**Pet Party**

**Immagine che contiene Elementi grafici, cuore, clipart, grafica

Descrizione generata automaticamente**

**Membri del team di sviluppo:**

Giacomo Sampaoli 975846 Giorgio Ianniciello 978893 Davide Provolo 838368

Indice

[Abstract 3](#_Toc138003823)

[Analisi dei requisiti - Requisiti del sistema 4](#_Toc138003824)

[Analisi del Dominio - Tabella dei requisiti 5](#_Toc138003825)

[Analisi del Domino – Vocabolario 7](#_Toc138003826)

[Analisi dei requisiti - Casi d’uso 9](#_Toc138003827)

[Scenari 10](#_Toc138003828)

[Analisi del Rischio - Tabella di valutazione dei beni 17](#_Toc138003829)

[Tabella Minacce e Controlli 18](#_Toc138003830)

[Analisi Tecnologica della sicurezza 19](#_Toc138003831)

[Security Use Case e Misuse Case 21](#_Toc138003832)

[Requisiti di protezione dei dati 28](#_Toc138003833)

[Requisiti di sistema aggiornati 29](#_Toc138003834)

[Glossario aggiornato 30](#_Toc138003835)

[Casi d’uso aggiornati 31](#_Toc138003836)

[Analisi del problema 33](#_Toc138003837)

[Analisi documento dei requisiti: tabella delle funzionalità 33](#_Toc138003838)

[Analisi dei vincoli - Tabella dei vincoli 46](#_Toc138003839)

[Analisi delle interazioni – Tabella Maschere 47](#_Toc138003840)

[Analisi Ruoli e responsabilità – Tabella Ruoli 52](#_Toc138003841)

[Scomposizione del problema 55](#_Toc138003842)

[Modello del dominio 57](#_Toc138003843)

[Architettura Logica 60](#_Toc138003844)

[Comportamento 69](#_Toc138003845)

[Piano di Lavoro 70](#_Toc138003846)

[Sviluppi Futuri 71](#_Toc138003847)

[Piano del Collaudo 72](#_Toc138003848)

[Progettazione 79](#_Toc138003849)

[Diagramma dei Package 81](#_Toc138003850)

[Diagramma dei componenti 82](#_Toc138003851)

[Progettazione di dettaglio 83](#_Toc138003852)

[Interazione 94](#_Toc138003853)

[Comportamento 101](#_Toc138003854)

[Progettazione della Persistenza 102](#_Toc138003855)

[Progettazione del collaudo 103](#_Toc138003856)

[Deployment 108](#_Toc138003857)

# Abstract

L’obiettivo che si pone l’applicazione **Pet Party** è quello di mettere in contatto tutti gli amanti degli animali domestici e di tenere traccia delle attività dei loro compagni di giochi. Il sistema di condivisione di **Pet Party** permette ai padroni di trovare veterinari e toelettatori specializzati in varie specie e razze, di confrontarsi con altri utenti creando post dedicati ai loro amati animali, rendendo lo scambio di consigli utili più semplice che mai. L’applicazione, oltre alla possibilità di condivisione di cui sopra, permette agli utenti di tenere un registro delle attività giornaliere dei propri animali, come ad esempio i pasti nella giornata, le ore di svago o passeggiata e i periodi di sonno, tutto al fine di rendere la vita di animali e padroni più felice e sana.

# Analisi dei requisiti - Requisiti del sistema

• Il software deve permettere agli utenti di registrarsi sulla piattaforma.

• Alla registrazione l’utente dovrà poter scegliere se usufruire dell’applicazione come amante degli animali oppure professionista

• Al momento della registrazione i professionisti dovranno fornire anche un documento di identità

• L’amministratore, l’assistenza e gli utenti potranno accedere all’applicazione solo dopo aver fornito le proprie credenziali: username e password

• I professionisti possono essere: veterinario, addestratore, toelettatore, etc …

• I professionisti forniranno informazioni aggiuntive quali specializzazione della specie, numero di telefono, orario di lavoro, tariffa, zona di interesse

• Chiunque può registrare sotto il proprio profilo uno o più animali domestici

• Ogni utente può condividere con i propri seguaci dei contenuti riguardanti i propri animali domestici e visualizzare i contenuti degli utenti seguiti

• Gli utenti possono interagire con gli altri utenti tramite apprezzamenti ai contenuti, commenti ai contenuti

• Ogni utenti dispone di un’area privata dove registrare le informazioni relativi ai propri animali domestici

• L’assistenza ha il compito di supervisionare le segnalazioni degli utenti e verificare l’identità dei professionisti

• I membri dell’assistenza sono selezionati dall’amministratori

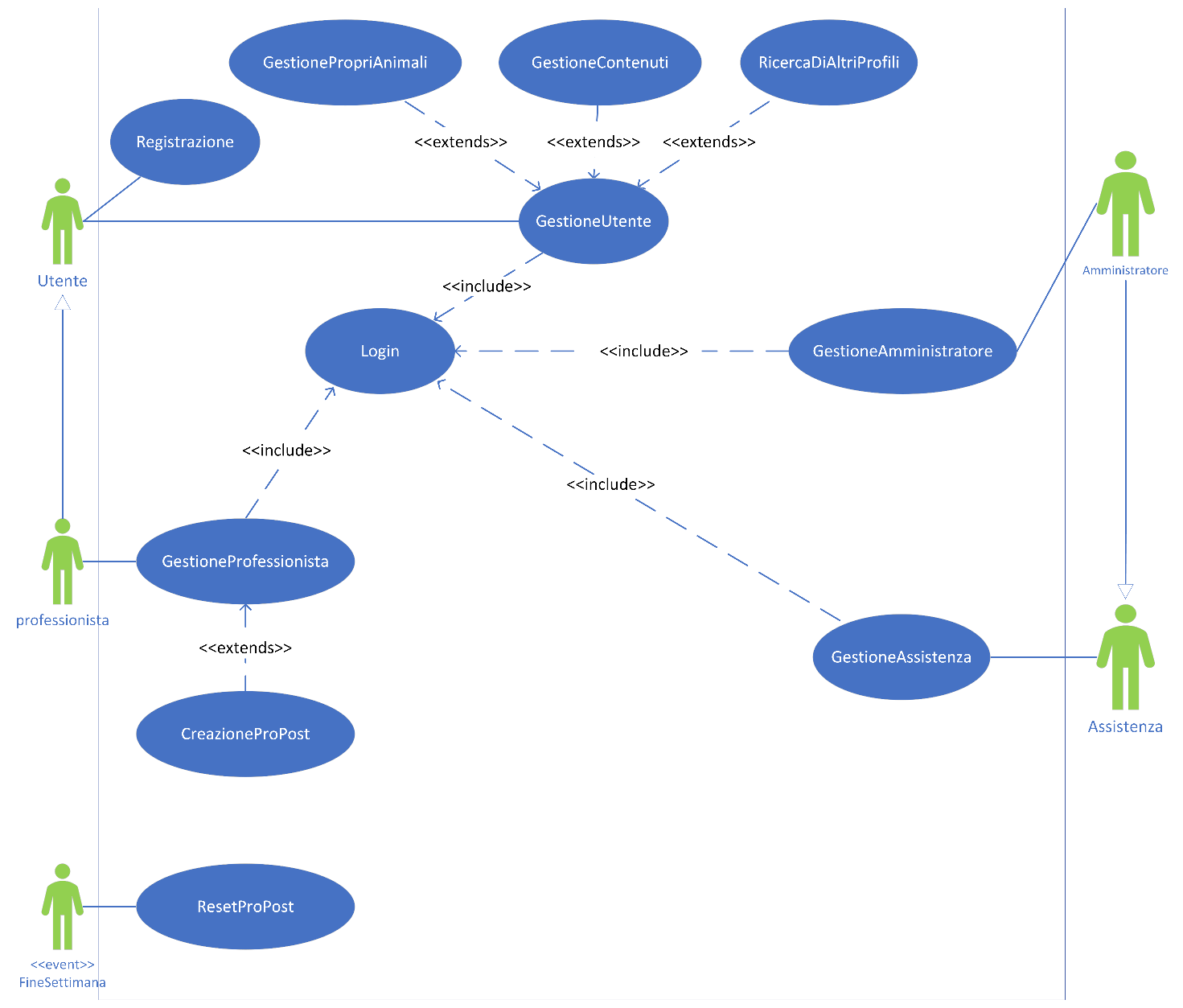
# Analisi del Dominio - Tabella dei requisiti

|  |  |  |
| --- | --- | --- |
| ID | REQUISITO | TIPO |
| R1F | L’utilizzo del servizio è possibile solo previa registrazione degli utenti | Funzionale |
| R2F | In fase di registrazione è necessaria la distinzione tra Utente Standard e Utente Professionista | Funzionale |
| R3F | Per la registrazione dei Professionisti è necessario caricare un documento di identità | Funzionale |
| R4F | Per la registrazione è necessario specificare nome, cognome, username, password | Funzionale |
| R5F | Gli utenti sono identificati tramite username | Funzionale |
| R6F | Gli utenti sono autenticati tramite username e password | Funzionale |
| R7F | Gli utenti possono registrare o modificare i dati dei propri animali | Funzionale |
| R8F | Possibilità di creare contenuti visibili ai propri seguaci | Funzionale |
| R9F | Possibilità di seguire altri utenti e visualizzare i loro contenuti | Funzionale |
| R10F | Possibilità di mettersi in contatto con altri utenti | Funzionale |
| R11F | Area privata in cui inserire le informazioni dei propri animali | Funzionale |
| R12F | Gli utenti possono commentare i contenuti | Funzionale |
| R13F | Gli utenti possono segnalare i commenti e i contenuti | Funzionale |
| R14F | Gli utenti possono apprezzare i contenuti degli altri utenti (anche i propri) | Funzionale |
| R15F | L’assistenza può bloccare/sbloccare utenti della piattaforma | Funzionale |
| R16F | L’assistenza può rimuovere i commenti e i post | Funzionale |
| R17F | L’assistenza può confermare o rifiutare richieste di registrazione di professionisti | Funzionale |
| R18F | Possibilità di cercare un padrone o animale | Funzionale |
| R19F | L’amministratore possiede tutti i privilegi, tra cui la creazione/eliminazione di account assistenza e funzionalità dell’assistenza | Funzionale |
| R1NF | Velocità di ricerca profili | Non funzionale |
| R2NF | Velocità di navigazione dell’applicazione | Non funzionale |
| R3NF | Interfaccia intuitiva | Non funzionale |
| R4NF | Creazione dei contenuti intuitiva | Non funzionale |
| R5NF | Backup dei dati periodico | Non funzionale |
| R6NF | Portabilità | Non funzionale |
| R7NF | L’Applicazione dovrà essere rilasciata sotto forma di file .war | Non funzionale |
| R8NF | I dati sensibili dell’account devono essere protetti per mezzo di meccanismi di cifratura | Non funzionale |

# Analisi del Domino – Vocabolario

|  |  |  |
| --- | --- | --- |
| Voce | Definizione | Sinonimi |
| Utente | Qualsiasi persona che interagisce con l’applicazione. | Attore |
| Account | Oggetto univoco utilizzato per riconoscere gli utenti all’interno del sistema |  |
| Utente Standard | Utente che dopo essersi registrato avrà la possibilità di usufruire dei vari servizi resi disponibili dall’applicazione |  |
| Utente Professionista | Utente che, dopo conferma della propria identità da parte dell’Assistenza, avrà la possibilità di pubblicizzare il proprio servizio riguardante gli animali domestici |  |
| Amministratore | Utente che possiede tutti i privilegi | Admin |
| Assistenza | Utente che si occupa di gestire le segnalazioni degli utenti. Ha la possibilità di eliminare commenti/contenuti segnalati, bloccare/sbloccare utenti. L’Assistenza si occupa anche di confermare l’identità degli utenti Professionisti che provano a registrarsi all’applicazione |  |
| Commento | Opinione testuale espressa relativa ad un contenuto |  |
| Contenuto | Pubblicazione multimediale di un utente | Post |
| Credenziali | Username e password utilizzati per accedere all’applicazione |  |
| Username | Stringa alfanumerica |  |
| Password | Stringa di riconoscimento fornita dall’utente all’applicazione per poter accedere |  |
| Documento di identità | Documento con la finalità di dimostrare l’identità dell’utente Professionista |  |
| Login | Operazione necessaria per identificare ed autenticare un utente | Autenticazione |
| Registrazione | Fase in cui un nuovo utente deposita informazioni personali |  |
| Registrazione animali | Fase in cui gli utenti registrano i propri animali domestici |  |
| Seguace | Utente che ha deciso di rimanere aggiornato ai contenuti di un altro utente | Follower |
| Utente Attivo | Un utente Professionista autenticato che può usufruire dei servizi dell’applicazione |  |
| Utente bloccato | Un utente che non può accedere all’applicazione |  |
| Area privata | Sezione personale in cui inserire informazioni relative ai propri animali domestici e i propri recapiti |  |
| Specie | Classificazione degli animali domestici |  |
| Razza | Sottoclassificazione degli animali domestici |  |
| Post professionista | Metodo di creazione di Post esclusivo per i Professionisti volto alla sponsorizzazione della sua attività | ProPost |
| Pagina di Benvenuto | Prima pagina presentata dopo l’Autenticazione dove sono presentati i post degli utenti seguiti | Home, Home Page, Feed |
| Bloccare/Sbloccare | Azione volta alla disabilitazione/abilitazione di un account in seguito a comportamenti illeciti | Ban/Unban |
| Apprezzare | Azione volta all’esprimere un apprezzamento ad un contenuto di un altro utente | Like, Mi Piace |

# Analisi dei requisiti - Casi d’uso



# Scenari

|  |  |
| --- | --- |
| Titolo | Login |
| Descrizione | Permette di accedere al sistema |
| Attori | Amministratore, Assistenza, Utente Standard, Utente Professionista |
| Relazioni | GestioneUtente, GestioneAmministratore, GestioneAssistenza, GestioneProfessionisti |
| Precondizioni | L’attore che partecipa al caso d’uso deve essere registrato nel sistema |
| Postcondizioni | L’attore è autenticato nel sistema |
| Scenario Principale | 1. Il sistema fornisce un’interfaccia utente per l’inserimento di username e password. 2. L’attore inserisce le proprie credenziali e conferma. 3. Il sistema verifica le credenziali. 4. Dopo le opportune verifiche viene presentata una schermata consona al tipo di attore. |
| Scenari Alternativi | A)Credenziali inserite non riconosciute:  4. Il sistema notifica l’attore con un avviso consono.  5. Il sistema ripropone all’attore l’interfaccia per l’inserimento delle credenziali.  B)Account bloccato:  4. Il sistema notifica l’attore con un avviso relativo allo stato dell’account.  5. Il sistema ripropone all’attore l’interfaccia per l’inserimento delle credenziali.  C) La registrazione del Professionista non è stata ancora verificata:  4. Il sistema notifica che l’account è in fase di verifica.  5. Il sistema ripropone all’attore l’interfaccia per l’inserimento delle credenziali.  D) La registrazione del Professionista è stata negata:  4. Il sistema notifica che l’autenticazione del professionista è stata negata e che l’account verrà eliminato dal sistema.  5. Il sistema ripropone all’attore l’interfaccia per l’inserimento delle credenziali. |
| Requisiti Non funzionali | **R8NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneProfessionista |
| Descrizione | Gestione di funzioni relative agli orari di lavoro e ai suoi ambiti di specializzazione. |
| Attori | Utente Professionista |
| Relazioni | Login, CreazioneProPost |
| Precondizioni | Il Professionista deve essersi registrato sulla piattaforma e deve essere stato verificato dall’assistenza. |
| Postcondizioni |  |
| Scenario Principale | 1. Login. 2. Il Professionista avrà la possibilità di eseguire tutte le funzionalità disponibili per l’Utente Standard, e inoltre potrà: definire le proprie modalità di lavoro, ambiti di specializzazione e accedere alle funzionalità di CreazioneProPost |
| Scenari Alternativi |  |
| Requisiti Non funzionali |  |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | CreazioneProPost |
| Descrizione | Gestione di creazioni di contenuti ProPost riservato al professionista |
| Attori | Professionista |
| Relazioni | GestioneProfessionista |
| Precondizioni | Il Professionista deve essersi registrato sulla piattaforma e deve essere stato verificato dall’assistenza. |
| Postcondizioni | Pubblicazione del ProPost |
| Scenario Principale | Il Professionista ha la possibilità di creare ProPost |
| Scenari Alternativi | A) Il Professionista vuole caricare un ProPost che gli farebbe eccedere il limite settimanale:  3. Il sistema impedisce la pubblicazione del ProPost.  4. Il sistema notifica il Professionista evidenziando l’errore.  5. Il sistema presenta al Professionista la possibilità di creare un Post. |
| Requisiti Non funzionali |  |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneUtente |
| Descrizione | Permette di usufruire delle funzionalità base del sistema tramite la home che gli perdette di visualizzare i Post degli utenti che segue, i ProPost degli Utenti Professionisti sponsorizzati ed eventuali ProPost degli utenti Professionisti che segue |
| Attori | Utente Standard, Utente Professionista |
| Relazioni | GestioneAnimali, CondivisionePost, RicercaDiAltriProfili, Login |
| Precondizioni | L’Utente deve aver effettuato il login. |
| Postcondizioni |  |
| Scenario Principale | 1. L’utente può accedere alle funzionalità di: GestioneAnimali, CondivisionePost e RicercaDiAltriProfili. |
| Scenari Alternativi |  |
| Requisiti Non funzionali | **R1NF, R2NF, R4NF, R5NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | RicercaDiAltriProfili |
| Descrizione | Permette all’Utente di cercare altri utenti registrati. |
| Attori | Utente |
| Relazioni | GestioneUtente |
| Precondizioni |  |
| Postcondizioni | Viene mostrato all’Utente il risultato della ricerca. |
| Scenario Principale | 1A. L’Utente inserisce l’username del profilo che vuole visualizzare.  1B. L’Utente seleziona “ricerca avanzata” e specifica i campi di suo interesse (zona, razza, professionista/non professionista).   1. Ottiene una lista di profili attinenti ai criteri di ricerca. |
| Scenari Alternativi | A) La ricerca non produce risultati  2. Viene mostrato un messaggio che comunica l’assenza di profili inerenti ai criteri di ricerca. |
| Requisiti Non funzionali | **R3NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneContenuti |
| Descrizione | Permette all’Utente di condividere Post. |
| Attori | Utente Standard, Utente Professionista |
| Relazioni | GestioneUtente |
| Precondizioni | L’Utente deve aver registrato almeno un animale |
| Postcondizioni | Il Post è stato condiviso con gli altri utenti seguaci ed è stato salvato nel database. |
| Scenario Principale | 1. Il sistema presenta all’Utente una sezione dedicata alla creazione di Post. 2. L’Utente sceglie a quali animali appartiene il Post. 3. L’Utente crea un Post multimediale che può avere una descrizione. 4. Una volta pubblicato il Post il sistema presenta un messaggio di avvenuta pubblicazione all’Utente e lo porta alla Home. |
| Scenari Alternativi | A) L’Utente tenta di pubblicare un post vuoto o senza aver selezionato almeno un animale.  3. Il sistema impedisce la creazione del Post e riporta l’Utente alla sezione dedicata alla creazione di Post. |
| Requisiti Non funzionali |  |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | Registrazione |
| Descrizione | Permette di registrarsi alla piattaforma. |
| Attori | Utente Standard, Utente Professionista |
| Relazioni |  |
| Precondizioni | L’Utente non deve risultare già registrato. |
| Postcondizioni | L’Utente può essere autenticato dal sistema.  Se l’Utente si è registrato come Professionista dovrà aspettare l’esito dell’autenticazione da parte dell’Assistenza. |
| Scenario Principale | 1. L’Utente visita la pagina di registrazione. 2. La piattaforma presenta la possibilità di registrarsi come Utente Standard (A) o Professionista (B)   3A. L’Utente inserisce nome, cognome, username, password e numero di telefono (facoltativo).  3B. L’Utente inserisce oltre ai dati sopracitati, un documento d’identità.  4. L’Utente deve confermare la conclusione della registrazione.   1. Il sistema controlla l’univocità dell’username, che l’username sia appropriato e che la password sia sufficientemente complessa. Se queste condizioni vengono rispettate il sistema memorizza tutti i dati associati. 2. Viene visualizzato un messaggio di conferma di avvenuta registrazione e nel caso del professionista, viene aggiunto un messaggio di invito ad attendere la verifica e l’abilitazione definitiva dell’account da parte dell’assistenza. 3. Il sistema mostra la schermata di login. |
| Scenari Alternativi | 1. Se le condizioni di registrazione non vengono rispettate verrà visualizzata nuovamente la sezione di registrazione in aggiunta ad un messaggio di errore di mancata registrazione e la lista di condizioni non rispettate. |
| Requisiti Non funzionali | **R8NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneAnimali |
| Descrizione | Permette di gestire le informazioni dei propri animali |
| Attori | Utente |
| Relazioni | GestioneUtente |
| Precondizioni |  |
| Postcondizioni | Le informazioni degli animali vengono registrate nel database. |
| Scenario Principale | 1. L’Utente inserisce nome, specie e razza se necessaria. 2. L’Utente ha la possibilità di aggiungere ulteriori informazioni come: foto, peso, tratti distintivi, descrizione generica, etc … |
| Scenari Alternativi | 1. L’Utente prova a registrare un animale senza aver completato i parametri essenziali 2. Il sistema presenta un messaggio di errore e ti riporta alla pagina di GestioneAnimali. |
| Requisiti Non funzionali | **R8NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneAmministratore |
| Descrizione | Gestione di funzionalità riservate all’amministratore di sistema. |
| Attori | Amministratore |
| Relazioni | Login |
| Precondizioni | L’amministratore deve aver effettuato il Login con le proprie credenziali. |
| Postcondizioni |  |
| Scenario Principale | 1. Login 2. Il Sistema presenta una schermata che permette all’Amministratore di eseguire le funzionalità disponibili. 3. L’amministratore crea/elimina un account Assistenza oppure utilizza le funzioni disponibili di GestioneAssistenza. |
| Scenari Alternativi | 1. L’Amministratore cerca di creare un account Assistenza con un username esistente. 2. Viene notificato un messaggio di errore. 3. L’Amministratore viene riportato alla schermata principale |
| Requisiti Non funzionali | **R8NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneAssistenza |
| Descrizione | Gestione di funzionalità riservata all’Assistenza e all’Amministratore |
| Attori | Assistenza |
| Relazioni | Login |
| Precondizioni | L’Amministratore deve aver creato delle credenziali per account Assistenza |
| Postcondizioni |  |
| Scenario Principale | 1. Login 2. Il sistema presenta un’interfaccia che permette all’Assistenza di eseguire le funzionalità. 3. L’assistenza visualizza i commenti/post segnalati e può decidere se eliminare il commento/post se ritenuto inappropriato. Ha inoltre la possibilità di Ban/Unban del creatore del contenuto se ritenuto necessario. 4. L’assistenza verifica l’identità degli Utenti Professionisti |
| Scenari Alternativi | 1. L’identità del Professionista che ha richiesto la registrazione non può essere validata:   5. L’Assistenza rifiuta la richiesta di registrazione. |
| Requisiti Non funzionali | **R8NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | ResetProPost |
| Descrizione | Ristabilisce la quantità di ProPost pubblicabili dal Professionista in una settimana |
| Attori | FineSettimana «event» |
| Relazioni |  |
| Precondizioni |  |
| Postcondizioni | I Professionisti sono abilitati a pubblicare 2 ProPost |
| Scenario Principale | 1. Si verifica L’evento FineSettimana 2. Il Contatore associato ad ogni Professionista relativo ai ProPost viene ristabilito |
| Scenari Alternativi |  |
| Requisiti Non funzionali |  |
| Punti aperti |  |

# Analisi del Rischio - Tabella di valutazione dei beni

|  |  |  |
| --- | --- | --- |
| Bene | Valore | Esposizione |
| Sistema Informativo | Alto.  Supporto alla gestione della piattaforma | Alta.  Perdita di immagine e finanziaria. Costo di tempo per il ripristino dei dati da backup. |
| Informazioni sugli utenti | Medio.  Nome, cognome, e credenziali degli utenti | Media.  Perdita di immagine e costo per il ripristino di backup. |
| Informazioni dell’Assistenza | Alto.  Credenziali dell’Assistenza | Alta.  Perdita di immagine dovuta al fatto che gli utenti potrebbero essere bloccati dal possessore delle credenziali di Assistenza. |
| Informazioni dell’Amministratore | Alto.  Credenziali dell’Amministratore | Alta.  Perdita della gestione dell’applicazione |
| Documenti Professionisti | Alto. | Alta.  Perdita di immagine e problemi legali relativi al furto di identità dei Professionisti alla causa delle possibili frodi che potrebbero essere perpetuate dai possibili malfattori. |

# Tabella Minacce e Controlli

|  |  |  |  |
| --- | --- | --- | --- |
| Minaccia | Probabilità | Controllo | Fattibilità |
| Furto di identità Amministratore | Bassa: username e password non sono facili di identificare | Log degli accessi, accesso da macchine sicure | Basso costo |
| Furto di identità Assistenza | Bassa: username e password non sono facili di identificare | Log degli accessi | Basso costo |
| Furto di identità utente | Medio-Alta: ingegneria sociale | Log degli accessi | Medio-alto costo |
| DoS | Alta | Progettazione adeguata, controllo e limitazione degli accessi | Basso costo: impossibile prevenire un DoS |
| Furto dei documenti dei Professionisti | Media: i documenti possono essere intercettati tramite Man in The Middle oppure estrapolati dal database mentre sono in attesa di verifica | Progettazione adeguata, cifratura e log delle comunicazioni | Medio-alto: la cifratura è ibrida: chiavi asimmetriche e simmetriche |

# Analisi Tecnologica della sicurezza

|  |  |
| --- | --- |
| Tecnologia | Vulnerabilità |
| Cifratura delle comunicazioni | Utilizzo di una cifratura ibrida, simmetrica e asimmetrica, con le loro relative vulnerabilità:  **Cifratura simmetrica:**   * Tempo di vita della chiave * Più informazioni vengono cifrate con la stessa chiave, più materiale è offerto per l’analisi del testo * Memorizzazione della chiave * Lunghezza della chiave   **Cifratura asimmetrica:**   * Lunghezza della chiave * Memorizzazione della chiave privata. L’utilizzo di cifrature standard non recenti potrebbero causare vulnerabilità ulteriori. |
| Autenticazione tramite credenziali | * Social engineering: un utente potrebbe essere vittima di un attacco di social engineering, volto al furto delle credenziali. * Password cracking: attraverso metodologie di cracking delle password, l’attaccante risale alla password di un utente o amministratore di sistema. * Negligenza: l’utente volontariamente o per sbaglio fornisce le credenziali a terzi. * Password banali |
| Architettura Client/Server | * Attacco DDos/Dos: l’obiettivo dell’attacco è quello di negare il servizio offerto agli utenti. * Attacco Man in the Middle: un malintenzionato potrebbe riuscire a sottrarre informazioni o effettuare operazioni illecite. * Sniffing della comunicazione: un malintenzionato potrebbe riuscire ad intercettare una comunicazione ed a sottrarne informazioni. * Deadlock: il servitore potrebbe trovarsi in una situazione di stallo generata da una richiesta ciclica. * Esposizione dei client: i client sono esposti, dato che devono connettersi al servitore. |

# Security Use Case e Misuse Case

Immagine che contiene luce

Descrizione generata automaticamente con attendibilità media

**Immagine che contiene testo, Carattere, schermata, cerchio

Descrizione generata automaticamente**Immagine che contiene diagramma, testo, linea, cerchio

Descrizione generata automaticamente

|  |  |  |  |
| --- | --- | --- | --- |
| Titolo | ControlloAccesso | | |
| Descrizione | Gli accessi alla piattaforma devono essere controllati | | |
| Misuse case | Sniffing, FurtoCredenziali | | |
| Relazioni | GestioneUtente, GestioneAmministratore, GestioneAssistenza, GestioneProfessionista | | |
| Precondizioni | L’attaccante ha i mezzi per scoprire almeno l’username di account di qualsiasi tipo | | |
| Postcondizioni | L’applicazione impossibilita temporaneamente l’accesso dell’utente e notifica un tentativo di accesso fraudolento | | |
| Scenario principale | **Sistema** | | **Attaccante** |
|  | | Dopo aver individuato degli username tenta di accedere al sistema inserendo password tramite un attacco con dizionario |
| Controlla che le credenziali immesse siano corrette, se non lo sono, dopo un certo numero di tentativi, blocca l’accesso | |  |
| Scenario di un attacco avvenuto con successo | **Sistema** | **Attaccante** | |
|  | Attacco con dizionario riuscito | |
| Il Sistema controlla le credenziali e consente l’accesso al sistema |  | |
|  | Naviga tra le maschere del sistema e cerca di capirne più informazioni possibili | |
| Il Sistema scrive nel log tutte le operazioni eseguite dall’utente |  | |
| Il sistema monitora periodicamente il log alla ricerca di pattern di accesso atipici e se rilevati notifica un accesso fraudolento |  | |

|  |  |  |
| --- | --- | --- |
| Titolo | ProtezioneIdentità | |
| Relazioni | GestioneAmministratore, GestioneAssistenza, GestioneProfessionista | |
| Descrizione | Durante la fase di verifica del documento di un professionista va garantita la sicurezza della comunicazione | |
| Misuse Case | Sniffing, Frode | |
| Precondizioni | L’Attaccante ha i mezzi per intercettare le comunicazioni | |
| Postcondizioni | Il Sistema notifica un tentativo di danno all’integrità dei dati | |
| Scenario principale | **Sistema** | **Attaccante** |
| Controlla l’invio e la ricezione dei documenti e salva i dati nei log |  |
|  | Intercetta un documento inviato |
|  | Non riesce a rimuovere i meccanismi di sicurezza |
|  | Inoltra i documenti intercettati al destinatario |
| Il sistema notifica una anomalia nella ricezione di un documento |  |
| Scenario di un attacco avvenuto con successo | **Sistema** | **Attaccante** |
| Controlla l’invio e la ricezione dei documenti e salva i dati nei log |  |
|  | Intercetta un documento inviato |
|  | Riesce a rimuovere i meccanismi di sicurezza |
|  | Ottiene i dati sensibili, poi riapplica i meccanismi di sicurezza e ritrasmette i dati |
| Elabora i dati e procede salvando il tutto nei log |  |
| Monitora periodicamente i log  alla ricerca di pattern atipici e se rilevati notifica all’amministratore e all’assistenza l’avvenimento |  |

|  |  |  |
| --- | --- | --- |
| Titolo | ProtezioneIdentità | |
| Descrizione | Durante la fase di verifica del documento di un professionista va garantita la sicurezza del database | |
| Misuse Case | Spoofing, Frode | |
| Relazioni | GestioneAmministratore, GestioneAssistenza, GestioneProfessionista | |
| Precondizioni | L’Attaccante ha i mezzi per estrapolare i documenti dal database | |
| Postcondizioni | Il Sistema notifica un tentativo di accesso malevolo al database | |
| Scenario principale | **Sistema** | **Attaccante** |
| Salva i documenti in attesa di verifica nel database |  |
|  | Tenta l’accesso al database |
|  | Non riesce ad accedere al database |
| Registra i tentativi di accesso nel log |  |
| Monitora periodicamente il log alla ricerca di pattern atipici e se rilevati notifica un accesso fraudolento |  |
| Scenario di un attacco avvenuto con successo | **Sistema** | **Attaccante** |
| Salva i documenti in attesa di verifica nel database |  |
|  | Tenta l’accesso al database |
|  | riesce ad accedere al database |
|  | Ottiene i dati sensibili |
| Registra i tentativi di accesso nel log |  |
| Monitora periodicamente il log alla ricerca di pattern atipici e se rilevati notifica un accesso fraudolento |  |

|  |  |  |
| --- | --- | --- |
| Titolo | Disponibilità | |
| Descrizione | Il sistema deve essere sempre disponibile in tutte le sue funzionalità | |
| Misuse Case | DoS | |
| Relazioni | GestioneUtente, GestioneProfessionista | |
| Precondizioni | L’Attaccante ha la potenza necessaria per mettere in azione questo tipo di attacco in modo efficace | |
| Postcondizioni | Il sistema notifica quando il flusso di richieste è fuori dall’ordinario e attua delle misure di sicurezza | |
| Scenario principale | **Sistema** | **Attaccante** |
| Ha i meccanismi di protezione attivi |  |
|  | Il malintenzionato mette in azione l’attacco |
| Notifica il tentativo di DoS mentre cerca di ridurre l’efficacia dell’attacco |  |
| Scenario di un attacco avvenuto con successo | **Sistema** | **Attaccante** |
| Ha i meccanismi di protezione attivi |  |
|  | Il malintenzionato mette in azione l’attacco |
| Notifica il tentativo di DoS mentre cerca di ridurre l’efficacia dell’attacco |  |
|  | Il sistema non riesce a contenere l’attacco e va fuori uso |
| L’amministratore vede che il sistema è fuori uso e tenta il ripristino locale |  |

|  |  |  |
| --- | --- | --- |
| Titolo | GarantireProtezione | |
| Descrizione | In tutte le trasmissioni i dati devono essere protetti | |
| Misuse Case | Sniffing, ManInTheMiddle | |
| Relazioni | GestioneUtente, GestioneAmministratore, GestioneAssistenza, GestioneProfessionista | |
| Precondizioni | 1. L’Attaccante ha i mezzi per intercettare i messaggi nel sistema 2. L’attaccante ha i mezzi per modificare i messaggi 3. L’attaccante ha i mezzi per spedire il messaggio modificato al destinatario | |
| Postcondizioni | Il Sistema notifica un tentativo di danno all’integrità dei dati | |
| Scenario principale | **Sistema** | **Attaccante** |
| Controlla l’invio dei messaggi e salva i dati nel log |  |
|  | Intercetta un messaggio inviato |
|  | Non riesce a rimuovere i meccanismi di sicurezza |
|  | Genera un messaggio e tenta di inviarlo al destinatario |
| Si accorge che il messaggio non è pertinente e segnala all’amministratore e all’assistenza l’avvenimento |  |
| Scenario di un attacco avvenuto con successo | **Sistema** | **Attaccante** |
| Controlla l’invio e la ricezione dei messaggi e salva i dati nel log |  |
|  | Intercetta un messaggio inviato |
|  | Riesce a rimuovere i meccanismi di sicurezza |
|  | Modifica i dati, poi riapplica i meccanismi di sicurezza e ritrasmette i dati |
| Elabora i dati e procede salvando il tutto nei log |  |
| Monitora periodicamente il log alla ricerca di pattern atipici e se rilevati notifica all’amministratore e all’assistenza l’avvenimento |  |

# Requisiti di protezione dei dati

Dall’analisi del rischio si possono evincere i seguenti ulteriori requisiti:

1. Generazione di un log per tenere traccia di

* 1. tutte le azioni che si verificano all’interno del sistema
  2. le informazioni scambiate tra le parti del sistema (che vanno protette per evitare che un accesso fraudolento al sistema di log possa rivelare dati riservati)

2. Adottare meccanismi di analisi del log per

* 1. Identificare accessi sospetti
  2. Identificare incongruenze tra i messaggi in uscita e in entrata
  3. Individuare una corretta politica di controllo degli accessi
  4. Proteggere i dati memorizzati e trasmessi

3. Il sistema deve essere realizzato in modo da consentire il ripristino locale ad un evento da parte dell’amministratore, nel caso in cui l’attacco DoS sia andato a buon fine.

4. In seguito a tre tentativi di accesso ad un account con credenziali errate, il sistema deve provvedere a sospendere temporaneamente l’account.

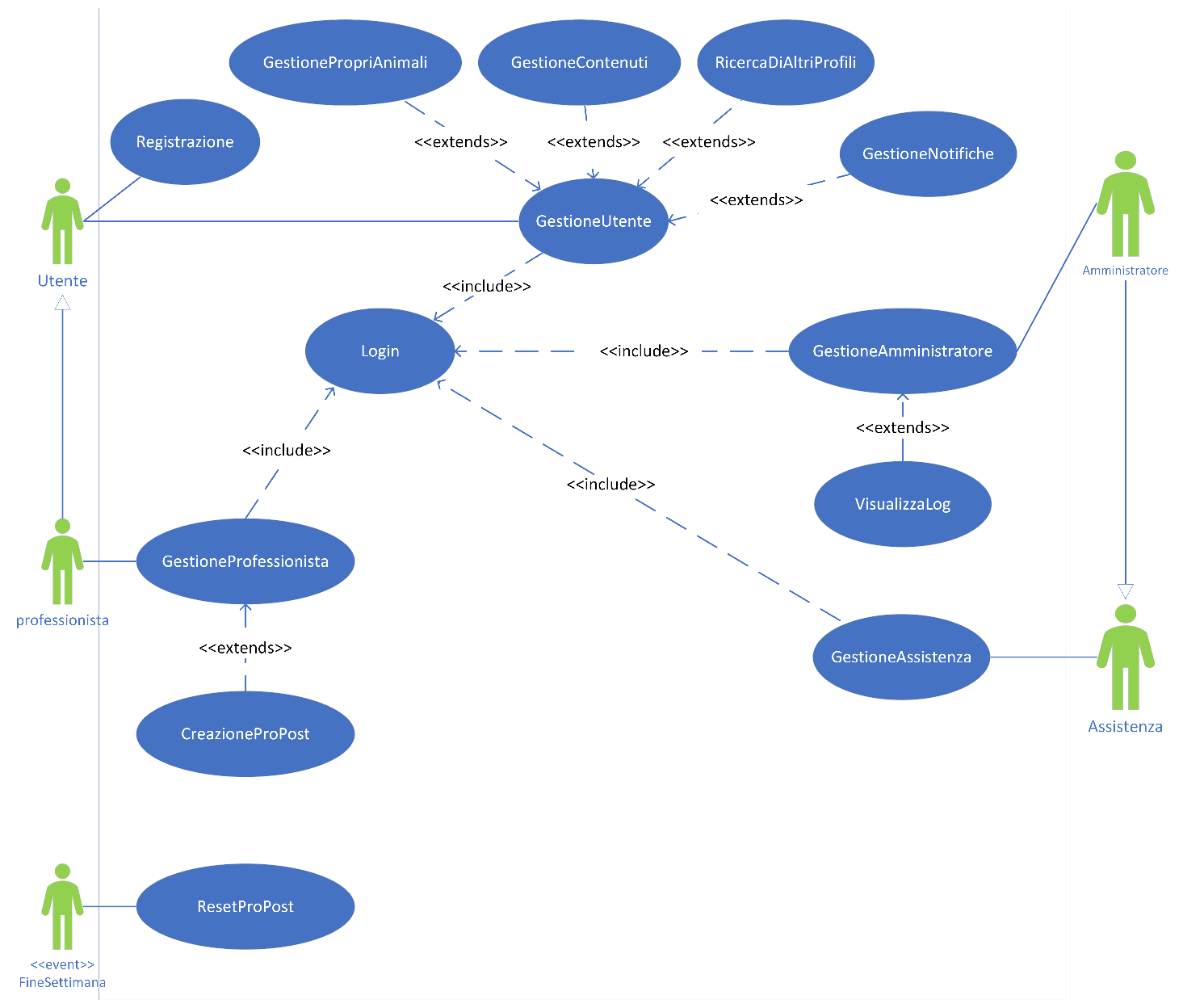
# Requisiti di sistema aggiornati

|  |  |  |
| --- | --- | --- |
| ID | REQUISITO | TIPO |
| R20F | Tramite un meccanismo di log il sistema memorizza tutte le operazioni eseguite sulla piattaforma | Funzionale |
| R21F | Devono essere presenti mezzi per l’analisi del log | Funzionale |
| R22F | Il sistema deve avere una struttura che lo renda facilmente ripristinabile in seguito ad un attacco DoS | Funzionale |
| R23F | Il sistema può bloccare automaticamente l’accesso ad un account | Funzionale |
| R24F | Il sistema notifica l’utente delle interazioni che lo riguardano | Funzionale |
| R25F | L’utente ha la possibilità di rimuovere un contenuto da lui creato | Funzionale |
| R26F | L’utente ha la possibilità di rimuovere un commento da lui creato | Funzionale |
| R27F | L’utente ha la possibilità di smettere di seguire utenti che seguiva in precedenza | Funzionale |
| R28F | L’utente ha la possibilità di togliere animali dal proprio profilo | Funzionale |
| R29F | Il professionista ha la possibilità di creare 2 ProPost a settimana | Funzionale |
| R30F | Il professionista deve definire le sue modalità lavorative e la sua specializzazione | Funzionale |
| R31F | Ogni tipo di utente ha la possibilità di modificare le informazioni del proprio profilo | Funzionale |
| R32F | Ogni Utente ha la possibilità di modificare le informazioni del profilo di un proprio animale | Funzionale |
| R9NF | Il log è visibile solo dall’amministratore | Non Funzionale |
| R10NF | L’account viene temporaneamente bloccato in seguito a 5 tentativi di accesso con credenziali errate consecutivi | Non Funzionale |
| R11NF | L’implementazione della chat verrà inserita in uno sviluppo futuro | Non Funzionale |

# Glossario aggiornato

|  |  |  |
| --- | --- | --- |
| Voce | Definizione | Sinonimi |
| Log | Registro dove vengono salate informazioni per risalire ad operazioni critiche svolte. Composto da una serie di voci | Registro |
| Voce di Log | Si tratta di una riga del log | Record di Log |
| Notifica | Messaggio che informa l’utente di un evento che lo coinvolge |  |
| Smettere di seguire | Smettere di seguire i contenuti di un altro utente | Unfollow |
| Zona | Provincia in cui opera il professionista |  |

# Casi d’uso aggiornati



|  |  |
| --- | --- |
| Titolo | VisualizzaLog |
| Descrizione | Permette di visualizzare i log del Sistema |
| Attori | Amministratore |
| Relazioni | GestioneAmministratore |
| Precondizioni | L’Amministratore ha eseguito l’accesso con le proprie credenziali |
| Postcondizioni |  |
| Scenario Principale | 1. Il Sistema offre una visualizzazione del log di sistema all’amministratore 2. L’amministratore può tornare alla schermata principale in qualsiasi momento |
| Scenari Alternativi |  |
| Requisiti Non funzionali | **R9NF** |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | GestioneNotifiche |
| Descrizione | Permette di visualizzare informazioni a proposito di nuovi like/commenti a propri post o di nuovi follower. |
| Attori | Utente |
| Relazioni | GestioneUtente |
| Precondizioni |  |
| Postcondizioni |  |
| Scenario Principale | 1. L’Utente accede alla lista delle nuove notifiche 2. Il sistema presenta all’Utente la lista delle sue notifiche relative a like, commenti o nuovi follower 3. Interagendo con le notifiche l’Utente può visionare il Post in questione ( nel caso in cui la notifica è relativa a like o commento ) oppure al profilo dell’altro Utente. |
| Scenari Alternativi |  |
| Requisiti Non funzionali | **R24F** |
| Punti aperti |  |

# Analisi del problema

# Analisi documento dei requisiti: tabella delle funzionalità

|  |  |  |  |
| --- | --- | --- | --- |
| Funzionalità | Tipo | Grado Complessità | Requisiti collegati |
| GestioneAssistenza | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R15F, R16F, R17F |
| BloccoUtente | Gestione dati | Semplice | R15F |
| SbloccoUtente | Gestione dati | Semplice | R15F |
| RimozioneCommento | Gestione dati | Semplice | R16F |
| EliminazionePost | Gestione dati | Semplice | R16F |
| AccettazioneProfessionista | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R17F |
| GestioneAmministratore | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R19F |
| CreazioneAccountAssistenza | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R19F |
| RimozioneAccountAssistenza | Gestione dati, interazione con l’esterno | Semplice | R19F |
| VisualizzaLog | Gestione dati | Semplice | R21F |
| ScritturaLog | Memorizzazione dati | Semplice | R20F |
| GestioneUtente | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R1F, R2F, R4F, R5F, R6F, R7F, R8F, R9F, R10F, R11F, R12F, R13F, R14F, R18F, R24F, R25F, R26F, R27F, R31F, R2NF, R4NF |
| SeguiUtente | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R9F |
| UnfollowUtente | Gestione dati, interazione con l’esterno | Semplice | R27F |
| ModificaProfilo | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R31F |
| GestioneNotifiche | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R24F |
| GestionePost | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R8F, R25F, R12F, R26F, R14F, R13F |
| CondivisionePost | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R8F |
| EliminazionePost | Gestione dati, interazione con l’esterno | Semplice | R25F |
| CreazioneCommento | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R12F |
| EliminazioneCommento | Gestione dati, interazione con l’esterno | Semplice | R26F |
| SegnalarePost | Memorizzazione dati, interazione con l’esterno | Semplice | R13F |
| SegnalareCommenti | Memorizzazione dati, interazione con l’esterno | Semplice | R13F |
| ApprezzareContenuti | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R14F |
| GestioneAnimali | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R7F, R11F,R32F |
| AggiungereAnimale | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R7F |
| TogliereAnimale | Gestione dati, interazione con l’esterno | Semplice | R28F |
| ModificaProfiloAnimale | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R32F |
| RicercaDiAltriProfili | Gestione dati, interazione con l’esterno | Complessa | R5F, R18F, R1NF |
| RicercaSemplice | Gestione dati, interazione con l’esterno | Semplice | R5F, R18F |
| RicercaAvanzata | Gestione dati, interazione con l’esterno | Semplice | R5F, R18F |
| Registrazione | Memorizzazione dati | Semplice | R1F, R2F, R3F, R4F |
| Login | Gestione dati, interazione con l’esterno | Semplice | R6F |
| ResetProPost | Gestione dati, interazione con l’esterno | Semplice | R29F |
| GestioneProfessionista | Memorizzazione dati, gestione dati, interazione con l’esterno | Complessa | R29F. R30F |
| CreazioneProPost | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R29F |
| ModificaProfiloProfessionista | Memorizzazione dati, gestione dati, interazione con l’esterno | Semplice | R30F |

**GestioneAmministratore: tabella informazioni/flusso**

CreazioneAccountAssistenza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Alto | Input | Non più di 32 caratteri |
| Password | Semplice | Molto Alto | Input | Non più di 32 caratteri |

RimozioneAccountAssistenza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Alto | Input | Non più di 32 caratteri |

VisualizzaLog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Record di lettura  **Composto da:**  Timestamp  Op. eseguita  Messaggio  Livello importanza | Composto  Semplice  Semplice  Semplice  Semplice | Alto  Medio  Alto  Molto Alto  Alto | Output  Output  Output  Output  Output | Esattamente 23 caratteri  Non Più di 256 caratteri  Non Più di 256 caratteri  **Enumeratore:**Basso,Medio,Alto |

ScritturaLog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Record di lettura  **Composto da:**  Timestamp  Op. eseguita  Messaggio  Livello importanza | Composto  Semplice  Semplice  Semplice  Semplice | Alto  Medio  Alto  Molto Alto  Alto | Input  Input  Input  Input  Input | Esattamente 23 caratteri  Non Più di 256 caratteri  Non Più di 256 caratteri  **Enumeratore:**Basso,Medio,Alto |

**GestioneAssistenza: tabella informazioni/flusso**

BloccoUtente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Username Assistente | Semplice | Alto | Input | Non più di 32 caratteri |

SbloccoUtente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Username Assistente | Semplice | Alto | Input | Non più di 32 caratteri |

RimozioneCommento

**Nota:** Funzione accessibile anche da utenti standard a patto che eseguano operazioni sui propri post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Alto | Input | Non più di 32 caratteri |
| IdCommento | Semplice | Alto | Input | Non più di 256 caratteri |

AccettazioneProfessionista

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Nome | Semplice | Medio | Input | Non più di 64 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 64 caratteri |
| Documento di identità | Semplice | Molto Alto | Input | Il documento deve essere valido e non scaduto |
| Username Assistente | Semplice | Alto | Input | Non più di 32 caratteri |

EliminazionePost

**Nota:** Funzione accessibile anche da utenti standard a patto che eseguano operazioni sui propri post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| IdPost | Semplice | Basso | Input | Non più di 128 caratteri |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |

**GestioneProfessionista: tabella informazioni/flusso**

CreazioneProPost

**Nota:** Nella creazione dei ProPost (e dei post in generale) si può pubblicare:

* Video con o senza testo descrittivo
* Immagine con o senza testo descrittivo

Inoltre, l’username deve obbligatoriamente appartenere ad un professionista.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| TimeStamp | Semplice | Basso | Input | Esattamente 23 caratteri |
| Contenuto  **Composto da:**  Immagine  Video  Testo | Composto  Semplice  Semplice  Semplice | Basso  Basso  Basso  Basso | Input  Input  Input  Input | Dimensione inferiore ai 8MB  Dimensione inferiore ai 128MB  Non più di 256 caratteri |

ModificaProfiloProfessionista

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Descrizione | Semplice | Basso | Input | Non più di 264 caratteri |
| Immagine Profilo | Semplice | Basso | Input | Non più di 8 MB |
| Zona | Semplice | Basso | Input | Non più di 64 caratteri |
| Telefono | Semplice | Basso | Input | Esattamente 10 caratteri |

**Nota:** le informazioni su tariffe e orari vengono inserite nella descrizione in quanto facoltative.

**GestioneUtente: tabella informazioni/flusso**

**Nota:** per le funzioni di EliminazionePost e EliminazioneCommenti vedere le tabelle presenti in GestioneAssistenza

GestioneNotifiche

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Descrizione | Semplice | Medio | Output | Non più di 256 caratteri |
| Username | Semplice | Medio | Input | Non più di 32 caratteri |

CondivisionePost

**Nota:** Nella creazione dei contenuti (post) si può pubblicare:

* Video con o senza testo descrittivo
* Immagine con o senza testo descrittivo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| TimeStamp | Semplice | Basso | Input | Esattamente 23 caratteri |
| Contenuto  **Composto da:**  Immagine  Video  Testo  Lista animali | Composto  Semplice  Semplice  Semplice  Semplice | Basso  Basso  Basso  Basso  Basso | Input  Input  Input  Input  Input | Dimensione inferiore ai 8MB  Dimensione inferiore ai 128MB  Non più di 256 caratteri  Minimo 1 animale |

CreazioneCommento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| IdCommento | Semplice | Basso | Input | Non più di 128 caratteri |
| TimeStamp | Semplice | Basso | Input | Esattamente 23 caratteri |
| Testo | Semplice | Basso | Input | Non più di 256 caratteri |
| IdPost | Semplice | Basso | Input | Non più di 128 caratteri |

SeguiUtente

**Nota:** L’utente “follower” non deve essere già tra i seguaci dell’utente “followed” e i due username non devono coincidere

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| UsernameFollower | Semplice | Basso | Input | Non più di 32 caratteri |
| UsernameFollowed | Semplice | Basso | Input | Non più di 32 caratteri |

UnfollowUtente

**Nota:** L’utente “follower” deve essere tra i seguaci dell’utente “followed” e i due username non devono coincidere

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| UsernameFollower | Semplice | Basso | Input | Non più di 32 caratteri |
| UsernameFollowed | Semplice | Basso | Input | Non più di 32 caratteri |

SegnalarePost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| IdPost | Semplice | Basso | Input | Non più di 128 caratteri |
| Motivo | Semplice | Basso | Input | Scelta dalla lista |
| TimeStamp | Semplice | Basso | Input | Esattamente 23 caratteri |

SegnalareCommenti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| IdCommento | Semplice | Basso | Input | Non più di 256 caratteri |
| Motivo | Semplice | Basso | Input | Scelta dalla lista |

ApprezzareContenuti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| IdPost | Semplice | Basso | Input | Non più di 128 caratteri |

ModificaProfilo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Nome | Semplice | Basso | Input | Non più di 64 caratteri |
| Cognome | Semplice | Basso | Input | Non più di 64 caratteri |
| Descrizione | Semplice | Basso | Input | Non più di 264 caratteri |
| Immagine Profilo | Semplice | Basso | Input | Non più di 8 MB |
| Telefono | Semplice | Basso | Input | Esattamente 10 caratteri |

**GestioneAnimali: tabella informazioni/flusso**

AggiungereAnimale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Nome | Semplice | Basso | Input | Non più di 64 caratteri |
| Razza | Semplice | Basso | Input | Non più di 64 caratteri |
| Specie | Semplice | Basso | Input | Non più di 64 caratteri |
| Sesso | Semplice | Basso | Input | **Enumerativo:** M o F |
| Descrizione | Semplice | Basso | Input | Non più di 256 |

TogliereAnimali

**Nota:** Gli animali nel sistema vengono identificati dalla coppia IdAnimale e Username del padrone che li registra nel sistema, un padrone può eliminare esclusivamente gli animali da lui registrati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| IdAnimale | Semplice | Basso | Input | Non più di 4 caratteri |

ModificaProfiloAnimali

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Nome | Semplice | Basso | Input | Non più di 64 caratteri |
| Descrizione | Semplice | Basso | Input | Non più di 264 caratteri |
| Immagine Profilo | Semplice | Basso | Input | Non più di 8 MB |

**RicercaDiAltriProfili: tabella informazioni/flusso**

RicercaSemplice

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Risultati | Semplice | Basso | Output | Set di Username |

RicercaAvanzata

**Nota:** è possibile effettuare una ricerca avanzata anche solo inserendo uno dei parametri in input, con il vincolo che se TipoProfilo non è “Professionista” non sarà possibile inserire Professione o Zona.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| TipoProfilo | Semplice | Basso | Input | **Enumerativo:** Professionista o standard |
| Specie | Semplice | Basso | Input | Non più di 64 caratteri |
| Professione | Semplice | Basso | Input | Non più di 64 caratteri |
| Risultati | Semplice | Basso | Output | Set di Username |
| Zona | Semplice | Basso | Input | Selezionabile da una lista di province |

**Registrazione: tabella informazioni/flusso**

Caso relativo al Professionista

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Molto Alto | Input | Non più di 32 caratteri |
| Nome | Semplice | Medio | Input | Non più di 64 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 64 caratteri |
| Documento | Semplice | Molto Alto | Input | Dimensione massima 4MB |
| Professione | Semplice | Basso | Input | Non più di 64 caratteri |
| Specializzazione | Semplice | Basso | Input | Non più di 64 caratteri |
| Telefono | Semplice | Basso | Input | Esattamente 10 caratteri |
| Zona | Semplice | Basso | Input | Non più di 128 caratteri |

Caso relativo all’ utente Standard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Molto Alto | Input | Non più di 32 caratteri |
| Nome | Semplice | Medio | Input | Non più di 64 caratteri |
| Cognome | Semplice | Medio | Input | Non più di 64 caratteri |
| Telefono | Semplice | Basso | Input | Esattamente 10 caratteri (facoltativo) |

**Login: tabella informazioni/flusso**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |
| Password | Semplice | Molto Alto | Input | Non più di 32 caratteri |

**ResetProPost: tabella informazioni/flusso**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Informazione** | **Tipo** | **Livello di protezione/privacy** | **Input/Output** | **Vincoli** |
| Username | Semplice | Basso | Input | Non più di 32 caratteri |

# Analisi dei vincoli - Tabella dei vincoli

|  |  |  |  |
| --- | --- | --- | --- |
| **Requisito** | **Categorie** | **Impatto** | **Funzionalità** |
| Velocità di ricerca profili | Tempo di risposta | Cercare di migliorare | Ricerca di altri profili |
| Velocità di navigazione dell’applicazione | Tempo di risposta | Cercare di migliorare | GestioneUtente, GestioneProfessionista |
| Interfaccia intuitiva | Usabilità | Cercare di migliorare | GestioneUtente, GestioneProfessionista |
| Creazione dei contenuti intuitiva | Usabilità | Cercare di migliorare | GestioneUtente, GestioneProfessionista |
| Backup dei dati periodici | Sicurezza, tempo di risposta | Peggiorano i tempi di risposta, migliorano la privacy e la persistenza dei dati | GestioneUtente, GestioneProfessionista,  GestioneAmministratore, GestioneAssistenza |
| Portabilità | Usabilità | Cercare di migliorare | GestioneUtente, GestioneProfessionista |
| I dati sensibili dell’account devono essere protetti per mezzo di meccanismi di cifratura | Sicurezza, tempo di risposta | Peggiorano i tempi di risposta, migliora la privacy dei dati | GestioneUtente, GestioneProfessionista,  GestioneAmministratore, GestioneAssistenza |

# Analisi delle interazioni – Tabella Maschere

|  |  |  |
| --- | --- | --- |
| Maschera | Informazioni | Funzionalità |
| View Registrazione | Nome, Cognome, Username, Password, tipo di account (standard/professionista), documento (solo se professionista), numero di telefono (obbligatorio se professionista), zona (solo se professionista), professione (solo se professionista), specializzazione (solo se professionista) | GestioneUtente, GestioneProfessionista |
| View Login | Username, Password | GestioneUtente, GestioneProfessionista,  GestioneAmministratore,  GestioneAssistenza |
| Home GestioneUtente | Lista dei post delle persone seguite e ProPost suggeriti, Possibilità di mettere like. Navigazione verso View Post,  Navigazione verso View Notifiche, navigazione verso View Ricerca, navigazione verso View Profilo Principale, Navigazione verso View Creazione Post,  Navigazione verso View Profilo del creatore di un Post | GestioneUtente,  GestioneProfessionista |
| View Notifiche | Informazioni relative a commenti o like a propri post e nuovi follower.  Navigazione verso Home GestioneUtente, Navigazione verso View Profilo, Navigazione verso View Post. | GestioneUtente,  GestioneProfessionista |
| View Creazione Post | Animale/i, contenuto multimediale, descrizione (non necessaria).  Navigazione verso Home GestioneUtente.  In caso di professionista, sarà disponibile la navigazione verso View CreazioneProPost | GestioneUtente,  GestioneProfessionista |
| View Creazione ProPost | Contenuto multimediale, descrizione (non necessaria).  Navigazione verso Home GestioneUtente.  Navigazione verso View Creazione Post. | GestioneProfessionista |
| View Profilo | Informazioni relative al profilo, lista di animali domestici registrati, Navigazione verso View Profilo Animale,  Possibilità di seguire/unfollow del profilo | GestioneUtente,  GestioneProfessionista |
| View Profilo Animale | Informazioni relative al profilo dell’animale, lista di Post dell’animale.  Navigazione verso View Profilo del proprietario.  Possibilità di mettere like ai post e di commentare i post dell’animale | GestioneUtente,  GestioneProfessionista |
| View Post | Mette in primo piano il post selezionato mostrando la lista dei commenti.  Possibilità di aggiungere un commento, di mettere mi piace al post, di segnalare il post, di segnalare un commento, Navigazione verso la View Segnalazione, Navigazione verso la View Home GestioneUtente | GestioneUtente,  GestioneProfessionista |
| View Segnalazione | Input per la specifica del motivo della segnalazione | GestioneUtente,  GestioneProfessionista |
| View Ricerca | Username per la ricerca, lista di eventuali risultati, navigazione verso View Profilo (dei profili risultanti), navigazione verso la View Ricerca Avanzata | RicercaDiAltriProfili |
| View Ricerca Avanzata | Possibilità di inserire come vincoli di ricerca: username, tipo profilo, specie, professione, zona.  Lista di eventuali risultati, navigazione verso View Profilo (dei profili risultanti), navigazione verso la View Ricerca | RicercaDiAltriProfili |
| View Profilo Personale | Informazioni relative al profilo, Navigazione verso View Modifica Profilo, Navigazione verso View Aggiungi Animale, lista di animali domestici registrati, Navigazione verso View Profilo Animale Personale, Navigazione View Notifiche | GestioneUtente |
| View Profilo Personale Professionista | Informazioni relative al profilo, Navigazione verso View Modifica Profilo Professionista, Navigazione verso View Aggiungi Animale, lista di animali domestici registrati o Propost pubblicati, Navigazione verso View Profilo Animale Personale, Navigazione View Notifiche. | Gestione Professionista |
| View Modifica Profilo | Possibilità di modificare: nome, cognome, descrizione, immagine profilo, telefono.  Navigazione verso la View Profilo Personale | GestioneUtente |
| View Modifica Profilo Professionista | Possibilità di modificare: descrizione, immagine profilo, zona, telefono.  Navigazione verso la View Profilo Personale | GestioneProfessionista |
| View Aggiungi Animale | Possibilità di inserire: nome, razza, specie, sesso, descrizione.  Navigazione verso la View Profilo Personale | GestioneAnimali |
| View Profilo Animale Personale | Informazioni relative al profilo dell’animale, Navigazione verso View Modifica Animale, lista di Post dell’animale,  possibilità di eliminare i Post.  Navigazione verso View Profilo Personale. | GestioneAnimali |
| View Modifica Animale | Possibilità di modificare: nome, descrizione, immagine profilo.  Navigazione verso la View Profilo Animale.  Possibilità di eliminare il profilo dell’animale | GestioneAnimali |
| Home GestioneAssistenza | Lista degli utenti bloccati, Lista segnalazione, possibilità di Navigazione verso View Accettazione Professionista, View Gestione Sban, View Gestione Ban e View Rimozione Commento/Post | GestioneAssistenza |
| View Accettazione Professionista | Username, Nome Professionista, Cognome Professionista, Documento d’identità | GestioneAssistenza |
| View Gestione Ban | Username Utente | GestioneAssistenza |
| View Gestione Sban | Username Utente | GestioneAssistenza |
| View Rimozione Commento/Post | IdCommento/IdPost | GestioneAssistenza |
| Home GestioneAmministratore | Lista account assistenza creati, Navigazione verso View Crea Assistenza, View Rimuovi Assistenza, View Visualizzazione Log, Home GestioneAssistenza | GestioneAmministratore |
| View Crea Assistenza | Username, Password | GestioneAmministratore |
| View Rimuovi Assistenza | Username | GestioneAmministratore |
| View Visualizzazione Log | Viene visualizzata la Lista dei Log | VisualizzaLog |

# Analisi Ruoli e responsabilità – Tabella Ruoli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ruolo | Responsabilità | Maschere | Riservatezza | Numerosità |
| Utente Standard | Garantire la privacy della password impostata | View Registrazione,  View Login,  Home GestioneUtente,  View CreazionePostUtente,  View Ricerca, View RicercaAvanzata,  View Segnalazione,  View Post,  View ProfiloPersonale,  View Profilo,  View ModificaProfilo,  View Aggiungi animale,  View ProfiloAnimalePersonale,  View ModificaAnimale,  View | È richiesto un basso/medio grado di riservatezza | Il numero può essere gestito da un amministratore di sistema, al più e limitato dalle risorse del sistema |
| Utente Professionista | Garantire la privacy della password impostata | View Registrazione,  View Login,  Home GestioneUtente,  View CreazionePost,  View CreazioneProPost,  View Ricerca, View RicercaAvanzata,  View Segnalazione,  View Post,  View ProfiloPersonaleProfessionista,  View Profilo,  View ModificaProfiloProfessionista,  View Aggiungi animale,  View ProfiloAnimalePersonale,  View ModificaAnimale. | È richiesto un medio grado di riservatezza | Il numero può essere gestito da un amministratore di sistema, al più e limitato dalle risorse del sistema |
| Assistenza | Visualizzazione di tutti gli Utenti e dei loro commenti | View Registrazione,  View Login, View Accettazione Professionista, View Gestione Ban, View Gestione Sban, View Rimozione Commento/Post | È richiesto un alto grado di riservatezza | Un team costituito da un numero considerevole di persone scelte dall’Amministratore |
| Amministratore | Gestione di tutte le informazioni relative agli utenti, agli assistenti ed al sistema | View Registrazione,  View Login, View Accettazione Professionista, View Gestione Ban, View Gestione Sban, View Rimozione Commento/Post,  Home GestioneAmministratore,  View ControlloAssistenza, View VisualizzaLog | È richiesto un alto grado di riservatezza | Un’unica persona |

**Utente Standard: tabella ruolo-informazioni**

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Nome Utente | Lettura/scrittura |
| Cognome Utente | Lettura/scrittura |
| Username | Lettura/scrittura |
| Password | Scrittura |
| Descrizione | Lettura/scrittura |
| Immagine | Lettura/scrittura |
| Telefono | Lettura/scrittura |
| Post | Lettura/scrittura |
| ProPost | Lettura |
| Commento | Lettura/scrittura |
| Messaggio Segnalazione | Scrittura |
| Time Stamp Segnalazione | Scrittura |
| Like | Lettura/scrittura |
| Animali | Lettura/scrittura |

**Utente Professionista: tabella ruolo-informazioni**

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Nome Utente | Lettura/scrittura |
| Cognome Utente | Lettura/scrittura |
| Username | Lettura/scrittura |
| Password | Scrittura |
| Descrizione | Lettura/scrittura |
| Immagine | Lettura/scrittura |
| Telefono | Lettura/scrittura |
| Post | Lettura/scrittura |
| ProPost | Lettura/Scrittura |
| Commento | Lettura/scrittura |
| Messaggio Segnalazione | Scrittura |
| Time Stamp Segnalazione | Scrittura |
| Like | Lettura/scrittura |
| Animali | Lettura/scrittura |
| Documento | Scrittura |
| Zona | Lettura/Scrittura |
| Specializzazione | Lettura/Scrittura |
| Professione | Lettura/Scrittura |

**Assistenza: tabella ruolo-informazioni**

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Username | Lettura/scrittura |
| Username Utente | Lettura |
| Nome Utente | Lettura |
| Cognome Utente | Lettura |
| Password | Scrittura |
| Post | Lettura |
| ProPost | Lettura |
| Commento | Lettura |
| Messaggio Segnalazione | Lettura |
| Time Stamp Segnalazione | Lettura |
| Documento Professionista | Lettura |
| Animale | Lettura |

**Amministrazione: tabella ruolo-informazioni**

|  |  |
| --- | --- |
| Informazione | Tipo di Accesso |
| Nome Utente | Lettura |
| Cognome Utente | Lettura |
| Username Amministratore | Lettura/scrittura |
| Password Amministratore | Scrittura |
| Username Assistente | Lettura/scrittura |
| Password Assistente | Scrittura |
| Post | Lettura |
| ProPost | Lettura |
| Commento | Lettura |
| Messaggio Segnalazione | Lettura |
| Time Stamp Segnalazione | Lettura |
| Voce Di Log | Lettura |

# Scomposizione del problema

**Tabella Scomposizione Funzionalità**

|  |  |
| --- | --- |
| Funzionalità | Scomposizione |
| GestioneUtente | GestioneNotifiche,  GestioneContenuti,  RicercaDiAltriProfili,  GestioneAnimali,  SeguiUtente,  UnfollowUtente,  ModificaProfilo |
| GestioneContenuti | CondivisionePost,  EliminazionePost,  CreazioneCommento,  EliminazioneCommento,  SegnalarePost,  SegnalareCommenti,  ApprezzareContenuti |
| GestioneAnimali | AggiungereAnimale,  TogliereAnimale,  ModificaProfiloAnimale |
| RicercaDiAltriProfili | RicercaSemplice,  RicercaAvanzata |
| GestioneAmministratore | CreazioneAcccountAssistenza,  RimozioneAcccountAssistenza |
| GestioneAssistenza | AccettazioneProfessionista,  BloccoUtente,  SbloccoUtente,  RimozioneCommento,  EliminazioneContenuto |

**Tabella Sotto-Funzionalità: GestioneUtente**

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-Funzionalità | Sotto-Funzionalità | Legame | Informazioni |
| UnfollowUtente | SeguiUtente | Non si può smettere di seguire un utente se non lo si è seguito in precedenza | Username Utente |

**Tabella Sotto-Funzionalità: GestioneContenuti**

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-Funzionalità | Sotto-Funzionalità | Legame | Informazioni |
| EliminazionePost | CondivisionePost | Non si può eliminare un post che non è stato condiviso | Username Utente, IdPost |
| EliminazioneCommento | CreazioneCommento | Non si può eliminare un commento che non è stato condiviso | Username Utente, IdCommento |
| SegnalarePost | CondivisionePost | Non si può segnalare un post che non è stato condiviso | Username Utente, IdPost |
| SegnalareCommento | CreazioneCommento | Non si può segnalare un commento che non è stato condiviso | Username Utente, IdCommento |

**Tabella Sotto-Funzionalità: GestioneAnimali**

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-Funzionalità | Sotto-Funzionalità | Legame | Informazioni |
| TogliereAnimale | AggiungereAnimale | Non si può rimuovere un animale che non è stato registrato | Username Utente, IdAnimale |

**Tabella Sotto-Funzionalità: GestioneAssistenza**

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-Funzionalità | Sotto-Funzionalità | Legame | Informazioni |
| BloccoUtente | SbloccoUtente | Non si può bloccare un utente che non è stato bloccato | Username Utente |

**Tabella Sotto-Funzionalità: GestioneAssistenza**

|  |  |  |  |
| --- | --- | --- | --- |
| Sotto-Funzionalità | Sotto-Funzionalità | Legame | Informazioni |
| RimozioneAccountAssistenza | CreazioneAccountAssistenza | Non si può rimuovere un assistente che non è stato registrato | Username Utente |

# Modello del dominio

Di seguito è riportato il diagramma delle classi del modello in relazione agli account.

Nel diagramma è presente una relazione di **overlapped** tra tutte le classi che ereditano da UtenteStandard: quindi tra la classe Professionista e le classi UtenteAttivo e UtenteBloccato.

Immagine che contiene schermata, testo, design

Descrizione generata automaticamente

Di seguito è riportato il diagramma delle classi del modello in relazione alla condivisione dei Post.

Immagine che contiene schermata, design

Descrizione generata automaticamente

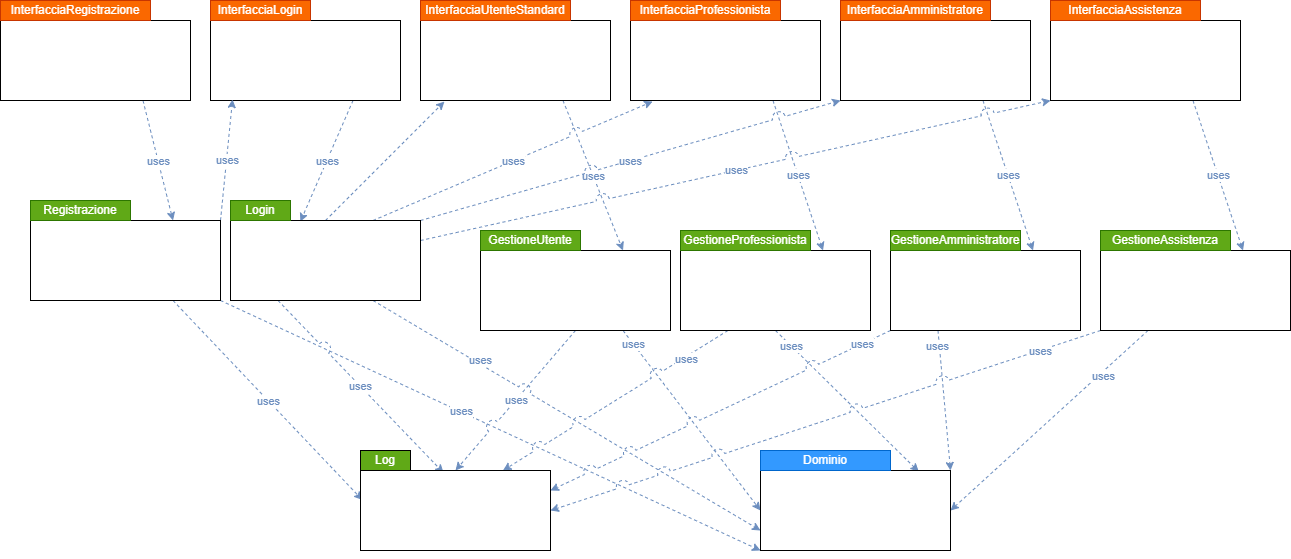
Di seguito è presente il diagramma delle classi del modello del dominio in relazione alla gestione dei Log.

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

# Architettura Logica

**Struttura**

*Diagramma dei package*

*Diagramma delle classi: Login*

*Immagine che contiene testo, schermata, design

Descrizione generata automaticamente*

*Diagramma delle classi: Registrazione*

*Immagine che contiene testo, schermata, design

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, design

Descrizione generata automaticamenteDiagramma delle classi: Amministratore/Assistenza*

*Immagine che contiene testo, diagramma, Piano, Parallelo

Descrizione generata automaticamenteDiagramma delle classi: Utente / Professionista*

*Immagine che contiene testo, diagramma, schermata, numero

Descrizione generata automaticamenteDiagramma di sequenza: Registrazione*

*Immagine che contiene testo, schermata, diagramma, Parallelo

Descrizione generata automaticamenteDiagramma di sequenza: Login*

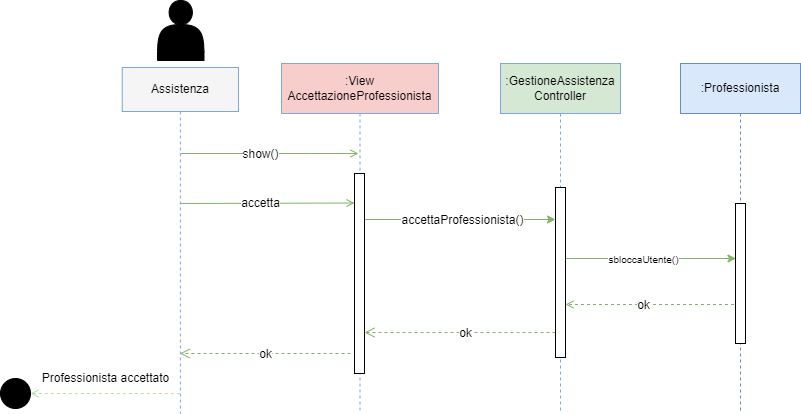
*Immagine che contiene testo, diagramma, schermata, linea

Descrizione generata automaticamenteDiagramma di sequenza: Creazione Account Assistenza*

Immagine che contiene testo, diagramma, schermata, design

Descrizione generata automaticamente

*Diagramma di sequenza: Accettazione Professionista*



*Diagramma di sequenza: Ricerca profili*

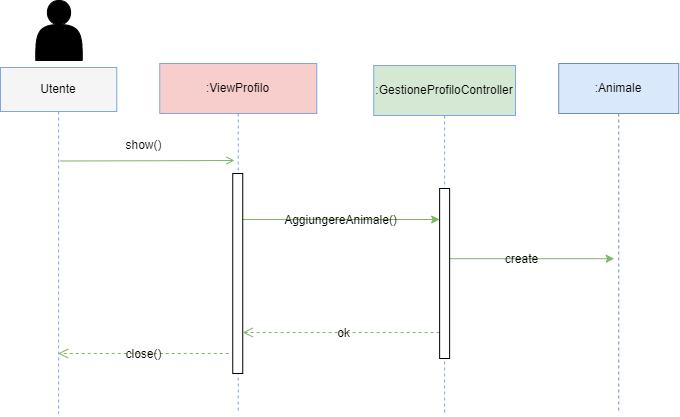
Immagine che contiene testo, diagramma, schermata, linea

Descrizione generata automaticamente

*Diagramma di sequenza: caricamento HomeUtente*

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente

*Diagramma di sequenza: Creazione ProfiloAnimale*

# Comportamento

*Diagramma di Stato: Utente Diagramma di Stato: Professionista*

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente**Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente**

# Piano di Lavoro

Il progetto e lo sviluppo del sistema sono assegnati a diversi team come indicato nella tabella

sottostante:

|  |  |
| --- | --- |
| **Nome Team** | **Composizione** |
| Team progettazione | Ianniciello, Provolo, Sampaoli |
| Team sviluppo | Ianniciello, Provolo, Sampaoli |
| Team DB | Ianniciello, Provolo, Sampaoli |
| Team sicurezza | Ianniciello, Provolo, Sampaoli |
| Team grafico | Ianniciello, Provolo, Sampaoli |

|  |  |  |
| --- | --- | --- |
| **Package** | **Progetto** | **Sviluppo** |
| Dominio | Team progettazione + Team DB | Team sviluppo + Team DB |
| GestioneUtente | Team progettazione | Team sviluppo |
| GestioneProfessionista | Team progettazione | Team sviluppo |
| GestioneAssistenza | Team progettazione | Team sviluppo |
| GestioneAmministratore | Team progettazione | Team sviluppo |
| Registrazione | Team sicurezza | Team sviluppo + Team sicurezza |
| Login | Team sicurezza | Team sviluppo + Team sicurezza |
| Log | Team sicurezza | Team sviluppo + Team sicurezza |
| InterfacciaRegistrazione | Team progettazione + Team grafico | Team grafico + Team sicurezza |
| interfacciaLogin | Team progettazione + Team grafico | Team grafico + Team sicurezza |
| interfacciaLog | Team progettazione + Team grafico | Team grafico + Team sicurezza |
| interfacciaUtente | Team progettazione + Team grafico | Team sviluppo + Team grafico |
| interfacciaProfessionista | Team progettazione + Team grafico | Team sviluppo + Team grafico |
| interfacciaAssistenza | Team progettazione + Team grafico | Team sviluppo + Team grafico |
| interfacciaAmministratore | Team progettazione + Team grafico | Team sviluppo + Team grafico |

I tempi di rilascio previsti sono i seguenti:

* Progettazione conclusa il 18/06/2023
* Sviluppo del prototipo entro due settimane dalla data di consegna della documentazione relativa alla progettazione
* Sviluppo delle singole parti con collaudo unitario entro una settimana rispetto alla fine della progettazione
* Integrazione e test dell’intero sistema entro una settimana rispetto alla fine dello sviluppo

# Sviluppi Futuri

* Possibilità di visualizzare all’interno del proprio profilo gli utenti seguiti e gli utenti seguaci
* Possibilità di reagire tramite like ai commenti sotto un Post

# Piano del Collaudo

Per garantire il corretto funzionamento del sistema sono necessari una gamma di test unitari e di integrazione che permettano la correttezza delle singole parti. Di seguito vengono riportati i casi più di interesse.

package testDominio;

import static org.junit.Assert.\*;

import java.sql.\*;

import java.util.\*;

import javax.sql.rowset.serial.SerialBlob;

import org.junit.\*;

import dominio.\*;

public class TestUtente

{

    private UtenteStandard u;

    private ProfiloUtente p;

    byte[] stream;

    Blob immagine;

    List<Animale>as;

    List<UtenteStandard>followers;

    List<UtenteStandard>followed;

    @Before

    public void setup() throws SQLException

    {

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        as=new ArrayList<Animale>();

        followers=new ArrayList<UtenteStandard>();

        followed=new ArrayList<UtenteStandard>();

        p=new ProfiloUtente("gianni","ilPiùCantato",immagine,"Morandi","3420476687",as);

        u=new UtenteStandard("gianni67",p,false,followers,followed);

    }

    @Test

    public void TestGetterUtenteStandard()

    {

        assertEquals(u.getProfilo(),p);

        assertEquals(u.getUsername(),"gianni67");

        assertFalse(u.isBloccato());

        assertEquals(u.getFollowers(),followers);

        assertEquals(u.getFollowed(),followed);

    }

    @Test

    public void TestSetterUtenteStandard()

    {

        u=new UtenteStandard();

        ProfiloUtente profilo =new ProfiloUtente("gianni","ilPiùCantato",immagine,"Morandi","3420476687",as);

        u.setProfilo(profilo);

        assertEquals(u.getProfilo(),profilo);

        u.setUsername("gianni");

        assertEquals(u.getUsername(),"gianni");

        u.setBloccato(true);

        assertTrue(u.isBloccato());

        u.setFollowers(followers);

        u.setFollowed(followed);

    }

}

package testDominio;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertFalse;

import static org.junit.Assert.assertTrue;

import java.sql.Blob;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import javax.sql.rowset.serial.SerialBlob;

import org.junit.Before;

import org.junit.Test;

import dominio.Animale;

import dominio.ProPost;

import dominio.Professionista;

import dominio.ProfiloProfessionista;

import dominio.UtenteStandard;

public class TestProfessionista

{

    private Professionista pro;

    private ProfiloProfessionista p;

    byte[] stream;

    Blob immagine;

    List<Animale>as;

    List<UtenteStandard>followers;

    List<UtenteStandard>followed;

    List<ProPost> proposts;

    @Before

    public void setup() throws SQLException

    {

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        as=new ArrayList<Animale>();

        followers=new ArrayList<UtenteStandard>();

        followed=new ArrayList<UtenteStandard>();

        proposts = new ArrayList<ProPost>();

        p=new ProfiloProfessionista("Carlo","gran veterinario, forte proprio",immagine,"crociani","Bologna","3420476687",as,proposts);

        pro=new Professionista("vet1",p,false,"veterinario","felini",true,followers,followed);

    }

    @Test

    public void TestGetterProfessionista()

    {

        assertEquals(pro.getProfilo(),p);

        assertEquals(pro.getUsername(),"vet1");

        assertFalse(pro.isBloccato());

        assertFalse(pro.isAbilitato());

        assertEquals(pro.getProfessione(),"veterinario");

        assertEquals(pro.getSpecializzazione(),"felini");

        assertEquals(pro.getFollowers(),followers);

        assertEquals(pro.getFollowed(),followed);

    }

    @Test

    public void TestSetterProfessionista()

    {

        pro=new Professionista();

        ProfiloProfessionista profilo =new ProfiloProfessionista("Carlo","gran veterinario, forte proprio",immagine,"crociani","Bologna","3420476687",as,proposts);

        pro.setProfilo(profilo);

        assertEquals(pro.getProfilo(),profilo);

        pro.setUsername("vet1");

        assertEquals(pro.getUsername(),"vet1");

        pro.setBloccato(true);

        assertTrue(pro.isBloccato());

        pro.setAbilitato(true);

        assertTrue(pro.isAbilitato());

        pro.setProfessione("veterinario");

        assertEquals(pro.getProfessione(),"veterinario");

        pro.setSpecializzazione("felini");

        assertEquals(pro.getSpecializzazione(),"felini");

        pro.setFollowers(followers);

        pro.setFollowed(followed);

        assertEquals(pro.getFollowers(),followers);

        assertEquals(pro.getFollowed(),followed);

    }

package testDominio;

import static org.junit.Assert.\*;

import java.sql.Blob;

import java.sql.Date;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import java.util.Set;

import java.util.TreeSet;

import org.junit.\*;

import dominio.Animale;

import dominio.Commento;

import dominio.Like;

import dominio.Post;

import dominio.PostAnimale;

import dominio.ProPost;

public class TestPost

{

    private Date dataCreazione;

    private Blob media;

    private List<Commento> commenti;

    private Set<Like>likes;

    private Set <Animale> animali;

    ProPost p;

    PostAnimale pa;

    @Before

    public void setup() throws SQLException

    {

        dataCreazione= new Date(0);

        commenti= new ArrayList<Commento>();

        likes = new TreeSet<Like>();

        animali = new TreeSet<Animale>();

        p = new ProPost(dataCreazione, media, "gran bel proPost", commenti,likes );

        pa= new PostAnimale(dataCreazione, media, "gran bel postAnimali", commenti,likes,animali);

    }

    @Test

    public void TestGetterPost()

    {

        assertEquals(p.getCommenti(),commenti);

        assertEquals(p.getDataCreazione(),dataCreazione);

        assertEquals(p.getDescrizione(),"gran bel proPost");

        assertEquals(p.getLikes(),likes);

        assertEquals(p.getMedia(),media);

        assertEquals(pa.getCommenti(),commenti);

        assertEquals(pa.getDataCreazione(),dataCreazione);

        assertEquals(pa.getDescrizione(),"gran bel postAnimali");

        assertEquals(pa.getLikes(),likes);

        assertEquals(pa.getMedia(),media);

        assertEquals(pa.getAnimali(),animali);

    }

    @Test

    public void TestSetterPost()

    {

        p= new ProPost();

        pa=new PostAnimale();

        p.setCommenti(commenti);

        p.setDataCreazione(dataCreazione);

        p.setDescrizione("gran bel proPost");

        p.setLikes(likes);

        p.setMedia(media);

        pa.setCommenti(commenti);

        pa.setDataCreazione(dataCreazione);

        pa.setDescrizione("gran bel proPost");

        pa.setLikes(likes);

        pa.setMedia(media);

        pa.setAnimali(animali);

        assertEquals(p.getCommenti(),commenti);

        assertEquals(p.getDataCreazione(),dataCreazione);

        assertEquals(p.getDescrizione(),"gran bel proPost");

        assertEquals(p.getLikes(),likes);

        assertEquals(p.getMedia(),media);

        assertEquals(pa.getCommenti(),commenti);

        assertEquals(pa.getDataCreazione(),dataCreazione);

        assertEquals(pa.getDescrizione(),"gran bel postAnimali");

        assertEquals(pa.getLikes(),likes);

        assertEquals(pa.getMedia(),media);

        assertEquals(pa.getAnimali(),animali);

    }

}

package testDominio;

import static org.junit.Assert.assertEquals;

import java.sql.Blob;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import javax.sql.rowset.serial.SerialBlob;

import org.junit.Before;

import org.junit.Test;

import dominio.Animale;

import dominio.PostAnimale;

import dominio.Sesso;

public class TestAnimale

{

    private String razza;

    private String specie;

    private Sesso sex;

    byte[] stream;

    Blob immagine;

    private List<PostAnimale> posts=new ArrayList<PostAnimale>();

    Animale a;

    @Before

    public void setup() throws SQLException

    {

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        a=new Animale("Lugia", "ma quanto è forte?",immagine, "uccello","leggendario",Sesso.Maschio,posts);

    }

    @Test

    public void TestGetterUtenteStandard()

    {

        assertEquals(a.getDescrizione(),"ma quanto è forte?");

        assertEquals(a.getImage(),immagine);

        assertEquals(a.getNome(),"Lugia");

        assertEquals(a.getPosts(),posts);

        assertEquals(a.getRazza(),"uccello");

        assertEquals(a.getSex(),Sesso.Maschio);

        assertEquals(a.getSpecie(),"leggendario");

    }

    @Test

    public void TestSetterUtenteStandard()

    {

        a=new Animale();

        a.setDescrizione("ma quanto è forte?");

        a.setImage(immagine);

        a.setNome("Lugia");

        a.setPosts(posts);

        a.setRazza("uccello");

        a.setSex(Sesso.Femmina);

        a.setSpecie("leggendario");

        assertEquals(a.getDescrizione(),"ma quanto è forte?");

        assertEquals(a.getImage(),immagine);

        assertEquals(a.getNome(),"Lugia");

        assertEquals(a.getPosts(),posts);

        assertEquals(a.getRazza(),"uccello");

        assertEquals(a.getSex(),Sesso.Femmina);

        assertEquals(a.getSpecie(),"leggendario");

    }

}

# Progettazione

**Progettazione Architetturale**

**Requisiti non funzionali**

Nell’Analisi del Problema (Tabella Vincoli) sono emersi i seguenti requisiti non funzionali che impongono dei vincoli al sistema:

* 1. Semplicità di navigabilità dell’applicazione
  2. Protezione dei dati sensibili degli utenti per mezzo di meccanismi di cifratura
  3. Periodico backup dei dati
  4. Controllo degli accessi

Uno degli aspetti chiave dell’applicazione è l’usabilità: la piattaforma deve essere intuitiva e semplice da utilizzare per soddisfare i requisiti e le aspettative di un’ampia clientela. Allo stesso tempo si vuole che i dati degli utenti vengano tutelati sia in fase di accesso che in fase di registrazione per cui è fondamentale trovare il giusto compromesso tra semplicità e sicurezza dei dati. Ragion per cui si è deciso di utilizzare il protocollo HTTPS a livello applicativo. Il controllo degli accessi permette all’assistenza e all’amministratore di monitorare la piattaforma al fine di garantirne il lecito utilizzo. In particolare, permette ai medesimi di sfruttare i loro privilegi per far sì che chiunque usufruisca dell’applicazione lo possa fare in modo sicuro.

**Scelta dell’architettura**

Dal punto di vista architetturale, la scelta più adatta è risultata essere l’architettura client/server a tre livelli rappresentati da database, application server e client.

**L1 – Client**

Il Client avrà modo di connettersi al server tramite una connessione TLS, il che permetterà di usufruire delle funzionalità presenti nel server mantenendo un livello di sicurezza adeguato.

**L2 – Server**

Per quanto riguarda il lato Server è stato ritenuto opportuno avere più server specializzati in funzionalità diverse:

* + Un server per la gestione dei Log
  + Server per le funzionalità dell’Amministratore e dell’Assistenza
  + Server per le funzionalità dell’Utente Standard e dell’Utente Professionista
  + Un server per la gestione dei Login e della Registrazione

**L3 – Database**

Al fine di mantenere i dati persistenti si è scelto l’utilizzo di un server dedicato nel quale sarà installato un opportuno DBMS che gestirà la base dati della piattaforma Pet Party (il cui schema di dati si può trovare nelle pagine successive) ed un ulteriore server sarà utilizzato per la memorizzazione dei log. La separazione tra log e database è fondamentale in modo da evitare che eventuali compromissioni di una macchina alterino anche l’altra. È sempre necessario essere in una condizione che permetta di riportare i dati del sistema ad uno stato consistente ed integro. La comunicazione tra l’applicazione e il server DBMS utilizzerà metodi CRUD implementati tramiti API “hibernate”.

In aggiunta si è ritenuto utile il **Pattern Broker** per gestire la sessione e per aggiungere, in questo, astrazione e protezione. L’utilizzo del broker è particolarmente utile a disaccoppiare il client dai server poiché nasconde questi ultimi. In questo modo risulterà più semplice aggiungere o rimuovere server. Infine, per garantire la sicurezza nella comunicazione tra client e server si è deciso di utilizzare il protocollo TLS che, essendo un protocollo crittografico, permette una comunicazione sicura dalla sorgente al destinatario (end-to-end) su reti TCP/IP fornendo autenticazione, integrità e protezione dei dati, operando al di sopra del livello di trasporto. Si è deciso di utilizzare il pattern MVP per separare i componenti dall’applicazione.

# Diagramma dei Package

Nella figura sottostante è riportata l’Architettura del Sistema organizzata attraverso un diagramma dei package.

**Immagine che contiene schermata, Policromia, testo, design

Descrizione generata automaticamente**

# Diagramma dei componenti

Immagine che contiene schermata, testo, diagramma, linea

Descrizione generata automaticamenteNella figura sottostante è riportata l’Architettura del Sistema organizzata attraverso un diagramma dei componenti.

**Scelte tecnologiche**

Poiché si lavora in Java con diversi tipi di file: JavaServer Pages, servlet, classi Java, XML ecc. si è deciso di utilizzare per il Deployment l’archivio con estensione **.war**.

# Progettazione di dettaglio

Nel seguito si riportano i diagrammi di dettaglio delle varie parti del Sistema

**Struttura**

*Diagramma di Dettaglio: Dominio – Account*

***Immagine che contiene schermata, testo, design

Descrizione generata automaticamente***

***Immagine che contiene schermata, nero, design

Descrizione generata automaticamente****Diagramma di Dettaglio: Dominio – Post*

*Diagramma di Dettaglio: Dominio – Log*

***Immagine che contiene schermata, testo, design

Descrizione generata automaticamente***

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamenteDiagramma di Dettaglio: Interfacce*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamenteImmagine che contiene testo, schermata

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente*

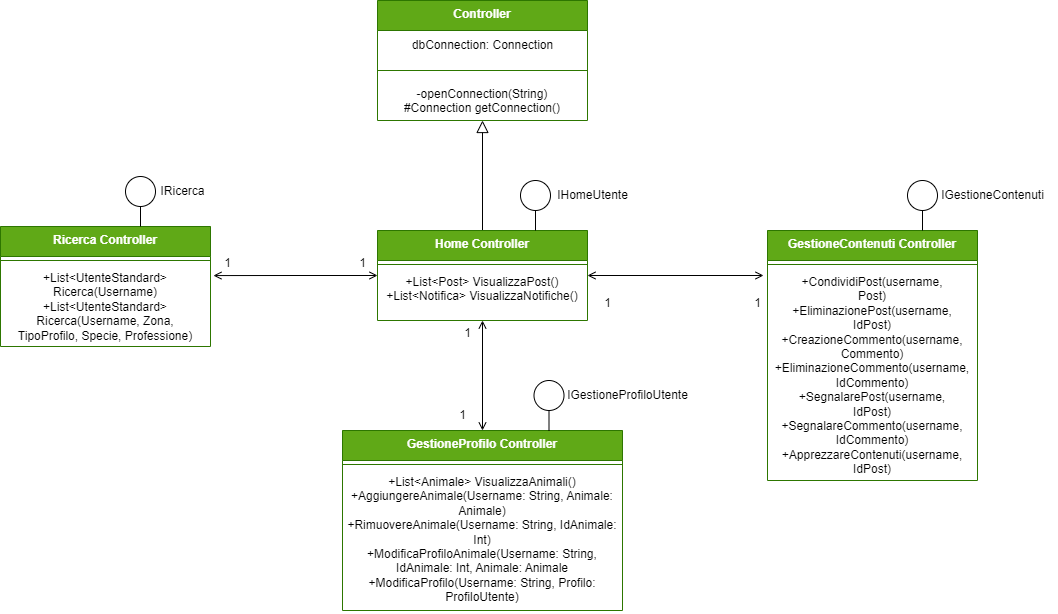
*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamente*

*Immagine che contiene testo, schermata, Carattere, verde

Descrizione generata automaticamente*

*Diagramma di dettaglio: GestioneUtente*

**

*Diagramma di dettaglio: GestioneProfessionista*

*Immagine che contiene schermata, testo

Descrizione generata automaticamente*

*Diagramma di dettaglio: GestioneAmministratore*

*Immagine che contiene testo, schermata, luna

Descrizione generata automaticamente*

*Diagramma di dettaglio: GestioneAssistenza*

*Immagine che contiene testo, schermata, luna

Descrizione generata automaticamente*

*Diagramma di dettaglio: GestioneLog*

*Immagine che contiene schermata, verde, testo, Carattere

Descrizione generata automaticamente*

Si è fattorizzata nella classe Controller tutta la gestione della persistenza dei dati in modo da non dover replicare la scrittura e lettura su DB. Solo i controller di gestione ereditano da Controller, gli altri controller accedono ai servizi di Controller attraverso le associazioni bi-direzionali

Ogni controller utilizza il Logger per la scrittura dei Log, questa relazione con tutti i controller non è stata riportata nei diagrammi precedenti in quanto avrebbe reso gli stessi più pesanti.

*Diagramma di dettaglio: Broker*

*Immagine che contiene testo, schermata, biglietto da visita, Carattere

Descrizione generata automaticamente*

Qui si ha la struttura del Broker: l’entry point è **FiltroRichieste**, che manipola le richieste che arrivano dai client aggiungendo la sessione e verificando che la sessione sia attiva, smista poi le richieste al server corretto e restituisce le risposte ai Client.

*Diagramma di dettaglio: view rese disponibili agli utentiStandard/utentiProfessionisti*

**

*Diagramma di dettaglio: view rese disponibili all’Amministratore/Assistenti*

*Immagine che contiene testo, schermata, design

Descrizione generata automaticamente*

**Alcune interfacce**

*Pagina iniziale*

*Immagine che contiene schermata, cartone animato, mammifero, Cartoni animati

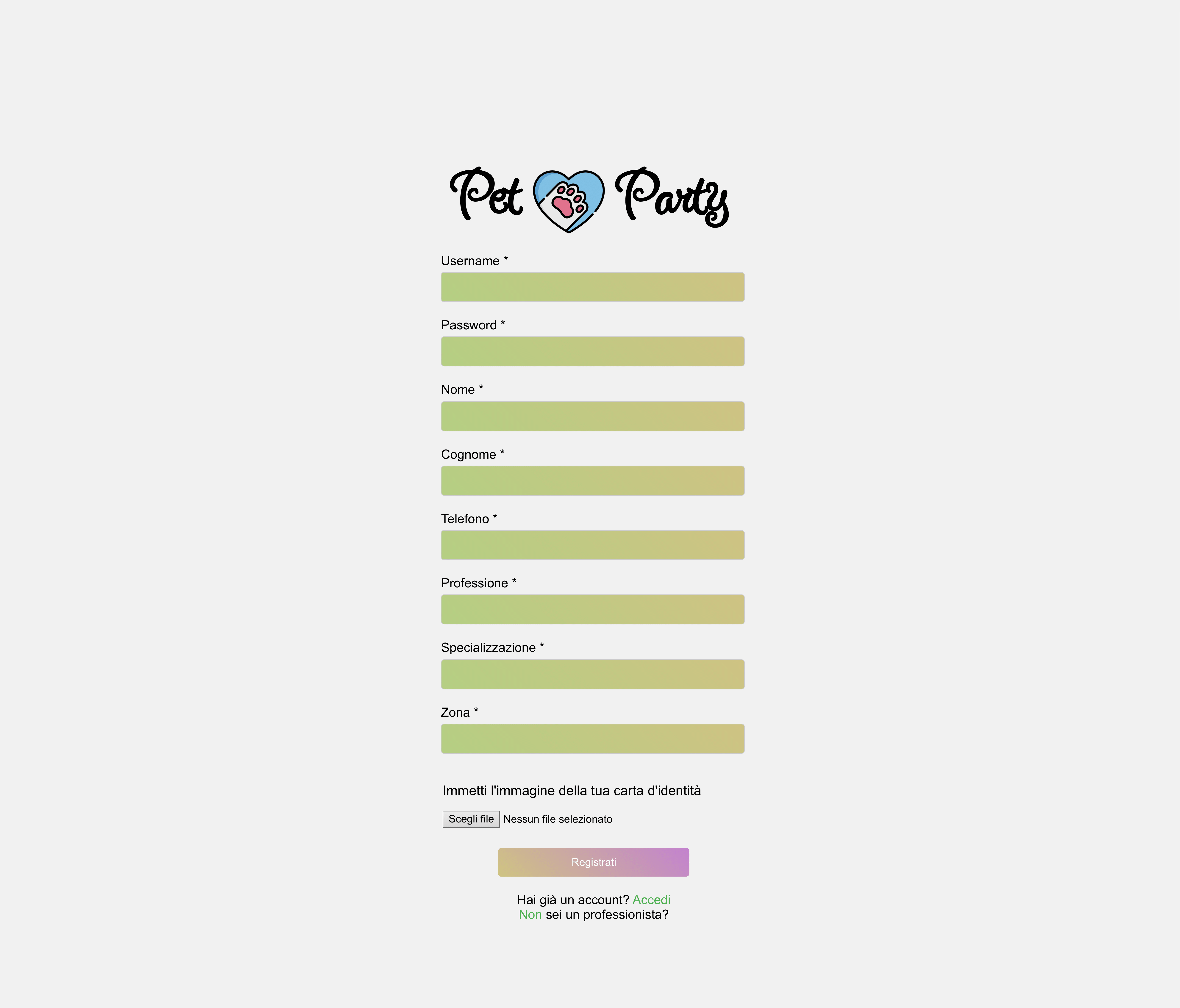
Descrizione generata automaticamente*

*Pagina registrazione*

*Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente*

*Pagina registrazione per professionisti*

**

*Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamentePagina Login*

# Interazione

*Diagramma di sequenza: Registrazione*

*Immagine che contiene schermata, quadrato

Descrizione generata automaticamente*

*Diagramma di sequenza: Login*

***Immagine che contiene schermata, testo, design

Descrizione generata automaticamente***

Immagine che contiene schermata, testo, software, Software multimediale

Descrizione generata automaticamenteDiagramma di sequenza: Caricamento Home

*Diagramma di sequenza: Accettazione Professionista*

Immagine che contiene schermata, testo, Software multimediale, software

Descrizione generata automaticamente

*Diagramma di sequenza: Creazione Assistenza*

Immagine che contiene schermata, testo, diagramma, Software multimediale

Descrizione generata automaticamente

*Diagramma di sequenza: Creazione Animale*

*Immagine che contiene testo, diagramma, schermata, Parallelo

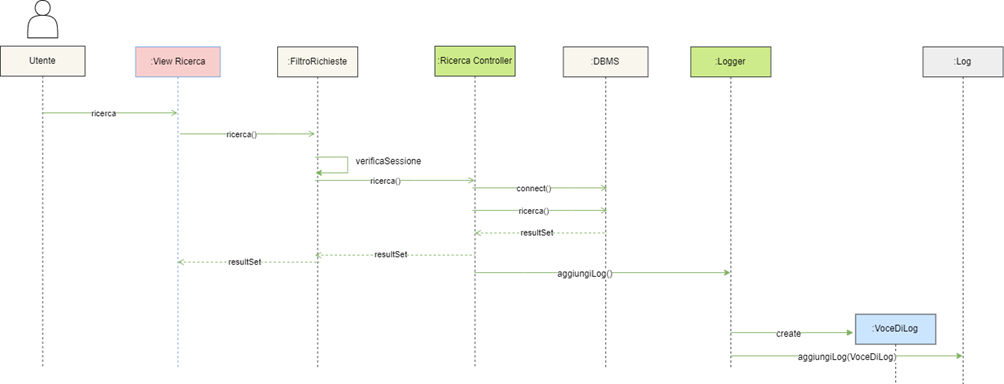
Descrizione generata automaticamente*

*Diagramma di sequenza: Visualizzazione Log*

*Immagine che contiene schermata, testo, design

Descrizione generata automaticamente*

*Diagramma di sequenza: Ricerca*



# Comportamento

*Diagramma di stato: Professionista*

*Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente*

Un professionista disabilitato sbloccato va inteso come un Professionista che si è appena registrato ma è in attesa di essere verificato. Se dopo il login il Professionista si trova nello stato Disabilitato Bloccato, verrà informato che la verifica delle sue credenziali è stata rifiutata ed il suo Account verrà eliminato.

# Progettazione della Persistenza

# Progettazione del collaudo

package testCollaudo;

import static org.junit.Assert.assertTrue;

import java.sql.Blob;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import javax.sql.rowset.serial.SerialBlob;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.junit.Before;

import org.junit.jupiter.api.Test;

import dominio.Animale;

import dominio.ProPost;

import dominio.Professionista;

import dominio.ProfiloProfessionista;

import dominio.ProfiloUtente;

import dominio.UtenteStandard;

import gestioneUtente.RicercaController;

import it.unibo.paw.hibernate.util.HibernateUtil;

public class TestRicercaUtenti

{

    private UtenteStandard u;

    private ProfiloUtente p;

    byte[] stream;

    Blob immagine;

    List<Animale>as;

    List<UtenteStandard>followers;

    List<UtenteStandard>followed;

    private Professionista pro;

    private ProfiloProfessionista profpro;

    List<ProPost> proposts;

    Session session = null;

    Transaction tx = null;

    @Before

    public void setup() throws SQLException

    {

        session = HibernateUtil.getSessionFactory().openSession();

        tx = session.beginTransaction();

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        as=new ArrayList<Animale>();

        followers=new ArrayList<UtenteStandard>();

        followed=new ArrayList<UtenteStandard>();

        p=new ProfiloUtente("gianni","ilPiùCantato",immagine,"Morandi","3420476687",as);

        u=new UtenteStandard("gianni67",p,false,followers,followed);

        session.persist(u);

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        as=new ArrayList<Animale>();

        followers=new ArrayList<UtenteStandard>();

        followed=new ArrayList<UtenteStandard>();

        proposts = new ArrayList<ProPost>();

        profpro=new ProfiloProfessionista("Carlo","gran veterinario, forte proprio",immagine,"crociani","Bologna","3420476687",as,proposts);

        pro=new Professionista("vet1",profpro,false,"veterinario","felini",true,followers,followed);

        session.persist(pro);

        tx.commit();

    }

    @Test

    public void testRicerca()

    {

        RicercaController rc = new RicercaController();

        assertTrue(rc.Ricerca("gianni67").contains(u));

    }

    public void testRicercaAvanzata()

    {

        RicercaController rc = new RicercaController();

        assertTrue(rc.Ricerca("vet1","Bologna","professionista","felini","veterinario").contains(pro));

    }

}

package testCollaudo;

import static org.junit.Assert.assertEquals;

import java.sql.Blob;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import javax.sql.rowset.serial.SerialBlob;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.junit.Before;

import org.junit.jupiter.api.Test;

import dominio.Animale;

import dominio.PostAnimale;

import dominio.Sesso;

import gestioneUtente.GestioneProfiloController;

import it.unibo.paw.hibernate.Course;

public class TestRegistraAnimale

{

    private String razza;

    private String specie;

    private Sesso sex;

    byte[] stream;

    Blob immagine;

    private List<PostAnimale> posts=new ArrayList<PostAnimale>();

    private int id=1;

    Animale a;

    Session session = null;

    Transaction tx = null;

    @Before

    public void setup() throws SQLException

    {

        stream=new byte[7];

        immagine = new SerialBlob(stream);

        a=new Animale(id,"Lugia", "ma quanto è forte?",immagine, "uccello","leggendario",Sesso.Maschio,posts);

    }

    @Test

    public void TestRegistraAnimale()

    {

        GestioneProfiloController gpc=new GestioneProfiloController();

        gpc.AggiungereAnimale("gianni67", a);

        Query query = session.createQuery("from " + Animale.class.getSimpleName() + " where id = ?");

        query.setInteger(0, id);

        Animale aRes = (Animale) query.uniqueResult();

        assertEquals(a,aRes);

    }

package testCollaudo;

import static org.junit.Assert.\*;

import static org.junit.jupiter.api.Assertions.assertEquals;

import java.sql.Blob;

import java.sql.Date;

import java.util.ArrayList;

import java.util.List;

import java.util.Set;

import java.util.TreeSet;

import javax.sql.rowset.serial.SerialBlob;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import dominio.Animale;

import dominio.Commento;

import dominio.Like;

import dominio.PostAnimale;

import dominio.ProPost;

import dominio.ProfiloUtente;

import dominio.UtenteStandard;

import gestioneUtente.GestioneContenutiController;

public class TestNewPost

{

    Session session = null;

    Transaction tx = null;

    private UtenteStandard u;

    private ProfiloUtente p;

    byte[] stream;

    Blob immagine;

    List<Animale>as;

    List<UtenteStandard>followers;

    List<UtenteStandard>followed;

    private Date dataCreazione;

    private Blob media;

    private List<Commento> commenti;

    private Set<Like>likes;

    private Set <Animale> animali;

    PostAnimale pa;

    String username;

    @BeforeAll

    public void setup()

    {

        // inserimento informazioni necessarie per la creazione di un post

        dataCreazione= new Date(0);

        commenti= new ArrayList<Commento>();

        likes = new TreeSet<Like>();

        animali = new TreeSet<Animale>();

        pa= new PostAnimale(dataCreazione, media, "gran bel postAnimali", commenti,likes,animali);

    }

    @Test

    public void testCaricamento()

    {

        //caricamento del post tramite controller apposito:

        GestioneContenutiController gcc=new GestioneContenutiController();

        gcc.condividiPost(username, pa);

        // --- Utente -----------------------------

        session = HibernateUtil.getSessionFactory().openSession();

        tx = session.beginTransaction();

        username="gianni67";

        Query query = session.createQuery("from "+ UtenteStandard.class.getSimpleName()+" where username = ?");

        query.setString(0,username);

        u = (UtenteStandard) query.uniqueResult();

        tx.commit();

        //verifica che il post caricato sia presente nel profilo dell'utente creatore

        List<Animale> animaliU = u.getProfilo().getAnimali();

        List<PostAnimale> feed=new ArrayList<PostAnimale>();

        for (Animale a : animaliU)

        {

            feed.addAll(a.getPosts());

        }

        assertTrue(feed.contains(pa));

    }

}

# Deployment

*Immagine che contiene testo, schermata, design

Descrizione generata automaticamente*Artefatti

Deployment Type-Level