Università degli Studi di Salerno

Corso di Ingegneria del Software

BesTicket
ODD
Versione 1.0

LOGO PROGETTO

Data: 04/ 01/2018

Coordinatore del progetto:

Nome	Matricola
Andrea De Lucia	
Rita Francese	

Partecipanti:

Nome	Matricola
Myriam Imbriano	0512103618
Manuel Fuschetto	0512103648

Scritto da:	Myriam Imbriano, Manuel Fuschetto

Revision History

Data	Versione	Descrizione	Autore
04/01/2018	1.0	Object Design Document	Myriam Imbriano, Manuel Fuschetto

Indice

1. Introduzione

1.1 Object Design Trade-Offs

Comprensibilità vs Tempo:

Il nostro progetto dovrà essere manutenibile quindi il codice dovrà essere quanto più comprensibile possibile per facilitare eventuali future modifiche. I vari metodi saranno quindi accompagnati da commenti che ne semplificheranno la comprensione. Ovviamente questa caratteristica aggiungerà un incremento di tempo allo sviluppo del nostro progetto.

Interfaccia vs Usabilità:

L'interfaccia del sito è stata realizzata in modo da essere semplice e intuitiva, grazie alle disposizioni dei vari form e pulsanti, tutti a portata di mano, si ha la possibilità, tramite la home, di accedere a quasi tutti i servizi del Sito.

Sicurezza vs Efficienza:

La sicurezza rappresenta uno degli aspetti importanti del sistema. Tuttavia, dati i tempi di sviluppo molto limitati, ci limiteremo ad implementare sistemi di sicurezza basati su username e password degli utenti.

1.2 Linee Guida per la Documentazione delle Interfacce

Per rendere il codice più estensibile e manutenibile, prima dell'implementazione della logica del sistema, è opportuno sottomettere le regole di implementazione in modo che eventuali correzioni nella logica dell'applicazione possono essere apportate prima di imbattersi nella sintassi degli strumenti scelti.

Commenti
I commenti di implementazione sono dei mezzi per commentare una particolare implementazione. I commenti dovrebbero essere usati per fornire informazioni aggiuntive che sono prontamente disponibili nel codice stesso.
I metodi devono essere preceduti da un commento e inoltre bisogna commentare, giustificare le eventuali decisioni o i calcoli.

Dichiarazioni
Non dichiarare le variabili al loro primo uso, può portare incomprensioni al programmatore e rendere la manutenibilità del codice più complessa e di difficile comprensione.

Indentazione
L'indentazione deve essere effettuata con un TAB e a prescindere dal linguaggio usato per la produzione del codice, ogni istruzione deve essere opportunamente indentata.
Es.

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Deve essere sostituita da:

```
<html>
        <head>
        </head>
        <body>
        </body>
</html>
```

Parentesi
A prescindere dalle istruzioni che seguono un IF o ciclo FOR e WHILE, è necessario, laddove ci fosse anche una sola istruzione, riportare il blocco di istruzioni tra parentesi graffe.

Script Javascript
Gli script che svolgono funzioni distinte dal funzionamento di una pagina, dovrebbero essere collocati in file separati.

1.3 Definizioni, Acronimi e Abbreviazioni

BesTicket: Nome del Sistema in Sviluppo;

Utente non Registrato: Utente non Loggato che può usufruire soltanto della ricerca dei concerti;

Utente Registrato: Utente registrato e loggato che può usufruire di tutte le funzioni che la piattaforma mette a disposizione;

Amministratore Concerti: Amministratore registrato e loggato che può usufruire di tutte le funzioni che la piattaforma mette a disposizione;

Amministratore Utenti: Amministratori registrato e loggato che può usufruire di tutte le funzioni che la piattaforma mette a disposizione;

Form: Insieme di una o più Textbox che fornisce all'utente autenticato l'inserimento di determinati requisiti o altro;

JDBC: Java DataBase Connection.

DBMS: Database Management System, Sistema di gestione del database

Mysql: è il più diffuso database Open Source basato sul linguaggio SQL.

RAD: Requirements Analysis Document.

SDD: System Design Documents

ODD: Object Design Document.

HTML: Linguaggio di mark-up per pagine Web.

CSS: Linguaggio usato per definire la formattazione delle pagine web.

Javascript: Linguaggio di scripting orientato agli oggetti e agli eventi, comunemente usato nella programmazione Web lato client.

1.4 Riferimenti

Object Software Engineering – Using UML, Patterns and Java.
Documento RAD del progetto **BesTicket**.
Documento SDD del progetto **BesTicket.**

2. Package

Il Sistema utilizzerà un'architettura Three-Tier, in particolare:

- **Il Livello di Presentazione:** è il livello composto da tutti i Boundary Object, come le form, che gli utenti vedono e con i quali possono interagire;
- **Il Livello Applicazione:** è il livello composto dagli oggetti responsabili dell'elaborazione dati e di notificare cambiamenti al Livello di Presentazione. Inoltre questo livello comunicherà anche con il database tramite il Livello Database sottostante;
- **Il Livello Database:** è il livello che si occupa dell'immagazzinamento dei dati persistenti e del loro loro recupero dal database;

Il sottosistema Livello di Presentazione è stato suddiviso in quattro sottosistemi:

1. **Sottosistema Utente Non Registrato:** include tutte le interfacce grafiche a cui l'Utente non Registrato può accedere ovvero: visualizzare il catalogo, visualizzare informazioni del sito, visualizzare la pagina relativa alla Registrazione e la Homepage;
2. **Sottosistema Utente Registrato:** include tutte le interfacce grafiche a cui l'Utente Registrato può accedere ovvero: visualizzare il catalogo, visualizzare informazioni del sito, visualizzare il carrello e procedere successivamente all'acquisto, visualizzare le proprie informazioni personali e la Homepage;
3. **Sottosistema Amministratore Concerti:** include tutte le interfacce grafiche a cui l'Amministratore Concerti può accedere ovvero: visualizzare il catalogo, visualizzare

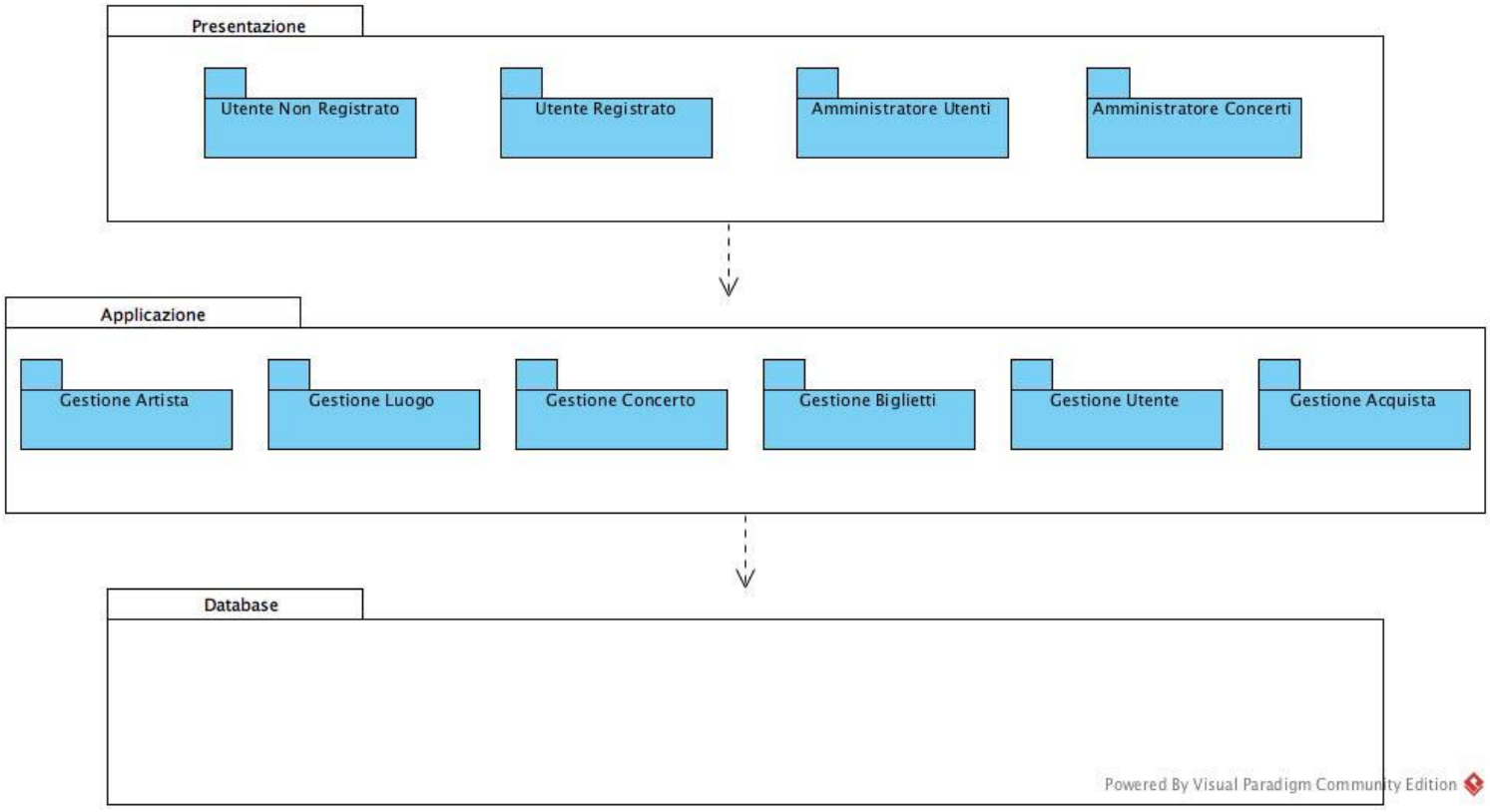
informazioni del sito, visualizzare le proprie informazioni personali, visualizzare le pagine relative all'amministrazione dei Concerti (es. aggiunta e rimozione di un Concerto o di un Luogo) e la Homepage;

4. **Sottosistema Amministratore Utenti:** include tutte le interfacce grafiche a cui l'Amministratore Utenti può accedere ovvero: visualizzare il catalogo, visualizzare informazioni del sito, visualizzare le proprie informazioni personali, visualizzare le pagine relative all'amministrazione degli Utenti (es. promozione e rimozione di un Utente) e la Homepage;

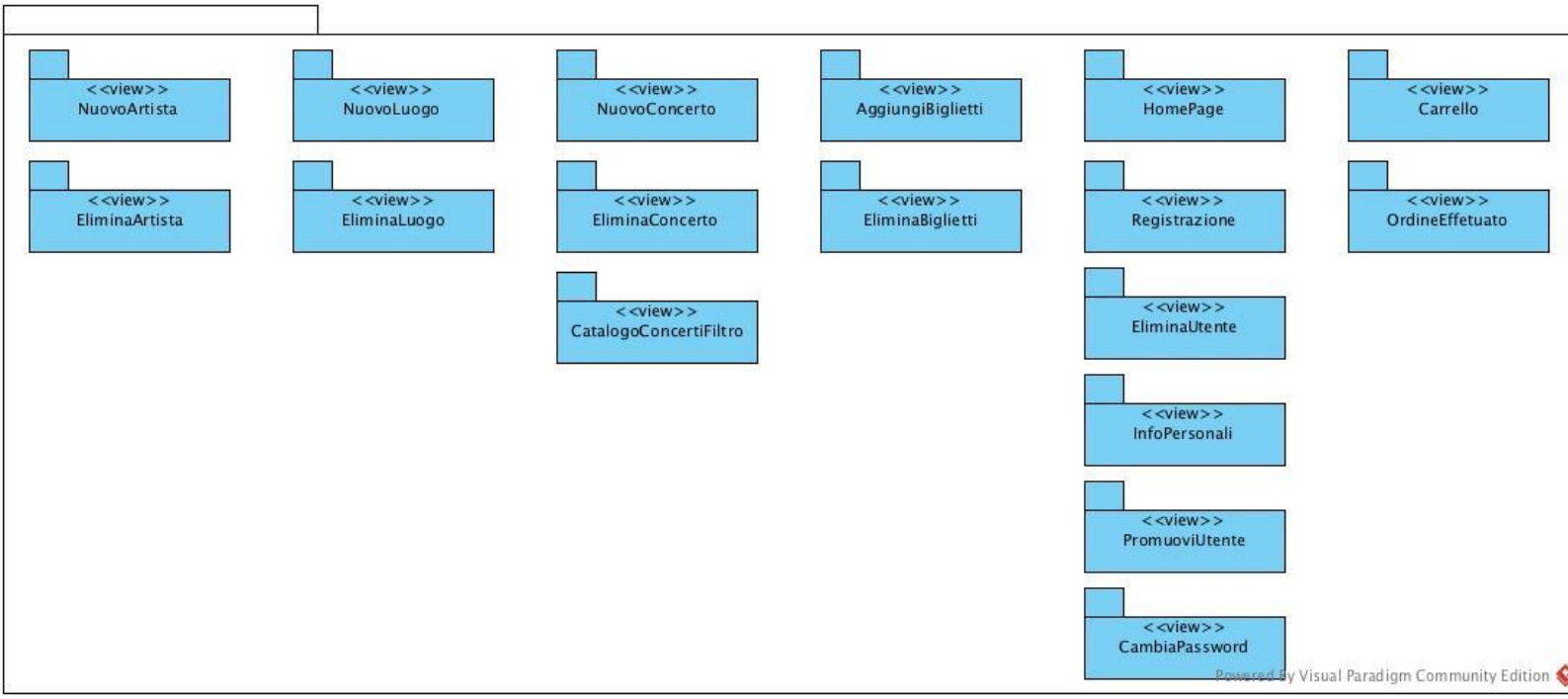
Il sottosistema Livello di Applicazione è stato suddiviso in vari sottosistemi:

1. **Sottosistema Gestione Concerti:** include tutte le operazioni che permettono l'aggiunta di un Concerto, l'eliminazione di un Concerto, la visualizzazione di un Concerto tramite un filtro e la visualizzazione delle informazioni relative a quel Concerto;
2. **Sottosistema Gestione Biglietti:** include tutte le operazioni che permettono l'aggiunta o l'eliminazione di biglietti per un determinato Concerto;
3. **Sottosistema Gestione Artista:** include tutte le operazioni che permettono l'aggiunta o l'eliminazione di un Artista;
4. **Sottosistema Gestione Luogo:** include tutte le operazioni che permettono l'aggiunta o l'eliminazione di un Luogo;
5. **Sottosistema Gestione Utente:** include tutte le operazioni che permettono la Registrazione di Utente, l'eliminazione di un Utente, promuovere un Utente ad Amministratore, di visualizzare le informazioni relative all'Utente, cambiare le proprie credenziali di Accesso, effettuare il login e il logout;
6. **Sottosistema Gestione Acquista:** include tutte le operazioni che permettono l'aggiunta di biglietti per un concerto nel carrello, svuotare il carrello, visualizzare i biglietti gia presenti nel carrello e procedere all'acquisto;

2.1 Package Generale

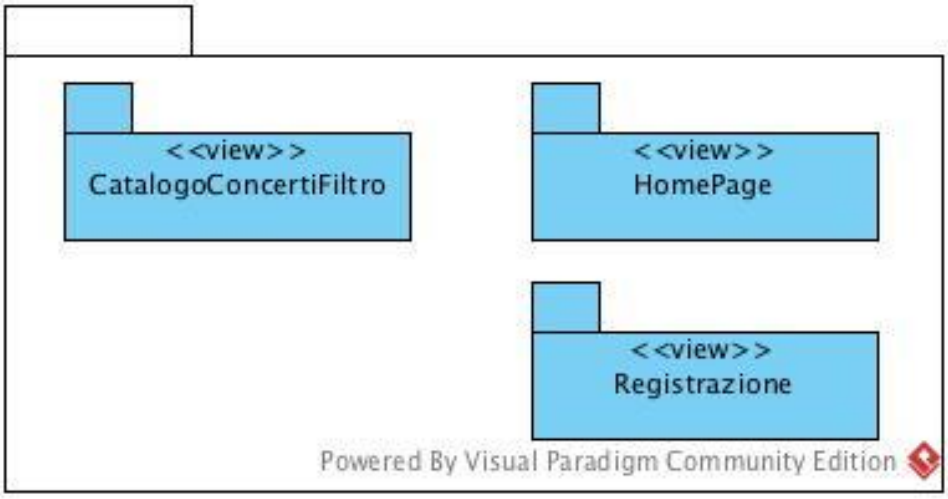


2.2 Package Presentazione Generale



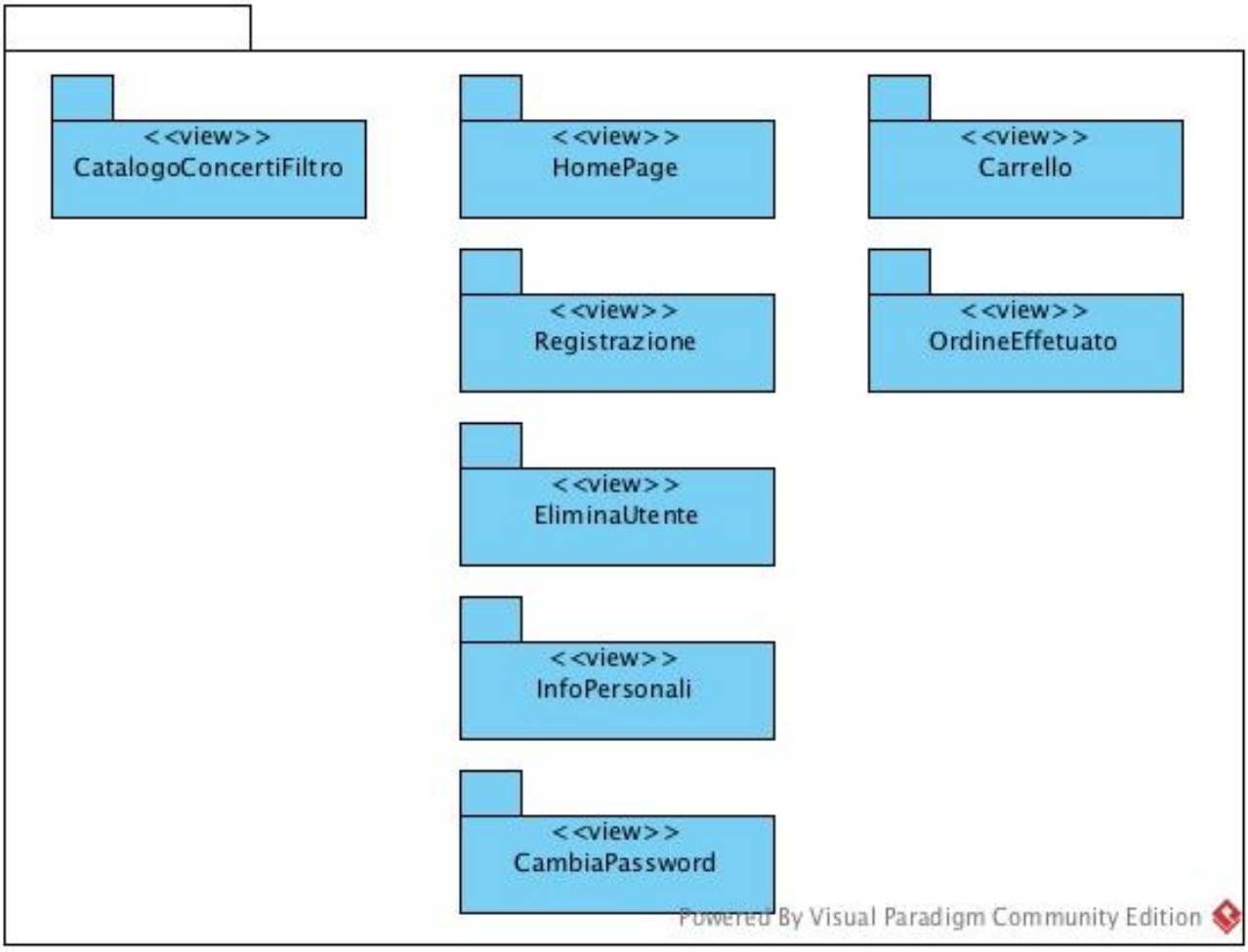
Il Diagramma descrive le interfacce delle varie sezioni del sistema:
GestioneArtista / GestioneLuogo / GestioneConcerto / GestioneBiglietti / GestioneUtente / GestioneAcquista

2.2.1 Package Presentazione Utente Non Registrato



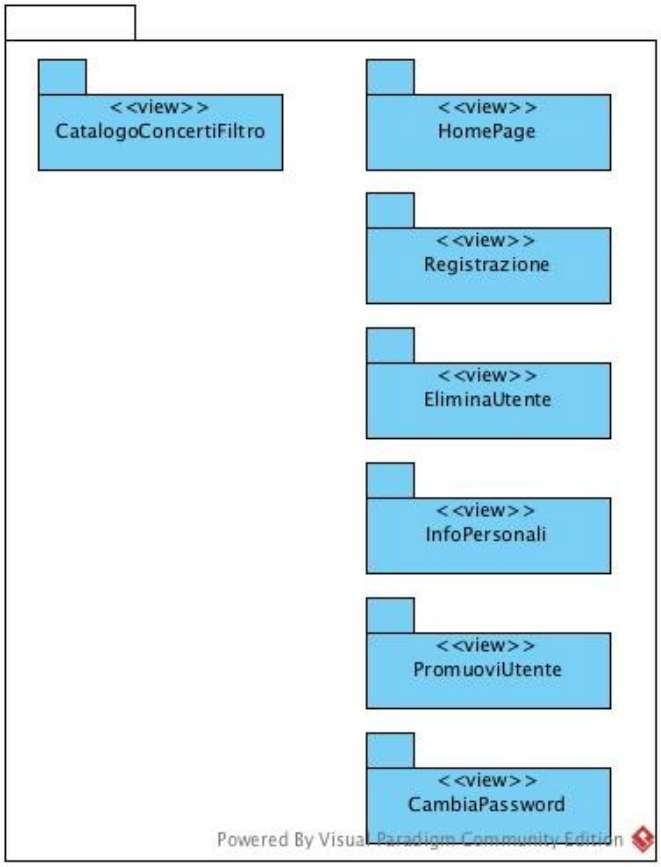
Il Diagramma descrive le interfacce per l'Utente Non Registrato delle varie sezioni del sistema: GestioneArtista / GestioneUtente

2.2.2 Package Presentazione Utente Registrato



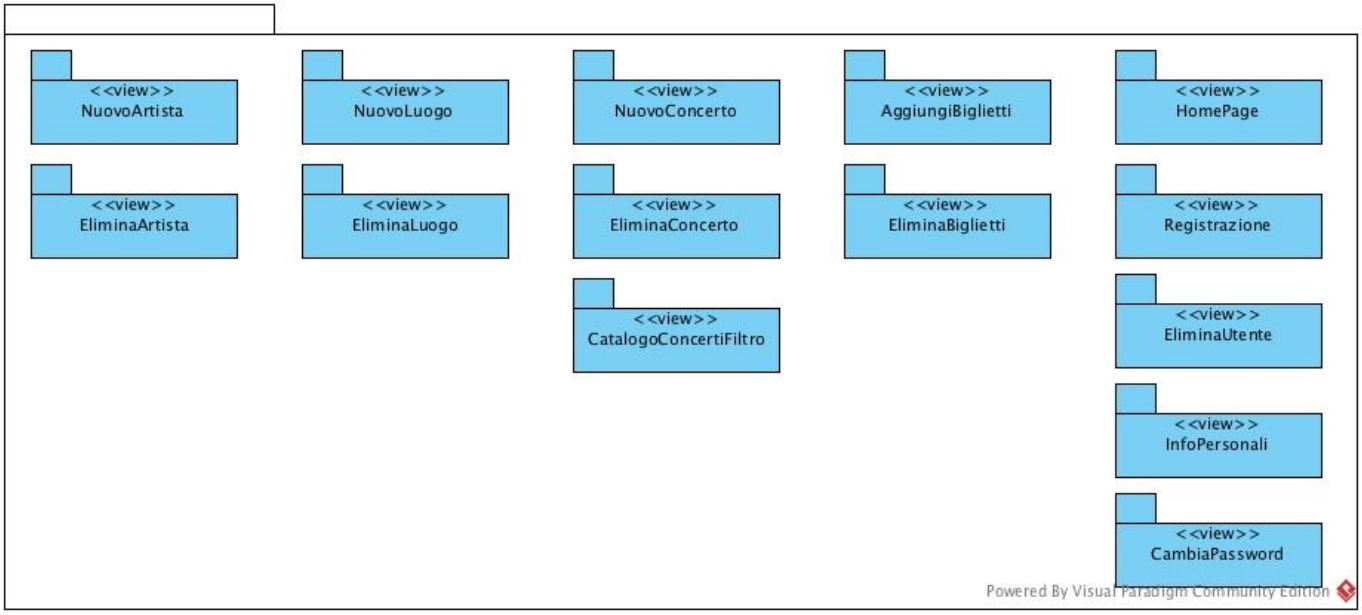
Il Diagramma descrive le interfacce per l'Utente Registrato delle varie sezioni del sistema: GestioneConcerto / GestioneUtente / GestioneAcquista

2.2.3 Package Presentazione Amministratore Utenti



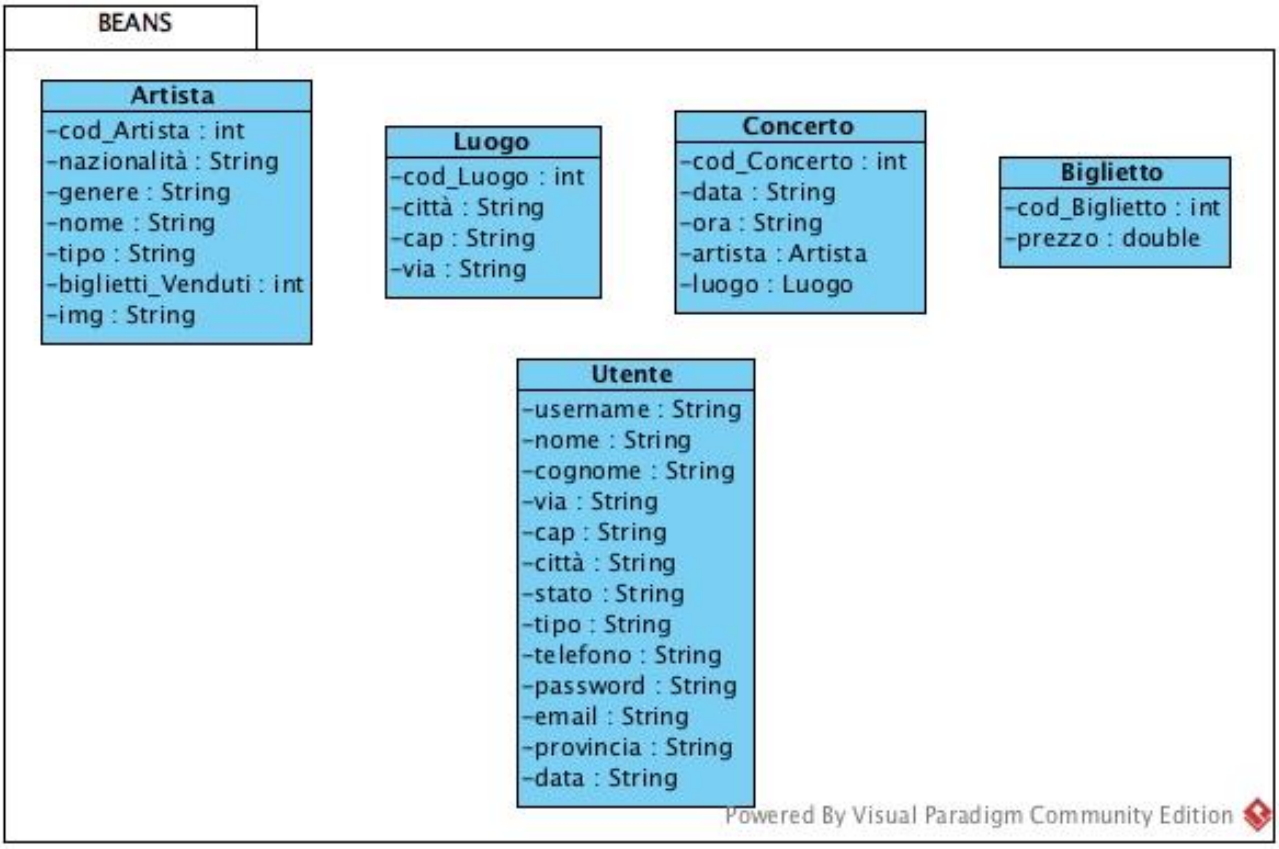
Il Diagramma descrive le interfacce per l'Amministratore Utenti delle varie sezioni del sistema: GestioneConcerto / GestioneUtente

2.2.4 Package Presentazione Amministratore Concerti



Il Diagramma descrive le interfacce per l'Amministratore Concerti delle varie sezioni del sistema: Gestione Artista / GestioneLuogo / GestioneConcerto / Gestione Biglietti / GestioneUtente

2.3 Package Entity



Il Package "Beans" contiene al suo interno delle Entity del nostro Sistema.

Ogni Entity è definita come segue:

Artista

Nome	Artista
Descrizione	Questa classe rappresenta l'Artista
Signature dei Metodi	+getCod_Artista(); +setCod_Artista(int cod_Artista); +getNazionalità(); +setNazionalità(String nazionalità); +getGenere(); +setGenere(String genere); +getNome(); +setNome(String nome); +getTipo(); +setTipo(String tipo); +getBiglietti_Venduti(); +setBiglietti_Venduti(int biglietti_Venduti); +getImg(); +setImg(String img);

Luogo

Nome	Luogo
Descrizione	Questa classe rappresenta il Luogo dove si tiene un Concerto
Signature dei Metodi	+getCod_Luogo(); +setCod_Luogo(int cod_Luogo); +getCittà(); +setCittà(String città); +getCAP(); +setCAP(int cap); +getVia(); +setVia(String via);

Concerto

Nome	Concerto
Descrizione	Questa classe rappresenta il Concerto
Signature dei Metodi	+getCod_Concerto(); +setCod_Concerto(int cod_Concerto); +getData(); +setData(String data); +getOra(); +setOra(String ora); +getArtista(); +setArtista(Artista artista); +getLuogo(); +setLuogo(Luogo luogo);

Biglietto

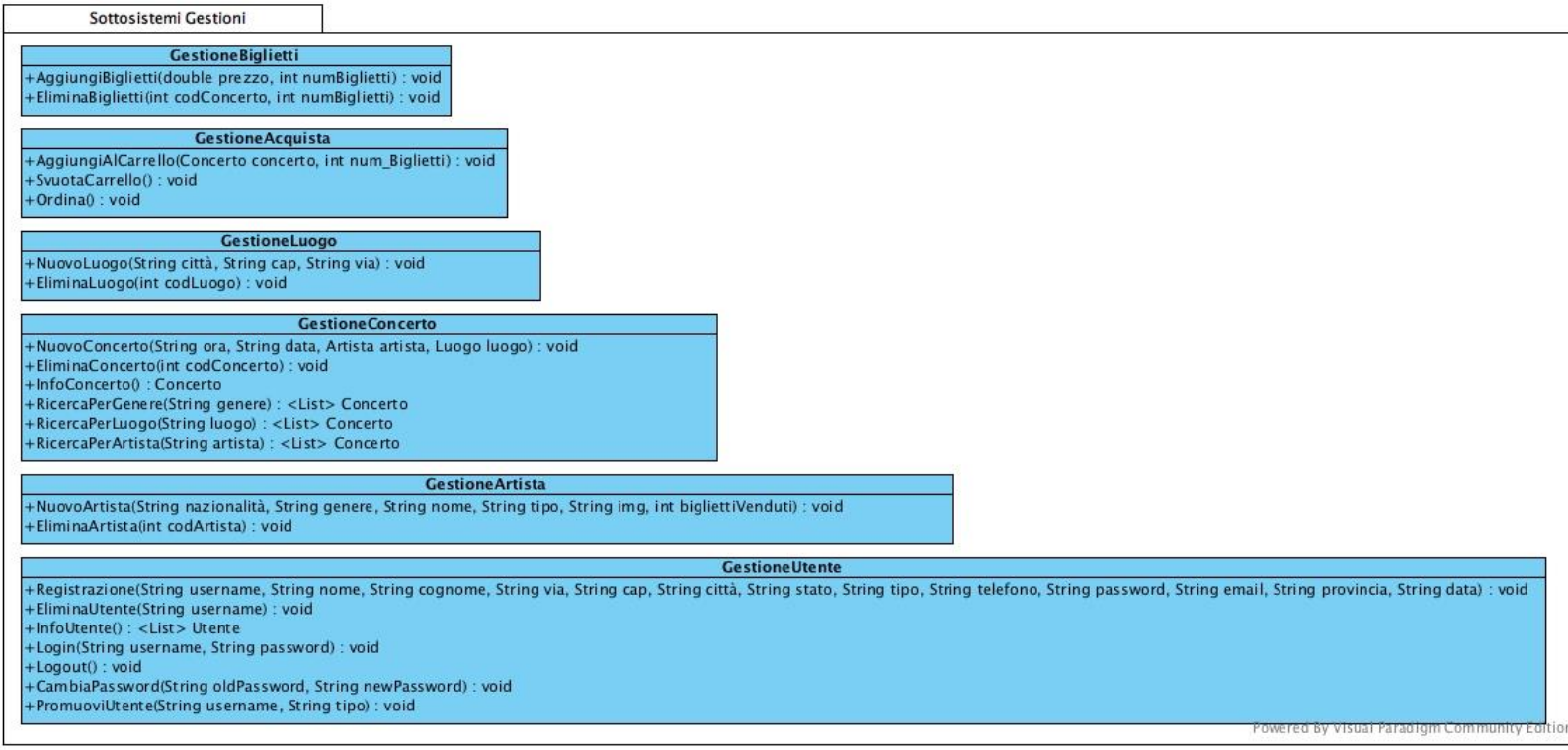
Nome	Biglietto
Descrizione	Questa classe rappresenta il biglietto di un determinato Concerto
Signature dei Metodi	+getCod_Biglietto(); +setCod_Biglietto(int cod_Biglietto); +getPrezzo(); +setPrezzo(double Prezzo);

Utente

Nome	Utente
Descrizione	Questa classe rappresenta l'Utente del sistema
Signature dei Metodi	+getUsername(); +setUsername(String username); +getNome(); +setNome(String Nome); +getCognome(); +setCognome(String Cognome); +getVia(); +setVia(String via); +getCAP(); +setCAP(String cap);

	+getCittà(); +setCittà(String città); +getStato(); +setStato(String stato); +getTipo(); +setTipo(String tipo); +getTelefono(); +set Telefono(String telefono); +getPassword(); +setPassword(String password); +getEmail(); +setEmail(String email); +getProvincia(); +setProvincia(String Provincia); +getData(); +setData(String Data);

2.4 Package Gestione



Il Package “Gestioni” fa parte dell’Applicazione e si focalizza sulle classi che si occuperanno di implementare la logica del sistema (Oggetti Control presenti nel RAD)

GestioneBiglietti

Nome	GestioneBiglietti
Descrizione	Questa classe rappresenta la gestione dei biglietti all'interno del sistema
Signature dei Metodi	+AggiungiBiglietti(double prezzo, int numBiglietti) : void +EliminaBiglietti(int codConcerto, int numBiglietti) : void

GestioneAcquista

Nome	GestioneAcquista
Descrizione	Questa classe rappresenta la gestione relativa al carrello nel sistema
Signature dei Metodi	+AggiungiAlCarrello(Concerto concerto, int num_Biglietti) : void +SvuotaCarrello() : void +Ordina() : void

GestioneLuogo

Nome	GestioneLuogo
Descrizione	Questa classe rappresenta la gestione del luogo relativo ad un concerto
Signature dei Metodi	+NuovoLuogo(String città, String cap, String via) : void +EliminaLuogo(int codLuogo) : void

Biglietto

Nome	Biglietto
Descrizione	Questa classe rappresenta il biglietto di un determinato Concerto
Signature dei Metodi	+getCod_Biglietto(); +setCod_Biglietto(int cod_Biglietto); +getPrezzo(); +setPrezzo(double Prezzo);

GestioneConcerto

Nome	GestioneConcerto
Descrizione	Questa classe rappresenta la gestione dei concerti all'interno del sistema
Signature dei Metodi	+NuovoConcerto(String ora, String data, Artista artista, Luogo luogo) : void +EliminaConcerto(int codConcerto) : void +InfoConcerto() : Concerto +RicercaPerGenere(String genere) : <List> Concerto +RicercaPerLuogo(String luogo) : <List> Concerto +RicercaPerArtista(String artista) : <List> Concerto

GestioneArtista

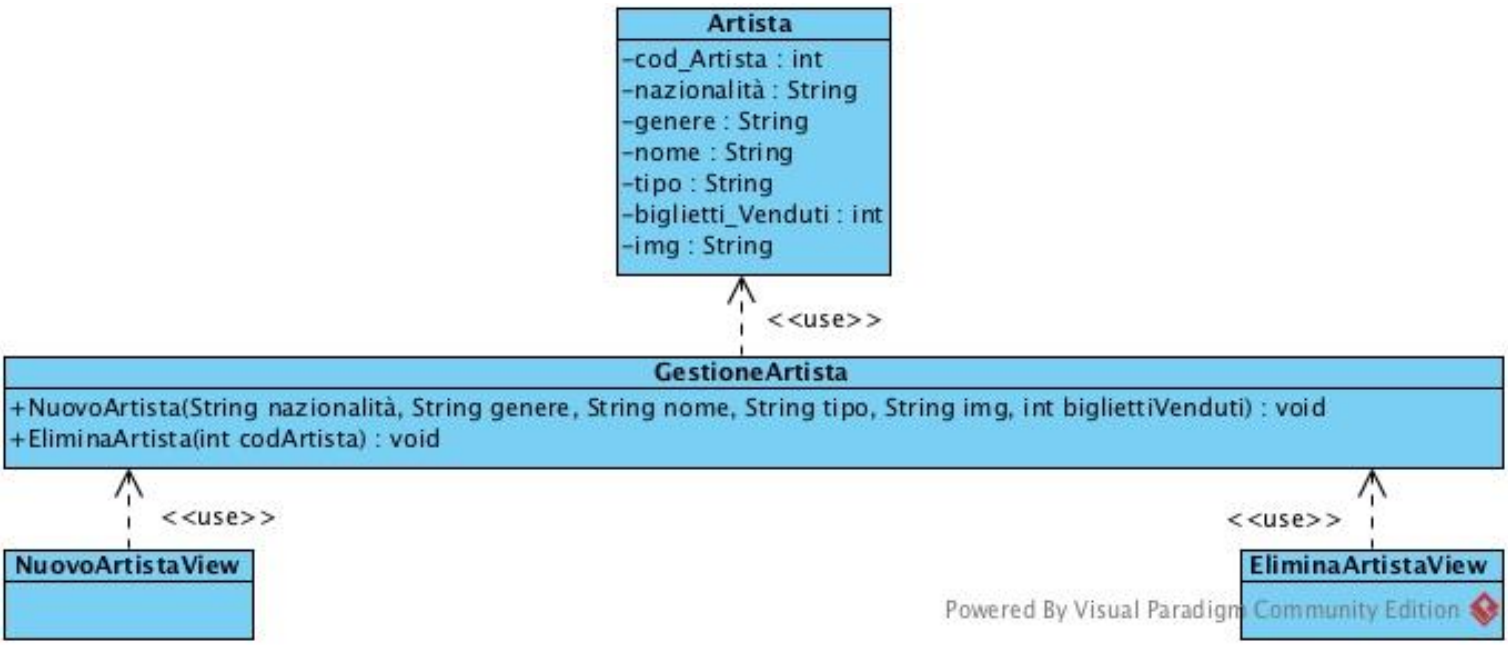
Nome	GestioneArtista
Descrizione	Questa classe rappresenta la gestione dell'Artista all'interno del Sistema
Signature dei Metodi	+NuovoArtista(String nazionalità, String genere, String nome, String tipo, String img, int biglietti Venduti) : void +EliminaArtista(int codArtista) : void

GestioneUtente

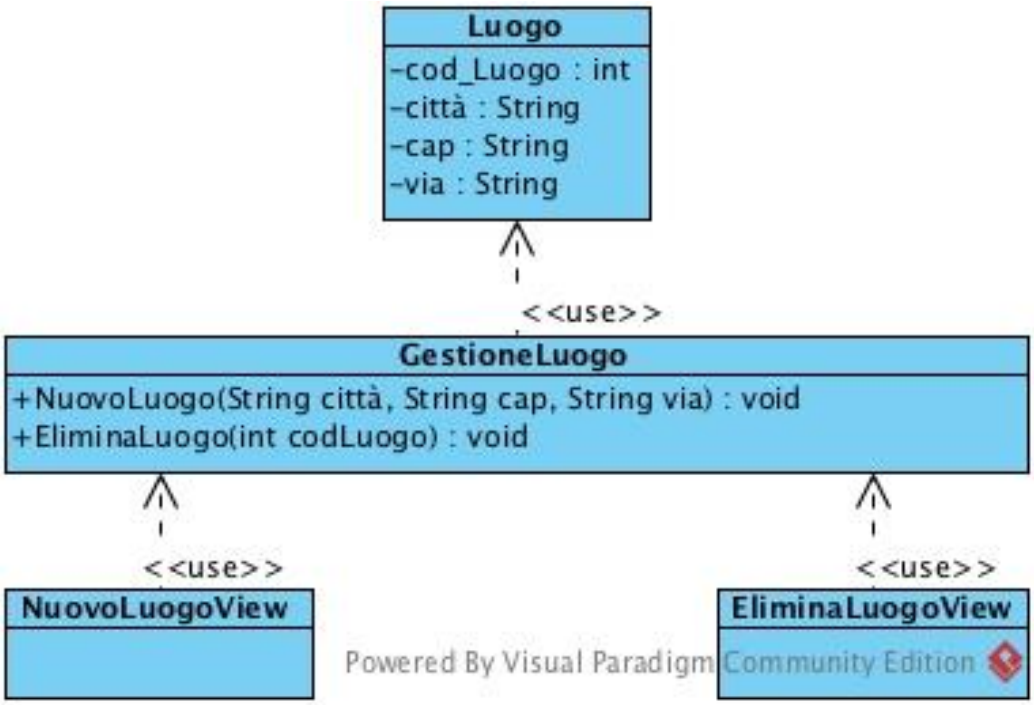
Nome	GestioneUtente
Descrizione	Questa classe rappresenta la gestione dell'utente all'interno del sistema
Signature dei Metodi	+Registrazione(String username, String nome, String cognome, String via, String cap, String città, String stato, String tipo, String telefono, String password, String email, String provincia, String data) : void +EliminaUtente(String username) : void +InfoUtente() : <List> Utente +Login(String username, String password) : void +Logout() : void +CambiaPassword(String oldPassword, String newPassword) : void +PromuoviUtente(String username, String tipo) : void

3. Class Interface

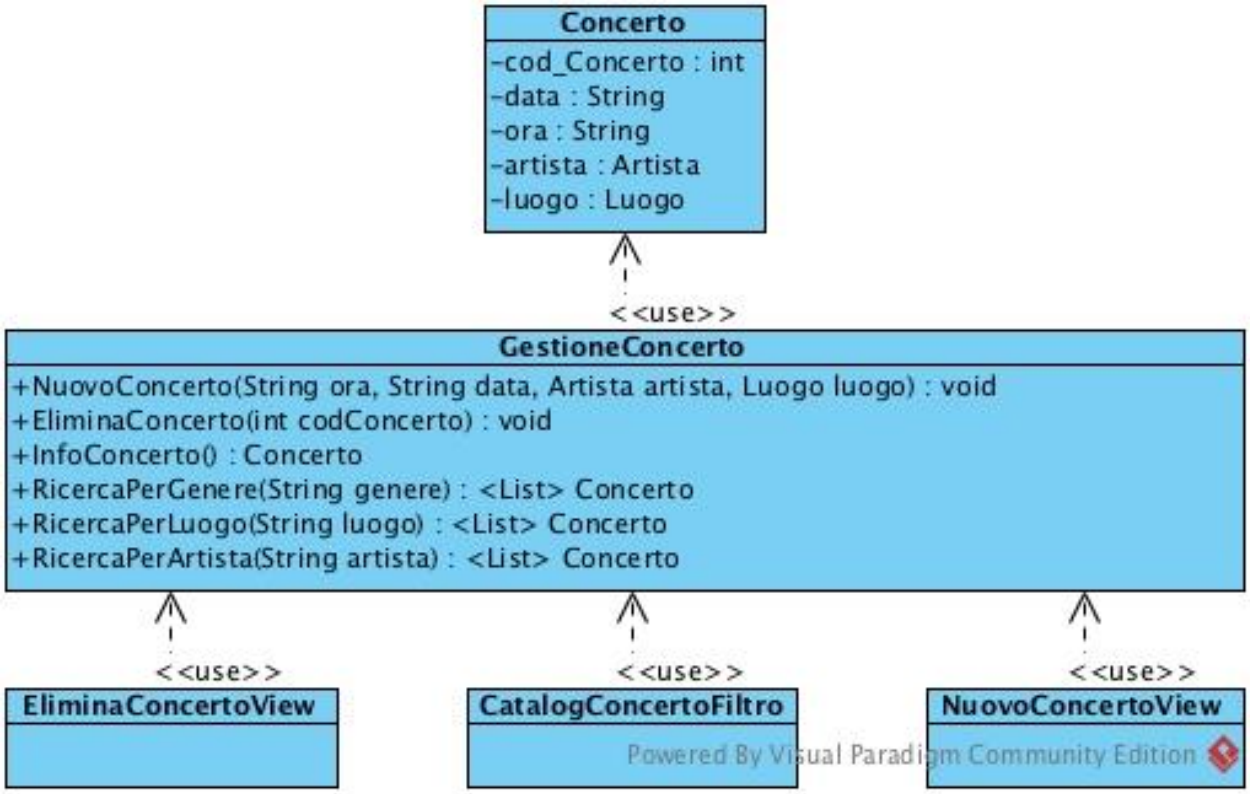
3.1 Gestione Artista



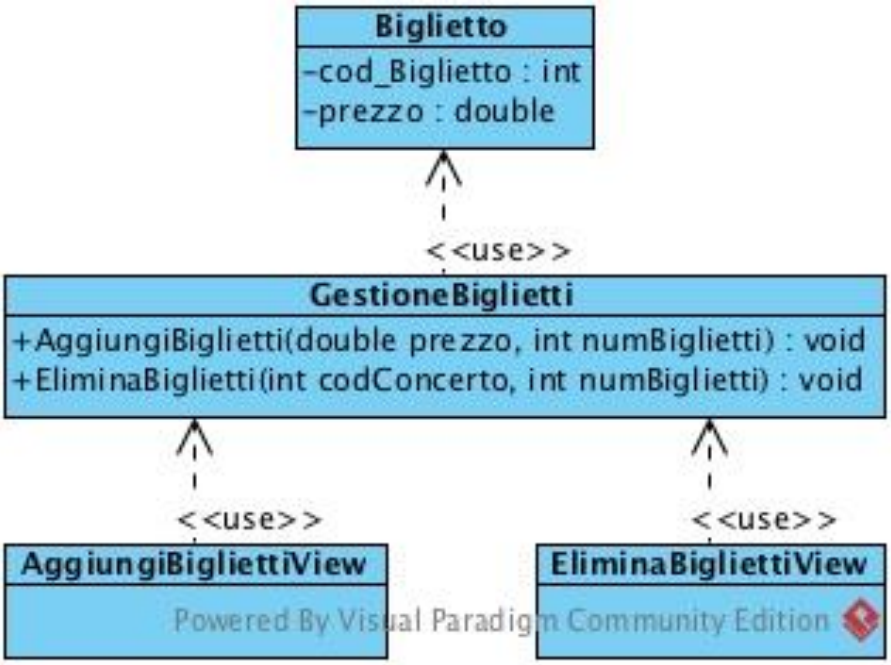
3.2 Gestione Luogo



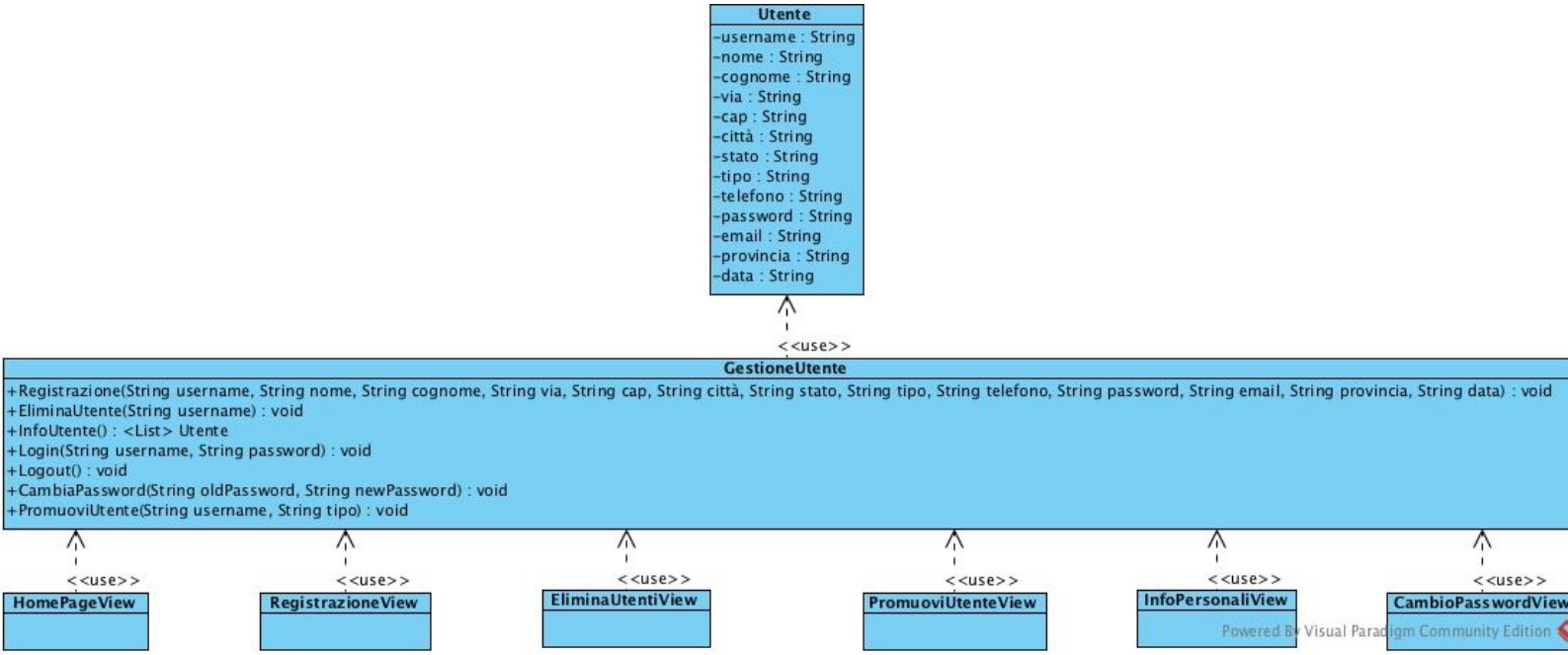
3.3 Gestione Concerto



3.4 Gestione Biglietti



3.5 Gestione Utente



3.6 Gestione Acquista