**1. Introduzione**

**1.1. Descrizione del Problema**

Grazie al nostro progetto, si creerà un collegamento che permetterà di ricongiungere gli appassionati alla fotografia.

Gli obiettivi principali del sistema sono quelli di offrire una piattaforma dove gli artisti possono pubblicare le loro opere d’arte e gli utenti possono appassionarsi alla fotografia grazie a un social semplice e funzionale. Per rendere questo possibile verrà creata una web Application con un’interfaccia chiara ed intuitiva.

Le funzionalità che il sistema offre sia all’Operatore che all’Amministratore sono:

* Gestione delle Segnalazioni:
* Bloccare un utente o un post,
* Ignorare una segnalazione.
* Gestione dei Messaggi Assistenza:
* Rispondere a un messaggio d’assistenza.
* Visualizzare le informazioni di un utente o un post.
* Ignorare un messaggio d’assistenza.

Inoltre il sistema offre le seguenti funzionalità all’Amministratore:

* Gestione Operatori:
* Visualizzare le informazioni di un operatore
* Rimuovere un operatore.
* Registrare un nuovo Operatore.

Le funzionalità che il sistema offre all’Artista sono:

* Gestione Abbonamento:
  + Sottoscrivere un abbonamento per passare all’ account Artista
* Gestione Carta di Credito:
  + Aggiungere e rimuovere una carta di credito.
* Gestione Statistiche:
* Visualizzare le statistiche generali dell’account.
* Visualizzare e analizzare le statistiche ad ogni foto messa in vendita.
* Gestione Foto:
* Vendere una foto e pubblicare un post per renderla visibile.
* Eliminare il Post.

Le funzionalità che il sistema sia all’Utente base che all’Artista sono:

* Gestione Profilo:
  + Modificare i dati del proprio account.
  + Effettuare il download delle immagini acquistate.
  + Visualizzare i propri acquisti.
* Gestione Post:
  + Aggiungere un Post.
  + Eliminare un Post.
* Gestione Interazioni:
  + Inserire l’interazione “Like” a un post.
  + Inserire un commento a un post.
  + Seguire un utente.
  + Inviare un messaggio a un utente.
  + Ricercare un utente.
  1. **Design Goals**

La Web App ClipShot punterà ad essere il più lineare ed intuitiva possibile. Per Far ciò si avvarrà di una struttura grafica chiara e completa, con bottoni, finestre di dialogo e icone. Cercherà inoltre di isolare le informazioni necessarie per indirizzare più facilmente l’utente verso la funzionalità da lui richiesta. Valore aggiunto all’applicazione sarà la sua semplicità, che permetterà anche agli utenti con scarsa conoscenza del sistema di portare a termine le loro operazione, evitando di inserire dati sbagliati o commettete errori durante l’utilizzo dell’applicazione.

Il sistema proposto rispetterà i seguenti criteri di design:

* + 1. **DG\_0 Criteri di Performance**
* DG\_0.1 Tempi di risposta: Il sistema deve garantire tempi di risposta brevi per ogni funzionalità. Mediamente una richiesta dovrà essere soddisfatta in un tempo non superiore ai 5 secondi. Ovviamente quest’ultimo può oscillare in base alla velocità di connessione.
* DG\_0.2 Throughput: Il sistema sarà capace di gestire una media di 200 utenti, consentendo loro di effettuare tutte le operazioni senza subire alcun rallentamento. L’applicazione sarà anche in grado di gestire un maggiore afflusso di utenti qualora sia necessario.
* DG\_0.3 Dati persistenti: Il sistema, per memorizzare tutti i dati relativi agli utenti e alle foto utilizzerà un database relazionale. La scelta di quest’ultimo farà in modo che le informazioni saranno il più velocemente possibile.
  + 1. **DG\_1 Criteri di Affidabilità**
* DG\_1.1 Robustezza: Il sistema mostrerà un messaggio che avvertirà l’utente nel caso in cui i dati siano mancanti o errati. Questo consentirà di non immettere dati sbagliati all’interno del database.
* DG\_1.2 Affidabilità: Il sistema garantirà il corretto svolgimento di tutte le funzionalità e cercherà di produrre l’output atteso evitando errori indesiderati.
* DG\_1.3 Disponibilità: Il sistema dovrà essere sempre fruibile agli utenti. Non sono previsti periodi di chiusura del sistema se non quelli per motivi di manutenzioni.
* DG\_1.4 Sicurezza: Il sistema, tramite username e password, riuscirà ad individuare il tipo di utente e gli permetterà di effettuare solo le operazioni appartenenti alla sua categoria.
  + 1. **DG\_2 Criteri di Costo**
* DG\_2.1 Costo: Prima dell’inizio dello sviluppo della piattaforma verranno valutati i costi di sviluppo del sistema, costi di manutenzione e costi di amministrazione.
  + 1. **DG\_3 Criteri di Mantenimento**
* DG\_3.1 Modificabilità: il sistema permetterà di apportare modifiche alle funzionalità già implementate senza la necessità di modificare i sottosistemi.
  + 1. **DG\_4 Criteri di Utente**
* DG\_4.1 Utilità: Il sistema punterà a soddisfare tutte le esigenze degli utenti sviluppando tutti i requisiti funzionali in fase di analisi.
* DG\_4.2 Usabilità: Il sistema dovrà essere chiaro ed intuitivo e anche l’utente con meno esperienza dovrà essere in grado di effettuare le operazioni desiderate. La piattaforma consentirà “l’autoapprendimento” cioè l’utente effettuerà le operazioni in maniera sempre più rapida con il maggiore utilizzo del sistema.
  1. **Definizioni, Acronimi e Abbrevazioni**

|  |  |
| --- | --- |
| Acronimo | Descrizione |
| RAD | Requirement Analysis Document |
| JDBC | Java DataBase Connectivity |
| DBMS | Database Management System |
| HTTP | HyperText Transfer Protocol |
| GUI | Graphic User Interface |
| DB | DataBase |

* 1. **Riferimenti**
* RAD ClipShot v1.0
* Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition
* https://it.wikipedia.org/
  1. **Panoramica**

Il documento si compone di 3 parti fondamentali:

1. Introduzione: In questa sezione verranno illustrati gli obiettivi del sistema proposto e saranno descritte, in modo non approfondito, le varie funzionalità messe a disposizione per gli attori. Verranno poi mostrati gli obiettivi del design, in particolar modo i criteri che il sistema dovrà rispettare.
2. Architettura del sistema proposto: In questa sezione verrà elaborata la macrocomposizione in sottosistemi, la gestione dei dati persistenti, il mapping hardware/software del sistema, il controllo degli accessi , sicurezza e le condizioni limite.
3. Servizi dei sottosistemi e Glossario: In questa sezione troveremo la descrizione per ogni servizio fornito del sottosistema e un glossario contenente una raccolta dei termini contenuti nel sistema proposto.

# **2. Architettura del sistema**

**2.1.** **Panoramica**

Il sistema prevederà l’utilizzo di un architettura ibrida tra Client- Server e Model-View-Controller.

Il Lato Client si occuperà delle pagine di presentazione del sistema(View)

Il Lato Server si occuperà della logica applicativa e di controllo (rispettivamente Model e Controller).

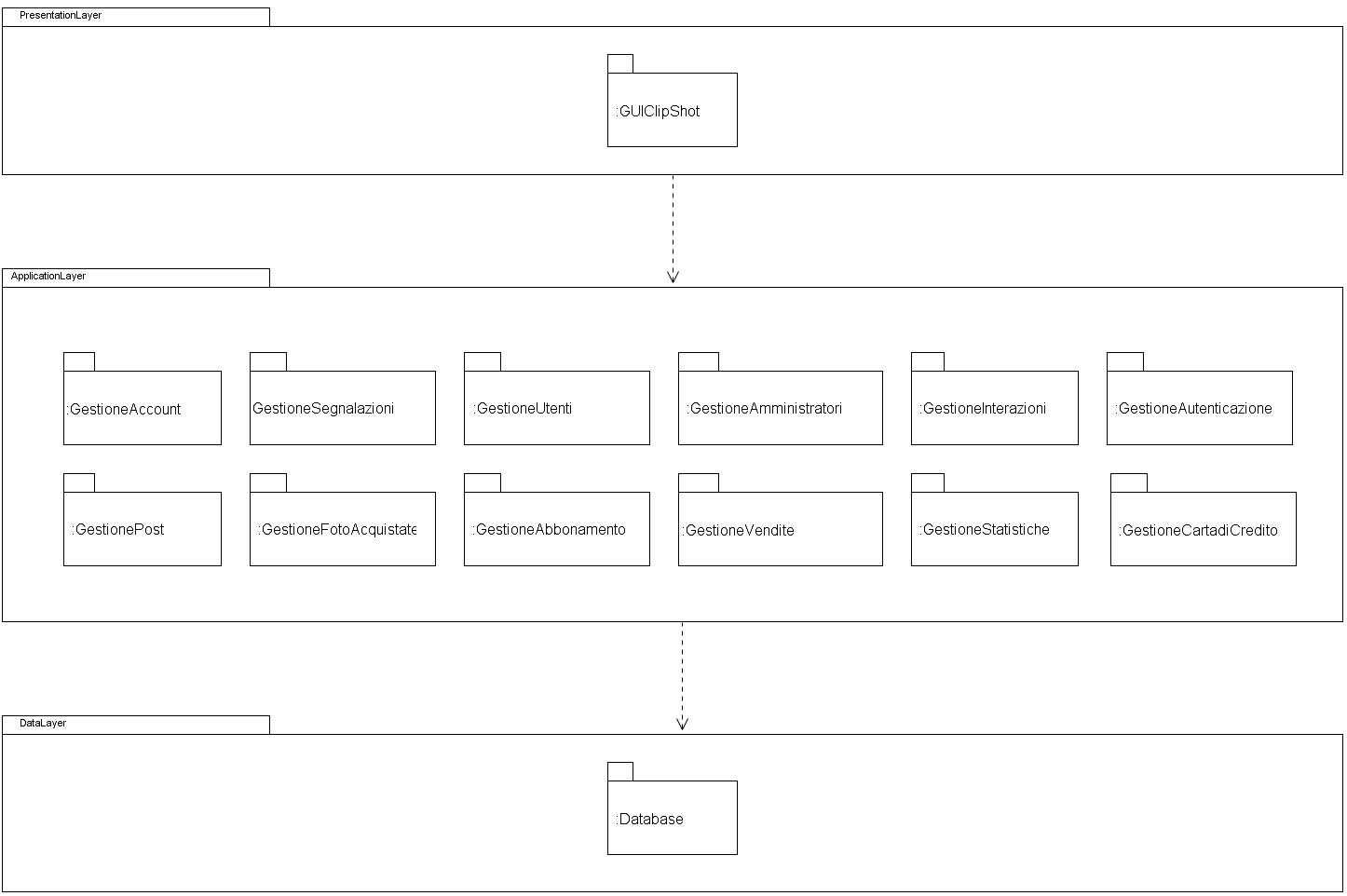
## **2.2 Decomposizione in sottosistemi**

### **2.2.1 Macrodecomposizione in sistemi**

La macro decomposizione divide il sistema nei seguenti sottosistemi:

1. Gestione Account.
2. Gestione Segnalazioni.
3. Gestione Utenti.
4. Gestione Amministratori.
5. Gestione Interazioni.
6. Gestione Autenticazione.
7. Gestione Post.
8. Gestione Foto Acquistate.
9. Gestione Abbonamento.
10. Gestione Carta di Credito.
11. Gestione Vendite.
12. Gestione Statistiche.

Di seguito è mostrato il diagramma in cui vengono evidenziate le relazioni di dipendenza tra i package di ogni sottosistema e tra i diversi sottosistemi.

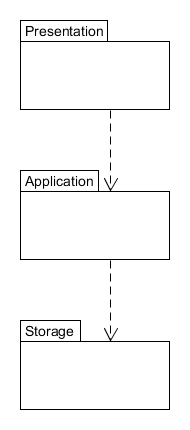


Gli utenti che useranno il sistema lo faranno dal proprio computer comunicando gli input all’interfaccia del Server Web i quali verranno gestiti dal Database in cui sono contenute tutte le informazioni dell’intero sistema. Il Database sarà gestito da un DBMS che si occupa di inserire, cercare e aggiornare i dati presenti al suo interno, elaborando la richiesta degli utenti da parte del Server. Il DBMS si occuperà anche di gestire gli accessi concorrenti al Database.

### **2.2.2 Microdecomposizione in sottosistemi**

Per semplificare la progettazione e lo sviluppo dell’applicazione, i sottosistemi saranno decomposti secondo lo schema previsto dall’architettura software Three-Layer, ovvero:

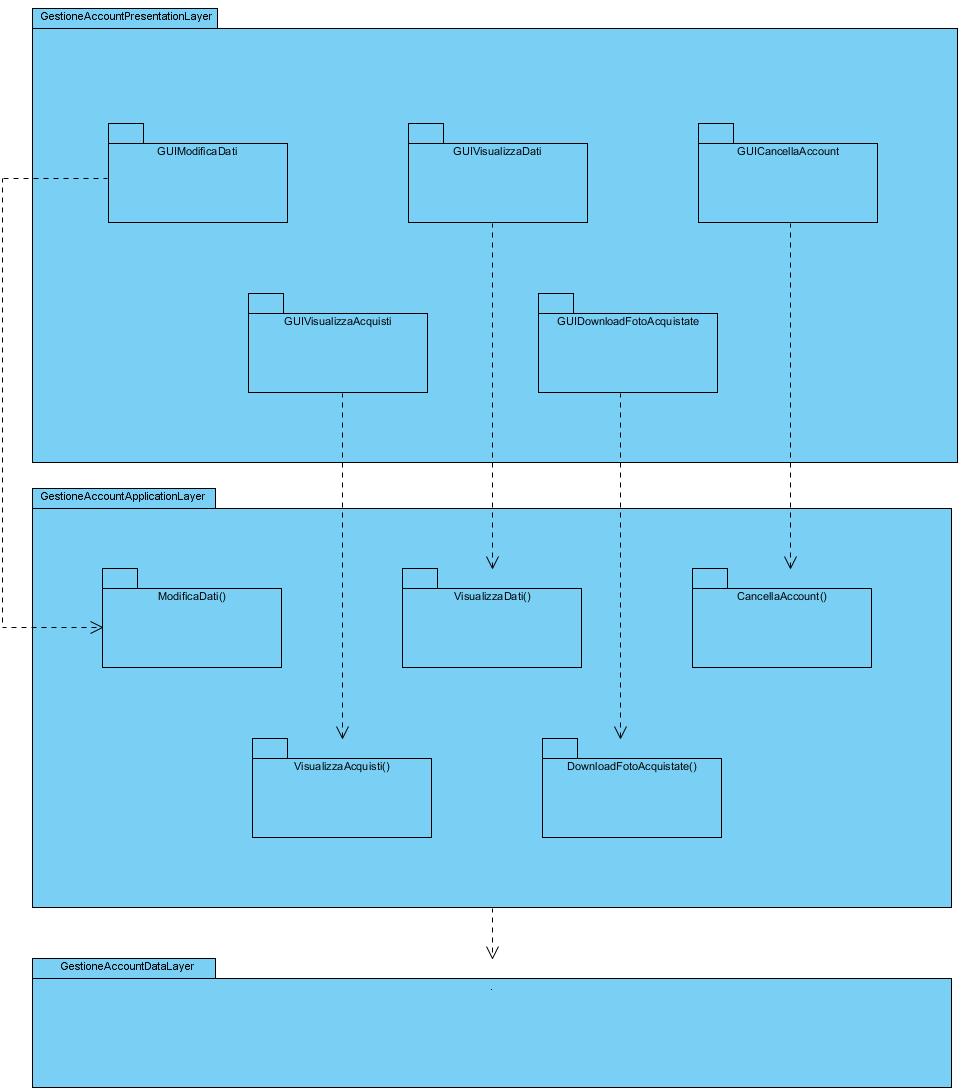
1. Presentation: si occupa di gestire l’interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l’utente.
2. Application: si occupa, dati gli input dagli utenti, di elaborarli e fornire dei risultati che verranno poi mostrati graficamente.
3. Storage: è la parte del sottosistema che interagisce col Database.



**2.2.3. Gestione Account**

Questo sottosistema si occupa della gestione dell’account, in particolare fornisce le funzioni di:

* Cancellazione account
* Visualizza dati account
* Modifica dati account
* Visualizza foto acquistate
* Download foto acquistate



**GestioneAccountPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUICancellaAccount: comprende le interfacce che consentono all’Utente Registrato di cancellarsi dal sistema.
* GUIVisualizzaDati: comprende le interfacce che consentono all’Utente Registrato di visualizzare le informazioni relative al proprio account.
* GUIModificaDati: comprende le interfacce che consentono all’Utente Registrato di modificare le informazioni relative al proprio account.
* GUIVisualizzaAcquisti: comprende le interfacce che consentono all’Utente Registrato di visualizzare le foto acquistate.
* GUIDownloadFotoAcquistate: comprende le interfacce che consentono all’utente Registrato di scaricare le foto acquistate.

**GestioneAccountApplicationLayer**

* CancellaAccount(): operazioni per cancellare il proprio account dal sistema.
* VisualizzaDati(): operazioni per la visualizzazione delle informazioni relative al proprio account.
* ModificaDati(): operazioni per la modifica delle informazioni al proprio account.
* VisualizzaAcquisti(): operazioni per la visualizzazione delle foto acquistate.
* DownloadFotoAcquistate(): operazioni per il download delle foto acquistate.

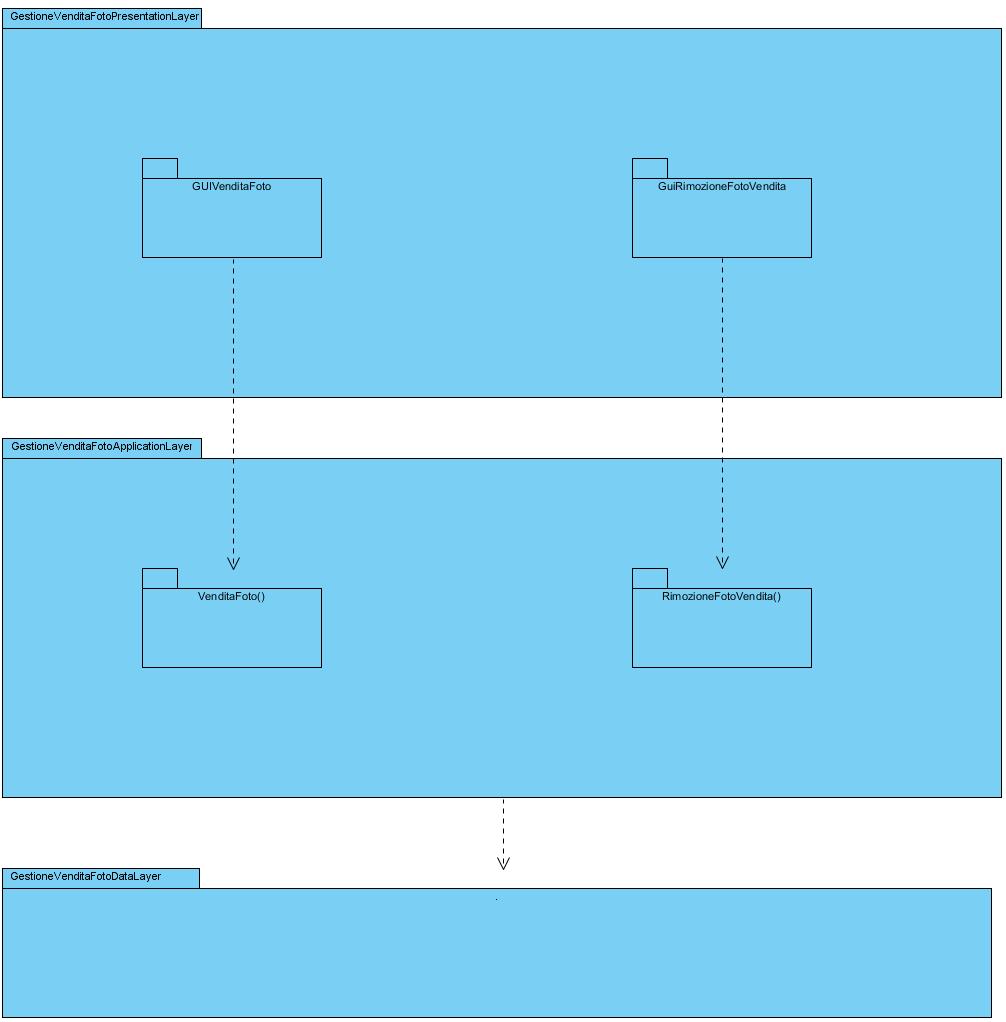
**GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi all’account.

**2.2.4 Gestione Vendita Foto**

Questo sottosistema si occupa della gestione della vendita delle foto, in particolare fornisce le funzioni di:

* Vendere una foto
* Rimuovere una foto



**GestioneVenditaFotoPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione della vendita delle foto. Comprende:

* GUIVenditaFoto: comprende le interfacce che consentono all’Artista di vendere una foto.
* GUIRimuoviFotoVendita: comprende le interfacce che consentono all’Artista di rimuovere una foto in vendita.

**GestioneVenditaFotoApplicationLayer**

* VendiFoto(): operazioni per vendere una foto.
* RimuoviFotoVendita(): operazioni per rimuovere una foto in vendita.

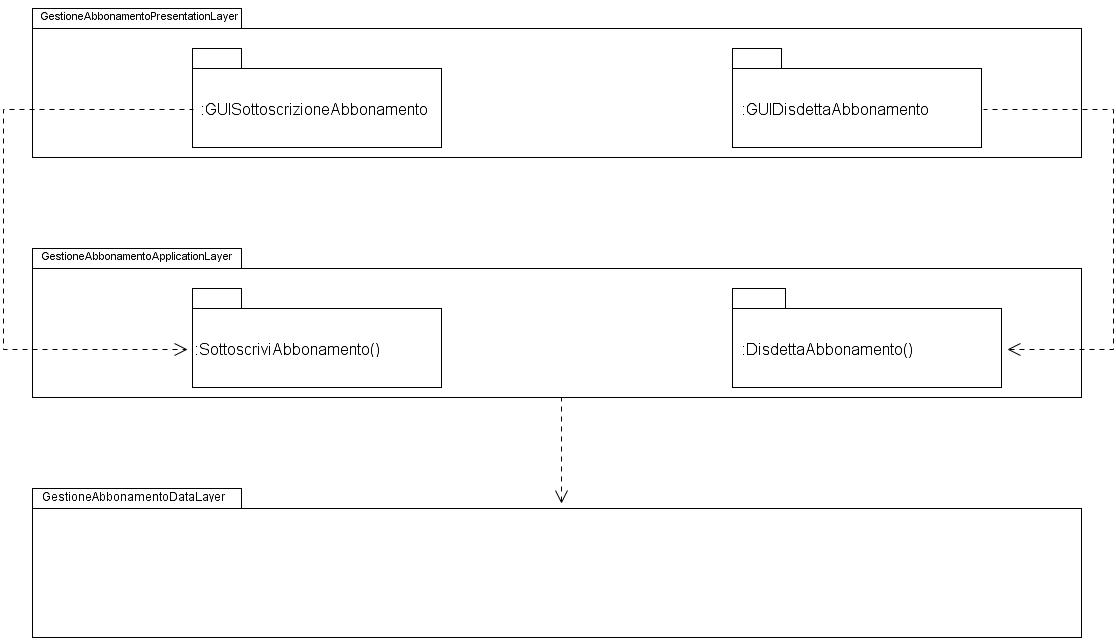
**GestioneVenditaFotoDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle foto in vendita.

## **2.2.5 Gestione Abbonamento**

Questo sottosistema si occupa della gestione dell’Abbonamento, in particolare fornisce le funzioni di:

* Sottoscrizione Abbonamento
* Disdetta Abbonamento



#### **GestioneAbbonamentoPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’Abbonamento. Comprende:

* GUISottoscrizioneAbbonamento: comprende le interfacce che consentono all’Utente Base di effettuare la Sottoscrizione dell’Abbonamento.
* GUIDisdettaAbbonamento: comprende le interfacce che consentono all’Artista di disdire l’Abbonamento.

#### **GestioneAbbonamentoApplicationLayer**

* SottoscriviAbbonamento(): operazioni per la Sottoscrizione dell’Abbonamento.
* DisdettaAbbonamento(): operazioni per la disdetta dell’Abbonamento.

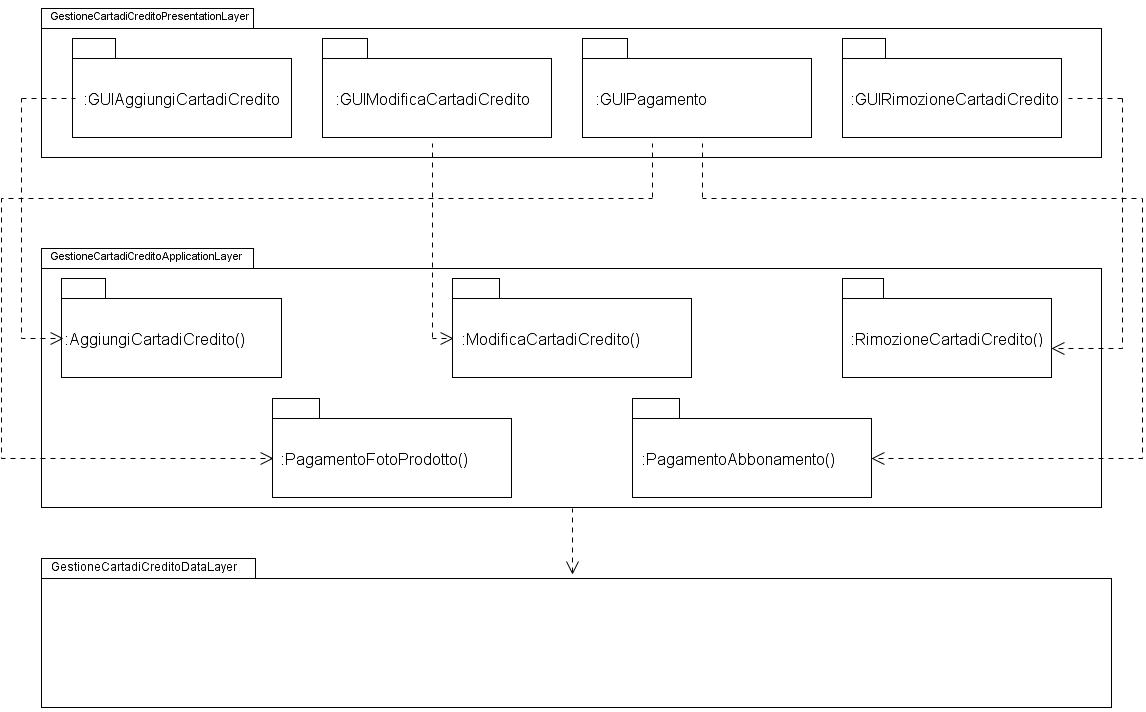
#### **GestioneAbbonamentoDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli Abbonamenti.

## **2.2.6 Gestione Carta di Credito**

Questo sottosistema si occupa della gestione della Carta di Credito, in particolare fornisce le funzioni di:

* Aggiunta Carta di Credito
* Modifica Carta di Credito
* Rimozione Carta di Credito
* Pagamento



#### **GestioneCartadiCreditoPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione della Carte di Credito. Comprende:

* GUIAggiuntaCartadiCredito: comprende le interfacce che consentono all’Utente Base di aggiungere la Carta di Credito.
* GUIModificaCartadiCredito: comprende le interfacce che consentono all’Utente Base e all’Artista di modificare la Carta di Credito.
* GUIRimozioneCartadiCredito: comprende le interfacce che consentono all’Utente Base e all’Artista di rimuovere la Carta di Credito.
* GUIPagamento: comprende le interfacce che consentono all’Utente Base e all’Artista di pagare rispettivamente le Foto Prodotto e l’Abbonamento.

#### **GestioneCartadiCreditoApplicationLayer**

* AggiungiCartadiCredito(): operazioni per l’aggiunta della Carta di Credito.
* ModificaCartadiCredito(): operazione per la modifica della Carta di Credito.
* RimuoviCartadiCredito(): operazioni per la rimozione della Carta di Credito.
* PagamentoFotoProdotto(): operazioni per il pagamento di una Foto Prodotto.
* PagamentoAbbonamento(): operazioni per il pagamento dell’Abbonamento.

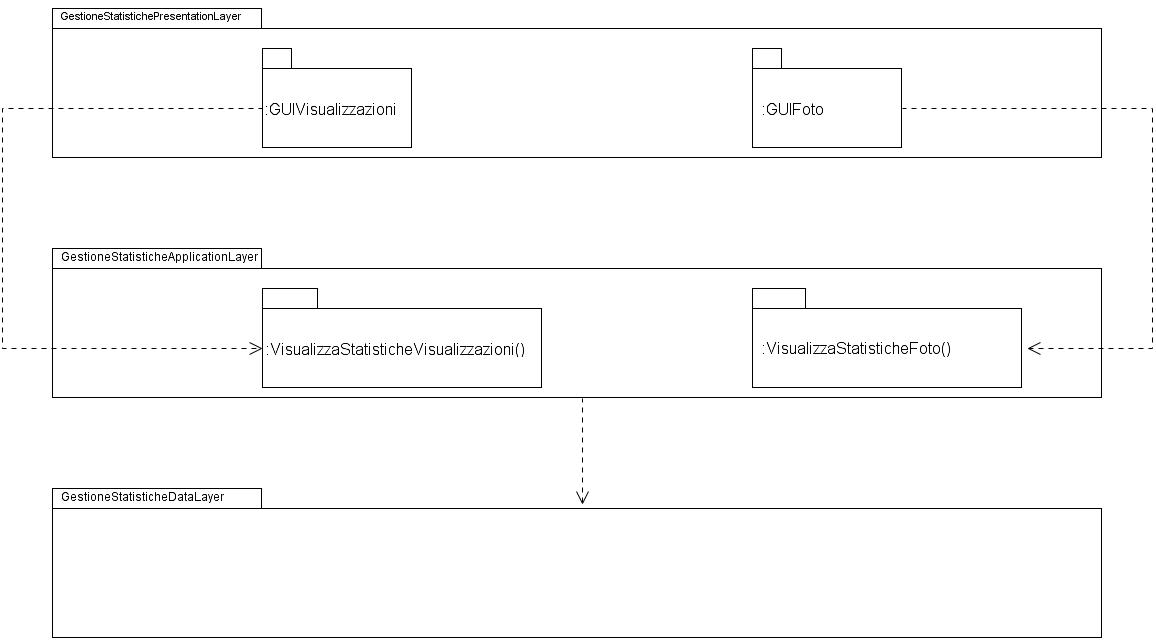
#### **GestioneCartadiCreditoDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle Carte di Credito.

### **2.2.7 Gestione Statistiche**

Questo sottosistema si occupa della gestione delle statistiche, in particolare fornisce le funzioni di:

* Visualizzazioni
* Foto



#### **GestioneStatistichePresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione delle statistiche. Comprende:

* GUIVisualizzazioni: comprende le interfacce che consentono all’Artista di visualizzare le statistiche relative alle visualizzazioni del proprio profilo.
* GUIFoto: comprende le interfacce che consentono all’Artista di visualizzare le statistiche relative alle foto.

#### **GestioneStatisticheApplicationLayer**

* VisualizzaStatisticheVisualizzazioni(): operazioni per visualizzare le statistiche relative alle visualizzazione del proprio profilo.
* VisualizzaStatisticheFoto: operazioni per visualizzare le statistiche relative alle foto.

#### **GestioneStatisticheDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli elementi analizzati dal sistema di Statistiche.

#### **2.2.8 Gestione Operatori:**

#### 

#### **GestioneOperatoriPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIListaOperatori: comprende le interfacce che consentono all’Amministratore di visualizzare la lista degli operatori.
* GUIInfoOperatore: comprende le interfacce che consentono all’Amministratore di visualizzare i dettagli di un operatore.
* GUICreaOperatore: comprende le interfacce che consentono all’Amministratore di registrare un nuovo operatore.
* GUIRimuoviOperatore: comprende le interfacce che consentono all’Amministratore di cancellare un operatore.

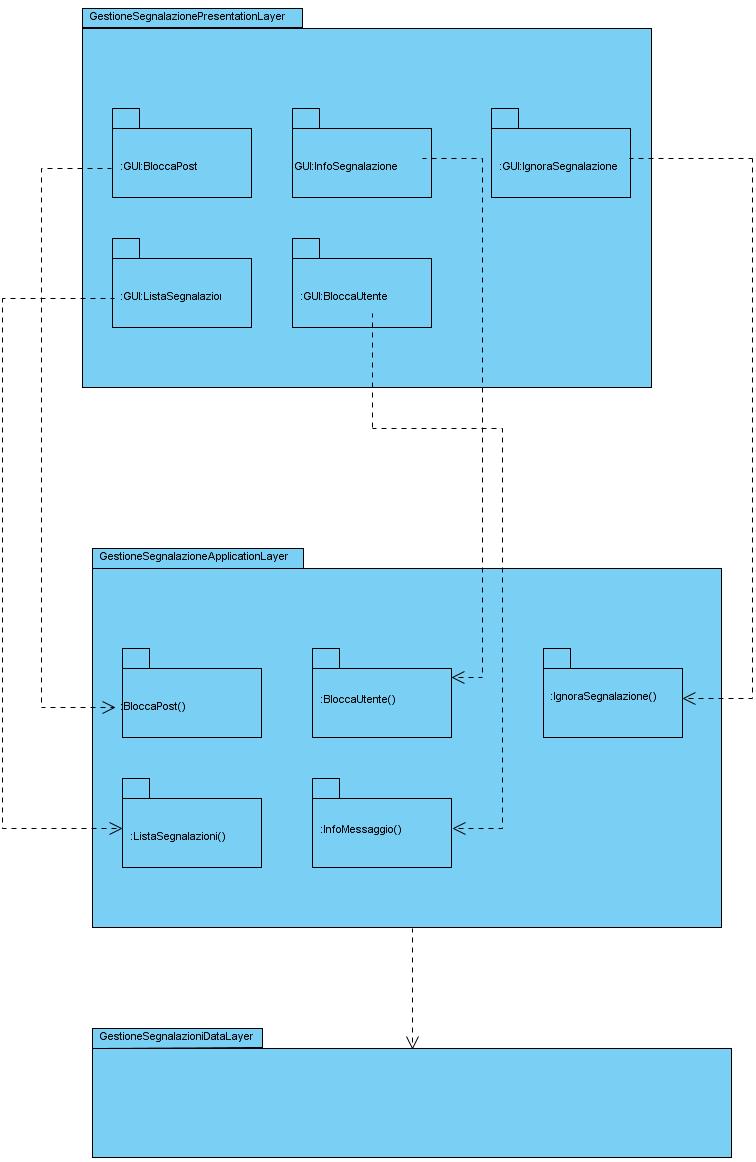
#### **GestioneOperatoriApplicationLayer**

* ListaOperatori(): operazioni per visualizzare tutti gli operatori.
* InfoOperatore(): operazioni per la visualizzazione delle informazioni relative ad un operatore.
* CreaOperatore(): operazioni per la registrazione di un nuovo operatore.
* RimuoviOperatore(): operazioni per la rimozione di un operatore dal sistema.

#### **GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli operatori.

**2.2.9 Gestione Segnalazioni**

****

#### **GestioneSegnalazioniPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIListaSegnalazioni: comprende le interfacce che consentono all’Operatore di visualizzare la lista delle segnalazioni.
* GUIInfoSegnalazione: comprende le interfacce che consentono all’Operatore di visualizzare i dettagli di una segnalazione.
* GUIIgnoraSegnalazione: comprende le interfacce che consentono all’Operatore di ignorare una segnalazione.
* GUIBloccaPost: comprende le interfacce che consentono all’Operatore di bloccare un post.
* GUIBloccaUtente: comprende le interfacce che consentono all’Operatore di bloccare un utente.

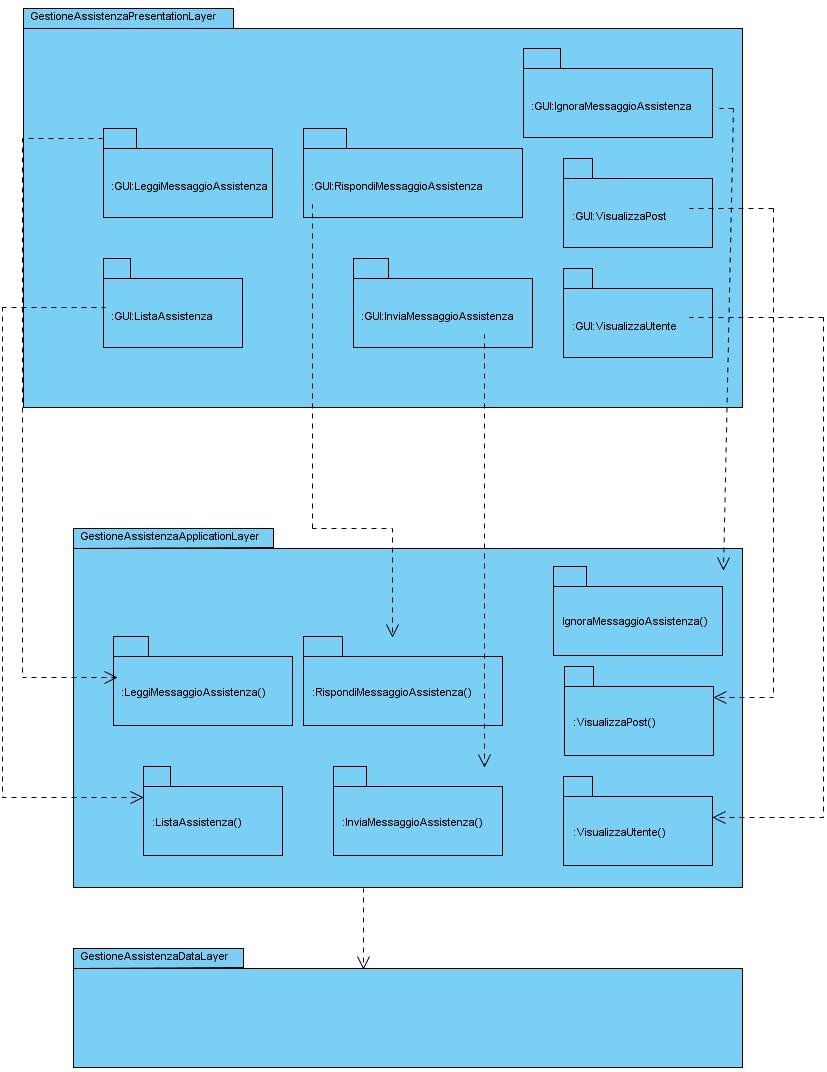
#### **GestioneSegnalazioniApplicationLayer**

* ListaSegnalazioni(): operazioni per visualizzare tutte le segnalazioni.
* InfoSegnalazione(): operazioni per la visualizzazione delle informazioni relative ad un segnalazione.
* IgnoraSegnalazione(): operazioni per ignorare una determinata segnalazione.
* BloccaPost(): operazioni per il blocco di un post dal sistema.
* BloccaUtente():operazione per il blocco di un utente dal sistema.

#### **GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle segnalazioni.

**2.2.10 Gestione Assistenza:**

****

#### **GestioneAssistenzaPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIListaAssistenza: comprende le interfacce che consentono all’Operatore di visualizzare la lista dei messaggi d’assistenza.
* GUILeggiMessaggioAssistenza: comprende le interfacce che consentono all’Operatore di visualizzare i dettagli di un messaggio d’assistenza.
* GUIIgnoraMessaggioAssistenza: comprende le interfacce che consentono all’Operatore di ignorare un messaggio d’assistenza.
* GUIRispondiMessaggioAssistenza: comprende le interfacce che consentono all’Operatore di rispondere a un messaggio d’assistenza.
* GUIInviaMessaggioAssistenza: comprende le interfacce che consentono all’Utente Base di inviare un messaggio d’assistenza.
* GUIVisualizzaPost:comprende le interfacce che consentono all’Operatore di visualizzare i dettagli di un post.
* GUIVisualizzaUtente:comprende le interfacce che consentono all’Operatore di visualizzare i dettagli di un utente.

#### **GestioneAssistenzaApplicationLayer**

* ListaAssistenza(): operazioni per visualizzare tutte i messaggi d’assistenza.
* LeggiMessaggioAssistenza(): operazioni per la visualizzazione delle informazioni relative ad un messaggio d’assistenza.
* IgnoraMessaggioAssistenza(): operazioni per ignorare un determinato messaggio d’assistenza.
* RispondiMessaggioAssistenza(): operazioni per rispondere a un messaggio d’assistenza.
* InviaMessaggioAssistenza(): operazioni per inviare un messaggio d’assistenza.
* VisualizzaPost(): operazioni per la visualizzazione dei dettagli di un post del sistema.
* VisualizzaUtente():operazione per la visualizzazione dei dettagli di un utente del sistema.

#### **GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle segnalazioni.