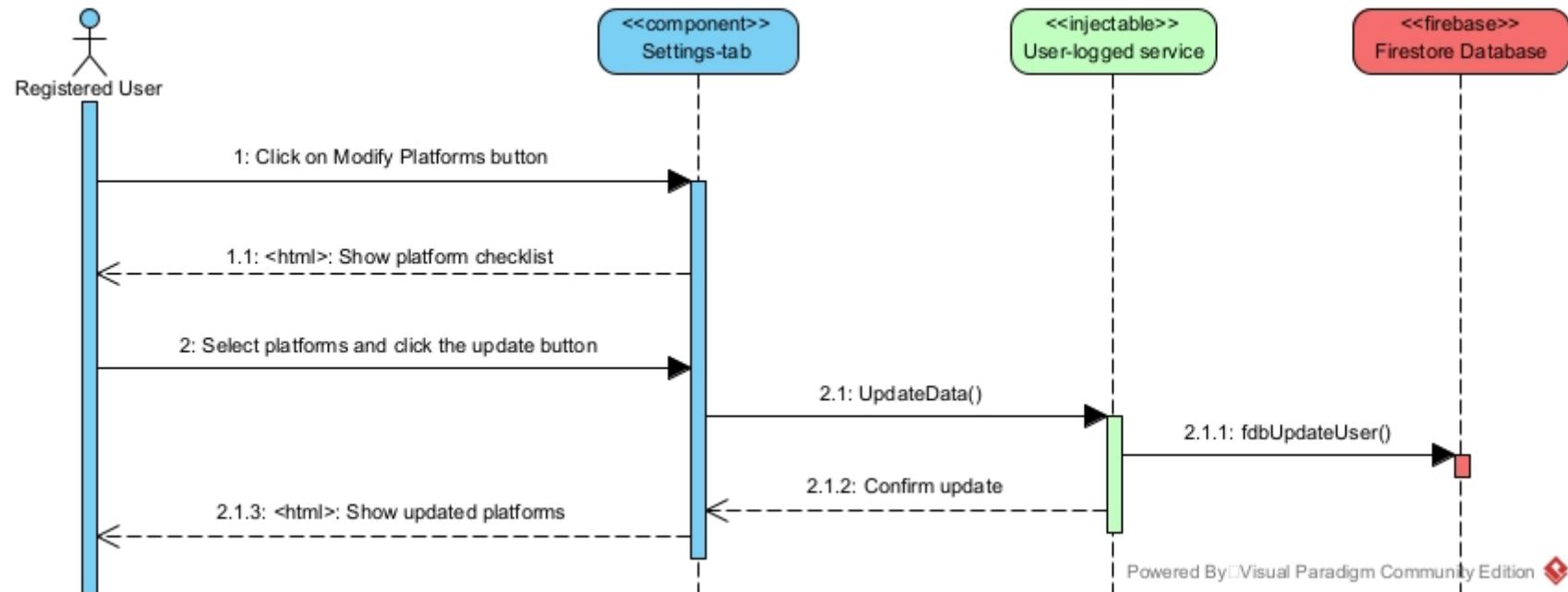
Dato che ogni videogioco nella raccolta è associato ad una determinata lista, si presuppone che il videogioco sia visibile solo nella lista in cui è stato inserito.
Inoltre, ad ogni videogioco è associato un proprio tasto *remove*

Modifica videogioco di una lista:



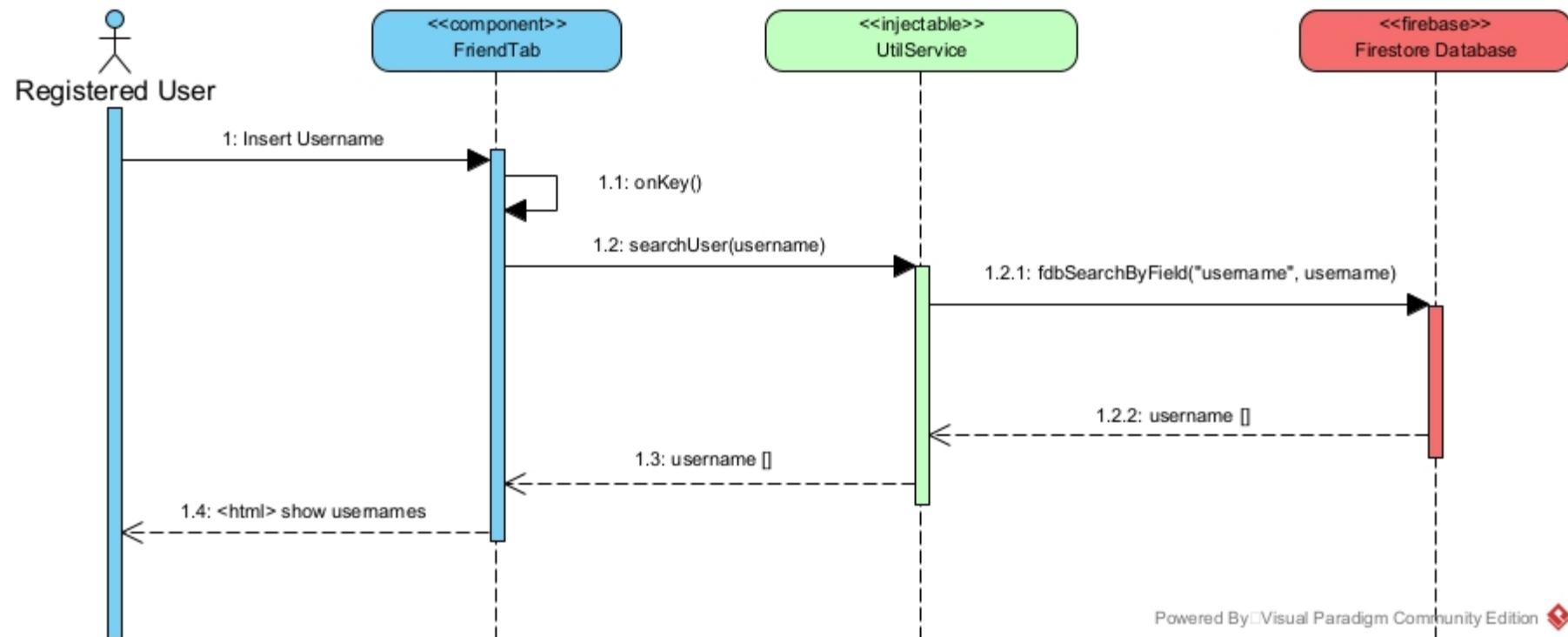
Questo caso d'uso presenta la modifica di un gioco che passa da una qualunque lista, alla lista dei giochi Completati. Questo, analogamente a Aggiungi gioco a una lista dà per scontata l'interazione in cui l'utente seleziona la lista giusta.

Aggiorna Preferenze (caso piattaforme):

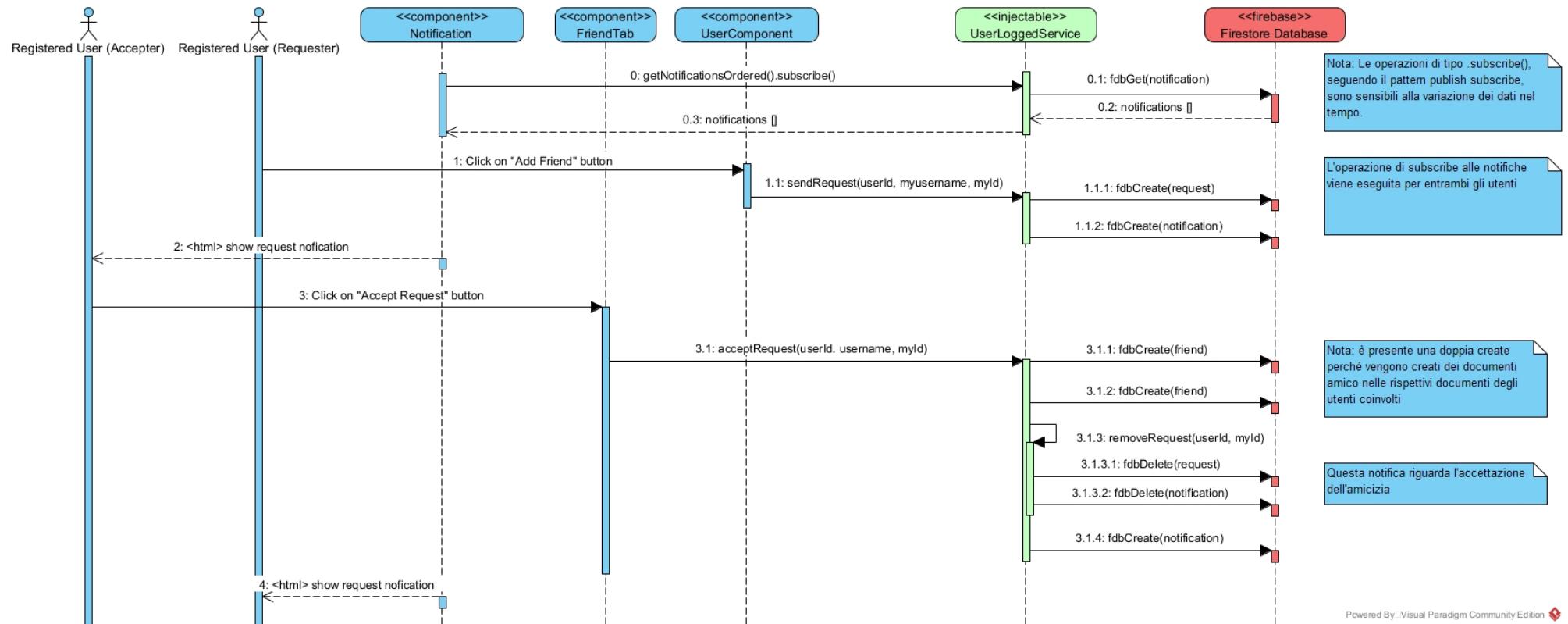


Il sequence diagram è uguale sia per la modifica della piattaforma che per quella dei generi preferiti

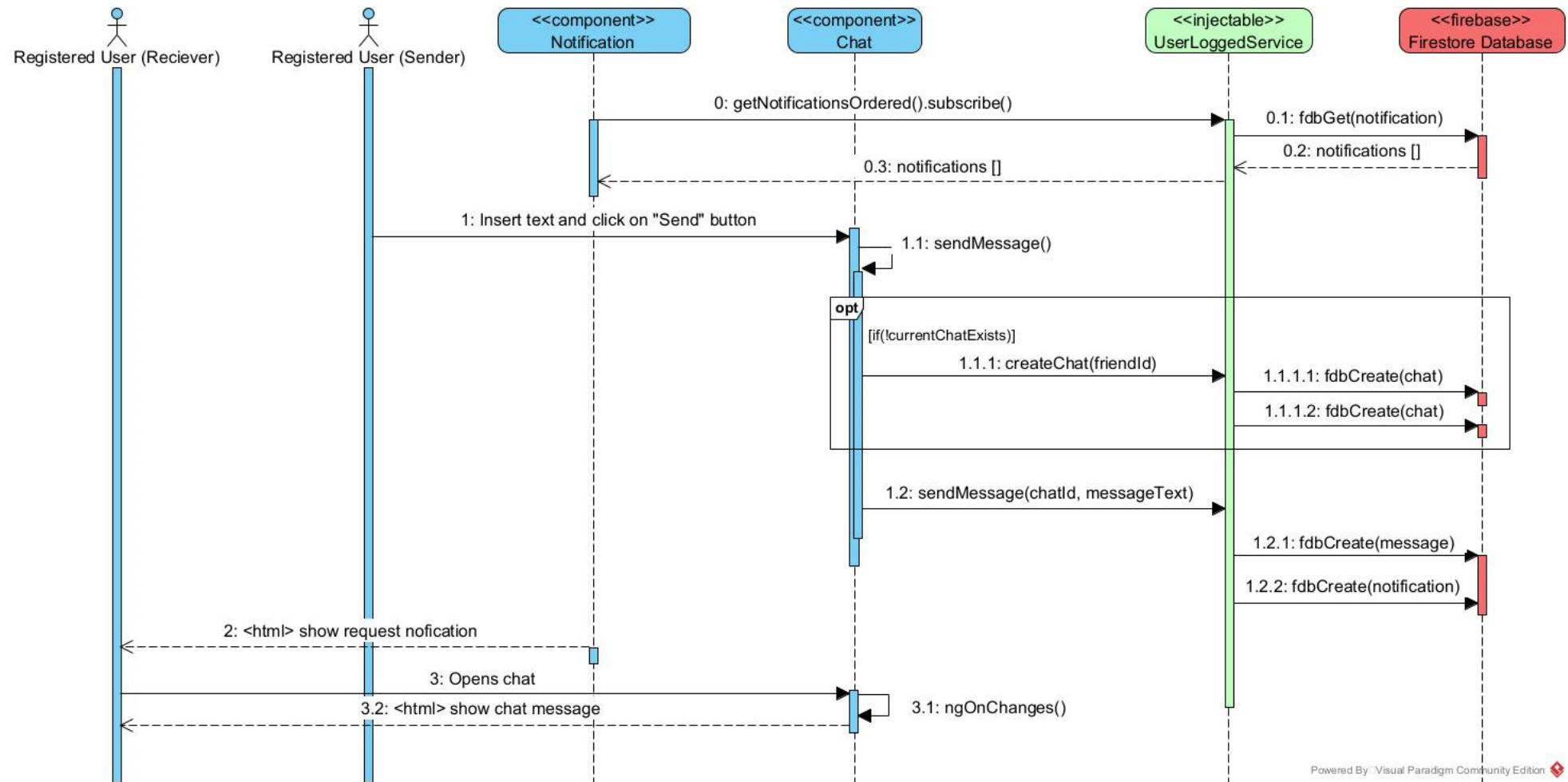
Ricerca Utente:



Aggiungi Amico:

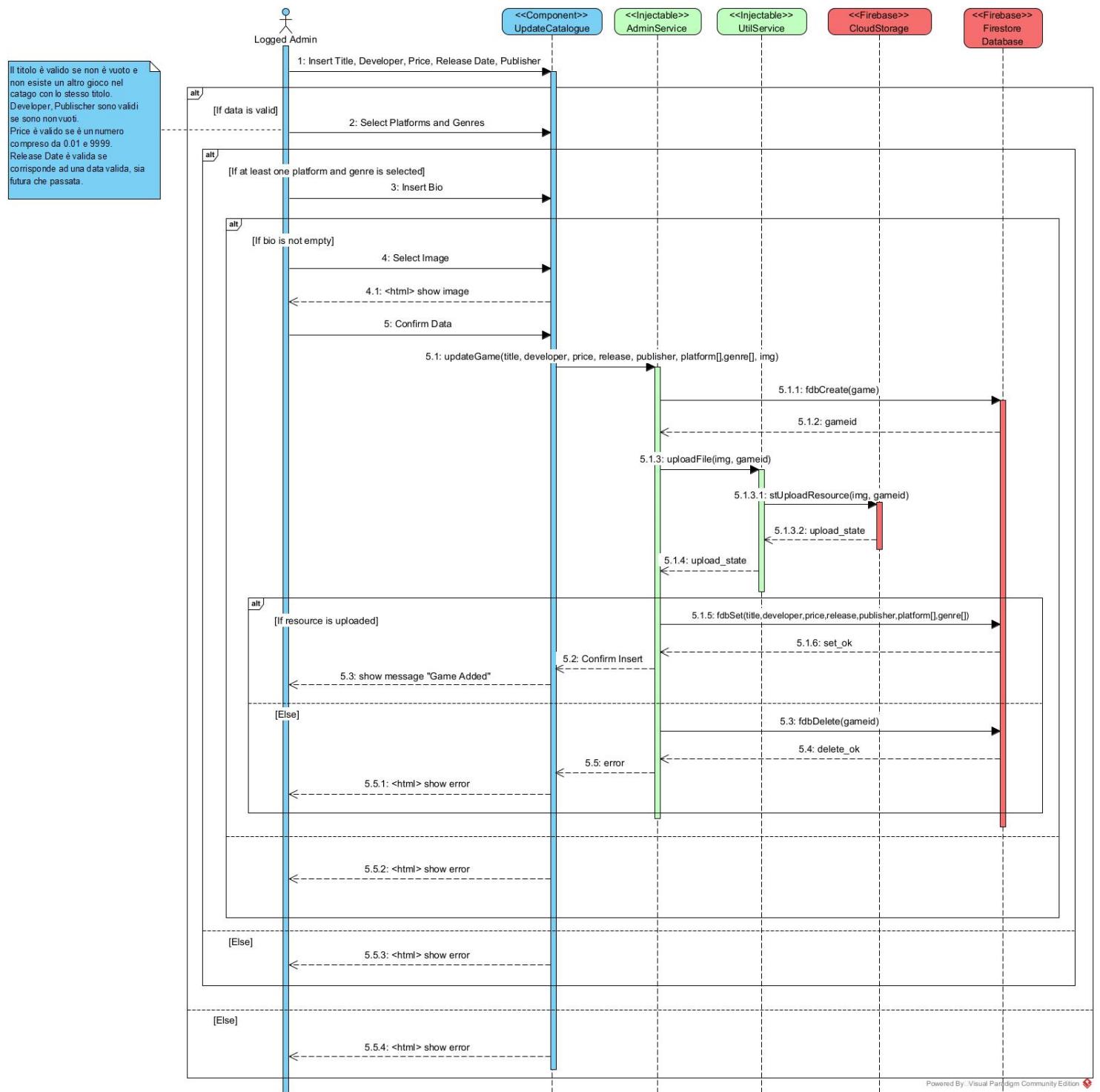


Invia Messaggio Privato:



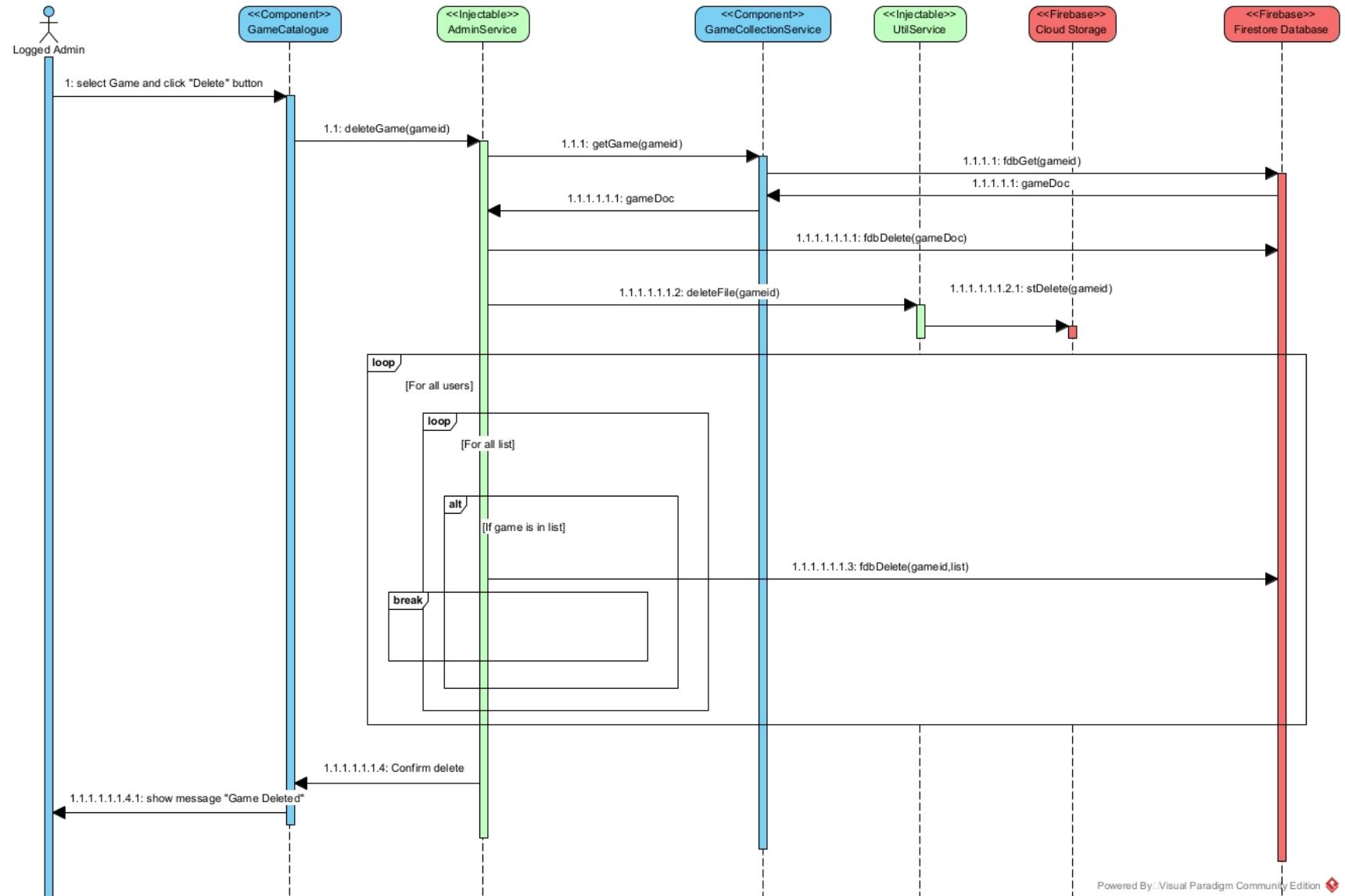
L'operazione `subscribe()`, seguendo il pattern publish/subscribe è un'operazione sensibile durante il tempo. In questo caso rappresenta un'operazione preliminare che viene eseguita ogniqualvolta un utente effettua il login.

Inserisci Videogioco:



La risorsa immagine viene caricata su CloudStorage avendo come nome l'id del documento su Firestore, per questo motivo viene creato prima il documento su Firestore.

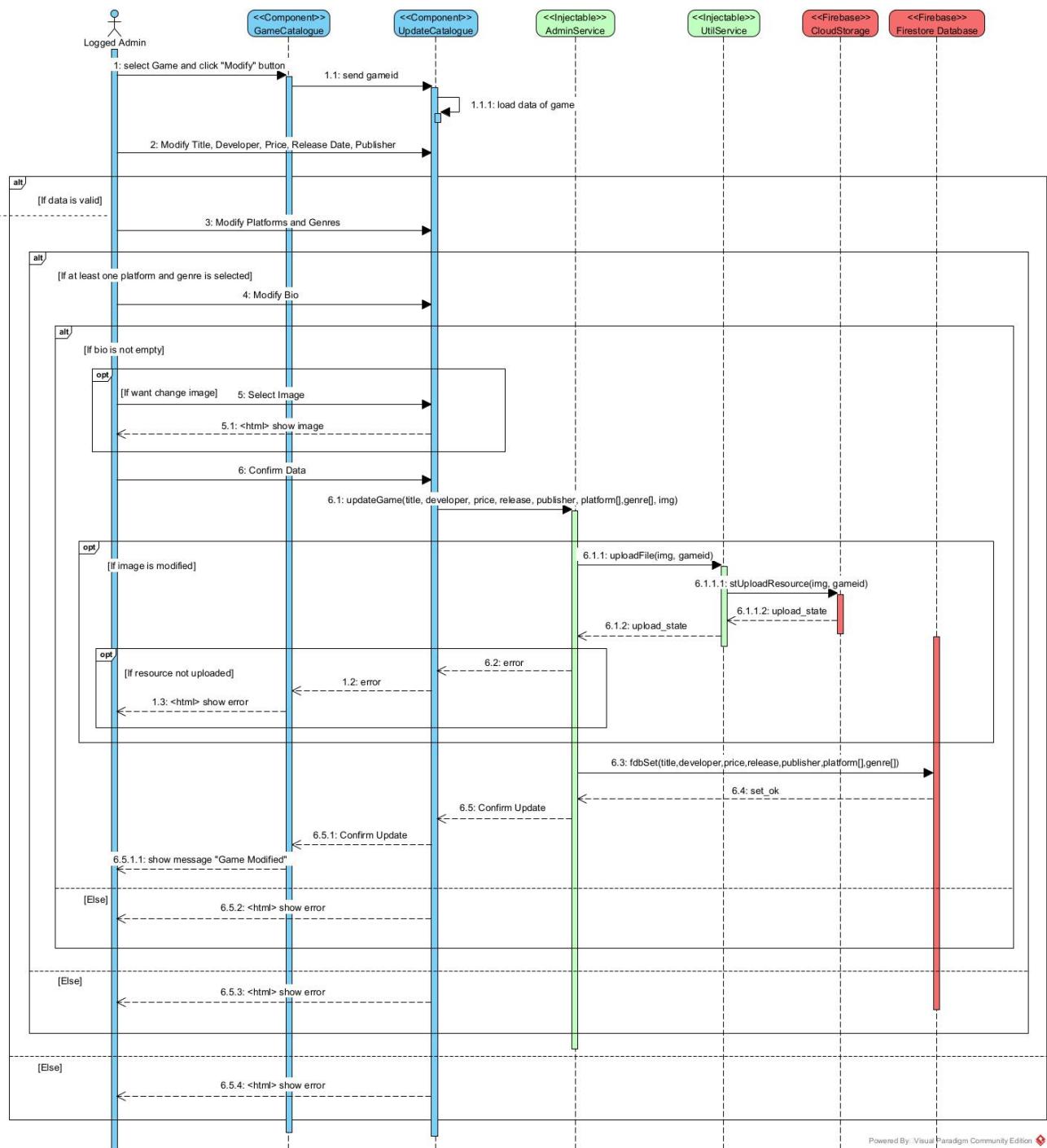
Elimina Videogame:



Quando viene eliminato un gioco dal catalogo, deve essere eliminato da ogni lista utente

Modifica Videogioco:

Il titolo è valido se non è vuoto e non esiste un altro gioco nel catalogo con lo stesso titolo.
 Developer, Publisher sono validi se sono non vuoti.
 Price è valido se è un numero compreso da 0.01 a 9999.
 Release Date è valida se corrisponde ad una data valida, sia futura che passata.



La risorsa immagine viene caricata su CloudStorage avente come nome l'id del documento su Firestore.