Sommario

Elaborazione analisi

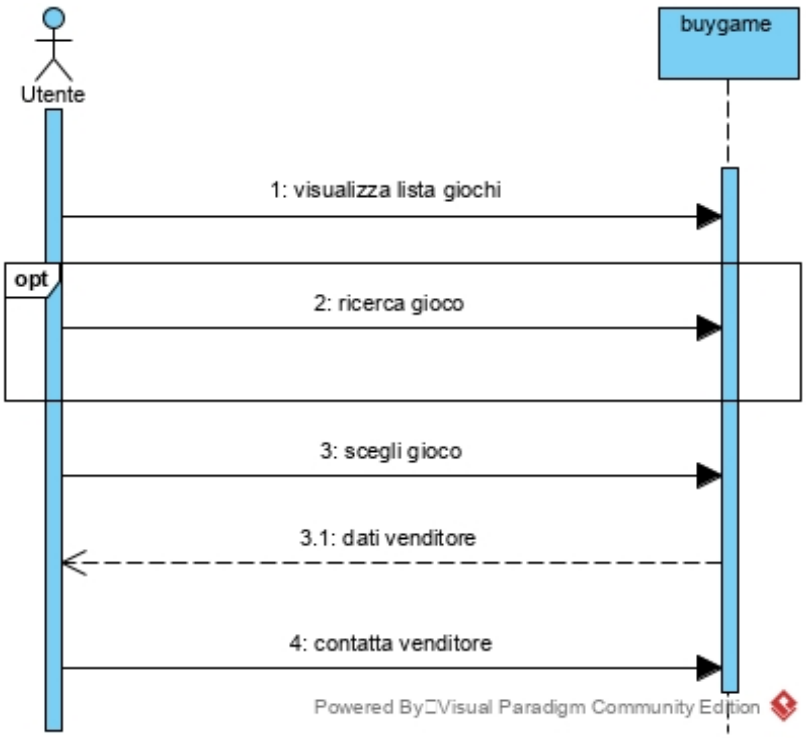
Descrizione dettagliata dei casi d'uso

Di seguito vengono riportati, in maniera dettagliata, i casi d'uso che si è scelto di implementare. Per ciascun caso d'uso vengono specificati attori, pre e post condizioni, scenario principale ed alternativo. Inoltre, vengono riportati gli SSD per descrivere ad alto livello come si intende implementare il caso d'uso specifico.

Acquista gioco

Caso d'uso	Acquista gioco
Attori primari	Utente
Attori secondari	Nessuno
Pre-condizioni	Nessuna
Scenario principale	1. Include(Visualizza lista giochi) #punto di estensione: Ricerca 2. L'utente seleziona un gioco dalla lista 3. Il sistema restituisce i dati del gioco e del venditore 4. L'utente seleziona e-mail o numero del venditore per contattarlo
Post-condizioni	Il cliente può compilare un'e-mail da inviare al venditore o avviare una chiamata per concludere l'acquisto
Scenari alternativi	Nessuno

System sequence diagram

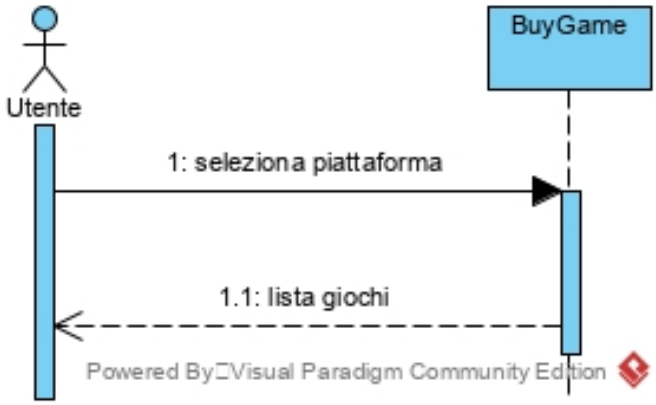


Visualizza lista giochi

Il seguente caso d'uso viene analizzato poiché è incluso nel caso d'uso precedente: Acquista gioco.

Caso d'uso	Visualizza lista giochi
Attori primari	Utente
Attori secondari	Nessuno
Pre-condizioni	Nessuna
Scenario principale	1. L'utente sceglie una tra le tre piattaforme disponibili 2. Il sistema preleva i giochi relativi solo alla piattaforma scelta e restituisce la lista
Post-condizioni	L'utente può leggere tutti gli annunci
Scenari alternativi	Nessuno

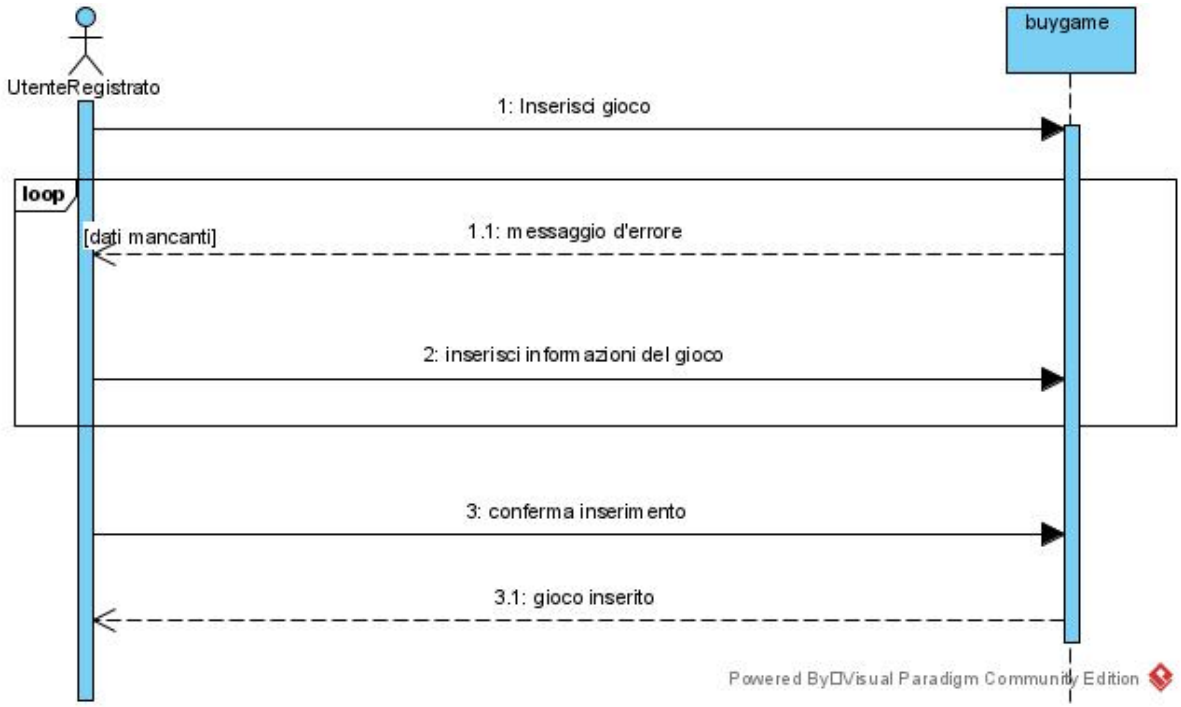
System sequence diagram



Inserisci gioco

Caso d'uso	Inserisci gioco
Attori primari	Utente registrato
Attori secondari	Nessuno
Pre-condizioni	L'utente deve essere registrato ed aver effettuato il login
Scenario principale	1. L'utente decide di creare un nuovo gioco 2. Il sistema permette di inserire i dati del gioco 3. L'utente conferma l'inserimento del gioco 4. Il sistema inserisce il gioco e restituisce una notifica di successo
Post-condizioni	Il gioco viene creato ed inserito nel database
Scenari alternativi	3.1 Il sistema rileva la mancanza di alcune informazioni 1 Il sistema restituisce un messaggio di errore e riporta l'utente al punto 2

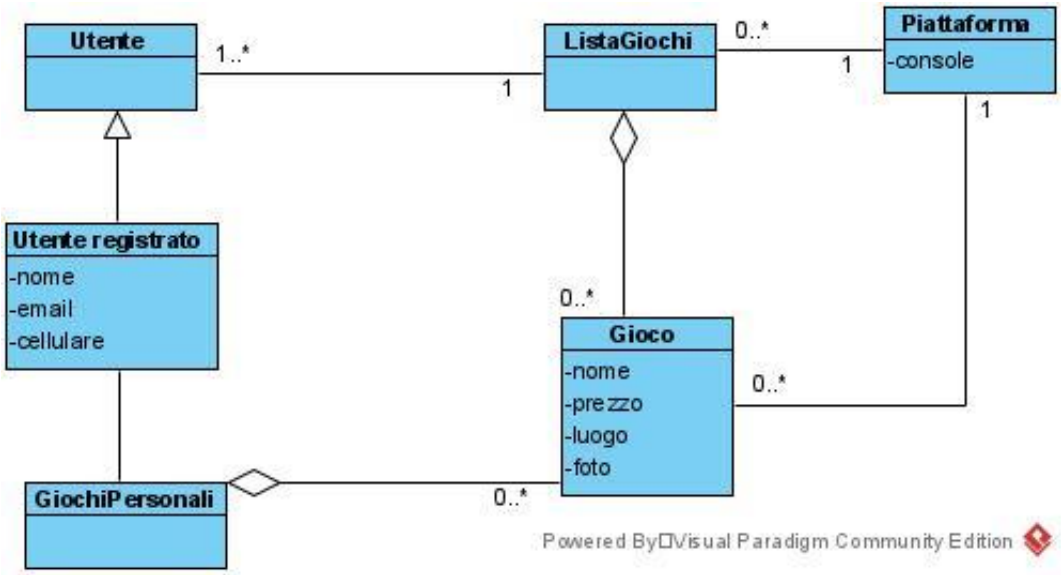
System sequence diagram



System domain model

Con il seguente modello descriviamo il dominio informativo del problema, al fine di evidenziare gli aspetti essenziali del sistema tralasciando, in fase di analisi, le operazioni definite su di esso.

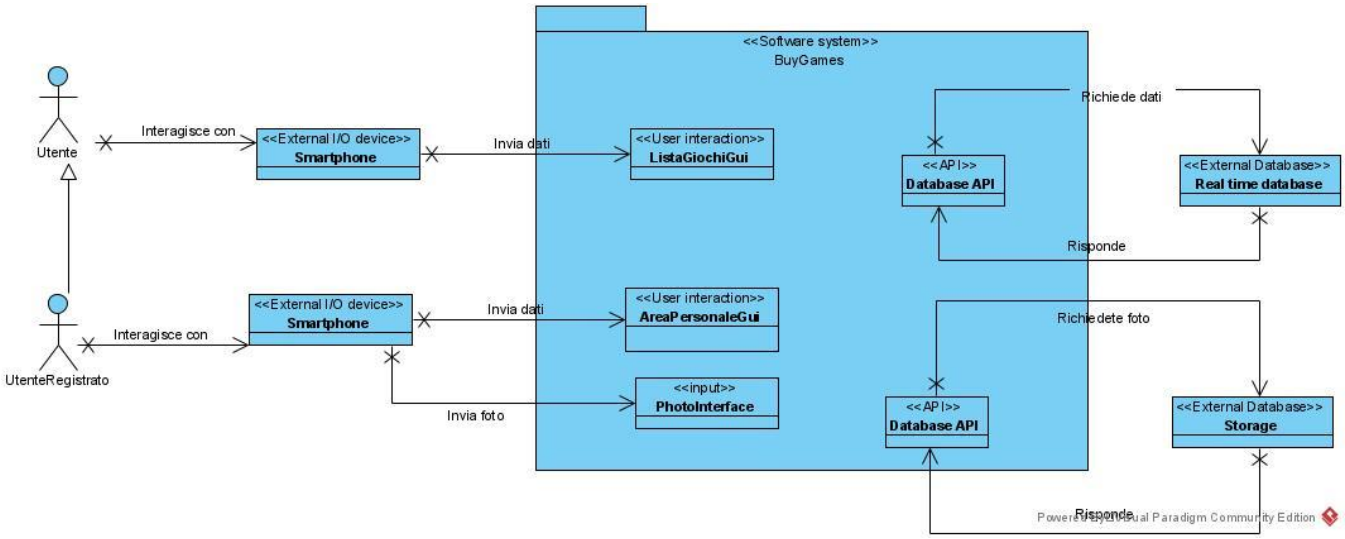
Si noti come nel diagramma sono presenti le classi utente ed utente registrato per mostrare come soltanto nel caso in cui l'utente sia registrato sia possibile accedere ad una lista di giochi personali; nel caso in cui l'utente non sia registrato è possibile accedere alla lista dei giochi messi in vendita e visibili a tutti. Inoltre, è presente una classe piattaforma associata alla lista giochi che consente di distinguere le differenti liste in base alla piattaforma selezionata.



Context diagram

L'obiettivo del seguente diagramma è quello di mostrare l'ambiente con cui il sistema interagisce. Le entità del contesto con cui il sistema interagisce sono umani ma anche altri sistemi fisici come lo smartphone, la sua fotocamera o un sistema di memorizzazione dei dati (RDBMS).

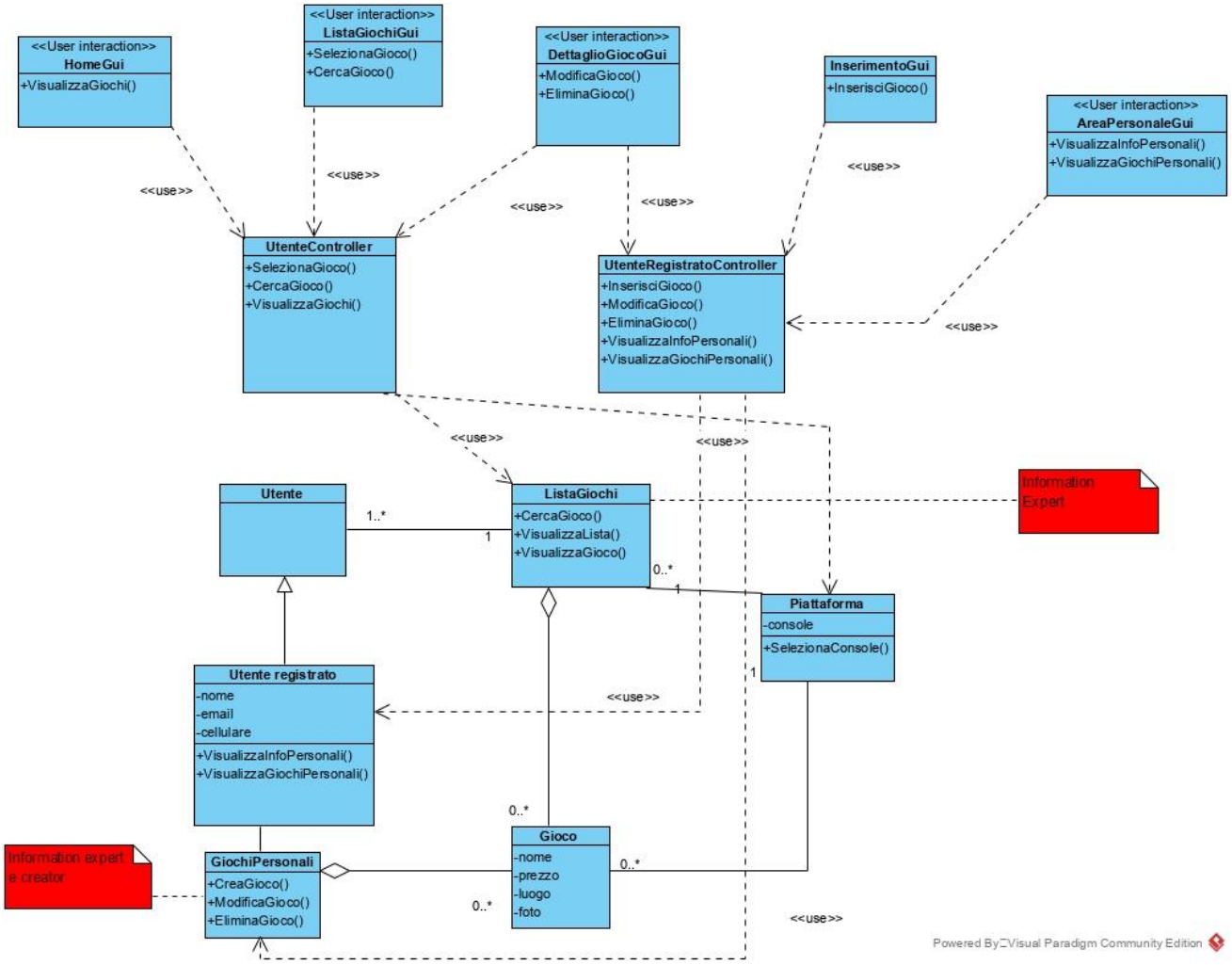
Formalmente, il contesto del sistema è descritto dal seguente diagramma degli oggetti.



Nome	Tipo	Descrizione
ListaGiochiGui	User interaction	Interfaccia che consente all'utente di scorrere la lista dei giochi messi in vendita dagli utenti.
AreaPersonaleGui	User interaction	Interfaccia che consente all'utente registrato di scorrere la lista dei giochi messi in vendita da se stesso.
Smartphone	External I/O device	Smartphone android su cui è installata l'applicazione e dotato di fotocamera.
Photo interface	Input	Interfaccia che consente di acquisire le foto scattate con la fotocamera dello smartphone.
External environment	External input	Ambiente esterno al quale vengono scattate le foto con lo smartphone.
DatabaseAPI	API	Interfaccia che gestisce la comunicazione con il database.
Real time database	External database	Database noSQL utilizzato per la memorizzazione delle informazioni degli utenti registrati e dei giochi.
Storage	External database	Database utilizzato per la memorizzazione delle foto dei vari giochi.

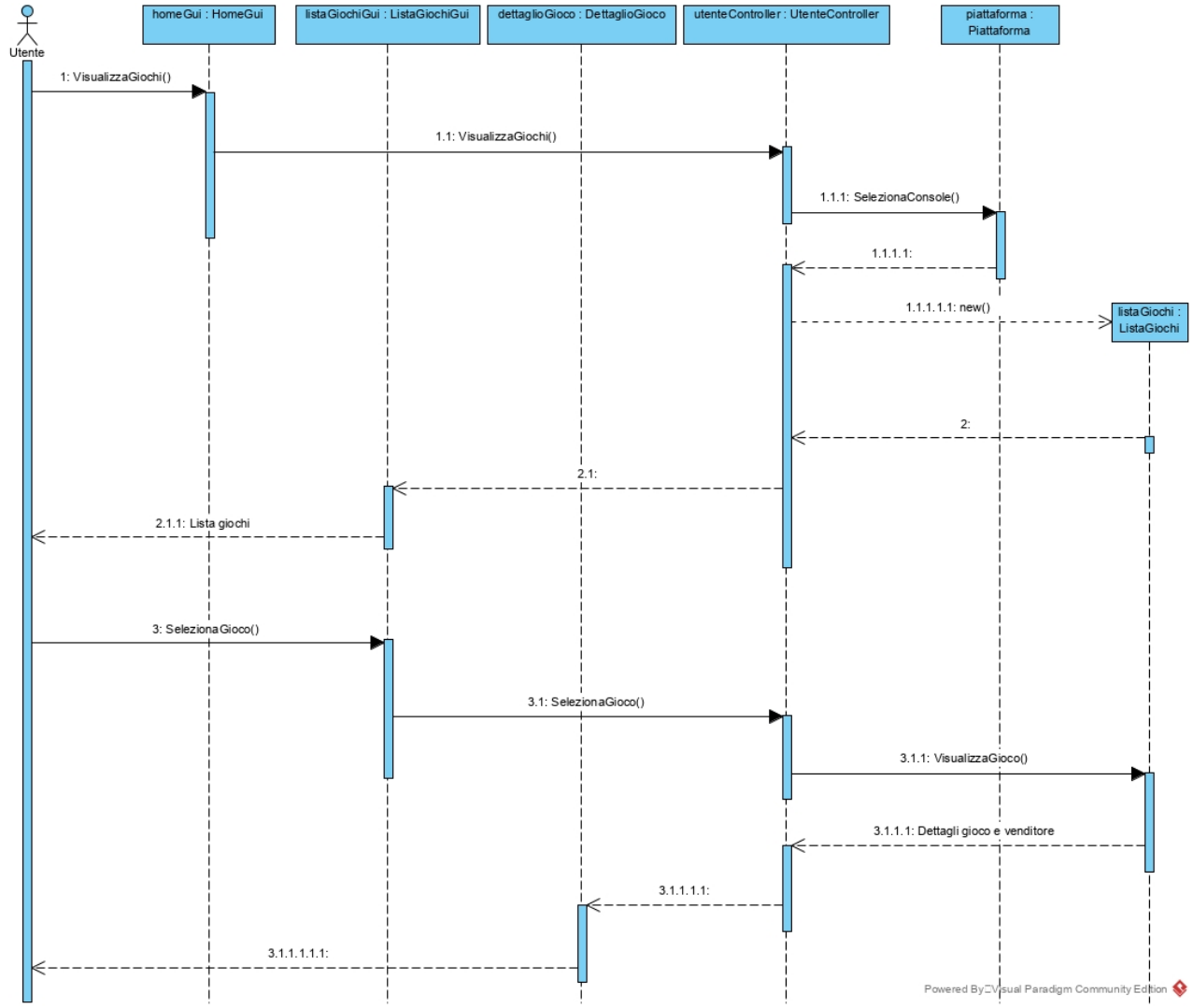
Grasp

Con il seguente diagramma sono state assegnate le varie responsabilità alle classi relative ai casi d'uso in formato dettagliato. Vengono illustrati i pattern utilizzati: in particolare si noti come il pattern *information expert* è utilizzato per modellare la relazione tra ListaGiochi e Gioco e tra GiochiPersonali e Gioco. Inoltre, tra GiochiPersonali e Gioco vi è una relazione modellata dal pattern *creator* poiché la responsabilità della creazione di nuovi giochi è affidata alla classe GiochiPersonali.

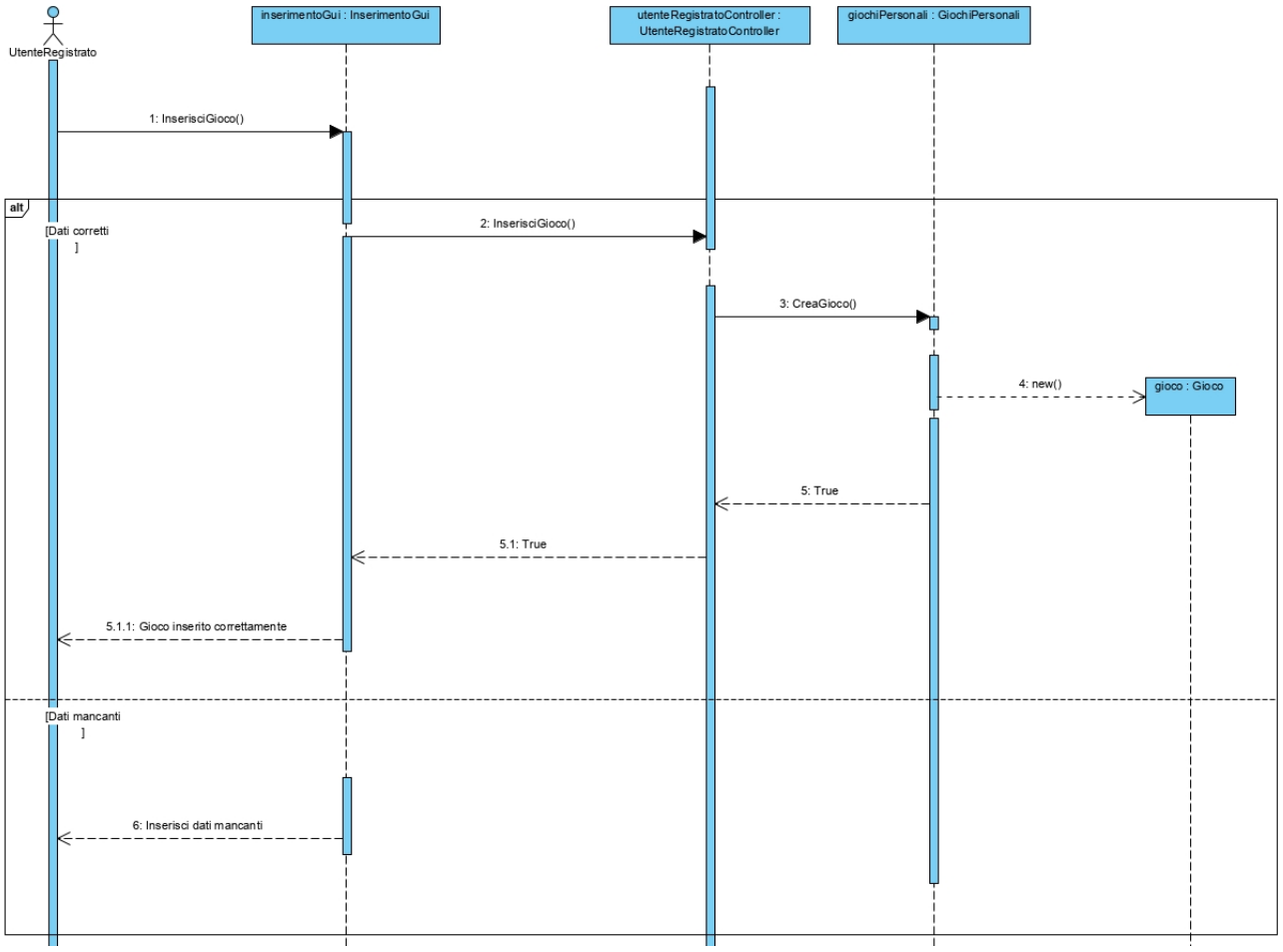


Sequence diagram di analisi

Acquista gioco



Inserisci annuncio

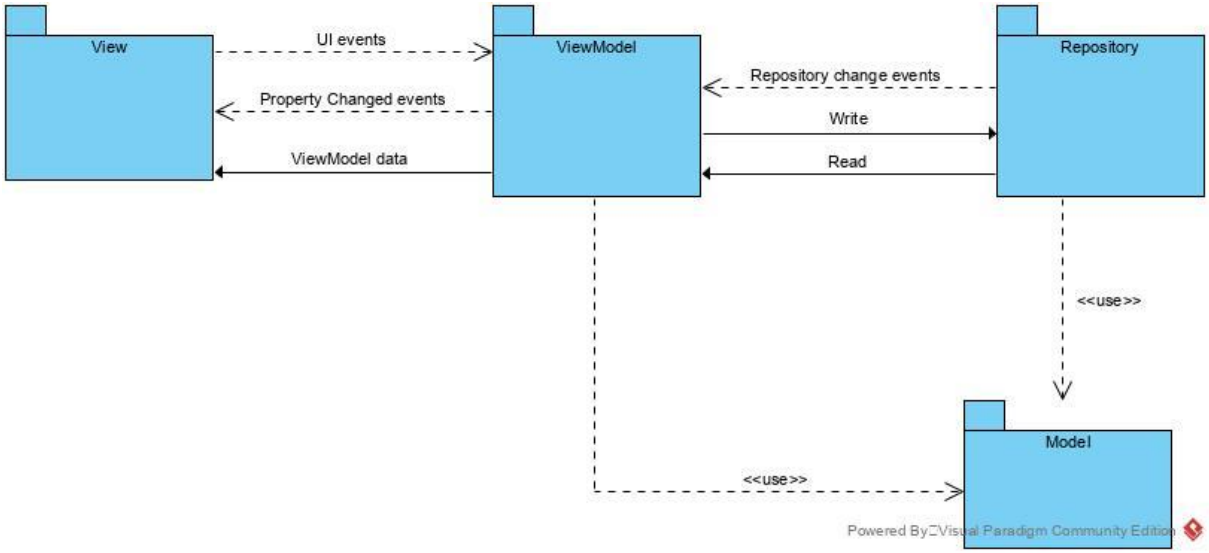


Architettura logica

In questo paragrafo viene riportata la progettazione di una soluzione al problema in termini di oggetti software che collaborano. In particolare, viene mostrata l'architettura logica del sistema, la quale modella la distribuzione delle classi in *package*. In questo documento si vuole utilizzare una prospettiva di alto livello, pertanto, non si vuole scendere nel dettaglio di come le classi siano organizzate nei vari package ma ci si vuole soffermare soltanto sulla struttura logica del software.

Il pattern scelto per realizzare la soluzione è il Model-View-ViewModel il quale, grazie alla divisione in tre livelli differenti, facilita la codifica e la testabilità dei singoli moduli.

Package diagram



Sarà solo il *package* repository a comunicare col *database*.