

soscial

Kevin Cecchini - 0001071530
Lorenzo Furlani - 0000924119
Maninder Singh - 0001078688

Sommario

Abstract.....	3
Analisi dei requisiti.....	4
Raccolta dei Requisiti.....	4
Tabella dei requisiti.....	5
Analisi del dominio.....	7
Analisi dei requisiti.....	9
Analisi del rischio.....	16
Analisi del problema.....	31
Analisi delle Funzionalità.....	32
Scomposizione funzionalità.....	36
Analisi dei Vincoli.....	37
Analisi delle interazioni.....	38
Analisi dei ruoli e delle responsabilità.....	40
Modello del Dominio.....	41
Architettura Logica.....	43
Struttura.....	43
Interazione.....	47
Comportamento.....	51
Piano di Lavoro.....	51
Piano del Collaudo.....	52
Progettazione.....	58
Progettazione Architetturale.....	58
Requisiti non funzionali.....	58
Scelta dell'architettura.....	59
Scelte tecnologiche.....	59
Progettazione di Dettaglio.....	65
Struttura.....	65
Interazione.....	77
Progettazione della persistenza.....	82
Progettazione del Collaudo.....	82
Progettazione del deployment.....	87
Deployment del sistema.....	87

Abstract

Il progetto prevede lo sviluppo di un software dedicato alla segnalazione e alla diffusione di informazioni utili riguardanti la città di Bologna, come traffico, incidenti e altri eventi rilevanti.

Gli utenti possono accedere al servizio tramite un'apposita registrazione. Una volta iscritti, hanno la possibilità di segnalare problemi o eventi direttamente attraverso l'applicazione.

Il processo di aggiornamento dei dati avviene in tempo reale:

- Le segnalazioni inviate dagli utenti vengono inoltrate a un pannello di amministrazione.
- Un admin verifica la validità delle informazioni ricevute.
- Ogni segnalazione deve essere accompagnata da una foto che funga da prova.
- Se l'admin ritiene attendibile la segnalazione, provvede ad aggiungere un evento al sistema. Al contempo dovrà essere aggiornata la mappa della città con le informazioni pertinenti.

Una volta modificata, la mappa si aggiorna automaticamente sull'interfaccia di ciascun utente, garantendo così una visualizzazione in tempo reale degli eventi.

La rimozione degli eventi avviene nel momento in cui l'admin riceve la prova che l'evento è stato risolto.

Il sistema deve assicurare trasparenza e tempestività, coinvolgendo attivamente i cittadini nella gestione delle informazioni urbane.

Analisi dei requisiti

Raccolta dei Requisiti

- L'utente, per usare l'applicazione, deve registrarsi. Inoltre deve autenticarsi ogni volta voglia utilizzare le funzionalità offerte dall'applicazione;
- L'utente accede al sistema con una coppia di credenziali chiamate username e password;
- Si prevede la presenza di un amministratore, che accede al sistema attraverso una coppia di credenziali chiamate username e password;
- L'utente può effettuare delle segnalazioni relative ad eventi di diversi tipi (frane, incidenti stradali, guasti di infrastrutture pubbliche ecc.), questi ultimi raggruppabili in categorie distinte (eventi naturali, richieste di interventi di manutenzione, ambiente/inquinamento, viabilità, eventi pubblici ecc.), inserendo una breve descrizione dell'evento, la categoria, il tipo ed il luogo dell'evento. Inoltre dovranno essere indicate delle prove fotografiche;
- L'utente può effettuare delle ricerche per eventi attualmente in corso e può visualizzarli su una mappa della città; inoltre ha la possibilità di affinare la ricerca tramite appositi filtri in base alla categoria di evento, al tipo di evento, luogo della città, ora, livello di gravità;
- Le segnalazioni dell'utente possono essere di 2 tipi:
 - segnalazione per notificare che un evento si è appena verificato;
 - segnalazione per notificare che un evento è stato risolto;
- Le segnalazioni dell'utente vengono registrate nel sistema e successivamente controllate e validate dall'amministratore;
- Il sistema associa ad ogni segnalazione un identificatore univoco, un'informazione di stato, data e ora;
- L'amministratore controlla la segnalazione e se la ritiene pertinente (in termini di descrizione, prove fotografiche, localizzazione ecc.) la accetta e provvede ad aggiungere (o rimuovere) l'evento. Segue l'aggiornamento della mappa della città che deve riflettere le modifiche apportate;
Nel caso in cui la segnalazione non sia considerata corretta, pertinente o completa, questa viene rifiutata e nell'eventualità di pluri-segnalazioni inappropriate il sistema provvederà a bloccare l'utente;
- Ad ogni evento deve essere associato, un identificatore univoco, un indice di gravità, una breve descrizione, un tipo, data, ora e luogo;
- Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (memoria, batteria e capacità di elaborazione), in modo da

garantire una buona esperienza utente anche su dispositivi con risorse limitate;

- Il sistema è distribuito e di natura client-server con la presenza di database dove memorizzare i dati;
- Possibili estensioni dell'applicazione per notifiche push e geolocalizzazione.

Tabella dei requisiti

ID REQUISITO	REQUISITO	TIPO
R1F	L'utente si registra inserendo nome, cognome, username, password	Funzionale
R2F	L'utente effettua il login fornendo una coppia di credenziali formata da username e password	Funzionale
R3F	L'amministratore effettua il login fornendo una coppia di credenziali formata da username e password	Funzionale
R4F	L'utente effettua segnalazioni allegando descrizione, luogo dell'evento, categoria e tipo di evento, fotografie, tipo di segnalazione	Funzionale
R5F	L'utente può cercare eventi attualmente in corso, eventualmente filtrarli e visualizzarli su una mappa	Funzionale
R6F	La segnalazione utente viene registrata nel sistema, il quale le assegna: id, informazione di stato iniziale, data e ora	Funzionale
R7F	L'amministratore verifica la segnalazione e, se pertinente, aggiunge (o rimuove) un evento, a cui segue un aggiornamento mappa. Altrimenti, la segnalazione viene rifiutata	Funzionale

R8F	Due tipi di segnalazioni: segnalazione per nuovo evento; segnalazione per evento risolto	Funzionale
R9F	Il sistema blocca un utente in caso di pluri-segnalazioni inappropriate	Funzionale
R9F	Filtri per la visualizzazione di eventi in base a categoria evento, tipo evento, luogo, ora, livello di gravità	Funzionale
R10F	Ad ogni evento è associato: identificatore, indice di gravità, descrizione, tipo, categoria, luogo, data, ora	Funzionale
R1NF	Velocità di ricerca dei dati	Non Funzionale
R2NF	Semplicità di navigazione tra le schermate	Non Funzionale
R3NF	Velocità di memorizzazione	Non Funzionale
R4NF	Il sistema sarà distribuito con la presenza di un qualche tipo di supporto di memorizzazione dati	Non Funzionale
R5NF	Il sistema deve garantire tempi di risposta rapidi, anche con un alto numero di utenti attivi o segnalazioni.	Non Funzionale
R6NF	Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo in modo da garantire una buona esperienza utente anche su dispositivi con risorse limitate.	Non Funzionale
R7NF	Ogni attore deve avere un username univoco	Non Funzionale
R8NF	Un utente bloccato per segnalazioni inappropriate non può effettuare nuove segnalazioni per un giorno e poi viene sbloccato dal Sistema	Non Funzionale

Analisi del dominio

Vocabolario

VOCE	DEFINIZIONE	SINONIMI
Applicazione	Prodotto software progettato per eseguire specifiche attività o funzioni	Programma, App
Utente	Persona o entità che interagisce con l'applicazione ed effettua segnalazioni	
Servizio	Strumento di segnalazione eventi	Funzionalità
Registrazione	Processo mediante il quale l'utente fornisce informazioni personali o di accesso	Iscrizione
Segnalazione	Attività con cui utente noto rende esplicito un evento/problema	Notifica
Segnalazione accettata	Segnalazione vagliata dall'amministratore e considerata corretta e pertinente	
Segnalazione rifiutata	Segnalazione vagliata dall'Amministratore e considerata incorretta e/o non pertinente	
Evento	Accadimento con causa naturale o umana che è oggetto delle Segnalazioni degli Utenti	Problema, Accadimento
Categoria Evento	Insieme omogeneo di tipi di Eventi, accomunati da una finalità, natura o contesto operativo simile	Classe di appartenenza di un Evento, Natura dell'Evento
Fotografia	Prova visiva da allegare alla Segnalazione	
Amministratore	Persona con privilegi elevati responsabile della gestione, configurazione e manutenzione del Sistema	Admin, Gestore

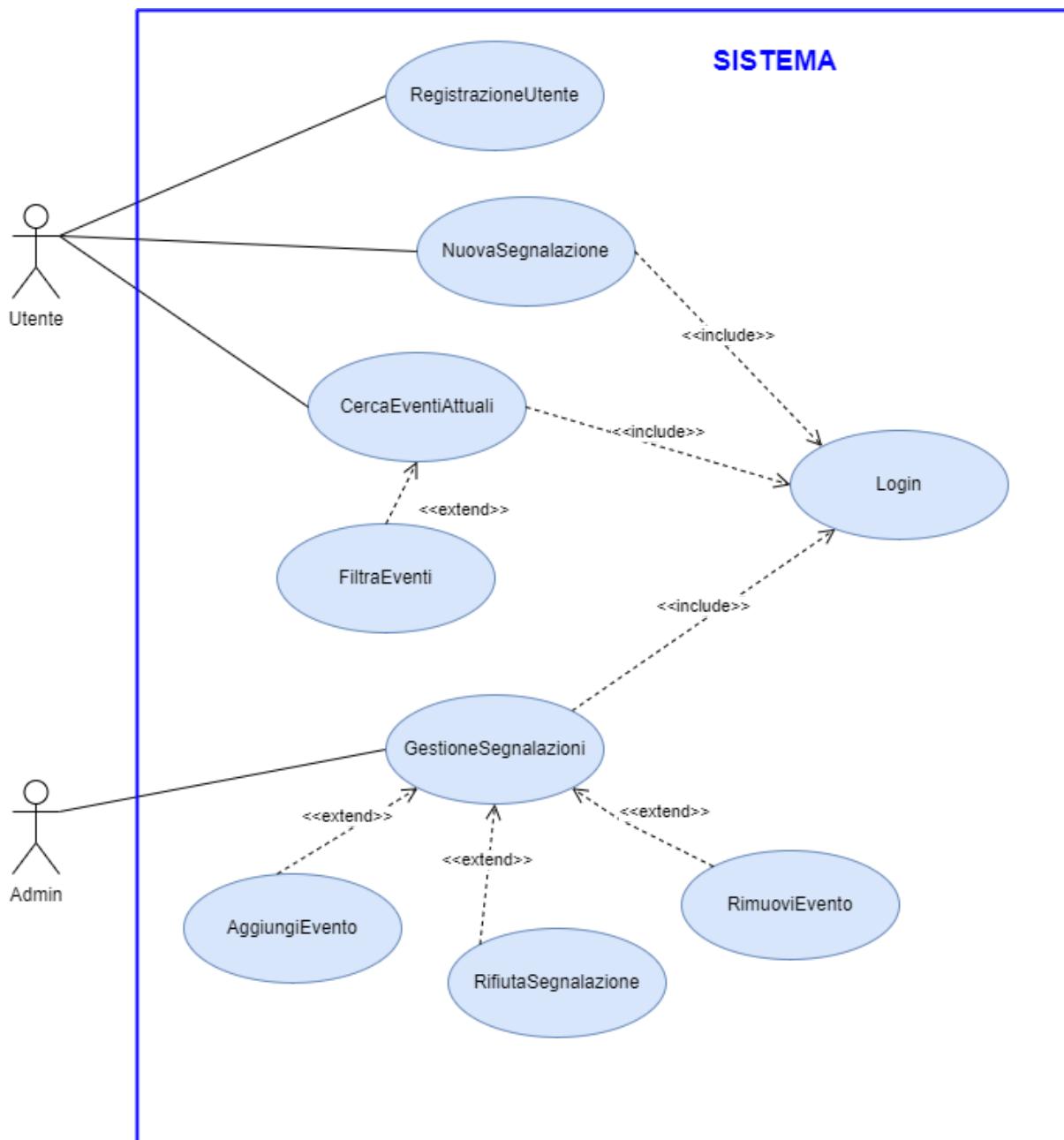
Mappa della città	Rappresentazione grafica digitale della città	
Credenziali	Insieme composto da username e password, necessari per accedere al Sistema	
Username	Parola formata dalla concatenazione di nome e cognome dell'Utente o dell'Admin	
Password	Codice alfanumerico di riconoscimento richiesto per accedere all'Applicazione	
Identificatore evento	Identifica un Evento all'interno del Sistema	idEvento
Identificatore segnalazione	Identifica una Segnalazione all'interno del Sistema	idSegnalazione
Filtro	Strumento che permette di restringere o personalizzare i risultati in base a determinati criteri	Setaccio, Criterio di filtraggio
Indice di gravità	Misura che quantifica l'impatto potenziale di un certo Evento	Livello di gravità, Valore di gravità
Informazione di stato	Fase in cui si trova la Segnalazione	Stato Segnalazione

Sistemi esterni

La mappa su cui vengono mostrati gli eventi non verrà sviluppata internamente ed ex novo, ma verranno adottate e integrate tecnologie di terze parti già consolidate, ottimizzando tempi e risorse.

Analisi dei requisiti

Diagramma dei casi d'uso



Scenari

Titolo	<i>Login</i>
Descrizione	Modalità con cui gli attori accedono al Sistema
Attori	Utente, Admin
Relazioni	GestioneSegnalazioni, Segnalazione, CercaEventiAttuali
Precondizioni	
Postcondizioni	L'attore risulta autenticato
Scenario principale	<ol style="list-style-type: none"> 1. L'Attore inserisce le credenziali di accesso al Sistema 2. Il Sistema verifica le credenziali inserite 3. Il Sistema autentica l'Attore
Scenari alternativi	<p>A) Le credenziali dell'Attore non sono valide:</p> <ol style="list-style-type: none"> 3. Il Sistema notifica l'errore 4. Il Sistema presenta la schermata di login <p>B) Utente non ancora registrato:</p> <ol style="list-style-type: none"> 5. Il Sistema notifica l'errore all'Utente 6. Il Sistema presenta la schermata di registrazione
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Ogni Attore deve avere un Username univoco (R7NF)
Punti aperti	

Titolo	<i>RegistrazioneUtente</i>
Descrizione	Un Utente si registra per usufruire dei servizi offerti dall'Applicazione
Attori	Utente
Relazioni	
Precondizioni	L'Utente non si è ancora registrato
Postcondizioni	L'Utente risulta registrato
Scenario principale	<ol style="list-style-type: none"> 1. L'Utente inserisce informazioni

	<p>personali come nome, cognome, Username e Password</p> <ol style="list-style-type: none"> 2. Il Sistema controlla che Username non risulti già utilizzato da un altro Utente 3. Il Sistema memorizza i dati del nuovo Utente, al quale assegna un identificatore (Id) 4. Il Sistema mostra all'Utente un messaggio di successo 5. Il Sistema mostra all'Utente la schermata di login
Scenari alternativi	<p>A) Username non univoco:</p> <ol style="list-style-type: none"> 6. Il Sistema notifica l'errore e presenta nuovamente la schermata di registrazione
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF) – Ogni attore deve avere un username univoco (R7NF)
Punti aperti	

Titolo	<i>CercaEventiAttuali</i>
Descrizione	Metodo di ricerca di eventi attualmente in corso
Attori	Utente
Relazioni	Login
Precondizioni	L'Utente deve essere registrato
Postcondizioni	Il Sistema mostra tutti gli eventi in corso in quel momento
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. L'Utente richiede la funzionalità di ricerca di eventi 3. Il Sistema mostra una schermata con la descrizione di tutti gli eventi in corso, con relativa localizzazione su una mappa della città 4. L'Utente scorre l'elenco degli eventi

	ed interagisce con la mappa
Scenari alternativi	A) Sessione scaduta 1. Login
Requisiti non funzionali	– Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Il sistema deve garantire tempi di risposta rapidi, anche con un alto numero di utenti attivi o segnalazioni (R5NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF)
Punti aperti	

Titolo	<i>FiltraEventi</i>
Descrizione	L'Utente può filtrare gli Eventi secondo certi Criteri di filtraggio
Attori	Utente
Relazioni	CercaEventiAttuali
Precondizioni	Devono essere presenti degli Eventi
Postcondizioni	L'Utente visualizza gli Eventi che rispondono ai Criteri di filtraggio
Scenario principale	1. L'Utente esplicita il Filtro 2. Il Sistema mostra gli Eventi filtrati
Scenari alternativi	A) Nessun Evento corrisponde al Criterio di filtraggio: 2. Il Sistema mostra un apposito messaggio
Requisiti non funzionali	– Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Il sistema deve garantire tempi di risposta rapidi, anche con un alto numero di utenti attivi o segnalazioni (R5NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF)
Punti aperti	

Titolo	<i>Nuova Segnalazione</i>
Descrizione	L'Utente segnala un Evento
Attori	Utente
Relazioni	Login
Precondizioni	L'utente deve essere registrato
Postcondizioni	L'Evento è stato registrato nel Sistema e verrà gestito e verificato dall'Admin
Scenario principale	<ol style="list-style-type: none"> 3. Login 4. L'Utente sceglie la funzionalità di segnalazione di Eventi 5. Il Sistema mostra una schermata per l'inserimento dei dati 6. L'Utente inserisce categoria, tipo e luogo dell' Evento, una descrizione, una o più Fotografie da allegare, il tipo di Segnalazione 7. Il Sistema registra la Segnalazione associandole un identificatore(idSegnalazione), un'Informazione di stato, la data e l'ora 8. Il Sistema invia messaggio di successo all'Utente
Scenari alternativi	A) Sessione scaduta <ol style="list-style-type: none"> 9. Login
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF) – Il sistema deve garantire tempi di risposta rapidi, anche con un alto numero di utenti attivi o segnalazioni (R5NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF) – Un utente bloccato per segnalazioni inappropriate non può effettuare nuove segnalazioni per un giorno e poi viene sbloccato dal Sistema (R8NF)
Punti aperti	

Titolo	<i>Gestione Segnalazioni</i>
Descrizione	Strumento di controllo, analisi e di gestione delle Segnalazioni dell'Utente
Attori	Admin
Relazioni	Login, AggiungiEvento, RimuoviEvento, RifiutaSegnalazione
Precondizioni	
Postcondizioni	
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. L'Admin controlla le nuove Segnalazioni per verificarne la correttezza 3. <ol style="list-style-type: none"> a. La Segnalazione è approvata e l'Admin aggiunge un nuovo Evento (AggiungiEvento) o rimuove un Evento risolto (RimuoviEvento) b. La Segnalazione è rifiutata (RifiutaSegnalazione)
Scenari alternativi	
Requisiti non funzionali	<ul style="list-style-type: none"> – Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF)
Punti aperti	

Titolo	<i>AggiungiEvento</i>
Descrizione	Viene aggiunto un Evento
Attori	Admin
Relazioni	GestioneSegnalazioni
Precondizioni	La Segnalazione è risultata pertinente, corretta, completa
Postcondizioni	L' Evento è stato aggiunto e lo Stato Segnalazione risulta modificato
Scenario principale	<ol style="list-style-type: none"> 1. L'Admin modifica lo Stato Segnalazione 2. L'Admin inserisce i dati dell'Evento:

	<p>descrizione, luogo, tipo, Indice di gravità, data, ora</p> <ol style="list-style-type: none"> 3. Il Sistema associa un identificatore (idEvento) all'Evento 4. Il Sistema registra l'Evento 5. Il Sistema aggiorna la Mappa
Scenari alternativi	
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF)
Punti aperti	

Titolo	<i>RimuoviEvento</i>
Descrizione	Viene rimosso un Evento
Attori	Admin
Relazioni	GestioneSegnalazioni
Precondizioni	Segnalazione è risultata pertinente,corretta,completa
Postcondizioni	Evento rimosso dal Sistema e lo Stato Segnalazione risulta modificato
Scenario principale	<ol style="list-style-type: none"> 1. L'Admin modifica lo Stato Segnalazione 2. L'Admin cancella l'Evento 3. Il Sistema registra le modifiche 4. Il Sistema aggiorna la Mappa
Scenari alternativi	
Requisiti non funzionali	<ul style="list-style-type: none"> – Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF)
Punti aperti	

Titolo	<i>RifiutaSegnalazione</i>
Descrizione	La Segnalazione non viene approvata
Attori	Admin
Relazioni	GestioneSegnalazioni

Precondizioni	
Postcondizioni	Non è stato creato/rimosso un Evento e lo Stato Segnalazione risulta modificato
Scenario principale	<ol style="list-style-type: none"> 1. L'Admin modifica lo Stato Segnalazione 2. Il Sistema registra la modifica
Scenari alternativi	
Requisiti non funzionali	<ul style="list-style-type: none"> – Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF) – Un utente bloccato per segnalazioni inappropriate non può effettuare nuove segnalazioni per un giorno e poi viene sbloccato dal Sistema (R8NF)
Punti aperti	

Analisi del rischio

Tabella Valutazione dei Beni

Bene	Valore	Esposizione
Sistema Informativo	Alta. Supporto e informazioni sensibili legate al funzionamento del software.	Alta. Perdita di immagine e rischio di sicurezza per gli utenti.
Informazioni relative agli Admin	Media. Credenziali(crittografate).	Media. Danni al team di gestione del software(seppur contenuti).
Informazioni relative agli Utenti	Alta. Credenziali(crittografate).	Alta. Perdita di immagine.
Informazioni relative agli Eventi	Alta. Eventi sulla Mappa nonché elemento principale del software.	Alta. Perdita di affidabilità quindi di immagine.
Informazioni relative alle Segnalazioni	Media. Segnalazioni di Eventi	Media. Problemi legati all'efficacia

	effettuate da parte di uno o più Utenti.	del software.
--	------------------------------------------	---------------

Tabella Minacce/Controlli

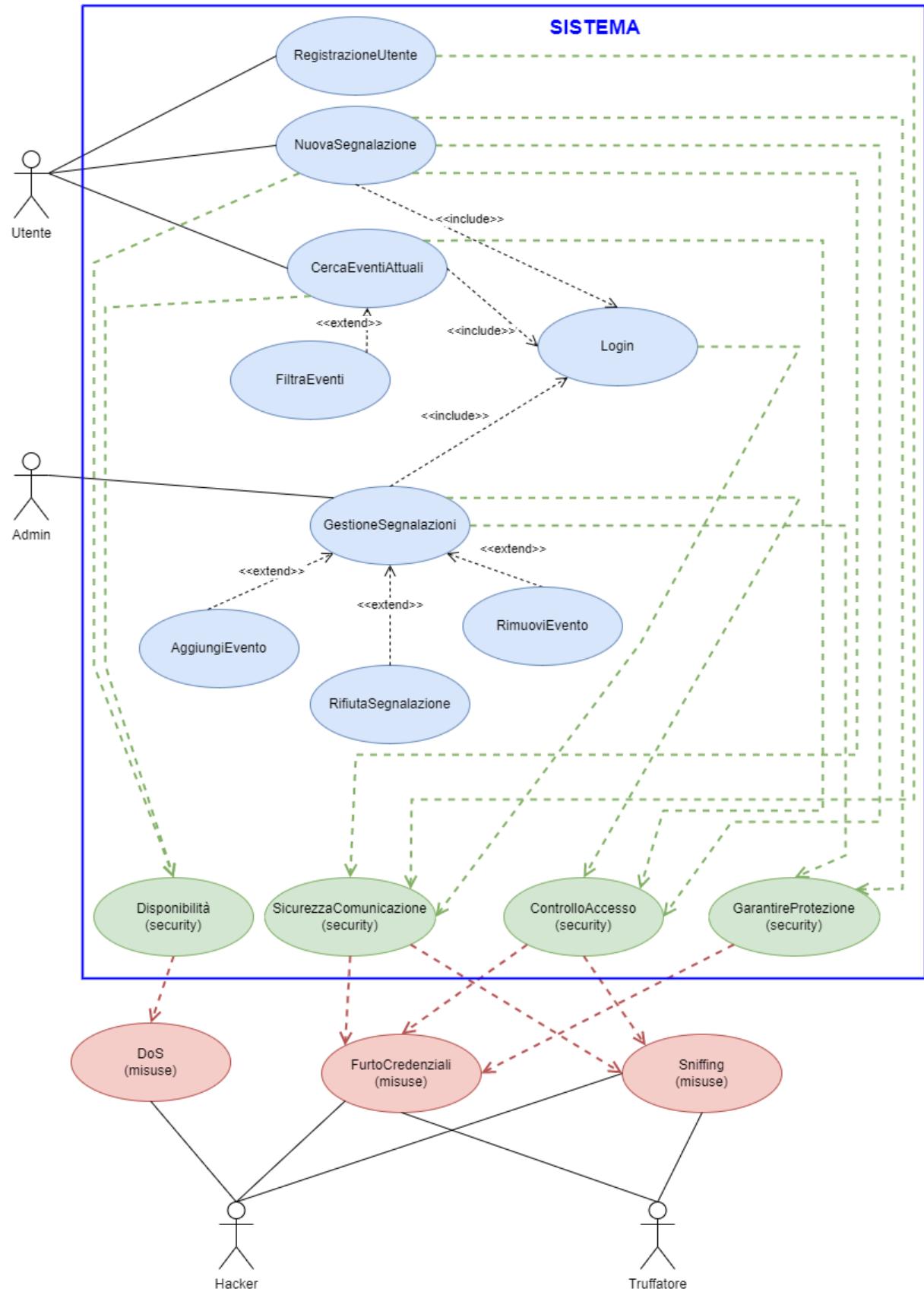
Minaccia	Probabilità	Controllo	Fattibilità
Furto credenziali Admin	Media	Log delle operazioni e accesso solo da macchine specifiche configurate appositamente	Basso costo
Furto credenziali Utenti	Alta. Username e Password Utenti spesso banali; ingegneria sociale	Log accessi e operazioni con tag per ogni utente	Costo basso
DoS	Media	Controllo e limitazione degli accessi	Costo basso. Impossibile prevenire questo tipo di attacco
Intercettazione comunicazioni	Alta. Il sistema è distribuito e client/server, con molte interazioni tra i diversi client ed il server	Cifratura delle comunicazioni e log di tutte le operazioni	Il costo dipende dal tipo di cifratura scelto. Se simmetrica basso, se asimmetrica più alto
Minacce alla sicurezza fisica: 1) Minacce ambientali(temperatura e umidità inappropriate, fuoco e fumo, danni causati dall'acqua, terremoti ed altri eventi climatici estremi) 2) Minacce tecniche legate ad energia elettrica	1) Medio-Alta, con dipendenza da: – Caratteristiche fisiche e condizioni di operatività delle apparecchiature – Caratteristiche climatiche e geologiche del luogo 2) Dipendenza da: – Stabilità della rete – Frequenza ed entità delle interruzioni	1) – Apparecchiature di controllo ambientale e sensori; – Sistemi di backup in diverse sedi ed aree geografiche; 2) – Gruppi di continuità – Generatori ausiliari 3) – Controllo dell'accesso fisico	1) Costo alto 2) Costo medio-alto 3) Costo medio-basso

3) Minacce causate dall'uomo: accesso fisico non autorizzato, furto, vandalismo, utilizzo improprio	3) Bassa. Rilevamento e controllo accessi hanno effetto deterrente	– Sistemi di rilevamento intrusioni – Logout automatico workstation dopo periodi di inattività	
-----------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------	---------------------------------------------------------------------------------------------------	--

Analisi Tecnologica della Sicurezza

Tecnologia	Vulnerabilità
Autenticazione tramite credenziali	<ul style="list-style-type: none"> • Password banali • Password cracking • Social engineering • Negligenza
Cifratura comunicazioni	<p>Le vulnerabilità dipendono dal tipo di cifratura.</p> <p>Cifratura simmetrica:</p> <ul style="list-style-type: none"> • Tempo di vita della chiave • Lunghezza della chiave • Memorizzazione della chiave • Più informazioni vengono cifrate con la stessa chiave, più materiale è offerto per l'analisi del testo a un attaccante <p>Cifratura asimmetrica:</p> <ul style="list-style-type: none"> • Lunghezza della chiave • Memorizzazione delle chiavi private
Architettura Client/Server	<ul style="list-style-type: none"> • Attacco DoS • Attacco Man in the Middle • Sniffing della comunicazione

Security Use Case e Misuse Case



Titolo	<i>Disponibilità</i>	
Descrizione	Il Sistema deve sempre garantire le funzionalità richieste	
Misuse case	DoS	
Relazioni		
Precondizioni	L'Attaccante dispone degli strumenti per mettere in atto questo tipo di attacco in modo efficace	
Postcondizioni	Il Sistema notifica quando il flusso di richieste eccede l'ordinario ed attua delle misure di sicurezza	
Scenario Principale	Sistema	Attaccante
	Ha meccanismi di protezione attivi	
		Mette in azione l'attacco
	Notifica tentativo di DoS mentre cerca di ridurre l'efficacia dell'attacco	
		Non riesce a portare a termine il misfatto
Scenario di attacco avvenuto con successo	Sistema	Attaccante
	Ha meccanismi di protezione attivi	
		Mette in azione l'attacco
	Notifica il tentativo di DoS mentre cerca di ridurre l'efficacia dell'attacco	
	Non riesce a contenere l'attacco e va fuori uso	
	L'Amministratore riceve notifica che il sistema è fuori uso e tenta un ripristino locale	

Titolo	<i>ControlloAccesso</i>	
Descrizione	Ogni accesso al Sistema deve essere controllato	
Misuse case	FurtoCredenziali	
Relazioni		
Precondizioni	L'Attaccante dispone degli strumenti per tentare di autenticarsi con le Credenziali di un Utente o Admin	
Postcondizioni	Il Sistema blocca il tentativo di accesso non autorizzato dell'Attaccante	
Scenario principale	Sistema	Attaccante
	Individua le Credenziali di un Utente o dell'Admin e tenta di accedere al sistema	
Scenario di attacco avvenuto con successo	Controlla le Credenziali immesse e blocca l'accesso nel caso in cui le Credenziali risultino errate dopo un certo numero di tentativi	
	Sistema	Attaccante
	Attacco riuscito	
	Controlla le credenziali immesse e consente l'accesso	
	Naviga tra le maschere del Sistema e cerca di carpire più informazioni possibili	
	Scrive in un log tutte le operazioni eseguite dall'Utente/Attaccante	

Titolo	<i>SicurezzaComunicazione</i>
Descrizione	I dati delle comunicazioni devono essere protetti
Misuse case	Sniffing, FurtoCredenziali
Relazioni	
Precondizioni	A. L'attaccante ha i mezzi per

	intercettare i messaggi B. L'attaccante ha i mezzi per modificare i messaggi C. L'attaccante ha i mezzi per spedire il messaggio modificato al destinatario	
Postcondizioni	Il sistema notifica un tentativo di frode	
Scenario principale	Sistema	Attaccante
	Si occupa di proteggere i messaggi che fluiscono tra le diverse parti e memorizza nel log i messaggi	
		Intercetta un messaggio
		Non riesce a rimuovere il meccanismo di protezione, così genera un nuovo messaggio e lo prova a mandare al destinatario
	Si accorge che il messaggio ricevuto non è pertinente e segnala un tentativo di frode	
Scenario di attacco avvenuto con successo	Sistema	Attaccante
	Si occupa di proteggere i messaggi che fluiscono tra le diverse parti e memorizza nel log i messaggi	
		Intercetta un messaggio nel Sistema
		Rimuove il meccanismo di protezione, modifica del messaggio, riapplica la protezione e manda il messaggio al destinatario
	Elabora il messaggio	

	e agisce di conseguenza, oltre a scrivere il messaggio nel log	
--	----------------------------------------------------------------	--

Titolo	<i>Garantire Protezione</i>							
Descrizione	Il sistema deve sempre garantire che utenti malintenzionati non possano accedere, analizzare e/o modificare dati del sistema							
Misuse case	FurtoCredenziali							
Relazioni								
Precondizioni	L'attaccante dispone dei mezzi per cercare e sfruttare eventuali vulnerabilità presenti nel sistema							
Postcondizioni	Il sistema blocca il tentativo di attacco ai dati da parte dell'attaccante							
Scenario principale	<table border="1"> <thead> <tr> <th>Sistema</th> <th>Attaccante</th> </tr> </thead> <tbody> <tr> <td>Ha meccanismi di protezione dei dati (cifratura dei dati)</td> <td>Ricerca delle vulnerabilità per superare le difese del sistema ma non ne trova</td> </tr> </tbody> </table>	Sistema	Attaccante	Ha meccanismi di protezione dei dati (cifratura dei dati)	Ricerca delle vulnerabilità per superare le difese del sistema ma non ne trova			
Sistema	Attaccante							
Ha meccanismi di protezione dei dati (cifratura dei dati)	Ricerca delle vulnerabilità per superare le difese del sistema ma non ne trova							
Scenario di attacco avvenuto con successo	<table border="1"> <thead> <tr> <th>Sistema</th> <th>Attaccante</th> </tr> </thead> <tbody> <tr> <td>Ha meccanismi di protezione dei dati (cifratura dei dati)</td> <td>Cerca e trova delle vulnerabilità per superare le difese del sistema</td> </tr> <tr> <td></td> <td>Accede ai dati, che però sono cifrati</td> </tr> </tbody> </table>	Sistema	Attaccante	Ha meccanismi di protezione dei dati (cifratura dei dati)	Cerca e trova delle vulnerabilità per superare le difese del sistema		Accede ai dati, che però sono cifrati	
Sistema	Attaccante							
Ha meccanismi di protezione dei dati (cifratura dei dati)	Cerca e trova delle vulnerabilità per superare le difese del sistema							
	Accede ai dati, che però sono cifrati							
Salva nel log gli accessi effettuati sui dati								

Requisiti di sicurezza aggiuntivi:

1) Creazione file di log per tracciare (RF11):

- tutte le azioni che avvengono sul sistema;
- i messaggi scambiati tra le parti del sistema (vanno protetti per evitare che un accesso fraudolento al sistema di log possa rivelare dati riservati).

2) Adozione di meccanismi di analisi del log per (RF12):

- Individuare azioni, comportamenti, accessi atipici o sospetti;
- identificare discrepanze tra i messaggi spediti e ricevuti.

L'Admin ha la possibilità di visualizzare il log (RF13), ma l'analisi dei log sarà automatizzata ed affidata ad un sistema esterno, il cui servizio potrà essere fornito anche da soluzioni di terze parti.

3) La password dell'utente e dell'admin deve essere almeno di 10 caratteri e non deve contenere informazioni personali come nome, cognome, anno di nascita ecc. (R8NF).

4) Individuazione di una corretta politica per il controllo degli accessi (R9NF).

5) I dati memorizzati e scambiati nel sistema devono essere protetti da un eventuale attaccante con accesso al sistema, eventualmente adottando una cifratura dei dati (R10NF).

6) Blocco temporaneo dell'accesso dopo tentativi multipli di Login falliti per prevenire accessi non autorizzati (RF14).

Aggiornamento requisiti del sistema:

ID	Requisito	Tipo
R11F	Creazione di un Log per tracciare tutte le azioni che avvengono sul sistema e tutte le interazioni tra le parti del Sistema	Funzionale
R12F	Analisi dei Log per individuare accessi e comportamenti sospetti e discrepanze tra messaggi spediti e ricevuti	Funzionale
R13F	Admin visualizza il Log	Funzionale

R14F	Blocco temporaneo dell'accesso dopo tentativi multipli di Login falliti	Funzionale
R8NF	La Password dell'Utente e dell'Admin deve essere almeno di 10 caratteri e non deve contenere riferimenti ad informazioni personali	Non Funzionale
R9NF	Individuazione corretta politica di controllo degli accessi	Non Funzionale
R10NF	La sicurezza dei dati scambiati e memorizzati nel Sistema deve essere garantita contro accessi non autorizzati	Non Funzionale
R11NF	L'analisi del Log verrà gestita da un sistema esterno	Non Funzionale

Aggiornamento Vocabolario

Voce	Definizione	Sinonimi
Log	Registro dove vengono salvate informazioni relative alle attività interne al Sistema ed i messaggi da/per il Sistema. E' composto da una serie di Entry di Log	Registro
Entry di Log	Elemento di un Log	Record di Log

Aggiunta nuovo caso d'uso:

Titolo	<i>VisualizzaLog</i>
Descrizione	L'Admin visualizza le voci del Log
Attori	Admin
Relazioni	Login
Precondizioni	

Postcondizioni	
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. Il Sistema mostra le ultime Entry aggiunte al Log, con la possibilità di filtrarle 3. L'Admin visualizza una Entry di Log
Scenari alternativi	
Requisiti non funzionali	<p>– La sicurezza dei dati scambiati e memorizzati nel sistema deve essere garantita contro accessi non autorizzati (R10NF)</p>
Punti aperti	

Aggiornamento casi d'uso: Le modifiche sono state evidenziate in rosso

Titolo	<i>Login</i>
Descrizione	Modalità con cui gli attori accedono al Sistema
Attori	Utente, Admin
Relazioni	GestioneSegnalazioni, Segnalazione, CercaEventiAttuali
Precondizioni	
Postcondizioni	L'attore risulta autenticato
Scenario principale	<ol style="list-style-type: none"> 1. L'Attore inserisce le credenziali di accesso al Sistema 2. Il Sistema verifica le credenziali inserite 3. Il Sistema autentica l'Attore
Scenari alternativi	<p>A) Le credenziali dell'Attore non sono valide:</p> <ol style="list-style-type: none"> 3. Il Sistema notifica l'errore 4. Il Sistema presenta la schermata di login <p>B) Utente non ancora registrato:</p> <ol style="list-style-type: none"> 3. Il Sistema notifica l'errore all'Utente

	<p>4. Il Sistema presenta la schermata di registrazione</p> <p>C) Credenziali Utente errate inserite più di tre volte:</p> <p>3. Il Sistema notifica l'Utente con un avviso</p> <p>4. Il Sistema presenta all'Utente la schermata di login dopo alcuni minuti</p>
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Ogni Attore deve avere un Username univoco (R7NF) – Individuazione corretta politica di controllo degli accessi (R9NF) – La sicurezza dei dati scambiati e memorizzati nel sistema deve essere garantita contro accessi non autorizzati (R10NF)
Punti aperti	

Titolo	<i>RegistrazioneUtente</i>
Descrizione	Un Utente si registra per usufruire dei servizi offerti dall'Applicazione
Attori	Utente
Relazioni	
Precondizioni	L'Utente non si è ancora registrato
Postcondizioni	L'Utente risulta registrato
Scenario principale	<ol style="list-style-type: none"> 1. L'Utente inserisce informazioni personali come nome, cognome, data di nascita, Username e Password 2. Il Sistema controlla che Username non risulti già utilizzato da un altro Utente 3. Il Sistema controlla che la Password sia di almeno 10 caratteri e non contenga riferimenti ad informazioni personali 4. Il Sistema memorizza i dati del nuovo Utente, al quale assegna un

	<p>identificatore (Id)</p> <p>5. Il Sistema mostra all'Utente un messaggio di successo</p> <p>6. Il Sistema mostra all'Utente la schermata di login</p>
Scenari alternativi	<p>A) Username non univoco:</p> <p>3. Il Sistema notifica l'errore e presenta nuovamente la schermata di registrazione</p> <p>B) Password non sicura</p> <p>4. Il Sistema mostra all'Utente un messaggio di errore e presenta nuovamente la schermata di registrazione</p>
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF) – Ogni attore deve avere un username univoco (R7NF) – La Password dell'Utente e dell'Admin deve essere almeno di 10 caratteri e non deve contenere riferimenti ad informazioni personali (R8NF) – La sicurezza dei dati scambiati e memorizzati nel sistema deve essere garantita contro accessi non autorizzati (R10NF)
Punti aperti	

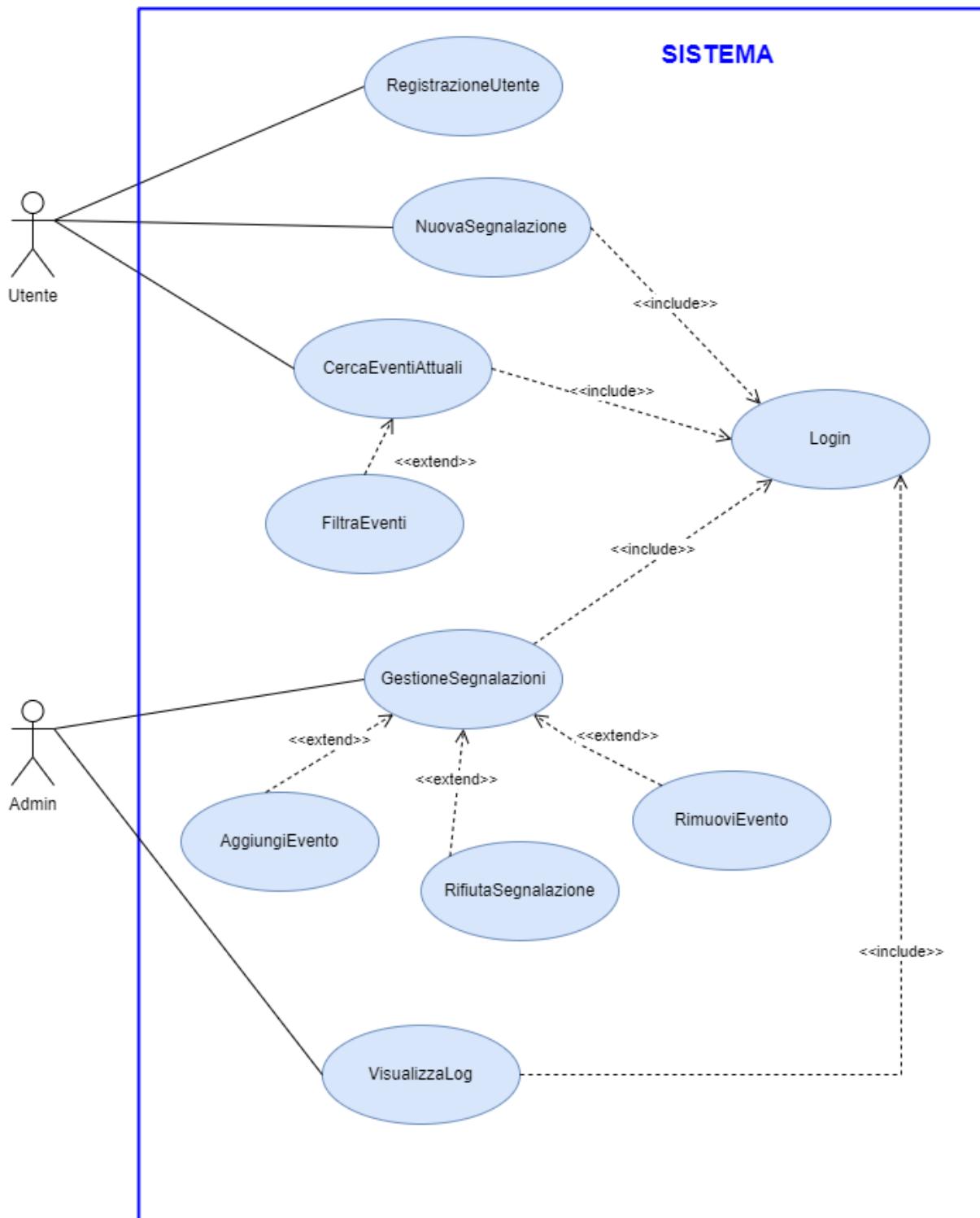
Titolo	<i>Nuova Segnalazione</i>
Descrizione	L'Utente segnala un Evento
Attori	Utente
Relazioni	Login
Precondizioni	L'utente deve essere registrato
Postcondizioni	L'Evento è stato registrato nel Sistema e verrà gestito e verificato dall'Admin
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. L'Utente sceglie la funzionalità di

	<p>segnalazione di Eventi</p> <ol style="list-style-type: none"> 3. Il Sistema mostra una schermata per l'inserimento dei dati 4. L'Utente inserisce categoria, tipo e luogo dell' Evento, una descrizione, una o più Fotografie da allegare, il tipo di Segnalazione 5. Il Sistema registra la Segnalazione associandole un identificatore(idSegnalazione), un'Informazione di stato, la data e l'ora 6. Il Sistema invia messaggio di successo all'Utente
Scenari alternativi	<p>A) Sessione scaduta</p> <ol style="list-style-type: none"> 1. Login
Requisiti non funzionali	<ul style="list-style-type: none"> – Semplicità di navigazione tra le schermate (R2NF) – Velocità di memorizzazione (R3NF) – Il sistema deve garantire tempi di risposta rapidi, anche con un alto numero di utenti attivi o segnalazioni (R5NF) – Il sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF) – Un utente bloccato per segnalazioni inappropriate non può effettuare nuove segnalazioni per un giorno e poi viene sbloccato dal Sistema (R8NF) – Individuazione corretta politica di controllo degli accessi (R9NF) – La sicurezza dei dati scambiati e memorizzati nel sistema deve essere garantita contro accessi non autorizzati (R10NF)
Punti aperti	

Titolo	<i>Gestione Segnalazioni</i>
Descrizione	Strumento di controllo, analisi e di gestione delle Segnalazioni dell'Utente
Attori	Admin
Relazioni	Login, AggiungiEvento, RimuoviEvento, RifiutaSegnalazione

Precondizioni	
Postcondizioni	
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. L'Admin controlla le nuove Segnalazioni per verificarne la correttezza 3. <ol style="list-style-type: none"> a. La Segnalazione è approvata e l'Admin aggiunge un nuovo Evento (AggiungiEvento) o rimuove un Evento risolto (RimuoviEvento) b. La Segnalazione è rifiutata (RifiutaSegnalazione)
Scenari alternativi	
Requisiti non funzionali	<ul style="list-style-type: none"> – Velocità di ricerca dei dati (R1NF) – Semplicità di navigazione tra le schermate (R2NF) – Individuazione corretta politica di controllo degli accessi (R9NF) – La sicurezza dei dati scambiati e memorizzati nel sistema deve essere garantita contro accessi non autorizzati (R10NF)
Punti aperti	

DIAGRAMMA DEI CASI D'USO AGGIORNATO



Analisi del problema

Analisi delle Funzionalità

Tabella delle funzionalità

Funzionalità	Tipo	Grado Complessità	Requisiti
RegistrazioneUtente	– Interazione esterno; – Memorizzazione dati	Semplice	R1F; R2NF, R3NF, R6NF, R7NF, R8NF, R10NF
NuovaSegnalazione	– Interazione esterno; – Memorizzazione dati	Semplice	R4F, R6F, R8F, R9F, R2NF, R3NF, R5NF, R6NF, R10NF
CercaEventiAttuali	– Interazione esterno; – Gestione dati	Semplice	R5F, R10F; R1NF, R2NF, R5NF, R6NF
Login	– Interazione esterno; – Gestione dati	Semplice	R2F, R14F, R3F R2NF, R7NF, R9NF, R10NF
GestioneSegnalazioni	– Gestione dati; – Memorizzazione dati	Complessa	R7F, R9F, R1NF, R2NF, R10NF
FiltraEventi	– Interazione esterno; – Gestione dati	Semplice	R5F, R9F, R10F; R1NF, R2NF, R5NF
ScritturaLog	– Memorizzazione dati	Semplice	R11F
VisualizzaLog	– Interazione esterno – Gestione dati	Semplice	R13F, R10NF

RegistrazioneUtente : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Username	Semplice	Medio	Input	Non più di 30 caratteri; Sono ammessi caratteri alfanumerici e trattini bassi
Password	Semplice	Alto	Input	Almeno 10

				caratteri(R8NF)
Nome Utente	Semplice	Medio	Input	Non più di 40 caratteri
Cognome utente	Semplice	Medio	Input	Non più di 40 caratteri
Id	Semplice	Medio	Input	Non più di 10 caratteri

NuovaSegnalazione : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Categoria Evento	Semplice	Basso	Input	Non più di 50 caratteri
Tipo Evento	Semplice	Basso	Input	Non più di 50 caratteri
Descrizione	Semplice	Basso	Input	Non più di 500 caratteri
Fotografie	Semplice	Medio	Input	Almeno una fotografia; Formati: JPG, PNG
Tipo Segnalazione	Semplice	Basso	Input	Non più di 20 caratteri
Luogo	Semplice	Basso	Input	Non più di 50 caratteri
IdSegnalazione	Semplice	Medio	Input	Non più di 10 caratteri
Stato Segnalazione	Semplice	Basso	Input	Non più di 20 caratteri
Data	Semplice	Basso	Input	Formato dd/mm/yyyy
Ora	Semplice	Basso	Input	Formato hh:mm:ss
Utente composto da:	Composto	Medio	Input	
- Nome Utente	Semplice	Medio	Input	Non più di 40 caratteri

- Cognome Utente	Semplice	Medio	Input	Non più di 40 caratteri
- Username Utente	Semplice	Medio	Input	Non più di 30 caratteri
- Id	Semplice	Medio	Input	Non più di 10 caratteri

CercaEventiAttuali : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Elenco eventi composto da:	Composto	Basso	Output	
Evento composto da:	Composto	Basso	Output	
- IdEvento	Semplice	Basso	Output	Non più di 10 caratteri
- Tipo Evento	Semplice	Basso	Output	Non più di 50 caratteri
- Categoria Evento	Semplice	Basso	Output	Non più di 50 caratteri
- Indice di gravità	Semplice	Basso	Output	Non più di 10 caratteri
- Descrizione	Semplice	Basso	Output	Non più di 500 caratteri
- Luogo	Semplice	Basso	Output	Non più di 50 caratteri
- Data	Semplice	Basso	Output	Formato dd/mm/yyyy
- Ora	Semplice	Basso	Output	Formato hh:mm:ss
Mappa	Semplice	Basso	Output	

GestioneSegnalazioni : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Segnalazione composto da:	Composto	Medio	Input	

- IdSegnalazione	Semplice	Medio	Input	Non più di 10 caratteri
- Utente composto(vedi sopra)	Semplice	Medio	Input	Non più di 40 caratteri
- Tipo evento	Semplice	Basso	Input	Non più di 50 caratteri
- Categoria Evento	Semplice	Basso	Input	Non più di 50 caratteri
- Luogo	Semplice	Basso	Input	Non più di 50 caratteri
- Descrizione	Semplice	Basso	Input	Non più di 500 caratteri
- Fotografie	Semplice	Medio	Input	Qualità immagine; Almeno una fotografia
- Tipo Segnalazione	Semplice	Basso	Input	Non più di 20 caratteri
- Stato Segnalazione	Semplice	Basso	Input	Non più di 20 caratteri
- Data	Semplice	Basso	Input	Formato dd/mm/yyyy
- Ora	Semplice	Basso	Input	Formato hh:mm:ss

FiltraEventi : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Elenco Eventi composto da: vedi CercaEventiAttuali	Composto	Basso	Input/Output	
Filtro composto da:	Composto	Basso	Input	
- Categoria Evento	Semplice	Basso	Input	Non più di 50 caratteri
- Tipo Evento	Semplice	Basso	Input	Non più di 50 caratteri

- Indice di gravità	Semplice	Basso	Input	Non più di 10 caratteri
- Ora	Semplice	Basso	Input	Formato hh:mm:ss
- Luogo	Semplice	Basso	Input	
Mappa	Semplice	Basso	Output	

ScritturaLog : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Entry di Log composto da:	Composto	Alto	Input	
- Data	Semplice	Basso	Input	
- Sorgente	Semplice	Medio	Input	
- Destinazione	Semplice	Medio	Input	
- Operazione	Semplice	Alto	Input	
- Messaggio	Semplice	Alto	Input	

VisualizzaLog : Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Entry di Log composto da: vedi ScritturaLog	Composto	Alto	Output	

Scomposizione funzionalità

Funzionalità	Scomposizione
GestioneSegnalazioni	AggiungiEvento, RifiutaSegnalazione, RimuoviEvento

Sottofunzionalità	Sottofunzionalità	Legame	Informazioni
RimuoviEvento	AggiungiEvento	Non si può rimuovere un Evento prima di averlo inserito	idEvento
RimuoviEvento	RifiutaSegnalazione	Non si può rimuovere un Evento per una Segnalazione che non sia stata accettata	idEvento, idSegnalazione
AggiungiEvento	RifiutaSegnalazione	Non si può aggiungere un Evento se la Segnalazione viene rifiutata	IdEvento, idSegnalazione

Analisi dei Vincoli

Tabella dei vincoli

Requisito	Categorie	Impatto	Funzionalità
Velocità ricerca dati (R1NF)	Tempo di risposta	Cerchiamo di migliorare	GestioneSegnalazioni CercaEventiAttuali FiltraEventi
Semplicità navigazione delle schermate (R2NF)	Usabilità	Cerchiamo di migliorare	RegistrazioneUtente NuovaSegnalazione Login CercaEventiAttuali GestioneSegnalazione FiltraEventi
Velocità memorizzazione dati (R3NF)	Tempo di risposta	Cerchiamo di migliorare	RegistrazioneUtente NuovaSegnalazione RimuoviEvento RifiutaSegnalazione AggiungiEvento GestioneSegnalazione
Scalabilità sistema per crescita segnalazioni, interazioni e richieste utenti (R5NF)	Tempo di risposta	Cerchiamo di migliorare	NuovaSegnalazione CercaEventiAttuali FiltraEventi
Il Sistema deve essere ottimizzato per l'uso efficiente delle risorse del dispositivo (R6NF)	Tempo di risposta; Usabilità	Cerchiamo di migliorare	RegistrazioneUtente CercaEventiAttuali FiltraEventi NuovaSegnalazione
Controllo degli accessi (R8NF, R9NF)	Sicurezza	Peggiorano il tempo di risposta, ma migliorano la privacy dei dati	NuovaSegnalazione CercaEventiAttuali GestioneSegnalazione Login
Protezione dei dati (R10NF)	Sicurezza	Peggiorano il tempo di risposta, ma migliorano la privacy dei dati	RegistrazioneUtente NuovaSegnalazione Login GestioneSegnalazione VisualizzaLog

Analisi delle interazioni

Tabella delle maschere

Maschera	Informazioni	Funzionalità
Home Admin	Elenco Segnalazioni	GestioneSegnalazioni VisualizzaLog
View GestioneSegnalazioni	Dettaglio Segnalazione	GestioneSegnalazioni
View AggiungiEvento	Evento	GestioneSegnalazioni
View RifiutaSegnalazione	Segnalazione	GestioneSegnalazioni
View RimuoviEvento	Evento	GestioneSegnalazioni
View Login	Username, Password	Login
Home Utente	Messaggio di benvenuto, elenco delle funzionalità	NuovaSegnalazione CercaEventiAttuali
View NuovaSegnalazione	Tipo Evento, descrizione, fotografie, luogo	NuovaSegnalazione
View RegistrazioneUtente	Username, Password, nome Utente, cognome Utente, data di nascita	RegistrazioneUtente
View CercaEventiAttuali	Elenco degli Eventi	CercaEventiAttuali
View FiltraEventi	Filtro Elenco degli Eventi	FiltraEventi
View Mappa	Eventi	CercaEventiAttuali FiltraEventi
View Log	Entry di Log	VisualizzaLog

Tabella Sistemi Esterni

Analizzatore log

Descrizione	Sistema che si occupa di analizzare i log al fine di individuare accessi sospetti e discrepanze tra messaggi spediti e ricevuti
Protocollo di interazione	Il sistema richiede i log alla piattaforma e, una volta ricevuti, esegue dei controlli e

	restituisce eventuali avvisi
Livello di protezione	Alto. Nei log sono presenti molte informazioni sensibili che, se compromesse, potrebbero causare seri danni al sistema

Sistema di gestione della Mappa

Descrizione	Sistema specializzato in funzionalità cartografiche, integrato nell'applicazione per offrire strumenti di navigazione e visualizzazione degli Eventi.
Protocollo di interazione	Quando un nuovo Evento viene aggiunto al Sistema, un controller dedicato si interfaccia con il framework di gestione della Mappa, trasmettendo le informazioni relative all'Evento. Il framework provvede quindi ad aggiornare la visualizzazione, rendendo visibile il nuovo Evento sulla Mappa in tempo reale.
Livello di protezione	Medio. Tale sistema esterno non tratta direttamente dati sensibili, ma mantiene un'interazione attiva con il Sistema principale. Di conseguenza, eventuali vulnerabilità nella componente esterna potrebbero influire sulla stabilità o sicurezza complessiva dell'applicazione.

Analisi dei ruoli e delle responsabilità

Tabella dei ruoli

Ruolo	Responsabilità	Maschere	Riservatezza	Numerosità
Utente	Visualizza Eventi in tempo reale, con la possibilità di	View CercaEventiAttuali View Login View RegistrazioneUtente	Richiesto un medio grado di riservatezza	Decine di migliaia di persone

	filtrare i risultati e visualizzarli su una Mappa; Effettua Segnalazioni	View NuovaSegnalazione Home Utente View FiltraEventi View Mappa		
Admin	Gestisce le Segnalazioni degli Utenti e l'aggiunta/rimozione di Eventi	Home GestioneSegnalazioni View AggiungiEvento View RimuoviEvento View RifiutaSegnalazione View Login View Log	Richiesto un alto grado di riservatezza	2,3 persone

Tabelle ruolo-informazione

Utente

Informazione	Tipo di Accesso
Nome Utente	Lettura/Scrittura
Cognome Utente	Lettura/Scrittura
Username Utente	Lettura/Scrittura
Password Utente	Scrittura
Segnalazione	Scrittura
Evento	Lettura

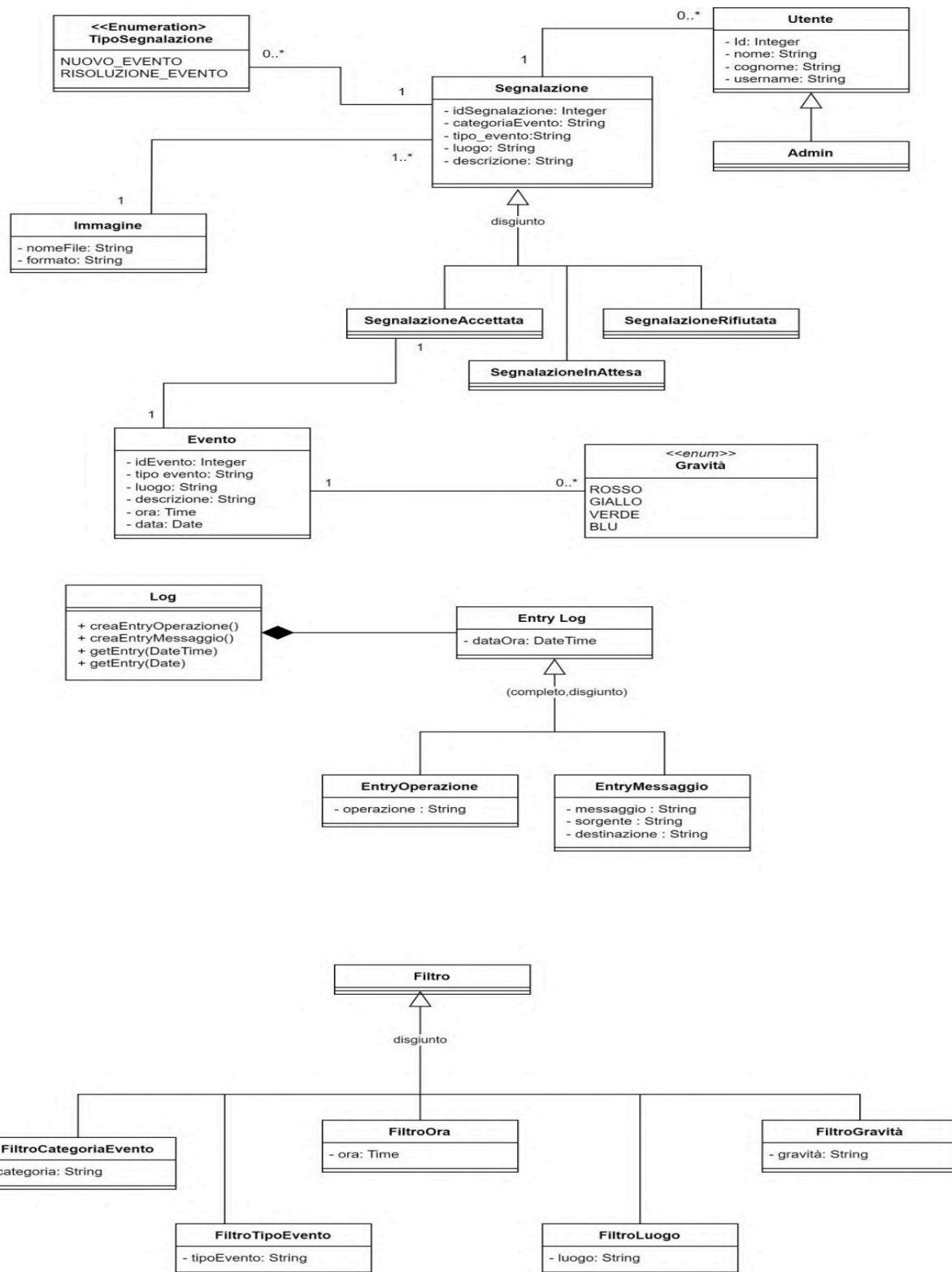
Admin

Informazione	Tipo di Accesso
Segnalazione	Lettura
Username Admin	Lettura/scrittura
Utente	Lettura
Password Admin	Scrittura
Evento	Scrittura
Entry di Log	Lettura

Modello del Dominio

Il seguente diagramma rappresenta le classi del modello del dominio. Le cardinalità delle associazioni sono state stabilite secondo la seguente

convenzione: la molteplicità di un'associazione rappresenta il numero minimo e massimo di istanze con cui una singola istanza di una determinata entità può partecipare all'associazione.



Architettura Logica

Struttura

Diagramma dei package

La struttura dei package individuata è conforme al pattern architetturale dell'Entity-control boundary, in cui:

- Gli Attori individuati nell'analisi dei requisiti interagiscono esclusivamente con i package di interfaccia (boundary), individuati in arancione
- Il modello (model), rappresentato dai package in azzurro, rappresenta la struttura logica dell'applicazione e le entità che interagiscono.
- I controller, evidenziati in verde, implementano la logica di business, servendo da legante tra le interfacce e il modello.

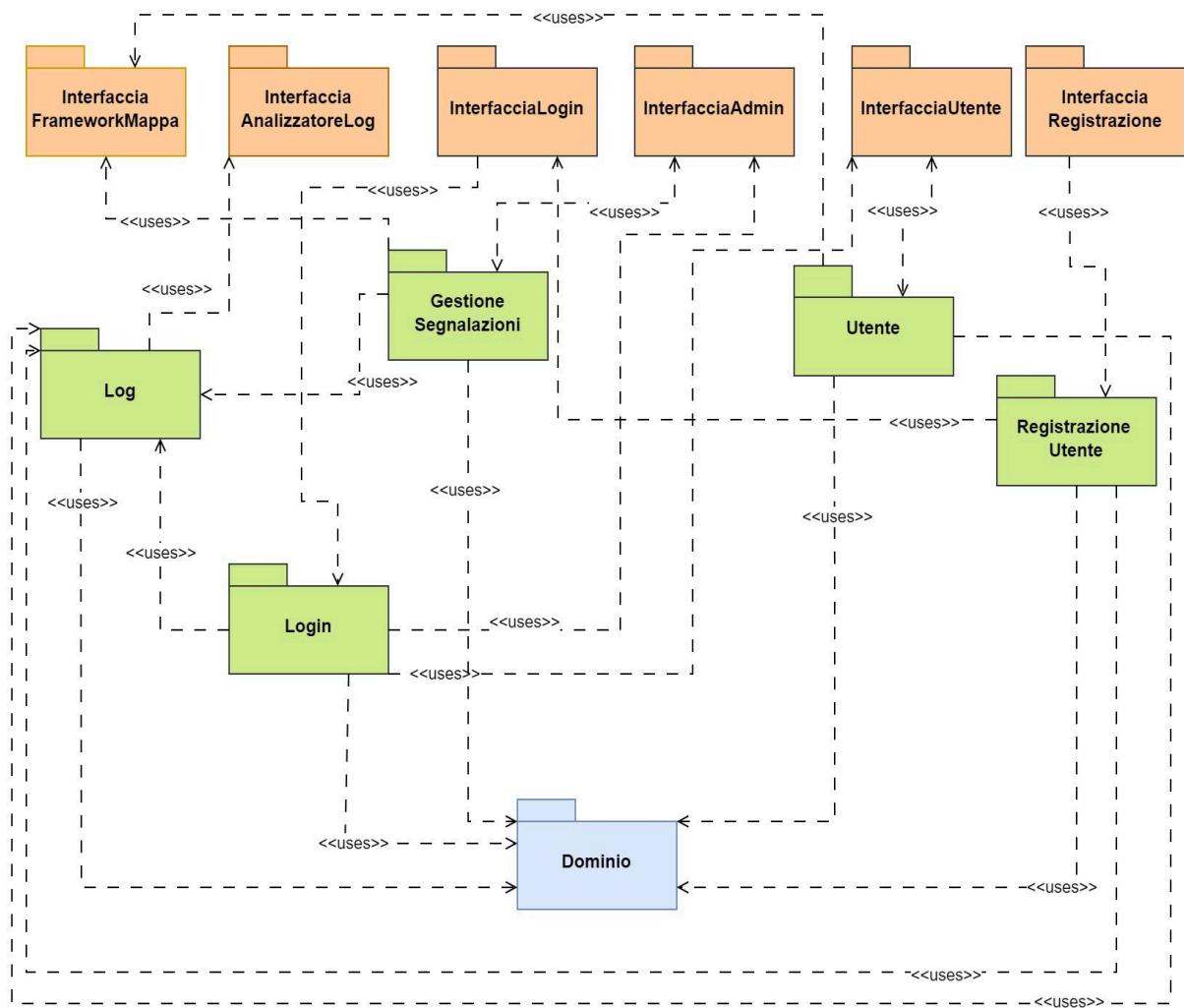


Diagramma delle classi: Dominio

Non viene riportato il diagramma delle classi associato al package Dominio in quanto è il modello del dominio creato nella fase precedente.

Diagramma delle classi: Registrazione e Login

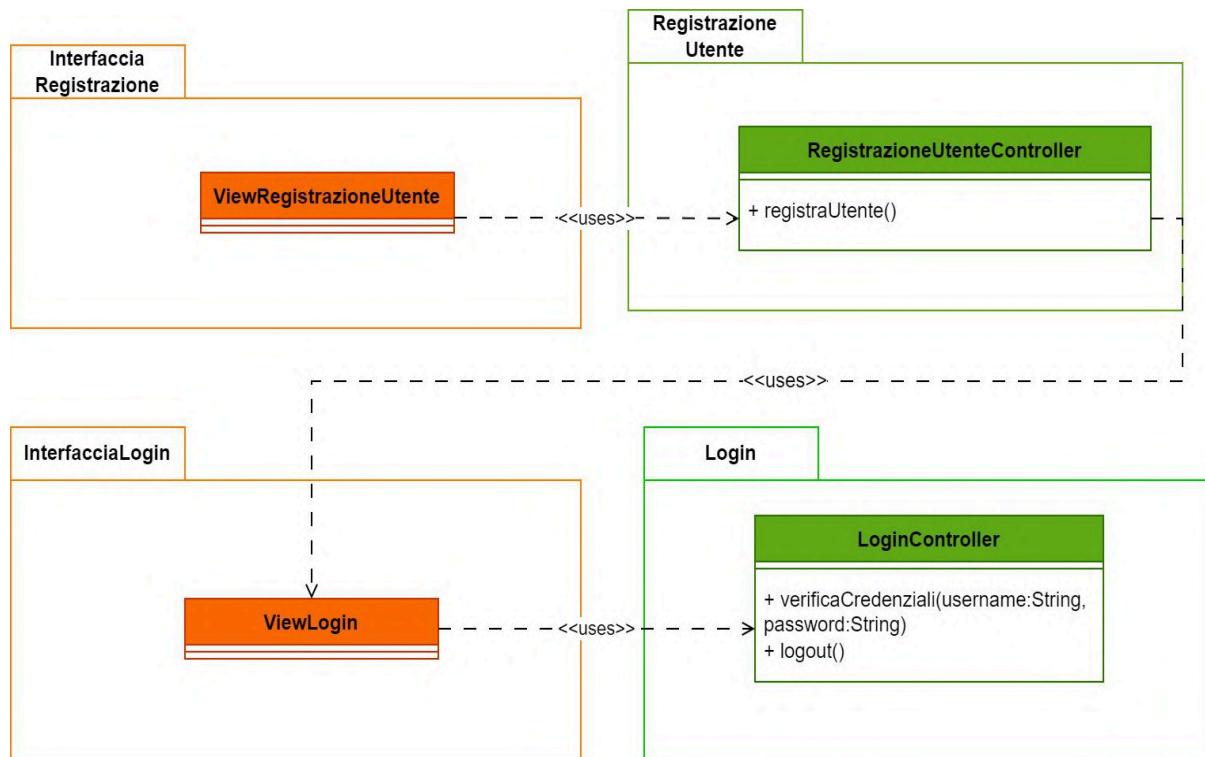


Diagramma delle classi: Scrittura Log

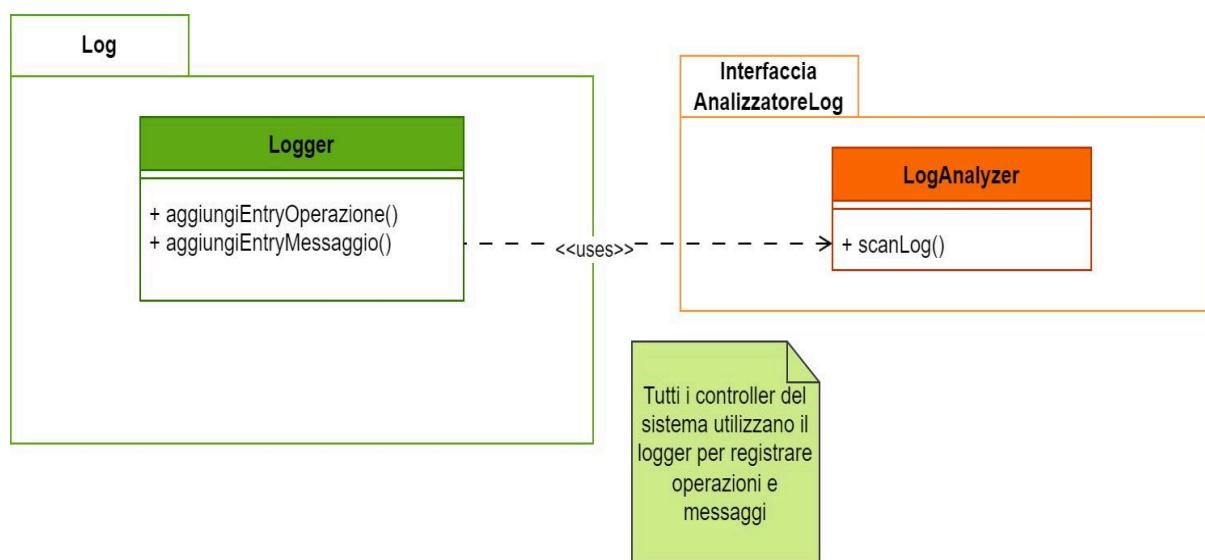


Diagramma delle classi: Admin

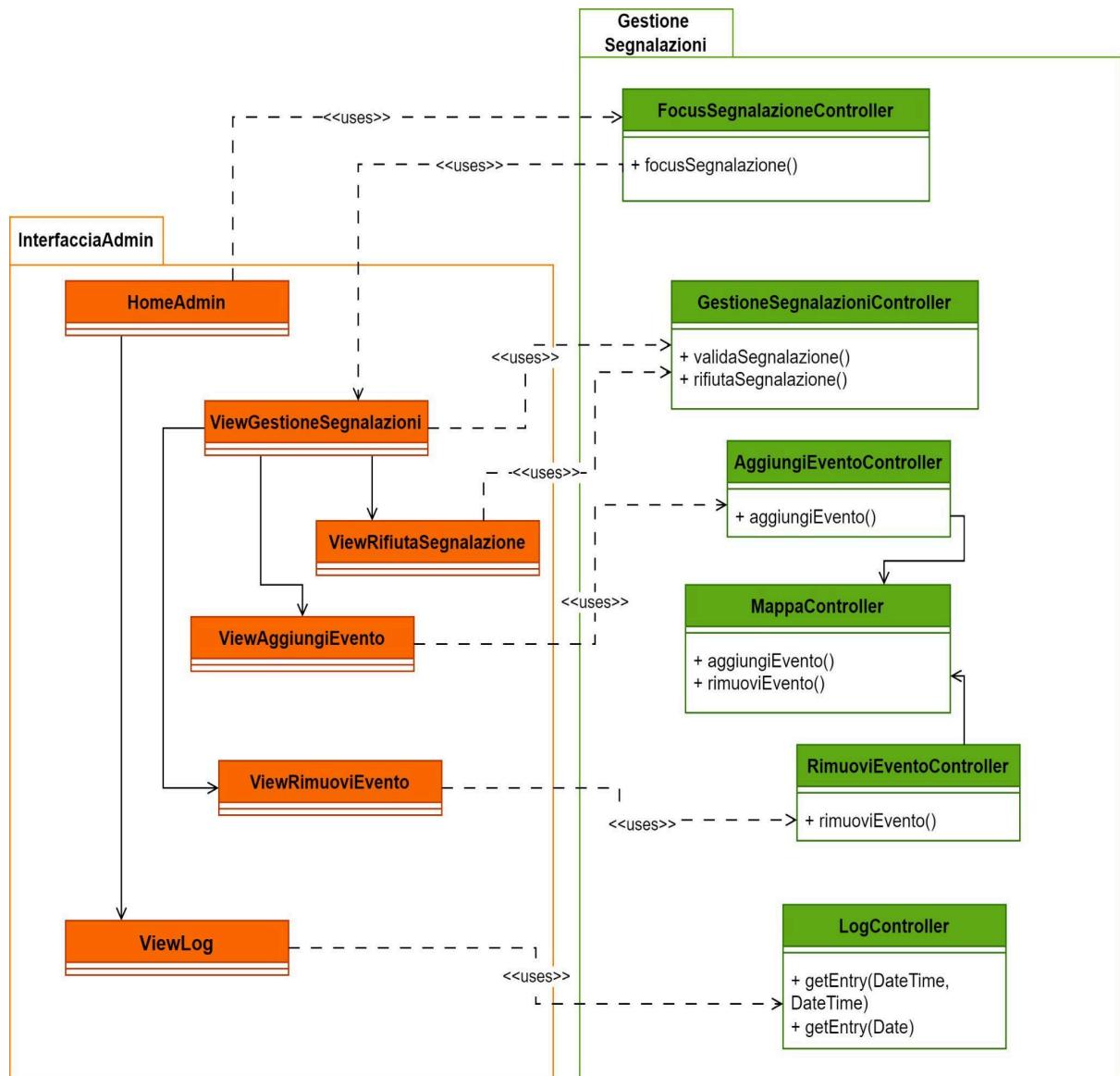
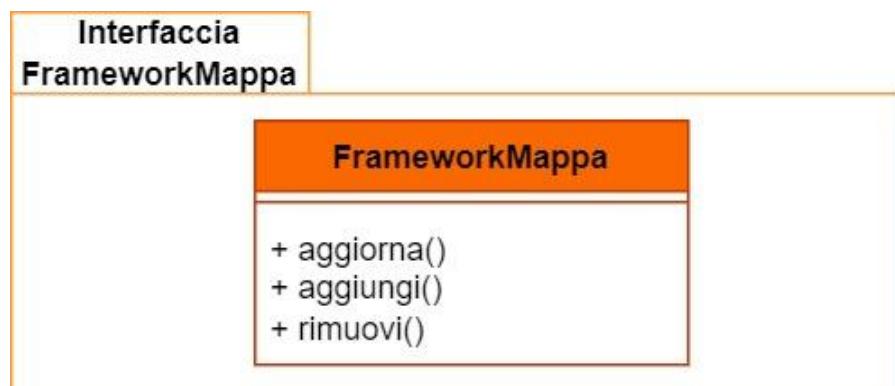
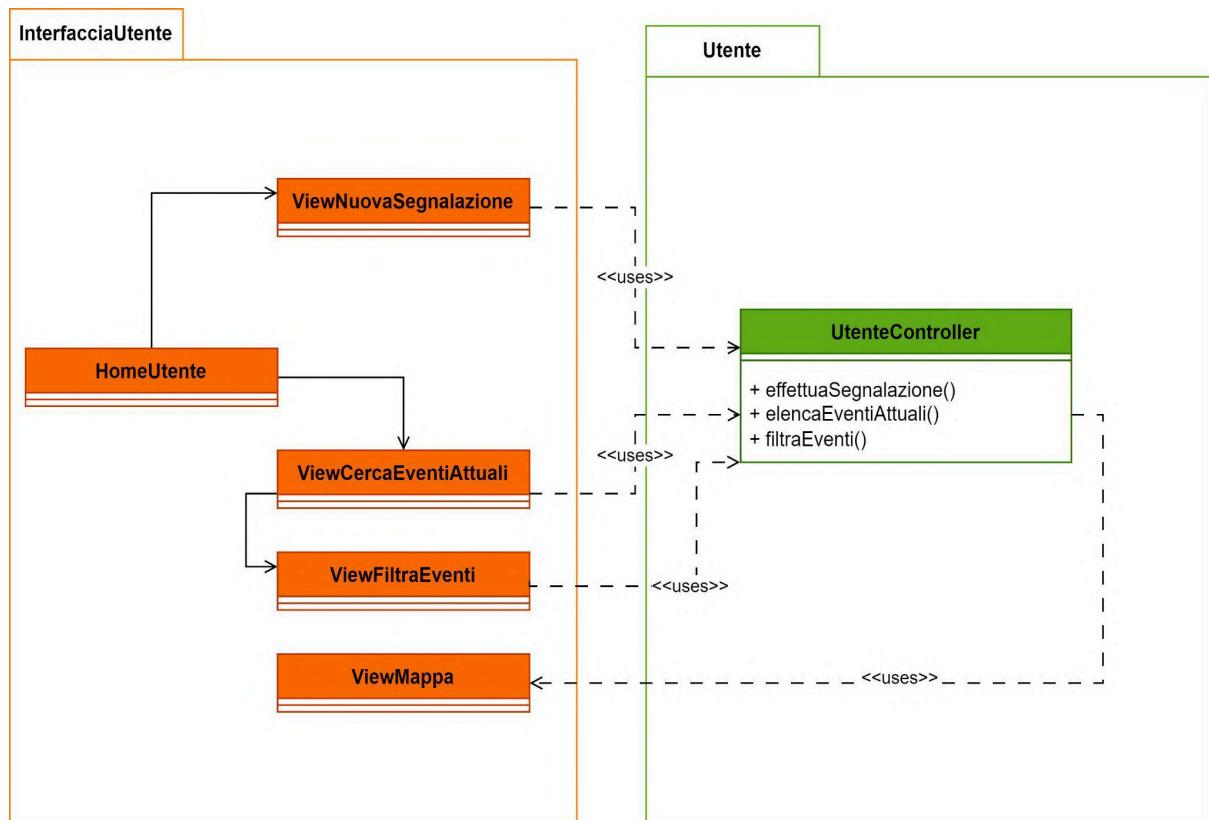


Diagramma delle classi: Framework Mappa



La classe *FrameworkMappa* è utilizzata dalle classi *MappaController* (nel package *GestioneSegnalazioni*) e *UtenteController* (nel package *Utente*). Essa gestisce l'aggiornamento in tempo reale della Mappa in seguito a modifiche del dominio, come l'aggiunta/rimozione di Eventi o l'applicazione di Filtri da parte dell'Utente.

Diagramma delle classi: Utente



Interazione

Diagrammi di sequenza

Diagramma di sequenza: Login con successo

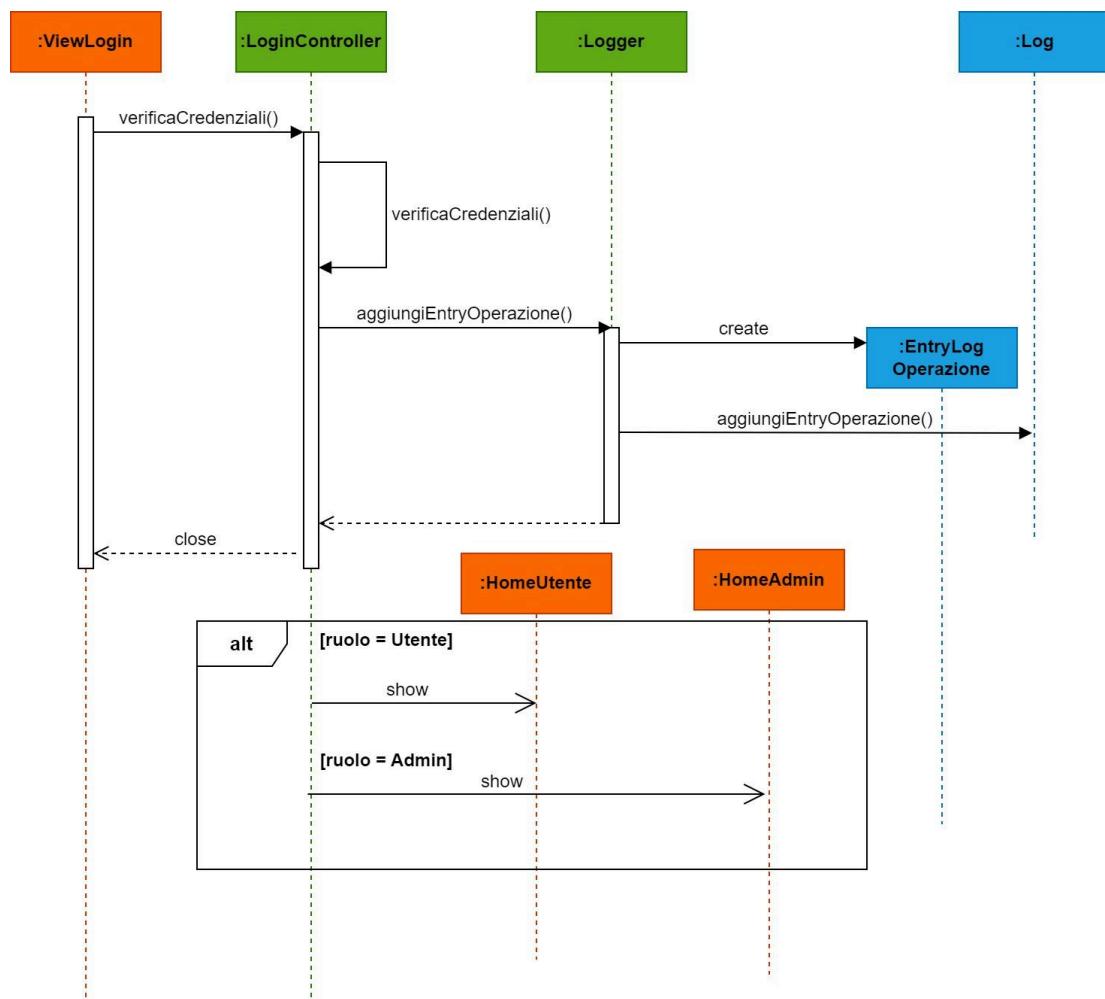


Diagramma di sequenza: NuovaSegnalazione

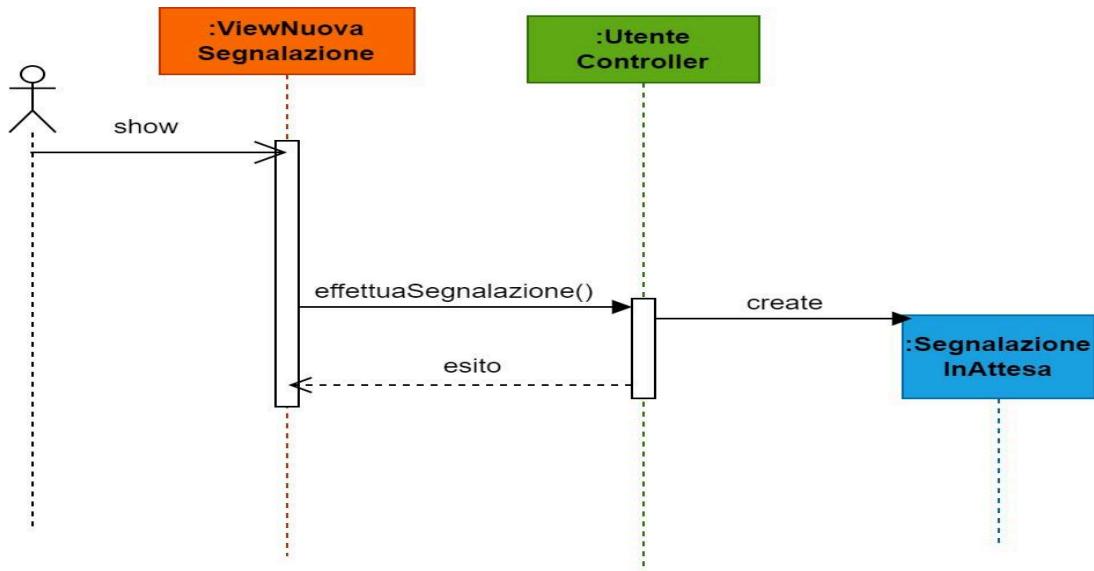


Diagramma di sequenza: CercaEventi e FiltraEventi

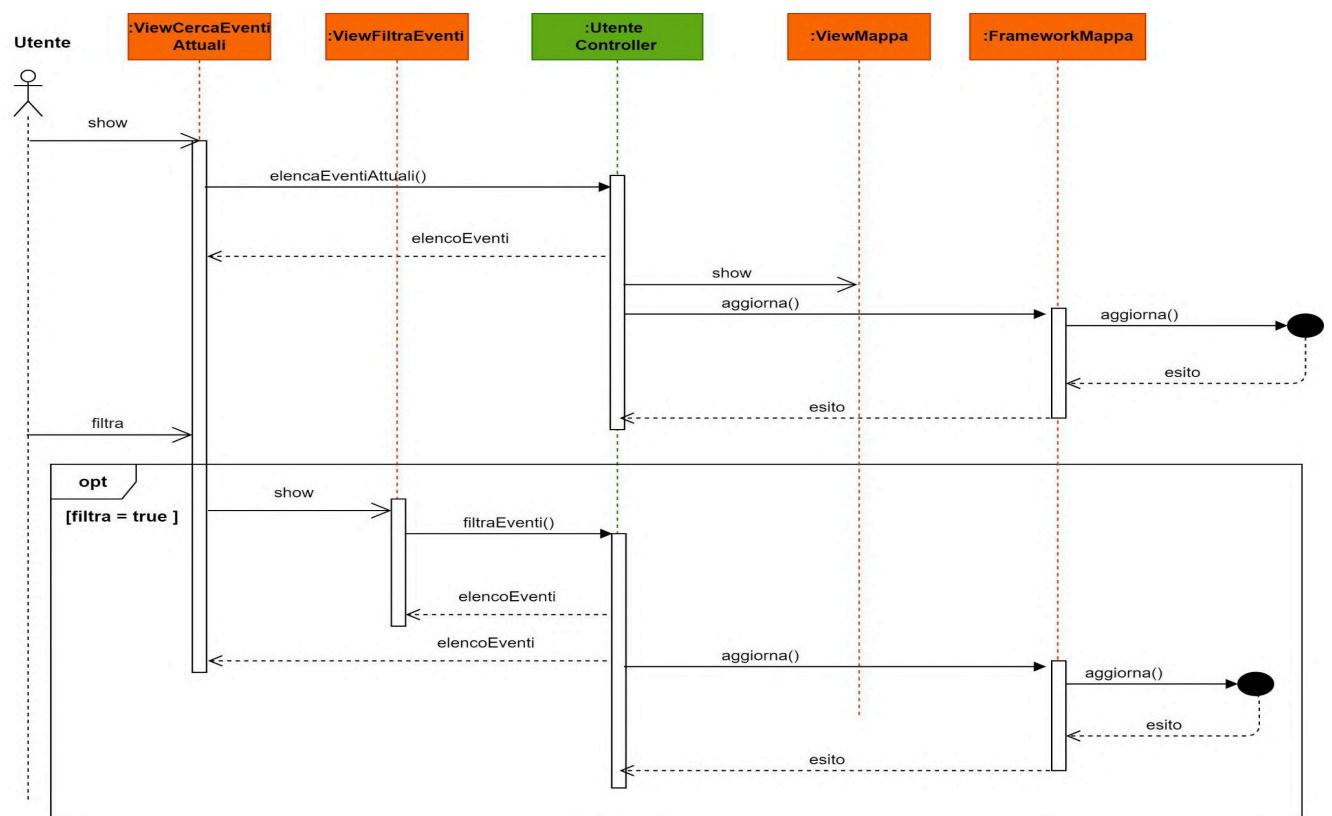


Diagramma di sequenza: RifiutaSegnalazione

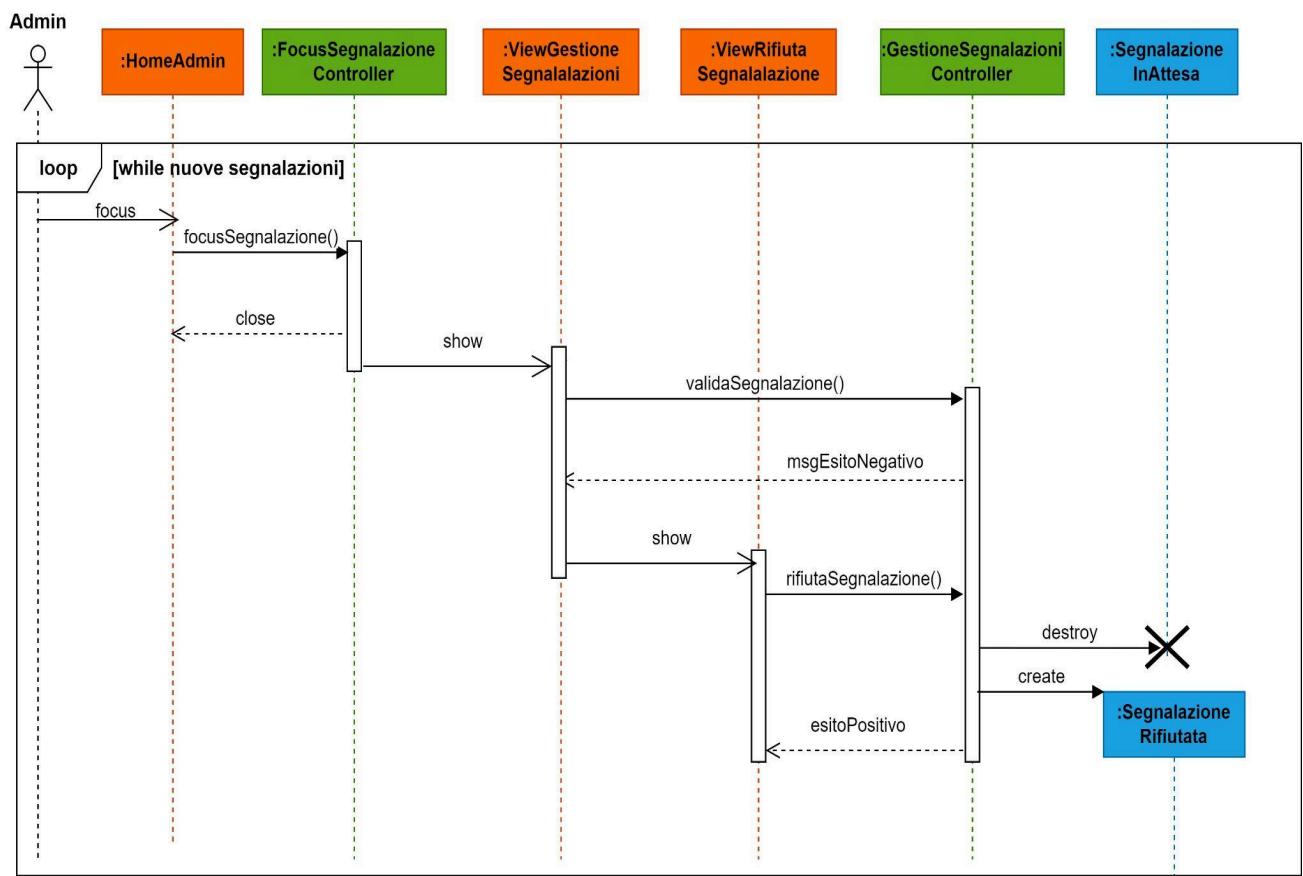
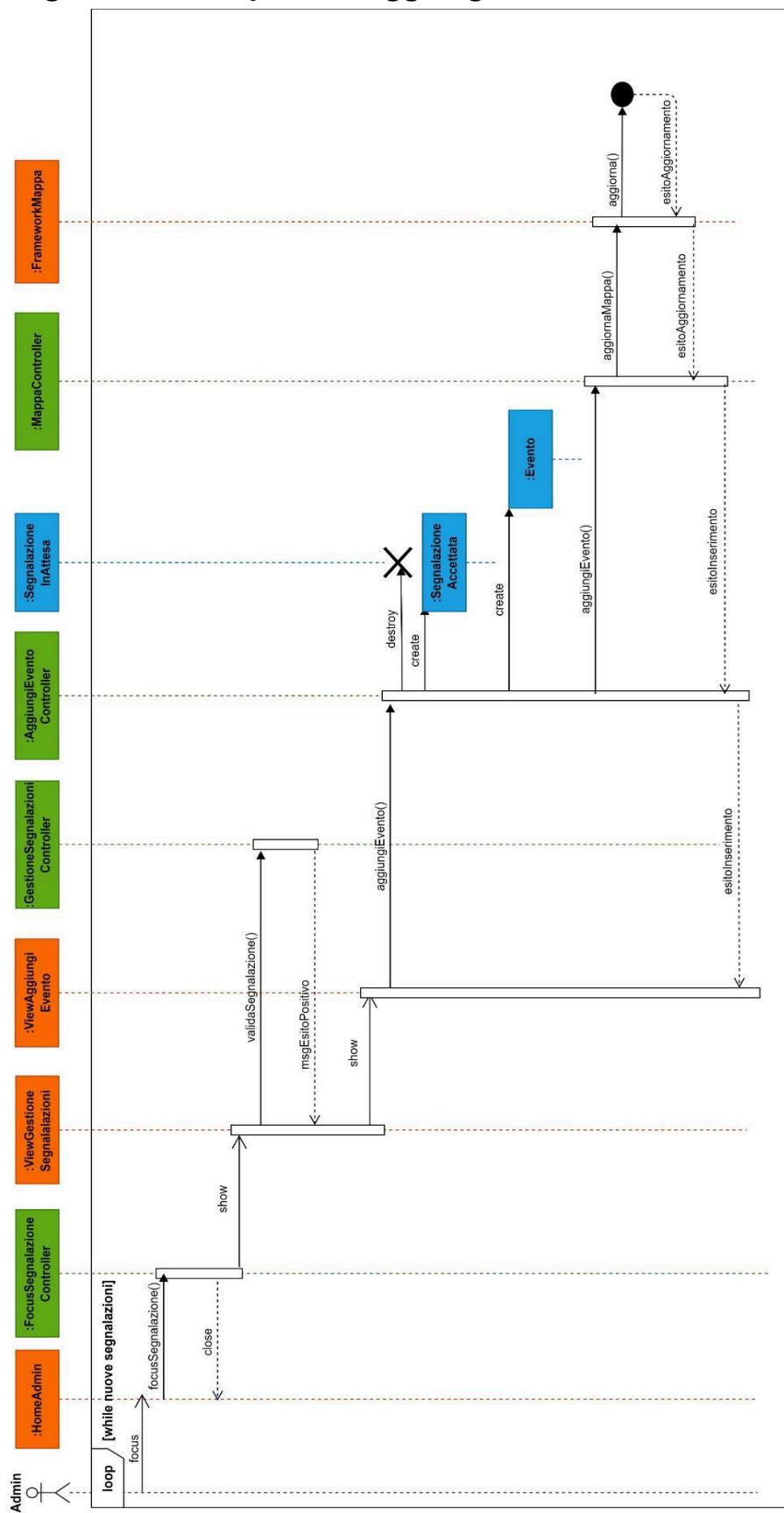


Diagramma di sequenza: AggiungiEvento



Comportamento

Dopo un'attenta valutazione, è stato deciso che non esistono diagrammi di stato rilevanti.

Piano di Lavoro

Il progetto e lo sviluppo del sistema sono assegnati a diversi team:

- Team di Progettazione (Cecchini, Furlani, Singh)
- Team di Sviluppo (Cecchini, Furlani, Singh)
- Team DB (Cecchini, Furlani, Singh)
- Team di Sicurezza (Cecchini, Furlani, Singh)
- Team Grafico (Cecchini, Furlani, Singh)

Package	Progettazione	Sviluppo
InterfacciaAdmin	Team di Progettazione Team Grafico	Team di Sviluppo Team Grafico
InterfacciaUtente	Team di Progettazione Team Grafico	Team di Sviluppo Team Grafico
InterfacciaLogin	Team di Progettazione Team Grafico Team di Sicurezza	Team di Sviluppo Team di Sicurezza Team Grafico
InterfacciaRegistrazione	Team di Progettazione Team Grafico Team di Sicurezza	Team di Sviluppo Team di Sicurezza Team Grafico
InterfacciaLog	Team di Progettazione Team Grafico Team di Sicurezza	Team di Sviluppo Team di Sicurezza Team Grafico
AnalizzatoreLog	Sistema esterno	Sistema esterno
Utente	Team di Progettazione	Team di Sviluppo
GestioneSegnalazioni	Team di Progettazione	Team di Sviluppo
Login	Team di Progettazione Team di Sicurezza	Team di Sviluppo Team di Sicurezza
Log	Team di Progettazione Team di Sicurezza	Team di Sviluppo Team di Sicurezza

Registrazione Utente	Team di Progettazione Team di Sicurezza	Team di Sviluppo Team di Sicurezza
Dominio	Team di Progettazione Team DB	Team di Sviluppo Team DB

Tempi di rilascio previsti:

- Progettazione entro 3 settimane dalla data odierna;
- Sviluppo delle singole parti con collaudo unitario entro 3 settimane rispetto al fine della progettazione;
- Integrazione e test dell'intero sistema entro 2 settimane rispetto alla fine dello sviluppo.

Prototipo

Il primo prototipo implementerà le funzionalità core dell'applicazione: la Segnalazione da parte dell'Utente e la gestione operativa delle Segnalazioni da parte dell'Admin.

Al contrario non saranno implementati:

- La gestione in tempo reale degli aggiornamenti della mappa;
- La visualizzazione dei log da parte dell'Admin;
- La ricerca di eventi da parte dell'Utente.

Sviluppi futuri

Il committente ha richiesto che nei prossimi anni si provveda anche ad aggiungere le seguenti funzionalità:

- ❖ Invio di notifiche push all'utente, allo scopo di comunicare variazioni di stato delle segnalazioni che ha effettuato;
- ❖ Servizi di geolocalizzazione, per identificare la posizione dell'utente ed assistarlo nelle segnalazioni o nelle ricerche degli eventi;
- ❖ Supporto ad altre lingue in aggiunta all'italiano.

Si richiede al Team progettazione di tenere conto di questi sviluppi futuri

Piano del Collaudo

Di seguito vengono riportati i test di alcune delle classi del dominio realizzati

mediante JUnit.

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

public class EventoTest {

    private Evento evento;

    @BeforeEach
    void setUp() {
        evento = new Evento(100, "Incidente", "Via Roma 12",
"Incidente tra auto e autobus. Detriti su carreggiata",
Gravita.GIALLO);
    }

    @Test
    void testCostruttore() {
        assertEquals(100, evento.getIdEvento());
        assertEquals("Incidente", evento.getTipoEvento());
        assertEquals("Via Roma 12", evento.getLuogo());
        assertEquals("Incidente tra auto e autobus. Detriti su
carreggiata",
            evento.getDescrizione());
        assertEquals(Gravita.GIALLO, evento.getGravita());
    }

    @Test
    void testSetIdEvento() {
        evento.setIdEvento(200);
        assertEquals(200, evento.getIdEvento());
    }

    @Test
    void testSetTipoEvento() {
        evento.setTipoEvento("Allagamento");
        assertEquals("Allagamento", evento.getTipoEvento());
    }
}
```

```
@Test
void testSetLuogo() {
    evento.setLuogo("Via Milano 12");
    assertEquals("Via Milano 12", evento.getLuogo());
}

@Test
void testSetDescrizione() {
    evento.setDescrizione("Strada non percorribile");
    assertEquals("Strada non percorribile",
    evento.getDescrizione());
}

@Test
void testSetGravita() {
    evento.setGravita(Gravita.GIALLO);
    assertEquals(Gravita.GIALLO, evento.getGravita());
}
}

public class SegnalazioneTest {

    private Segnalazione segnalazione;

    @BeforeEach
    void setUp() {

        segnalazione = new Segnalazione(10, new Utente(1, "Mario",
        "Rossi", "marioRossi74"), "Incidente stradale",
        "Incidente tra 2 auto. Carreggiata bloccata",
        TipoSegnalazione.NUOVO_EVENTO, "Via Emilia 233", new
        Immagine("fotoIncidente1.jpg"));

    }

    @Test
    void testCostruttore() {
        assertEquals(10, segnalazione.getIdSegnalazione());
        assertEquals(1, segnalazione.getAutore().getId());
    }
}
```

```
        assertEquals("Incidente stradale",
segnalazione.getTipoEvento());
        assertEquals("Incidente tra 2 auto. Carreggiata bloccata",
segnalazione.getDescrizione());
        assertEquals(TipoSegnalazione.NUOVO_EVENTO,
segnalazione.getTipoSegnalazione());
        assertEquals("Via Emilia 233", segnalazione.getLuogo());
        assertEquals("fotoIncidente1.jpg",
segnalazione.getImmagine().getNomeImmagine());
    }

    @Test
    void testSetId() {
        segnalazione.setIdSegnalazione(20);
        assertEquals(20, segnalazione.getIdSegnalazione());
    }

    @Test
    void testSetTipoEvento() {
        segnalazione.setTipoEvento("Frana");
        assertEquals("Frana", segnalazione.getTipoEvento());
    }

    @Test
    void testSetDescrizione() {
        segnalazione.setDescrizione("Frana di una parte della
collina");
        assertEquals("Frana di una parte della collina",
segnalazione.getDescrizione());
    }

    @Test
    void testSetTipoSegnalazione() {

        segnalazione.setTipoSegnalazione(TipoSegnalazione.NUOVO_EVENTO);
        assertEquals(TipoSegnalazione.NUOVO_EVENTO,
segnalazione.getTipoSegnalazione());
```

```
}

@Test
void testSetLuogo() {
    segnalazione.setLuogo("Via Marconi 23");
    assertEquals("Via Marconi 23", segnalazione.getLuogo());
}

@Test
void testSetNameFoto() {
    segnalazione.setNomeImmagine("fotoFrana.png");
    assertEquals("fotoFrana.png",
segnalazione.getFoto().getNomeImmagine());
}
}

public class UtenteTest {

    private Utente utente;

    @BeforeEach
    void setUp() {
        utente = new Utente(1, "Mario", "Rossi", "marioRossi74");
    }

    @Test
    void testCostruttore() {
        assertEquals(1, utente.getId());
        assertEquals("Mario", utente.getNome());
        assertEquals("Rossi", utente.getCognome());
        assertEquals("marioRossi74", utente.getUsername());
    }

    @Test
    void testSetIdUser() {
        utente.setId(2);
        assertEquals(2, utente.getId());
    }
}
```

```
@Test
void testSetNome() {
    utente.setNome("nuovoNome");
    assertEquals("nuovoNome", utente.getNome());
}

@Test
void testSetCognome() {
    utente.setCognome("nuovoCognome");
    assertEquals("nuovoCognome", utente.getCognome());
}

@Test
void testSetUsername() {
    utente.setUsername("nuovoUser");
    assertEquals("nuovoUser", utente.getUsername());
}

}
```

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:

- Sicurezza
- Controllo degli accessi
- Usabilità
- Tempo di risposta

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. A tal proposito verrà posta particolare attenzione alla progettazione delle interfacce utente ed alla fruibilità delle stesse anche su dispositivi mobile o dispositivi con risorse di calcolo e/o di memoria limitate.

Il sistema deve anche essere in grado di garantire un certo livello di scalabilità, garantendo simili tempi di risposta a richieste simultanee di molteplici utenti.

Notiamo come i due requisiti sopracitati (usabilità e tempo di risposta) siano in contrasto con i requisiti di sicurezza e di controllo degli accessi, che sono molto importanti in quanto la manomissione di informazioni riguardanti gli eventi, gli amministratori o gli utenti potrebbe condurre a gravi perdite d'immagine e di affidabilità come messo in luce dalla "Tabella Valutazione Beni". A tal proposito introduciamo un protocollo TLS di comunicazione sicura ed il controllo degli accessi, i quali permettono, rispettivamente:

- La trasmissione dei dati in sicurezza;
- L'accesso esclusivamente ad attori autorizzati.

L'utilizzo di questi approcci porta ad un lieve peggioramento dei tempi di risposta del sistema e ad una leggera diminuzione dell'usabilità del sistema da parte dell'utente che deve effettuare le procedure di registrazione e di autenticazione. D'altro canto, tale sistema si interfaccia esclusivamente con operatori umani, che spesso non percepiscono lievi scostamenti di qualche secondo nei tempi di risposta.

Scelta dell'architettura

Dal punto di vista architettonico, l'architettura più idonea per questo tipo di sistema è un'architettura client/server a tre livelli, organizzata nel modo seguente:

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
- Vari server per la logica applicativa (Server per le funzionalità dell'Amministratore, Server per le funzionalità dell'utente)
- Un server per la gestione dei Login
- Un server per la gestione della Registrazione

L3 - Persistenza

Per la gestione della persistenza si avrà un server dedicato nel quale sarà installato un opportuno DBMS che gestirà in modo completo la base dati dell'applicazione ed un ulteriore server sarà dedicato alla memorizzazione dei log. Tale scelta è stata presa in modo tale che eventuali compromissioni di una macchina non alterino l'altra.

La metodologia "forza bruta", con l'utilizzo di metodi CRUD, permetterà l'interfacciamento con il DBMS.

Scelte tecnologiche

L'applicazione sarà inizialmente disponibile in versione web, con successivo sviluppo mobile. L'architettura presuppone che i client siano serviti agli utenti direttamente dai server.

Inoltre, si è scelto di adottare il pattern *Broker* per la gestione della sessione, in quanto consente di introdurre un livello di astrazione e sicurezza. Questo approccio è particolarmente utile per disaccoppiare client e server, poiché il *Broker* funge da

mediatore tra le parti, semplificando l'aggiunta o la rimozione di server, nonché il riutilizzo di componenti progettati in precedenza o per sistemi differenti. Considerata la natura web dell'applicazione, si utilizzerà il *Broker* integrato nel server web.

Per lo sviluppo dell'applicazione utilizzeremo Node.js per la parte backend (lato server) e React per il frontend (interfaccia utente). In questo modo è possibile creare un'app moderna, scalabile e modulare.

Per quanto riguarda la mappa interattiva, utilizzeremo le API di Leaflet, una libreria JavaScript open source leggera e flessibile per la visualizzazione di mappe su web.

Per la geocodifica e reverse geocoding (cioè la conversione tra indirizzi e coordinate geografiche e viceversa), utilizzeremo un servizio esterno di geocoding API, che consente agli utenti di cercare un luogo e visualizzarlo sulla mappa, oppure ottenere un indirizzo a partire da una posizione geografica.

Per il deployment si farà uso di pacchetti `.war`.

Diagramma dei package

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

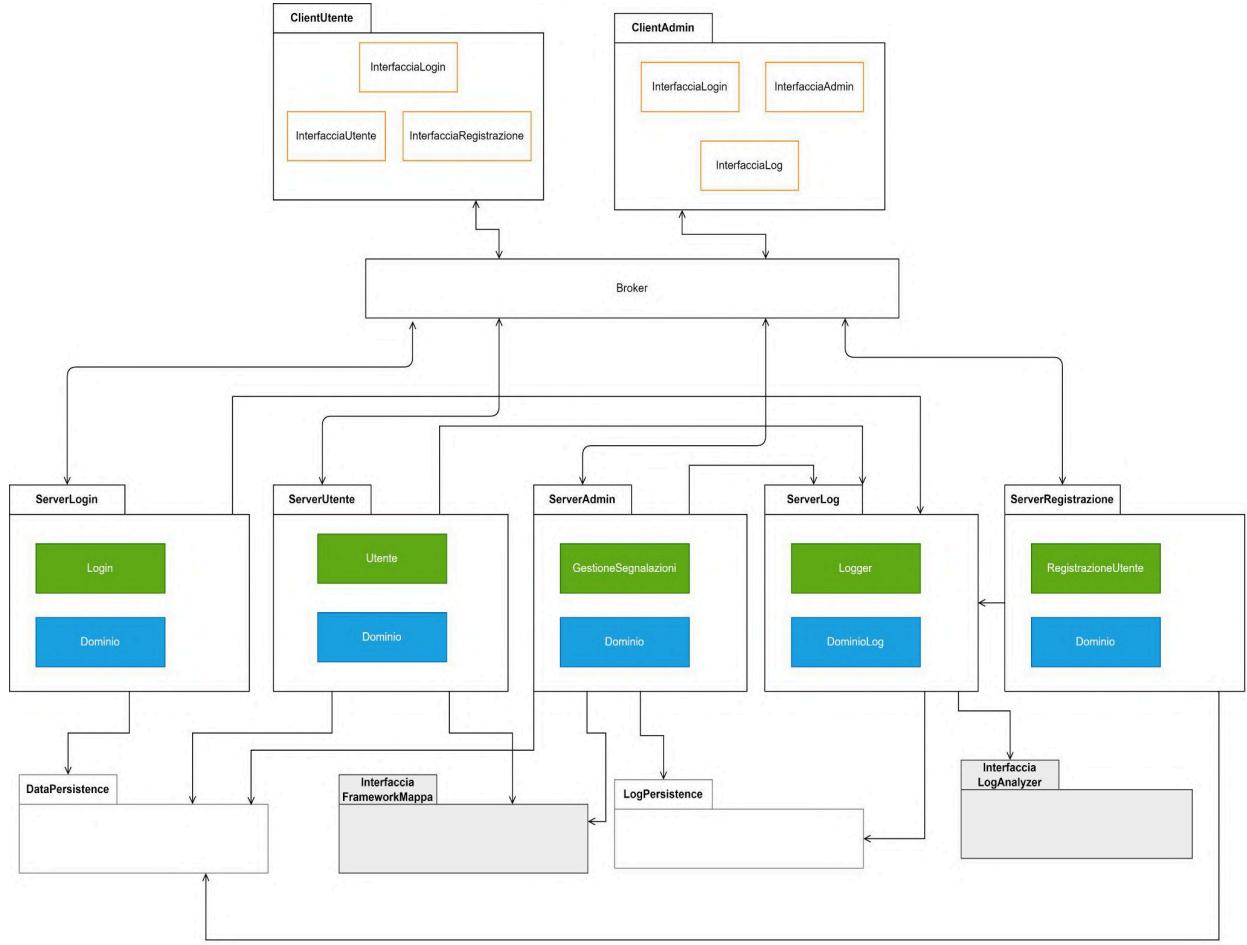
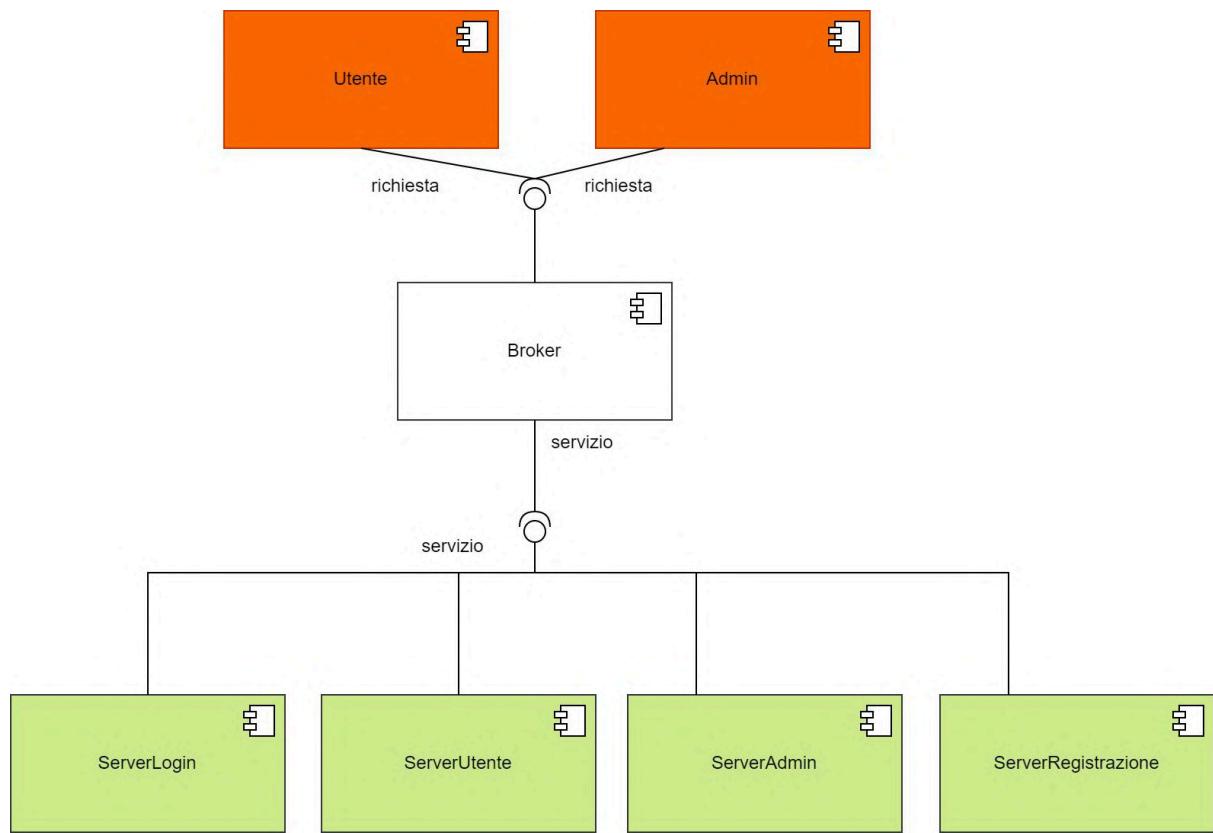
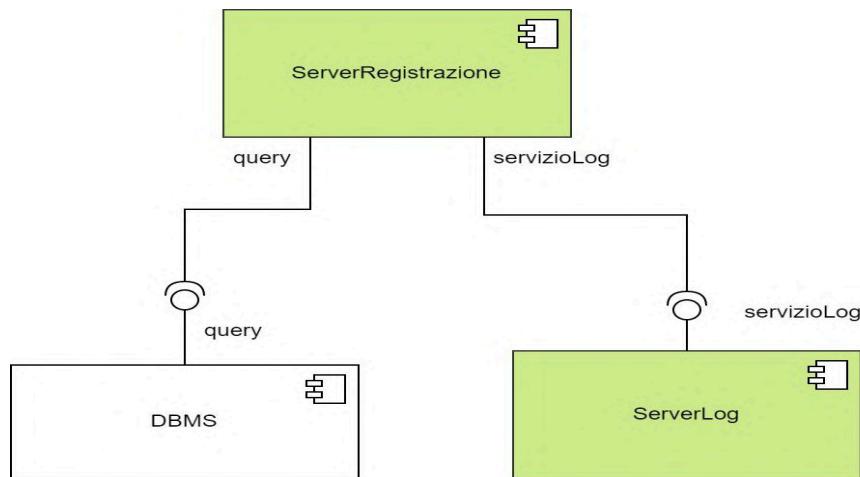


Diagramma dei componenti

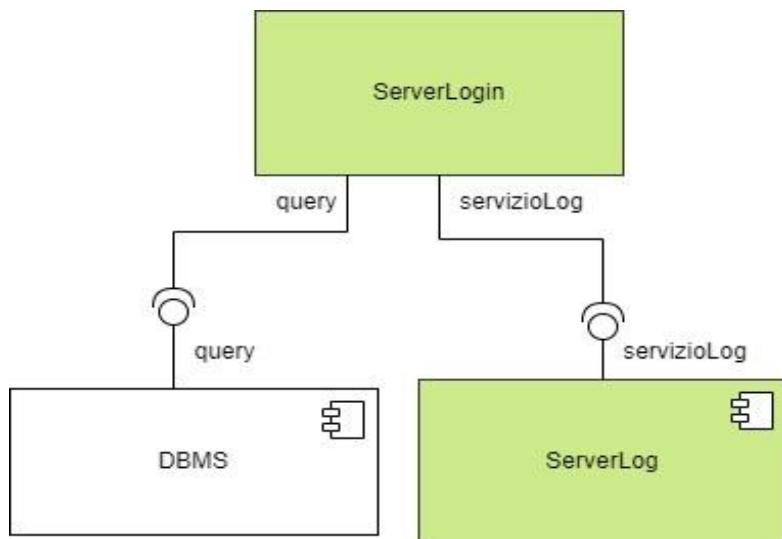
Nella figura sottostante è riportata l'Architettura del Sistema organizzata attraverso un diagramma dei componenti. Le richieste provenienti dai client vengono inviate al broker, il quale tramite connessioni multiple wiring le inoltra al server opportuno. Per ogni server verrà poi analizzato il legame con persistenza e interfacce verso l'esterno.



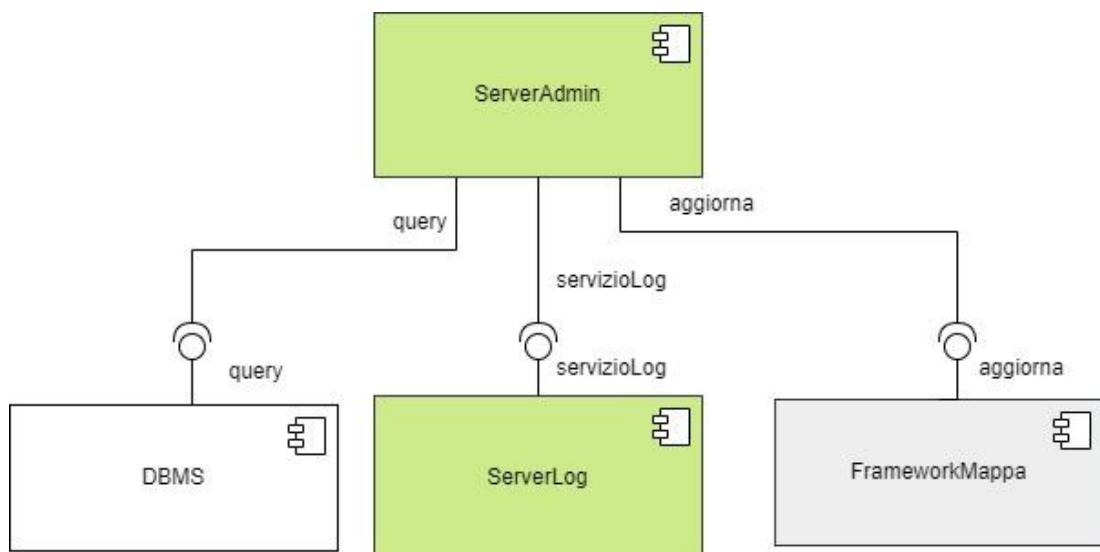
Il componente ServerRegistrazione:



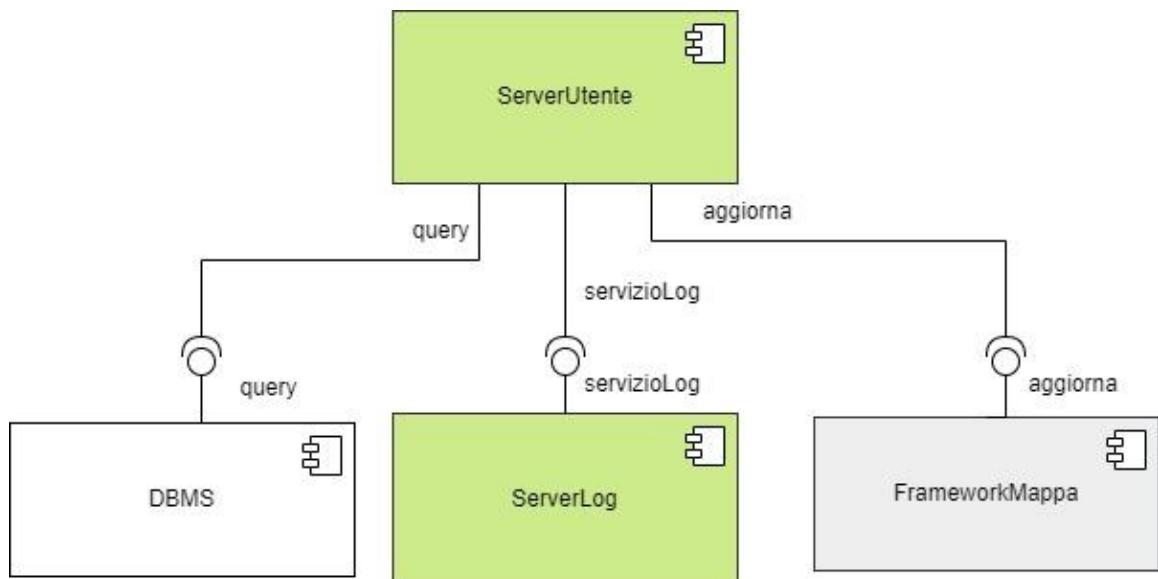
Il componente ServerLogin:



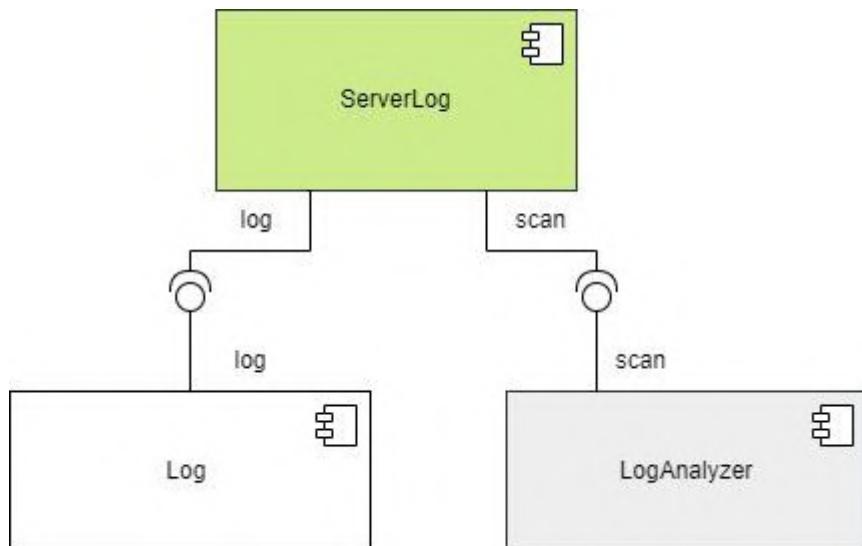
Il componente ServerAdmin:



Il componente ServerUtente:



Il componente ServerLog:

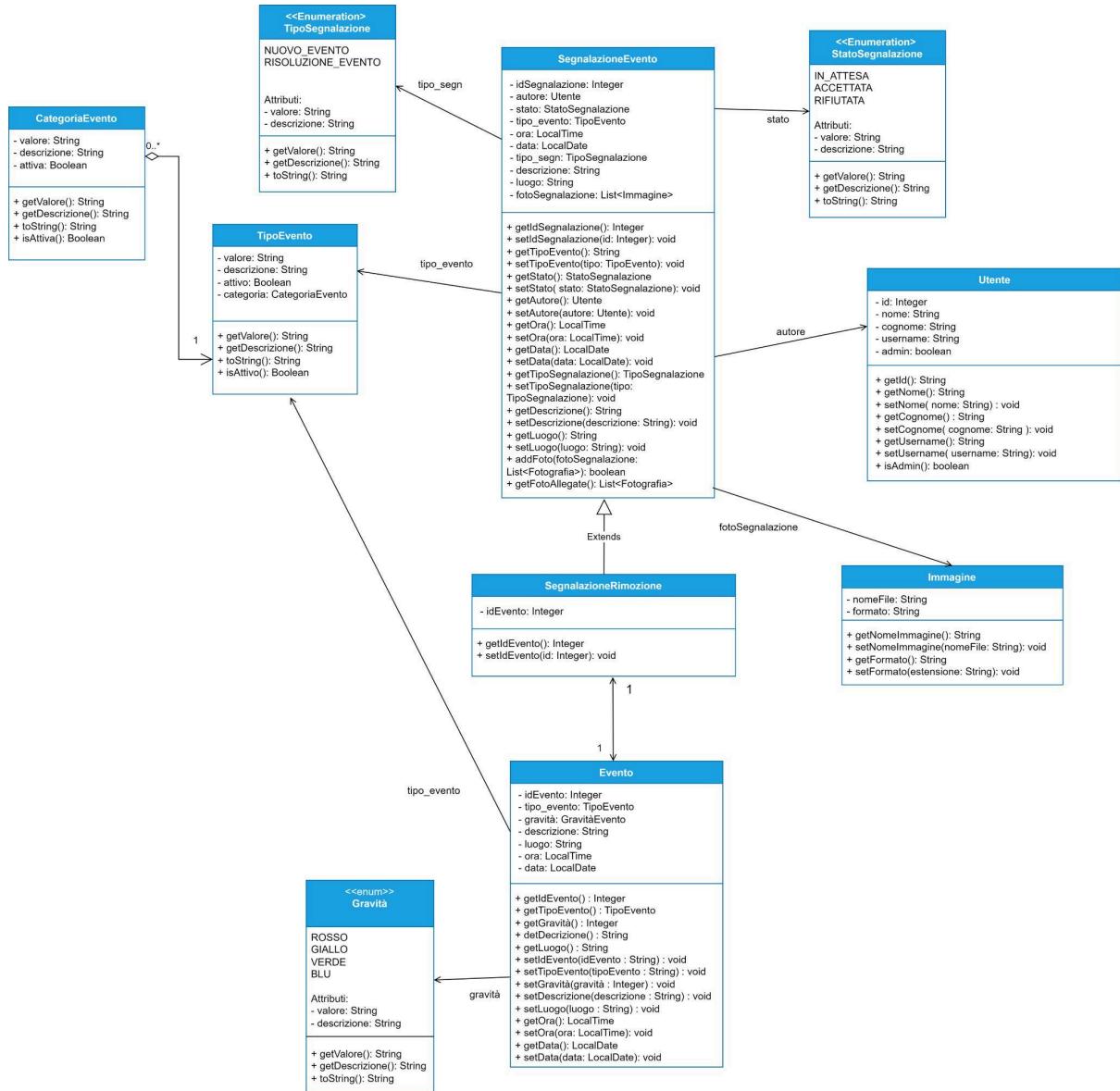


Progettazione di Dettaglio

Struttura

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

Diagramma di Dettaglio: Dominio



Nella fase di analisi, lo stato della segnalazione è stato rappresentato tramite un insieme di sottoclassi, ciascuna corrispondente a un possibile stato. In sede di progettazione, tale struttura è stata semplificata ricorrendo a un attributo enumerato definito direttamente nella classe **Segnalazione**, riducendo la complessità strutturale senza comprometterne la funzionalità.

Nel passaggio dal modello di dominio dell'analisi a quello di progettazione, gli attributi **tipoEvento** e **categoriaEvento** sono stati promossi a classi autonome. Di

conseguenza, è stato introdotto un vincolo secondo cui le istanze delle classi *TipoEvento* e *CategoriaEvento* devono essere create e gestite esclusivamente dal Sistema o dall'Amministratore.

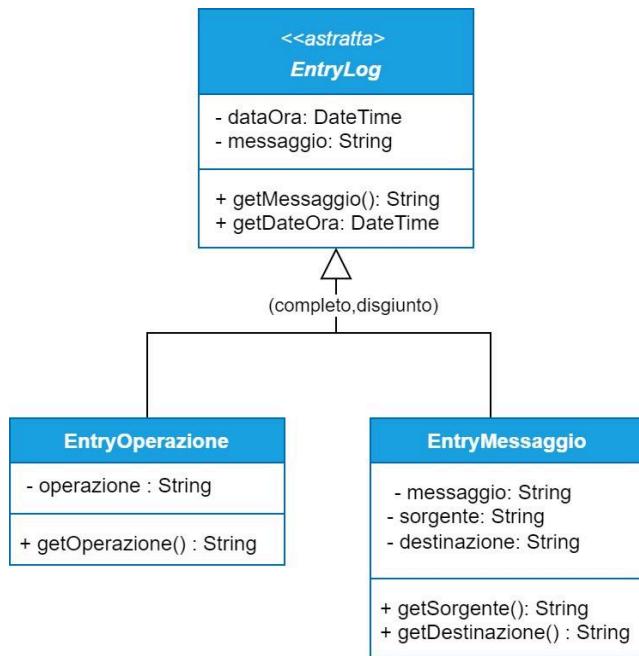
Agli Utenti è consentita unicamente la selezione tra le opzioni predefinite, ad esempio tramite una *combo box*.

Nelle classi *TipoEvento* e *CategoriaEvento* è stato introdotto un attributo booleano (*attivo / attiva*) con lo scopo di gestire la disponibilità di ciascun tipo o categoria. Questo consente, ad esempio, di disattivare la selezione di un certo tipo o categoria nelle nuove segnalazioni da parte degli utenti, o nell'inserimento di eventi da parte dell'Admin — ad esempio in seguito a una fusione con un'altra voce — senza tuttavia alterare le segnalazioni già registrate in passato.

La scelta di collocare l'attributo *livello di gravità* all'interno della classe *Evento* risponde alla necessità di associare livelli di gravità diversi a uno stesso evento. Se avessimo invece inserito questo attributo nella classe *TipoEvento*, non sarebbe stato possibile gestire tali variazioni.

Durante la progettazione, per distinguere tra i ruoli di Admin e Utente, è stato introdotto un attributo booleano *admin* nella classe *Utente*. Questa soluzione semplifica la gestione della persistenza mantenendo un'unica struttura dati condivisa e ottimizza il controllo dei permessi, consentendo una verifica rapida dello stato utente tramite tale attributo. Inoltre, rende possibile un sistema di autenticazione uniforme, in cui il processo di login è centralizzato e il riconoscimento del ruolo avviene subito dopo l'accesso, evitando duplicazioni di codice.

Nella figura seguente è rappresentata la struttura del dominio relativo al log.



Di seguito viene presentato il modello dei filtri, per il quale è stato adottato il pattern Strategy. Questo pattern consente di definire una famiglia di algoritmi intercambiabili, incapsulandoli all'interno di classi separate e permettendo al contesto di selezionare dinamicamente quale strategia utilizzare in base alle esigenze. Nel caso dei filtri, ciò significa che è possibile aggiungere, modificare o sostituire i criteri di filtraggio senza alterare il codice che li utilizza, migliorando così la flessibilità, la manutenibilità e la riusabilità del sistema.

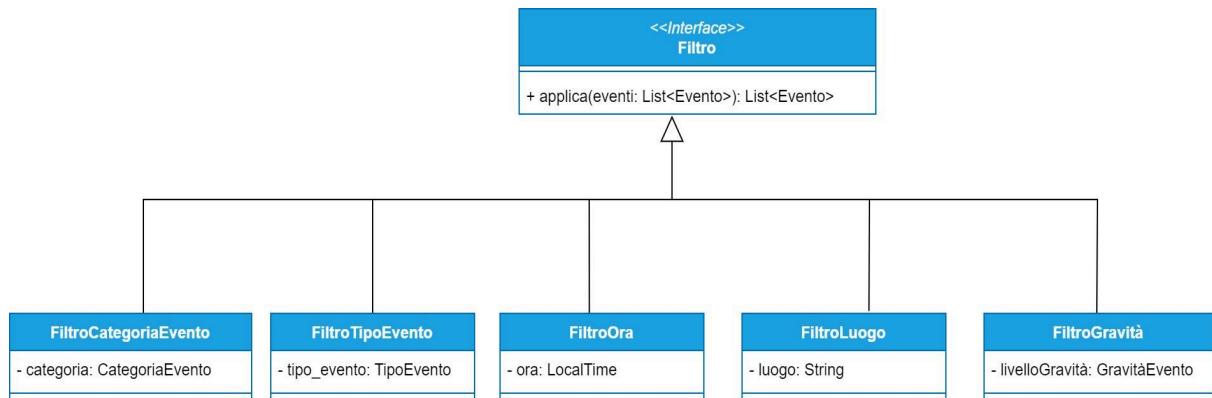
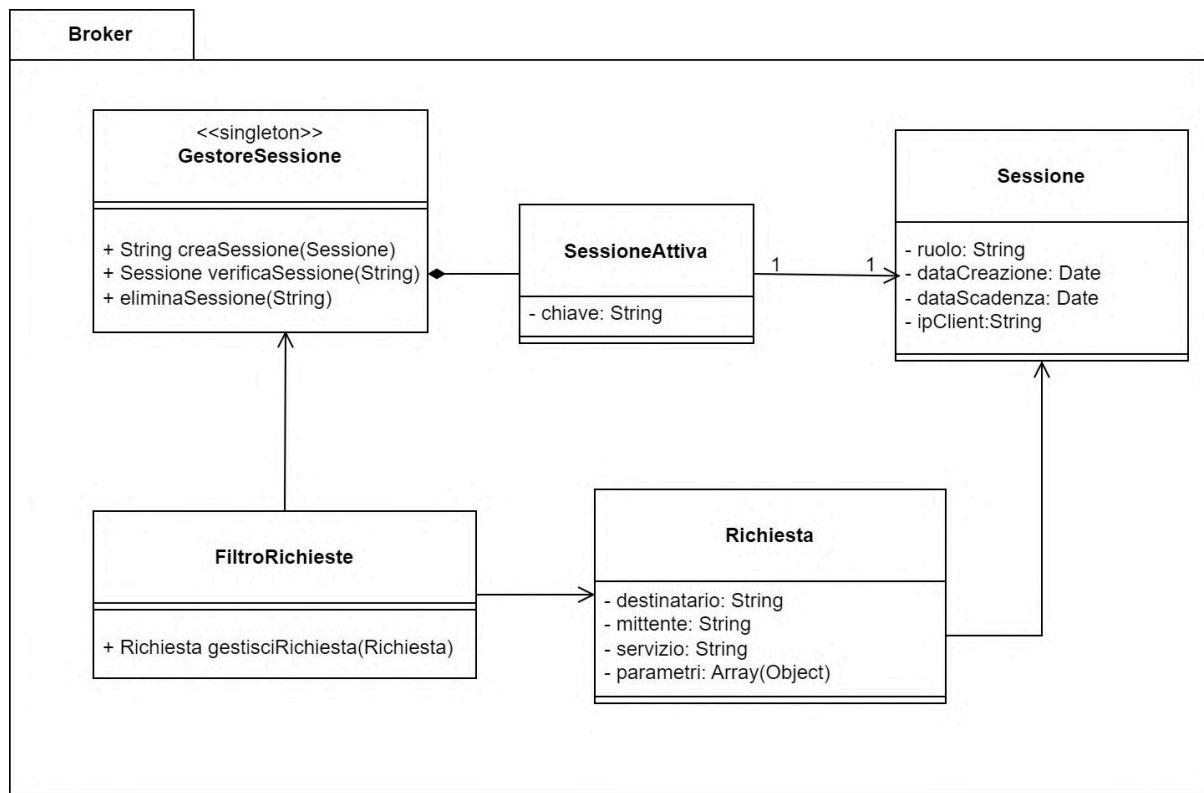


Diagramma di Dettaglio: Broker



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: Interfacce server

<code><<Interface>> ILoginController</code>	<code><<Interface>> IRegistrazioneUtenteController</code>	<code><<Interface>> IAggiungiEventoController</code>
+ verificaCredenziali(username: String, password: String) + logout()	+ registraUtente(nome: String, cognome: String, username: String, password: String)	+ aggiungiEvento(datiSegnalazione: Segnalazione): Boolean
<code><<Interface>> IGestioneSegnalazioniController</code>	<code><<Interface>> IMappaController</code>	<code><<Interface>> IRimuoviEventoController</code>
+ validaSegnalazione(Segnalazione): Boolean + rifiutaSegnalazione(Segnalazione): Boolean + getConnection(): Connection	+ aggiungiEvento(Evento): Boolean + rimuoviEvento(Evento): Boolean	+ rimuoviEvento(Evento): Boolean
<code><<Interface>> IFocusSegnalazioneController</code>	<code><<Interface>> ILogController</code>	<code><<Interface>> ILogger</code>
+ focusSegnalazione(id: Integer): Segnalazione	+ getEntry(DateTime, DateTime): List<EntryLog> + getEntry(Date): List<EntryLog>	+ aggiungiEntryOperazione(String) + aggiungiEntryMessaggio(String, String, String) + aggiungiMonitor(ILogMonitor) + rimuoviMonitor(ILogMonitor) + notifica()
<code><<Interface>> IUtenteController</code>	<code><<Interface>> ILogMonitor</code>	
+ effettuaSegnalazione(Segnalazione): Boolean + filtraEventi(Filtro): List<Evento> + elencaEventiAttuali(): List<Evento>	+ notificaAnalyzer(ILogger)	

Grazie all'utilizzo di queste interfacce sarà possibile applicare il Principio di Inversione delle Dipendenze, svincolando e nascondendo l'implementazione dei servizi offerti ai clienti mediante una dipendenza da sole astrazioni

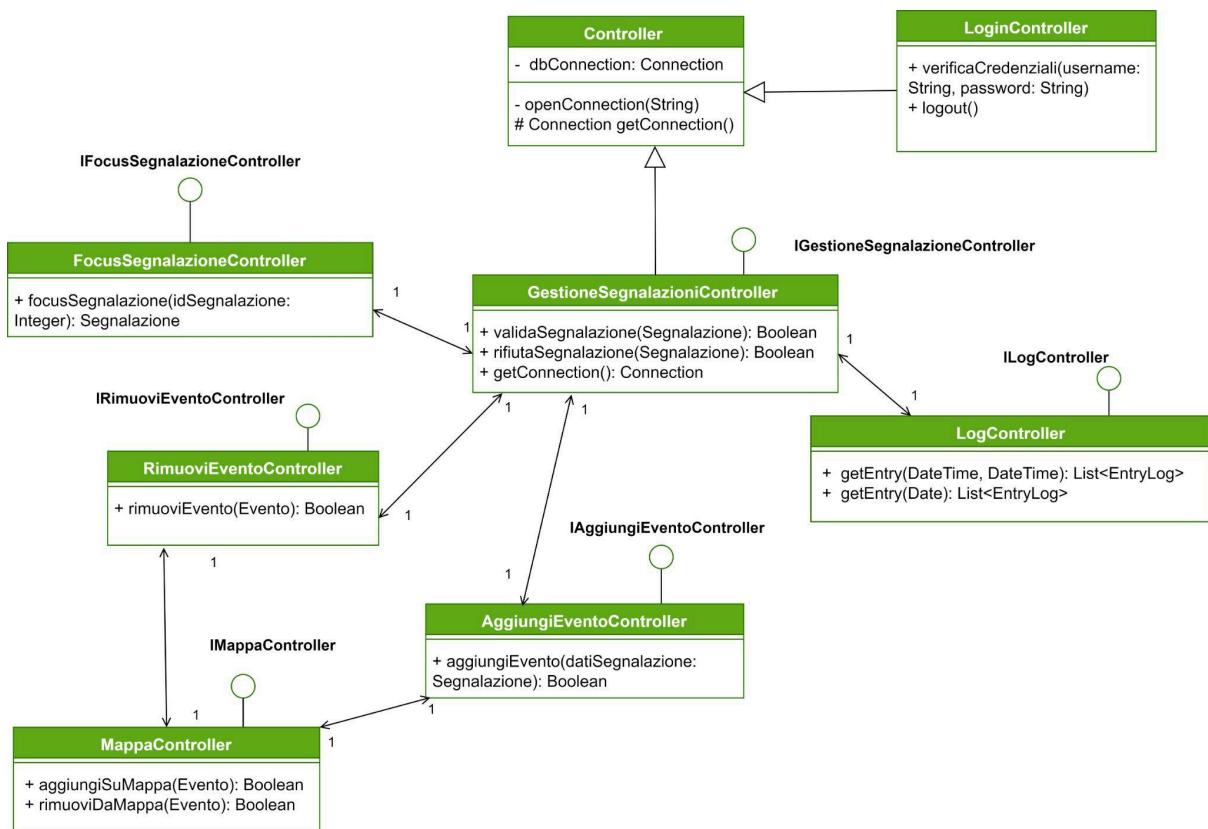
Diagramma di dettaglio: Controller

Admin

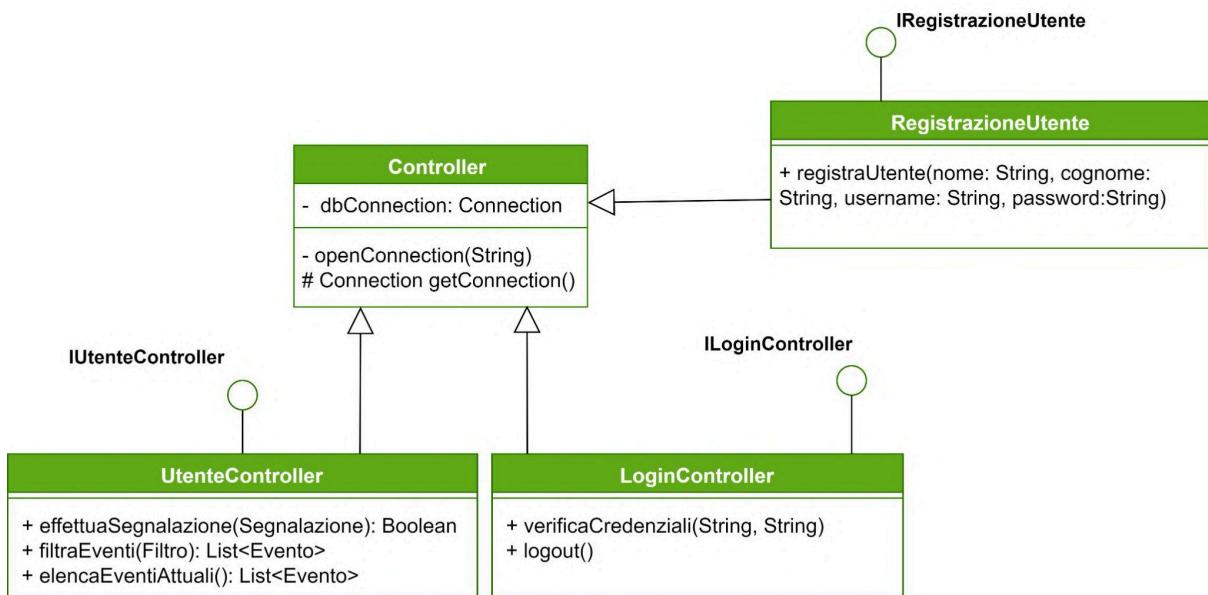
Si è scelto di centralizzare la gestione della persistenza dei dati all'interno della classe *Controller*, al fine di evitare duplicazioni nelle operazioni di lettura e scrittura sul database.

Solo *LoginController* e *GestioneSegnalazioniController* estendono direttamente *Controller*, mentre gli altri controller accedono ai suoi servizi tramite associazioni bidirezionali.

In caso di autenticazione riuscita, *LoginController* associa l'istanza dell'Admin alla sessione corrente, gestita dal *Broker*, in modo da mantenerne lo stato durante l'interazione con il sistema.



Utente



Controller è utilizzato come classe base, implementata da tutti gli altri controller, per la gestione della connessione al DBMS.

In caso di autenticazione riuscita, LoginController associa l'istanza dell'Utente alla sessione corrente, gestita dal Broker, in modo da mantenerne lo stato durante l'interazione con il sistema.

Logger

Tutti i controller utilizzano il Logger.

La classe *LogMonitor*, introdotta in fase di progettazione, ha il compito di monitorare i file scritti dal *Logger*. Per supportare questo comportamento è stato adottato il pattern Observer.

Quando il *Logger* scrive un'entry di log, notifica il *LogMonitor*, che a sua volta informa il sistema esterno *LogAnalyzer*. Quest'ultimo provvede autonomamente all'analisi dei file di log.

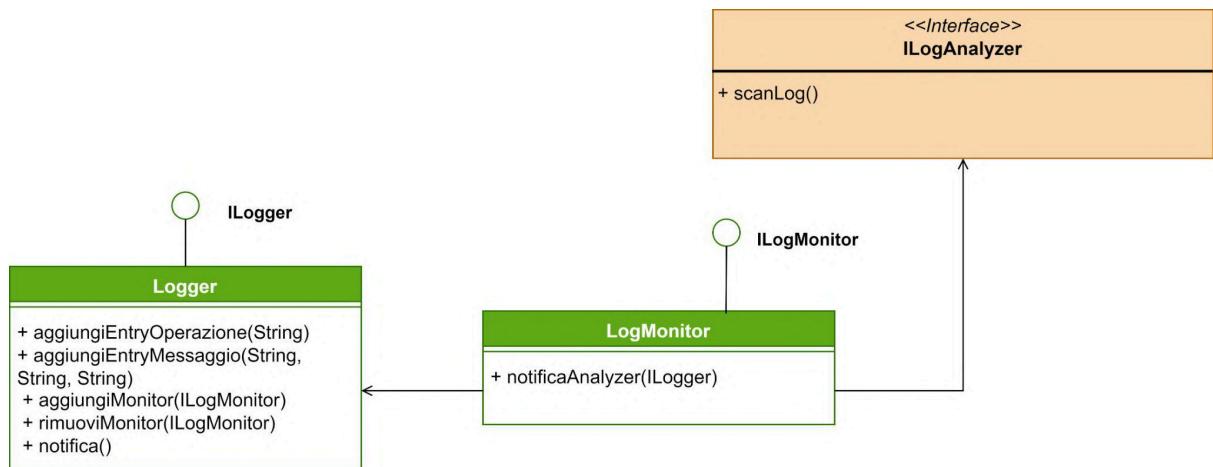
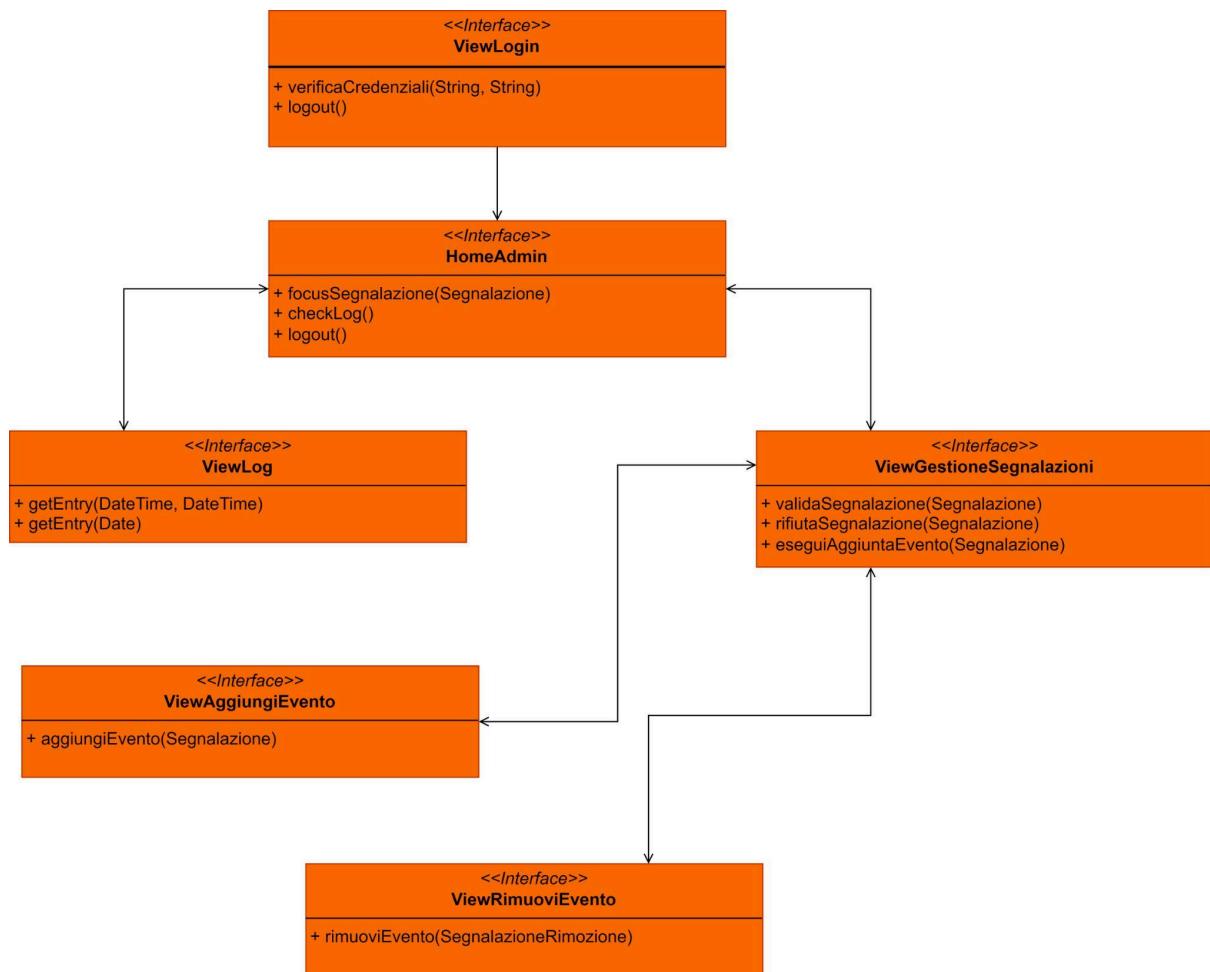


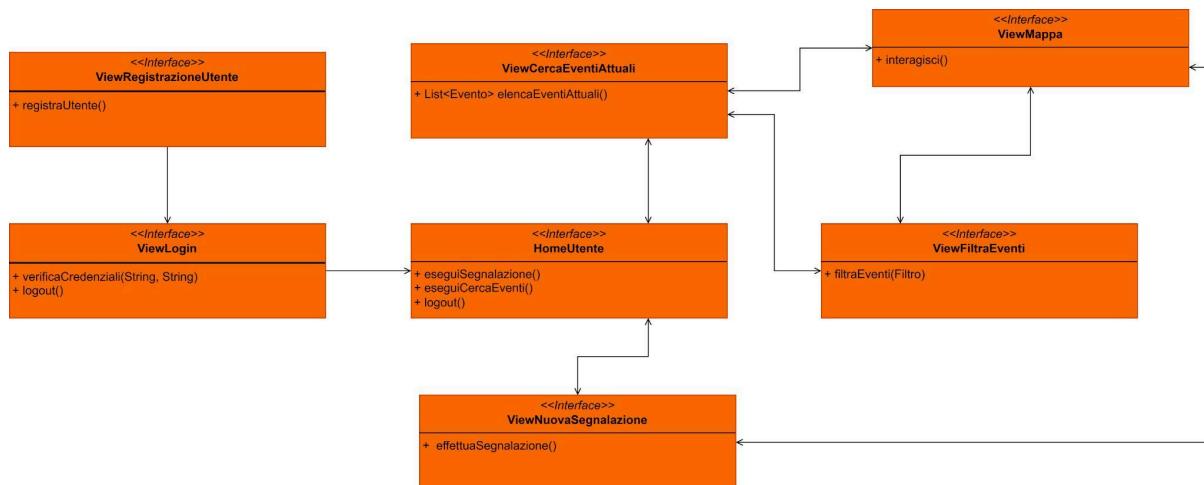
Diagramma di dettaglio: Client

Admin

Nella progettazione, a differenza di quanto previsto nell'analisi, la view per il rifiuto delle segnalazioni(*ViewRifiutaSegnalazione*) non è stata separata, ma incorporata direttamente nella *ViewGestioneSegnalazioni*, per motivi di praticità e per garantire all'utente un'interazione più immediata ed efficiente.

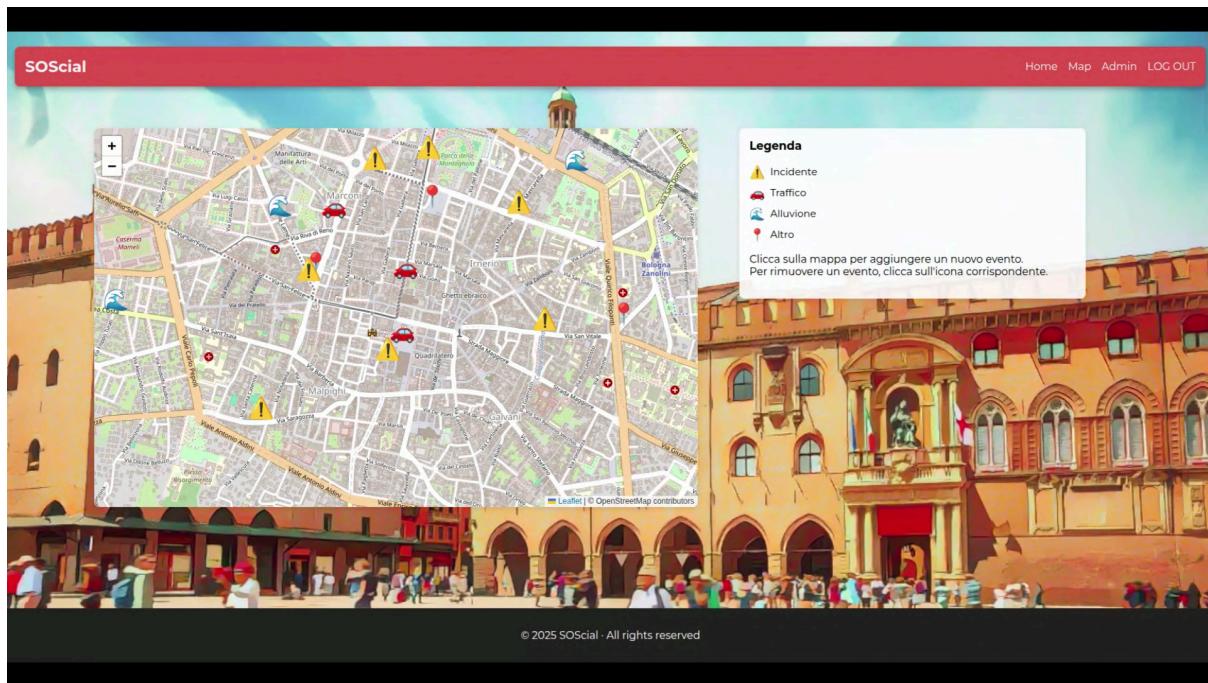


Utente

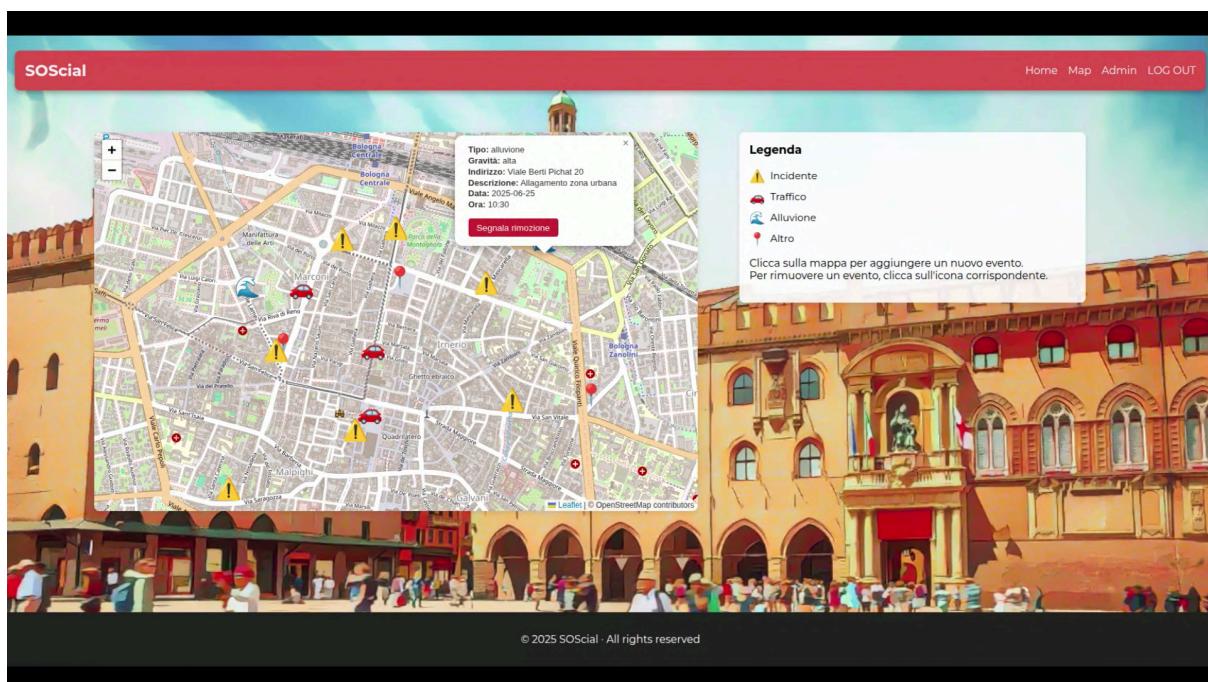


Interfacce Utente

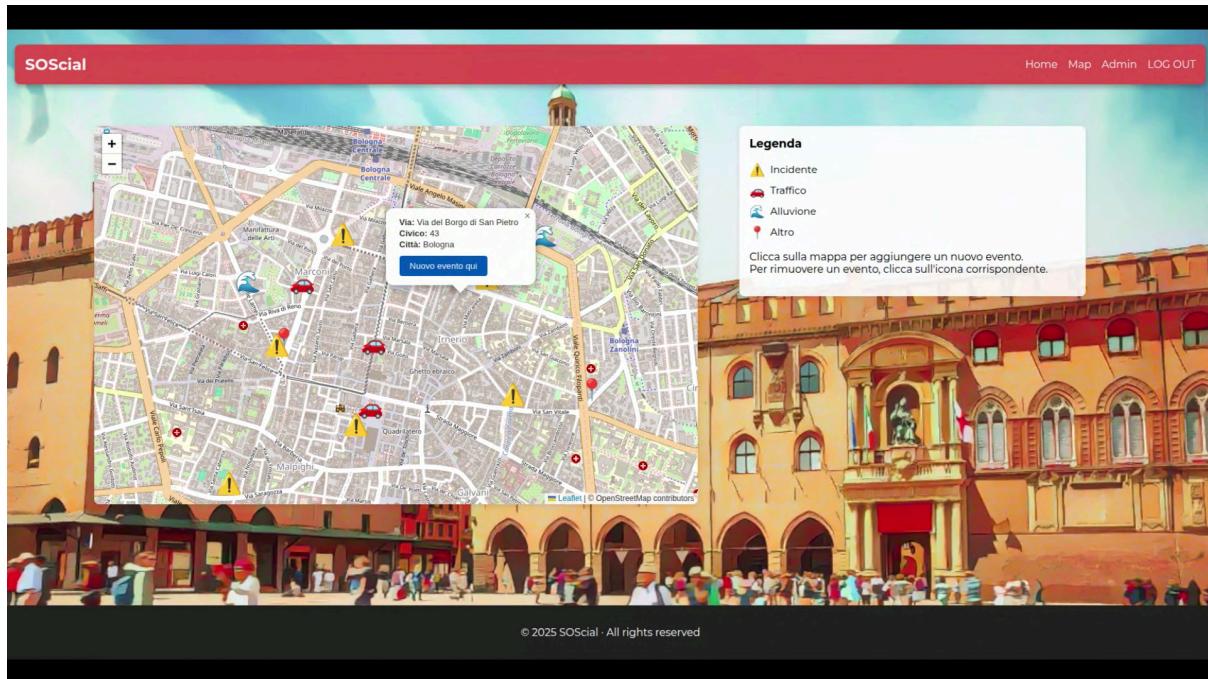
Mappa della città



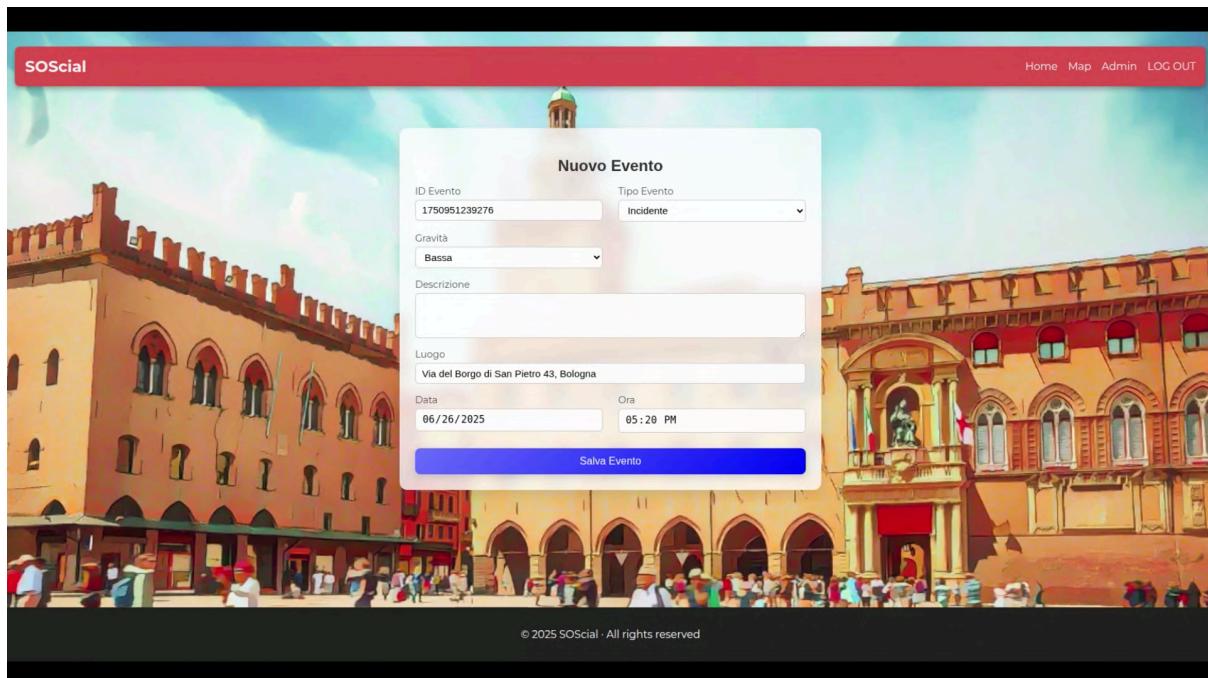
Segnalazione per un Evento risolto



Segnalazione di un nuovo Evento



Nuova Segnalazione



Interfacce Admin

HomeAdmin: lista Segnalazioni

The screenshot shows a web application interface titled "SOocial". At the top right, there are links for "Home", "Map", "Admin", and "LOG OUT". The main content area is titled "Lista Segnalazioni" and displays a list of four items:

- 1 - RIMOZIONE - idEvento:123
- 2 - RIMOZIONE - idEvento:456
- 3 - AGGIUNTA - incidente
- 4 - AGGIUNTA - traffico

The background of the page features a colorful illustration of a European town square with buildings, people, and a flag.

Dettaglio Segnalazione

The screenshot shows a detailed view of a report titled "Dettaglio Segnalazione". The top section displays basic information:
Tipo Segnalazione: aggiunta
ID Segnalazione: 4

The main content area is titled "Dettagli Aggiunta" and contains the following details:
Type: traffico
Gravità: media
Descrizione: Coda in tangenziale
Data: 2025-06-26
Ora: 09:30
Indirizzo: Viale Indipendenza 45

Below this, there is a thumbnail image of a traffic jam on a highway. Navigation arrows are visible on either side of the image. At the bottom of the detail view, there are two buttons: "Aggiungi Evento" (green) and "Rifiuta" (red).

At the very bottom of the page, there is a link: "← Torna alla Lista".

Interazione

Per una maggiore leggibilità sono state omesse le barre di attivazione sulle lifeline.

Diagramma di sequenza: Login con successo

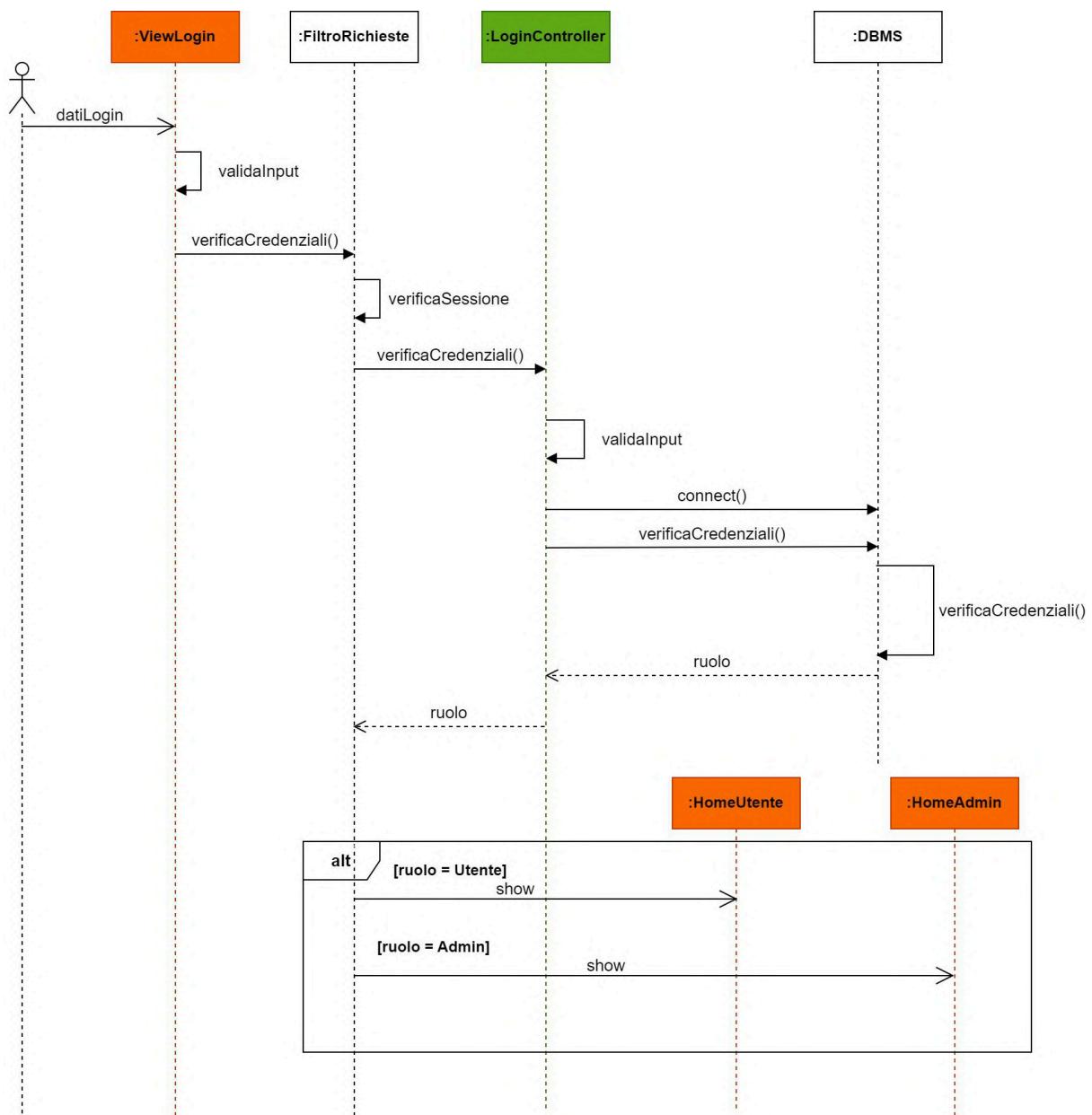


Diagramma di sequenza: NuovaSegnalazione

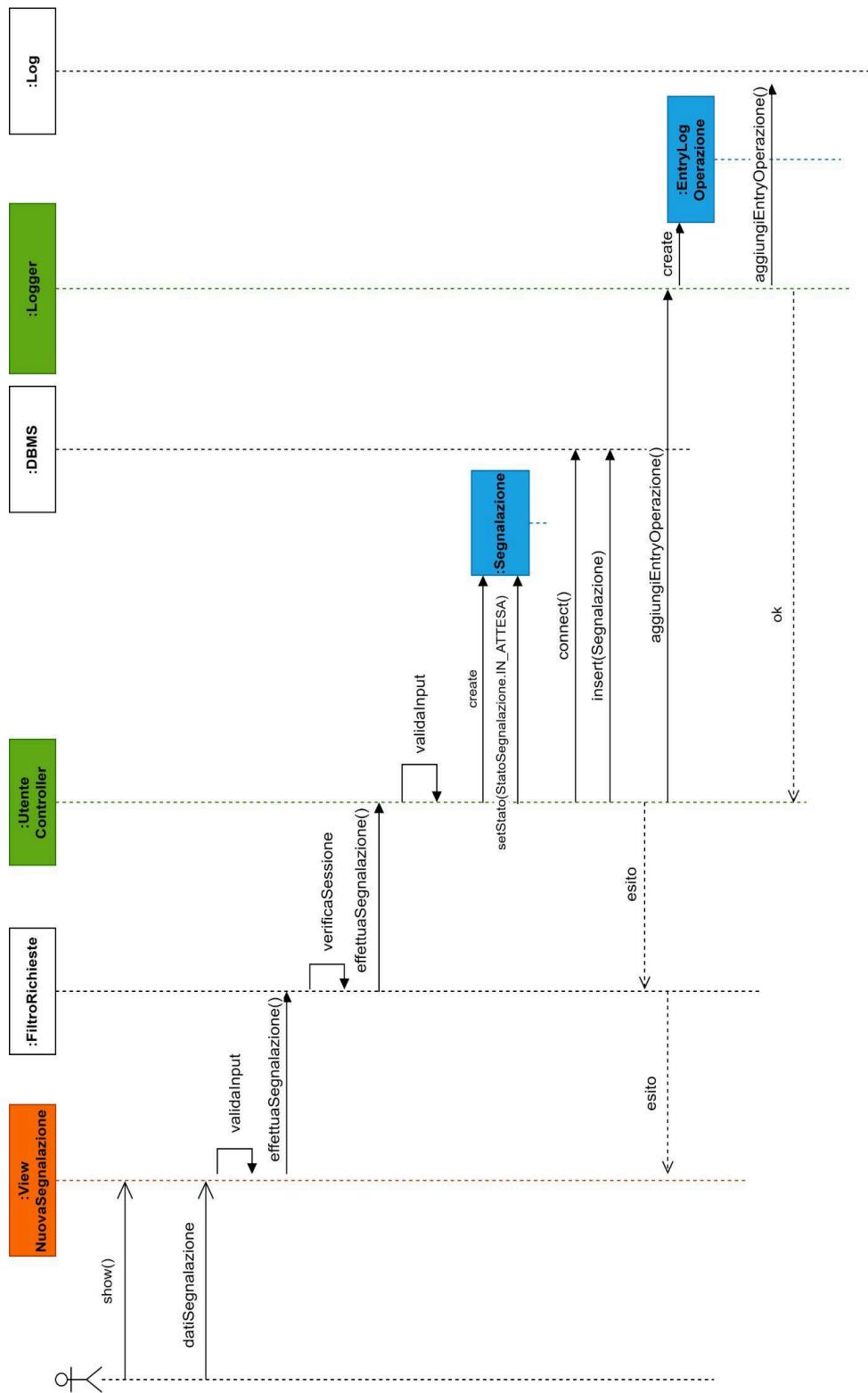


Diagramma di sequenza: CercaEventi e FiltraEventi

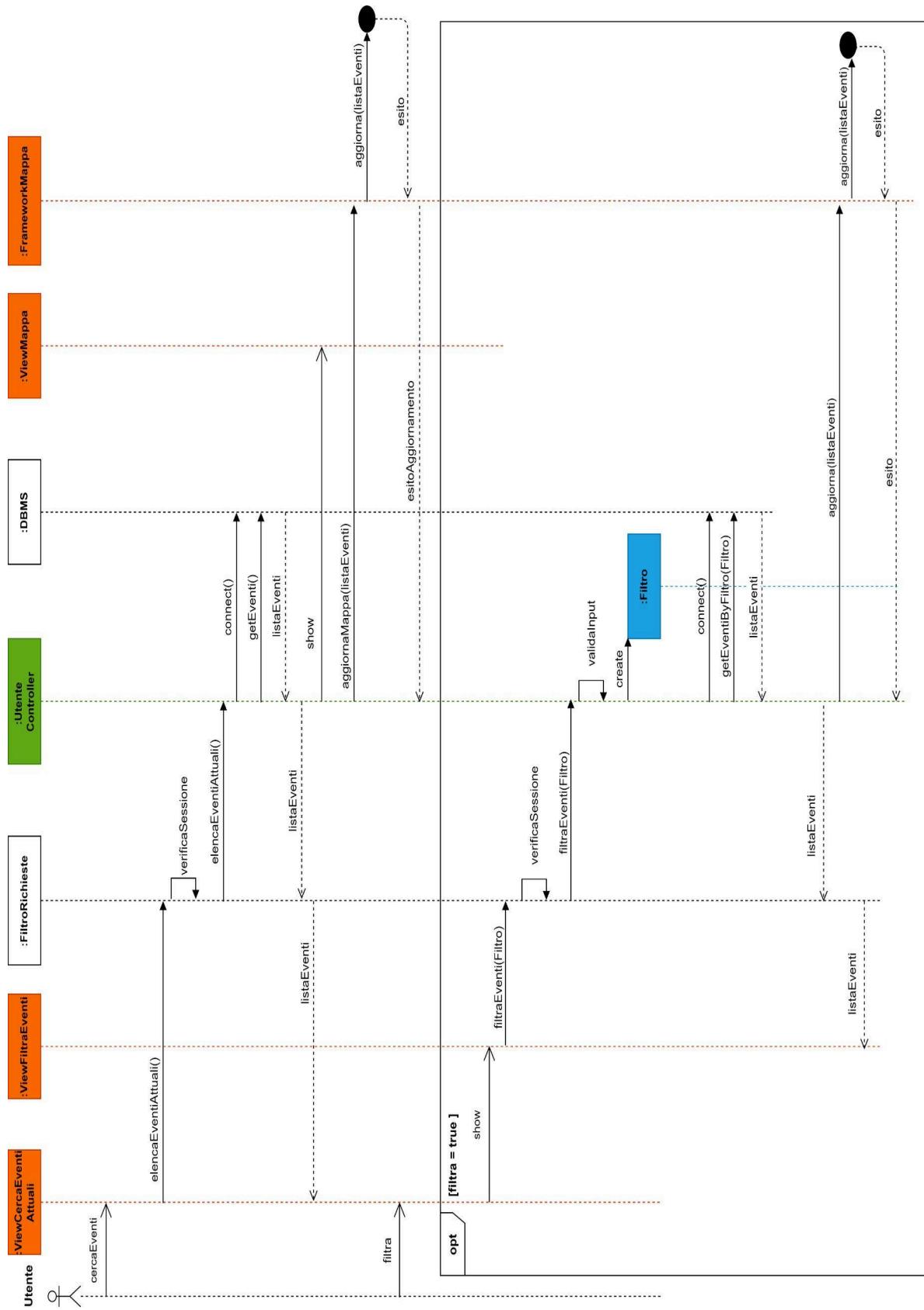
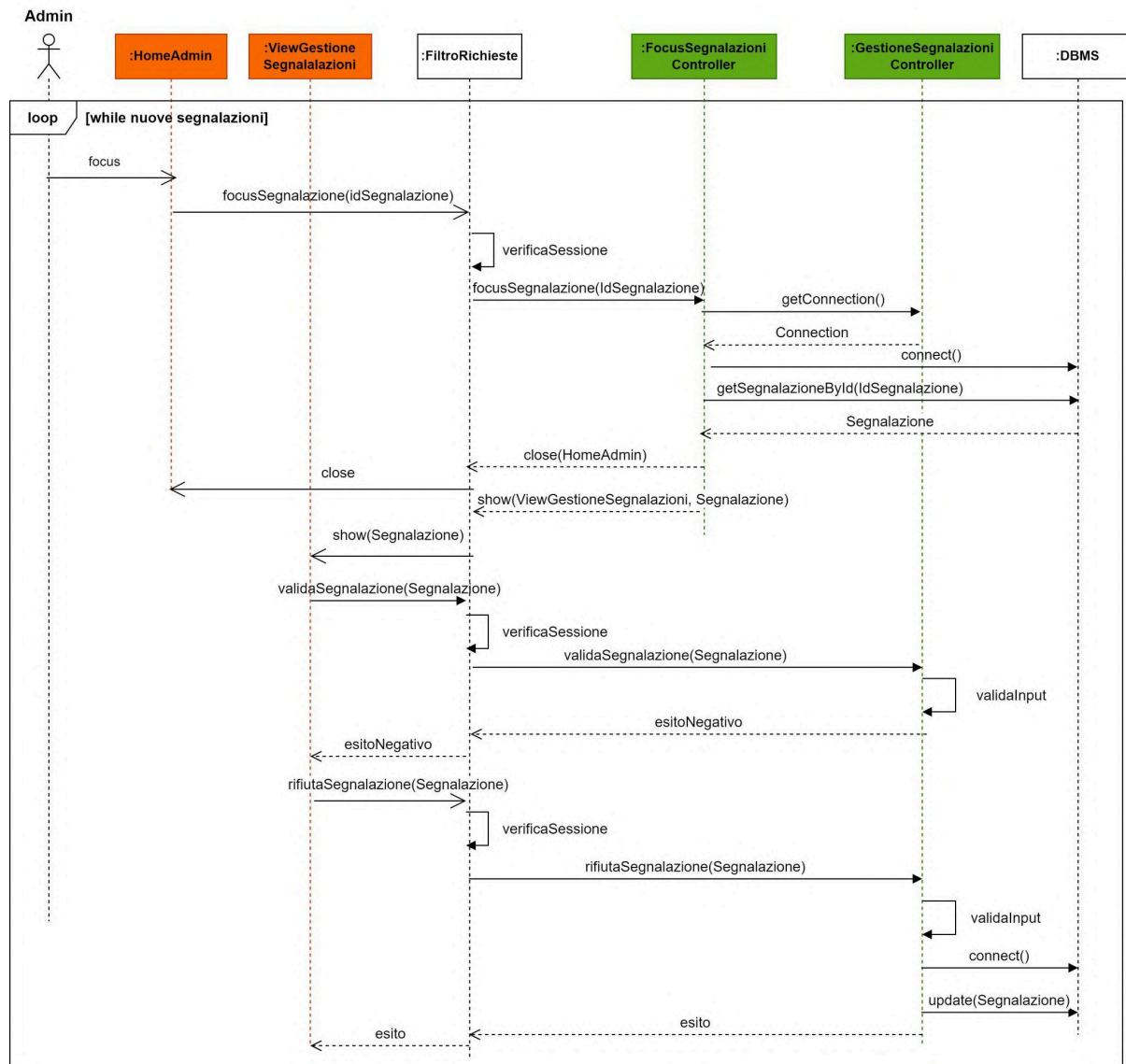


Diagramma di sequenza: RifiutaSegnalazione

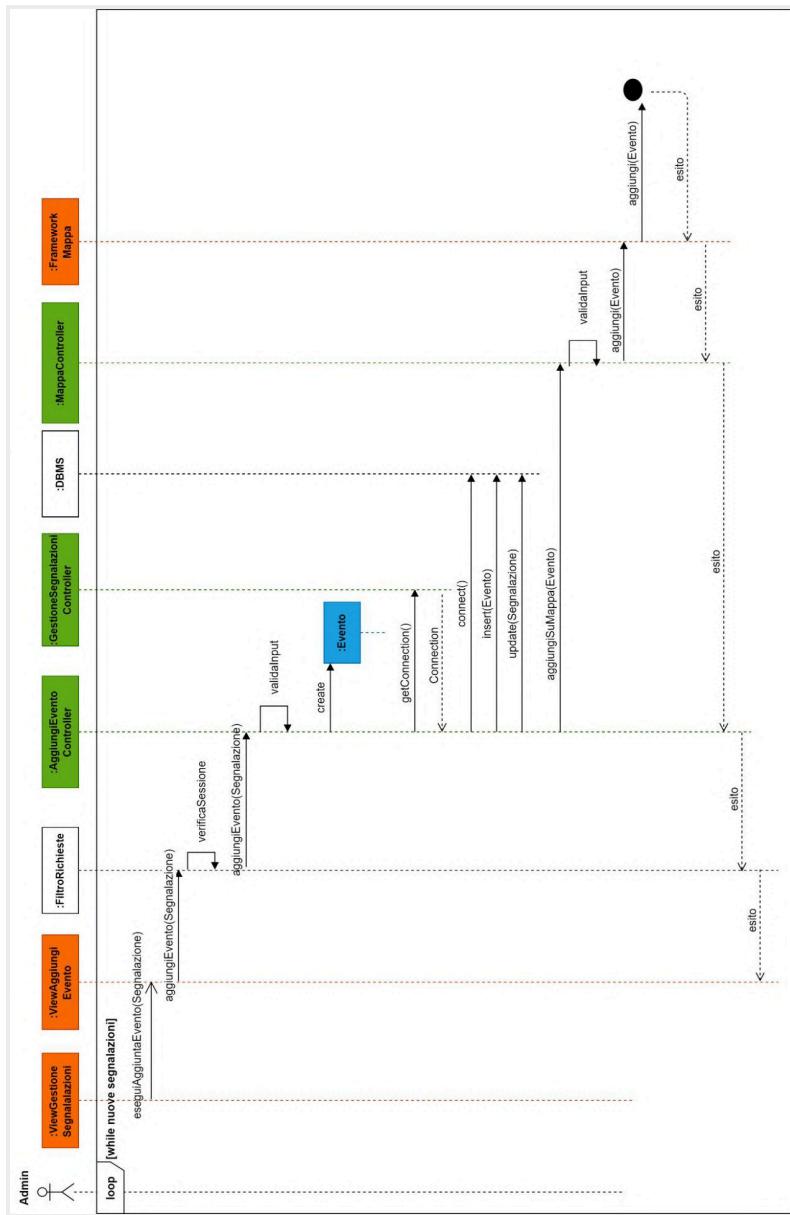


Nota: `update(Signalazione)` modella la modifica dello StatoSignalazione, che passa da IN_ATTESA a RIFIUTATA

Diagramma di sequenza: AggiungiEvento

Nel diagramma seguente non sono rappresentati né il focus su una segnalazione né il processo di validazione. Per una descrizione dettagliata di tali aspetti, si rimanda al diagramma di sequenza *RifiutaSegnalazione*.

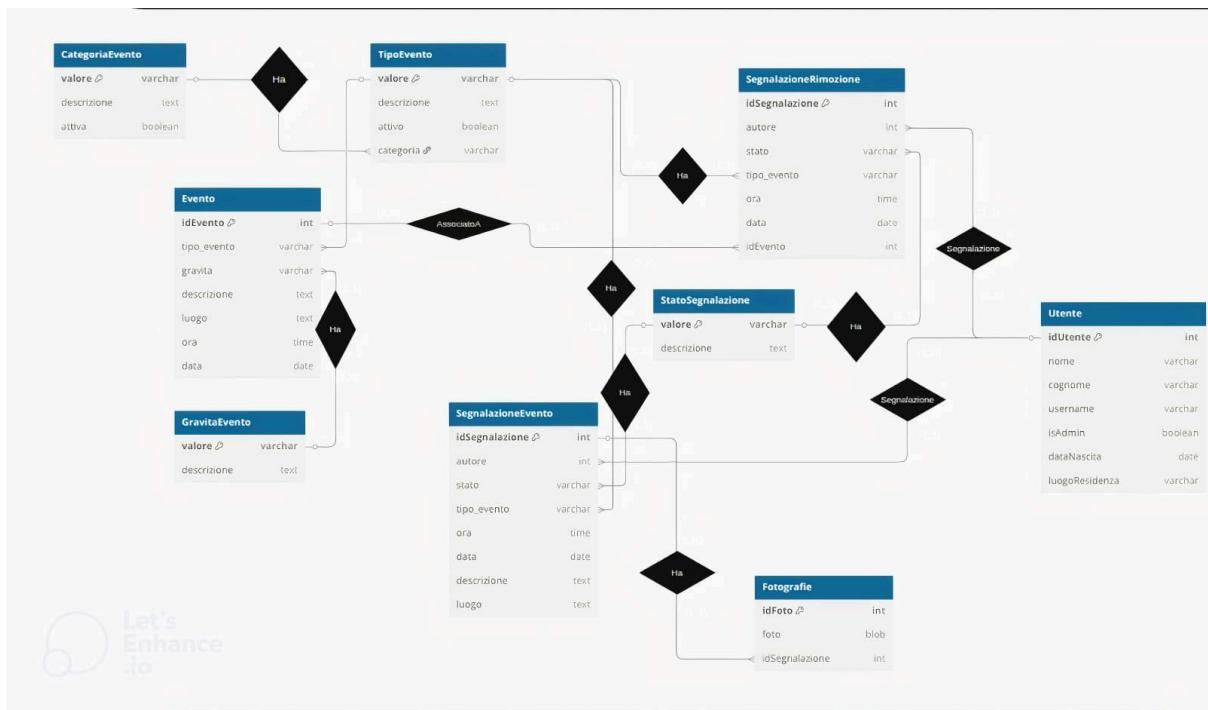
Ipotesi: si assume che la validazione della Segnalazione abbia esito positivo, permettendo all'Admin di procedere con l'inserimento di un nuovo Evento.



Nota: update(Segnalazione) modella la modifica dello StatoSegnalazione, che passa da IN_ATTESA ad ACCETTATA

Progettazione della persistenza

Diagramma ER



Formato file log

- Formato file per Log delle operazioni
DataOra - operazione - esecutore
- Formato file per Log dei messaggi
DataOra - messaggio protetto - invio/ricezione - autore

Progettazione del Collaudo

```

public class ValidaSegnalazioneTest {

    @Test
    void testValidaSegnalazione_creaEvento() {

        GestioneSegnalazioneController gestisciCtrl = new
        GestioneSegnalazioneController();

        UtenteController utenteCtrl = new UtenteController();

        SegnalazioneEvento seg = new SegnalazioneEvento();
    }
}

```

```

        seg.setIdSegnalazione(102);
        seg.setTipoEvento(new TipoEvento("Frana"));
        seg.setDescrizione("Frana in collina");
        seg.setLuogo("Via del Poggio");
        seg.setData(LocalDate.of(2025, 6, 10));
        seg.setOra(LocalTime.of(8, 0));
        seg.setStato(StatoSegnalazione.IN_ATTESA);

seg.setTipoSegnalazione(TipoSegnalazione.NUOVO_EVENTO);
    List<Immagine> fotografie = new LinkedList<>();
    seg.addFoto(fotografie.add(new
Immagine("fotoFrana.jpg")));

gestisciCtrl.validaSegnalazione(seg);

List<Evento> eventi =
utenteCtrl.elencaEventiAttuali();
assertEquals(1, eventi.size());

    Evento ev = eventi.get(0);
    assertEquals(seg.getTipoEvento(),
ev.getTipo_evento());
    assertEquals(seg.getDescrizione(),
ev.getDescrizione());
    assertEquals(seg.getLuogo(), ev.getLuogo());
    assertEquals(seg.getData(), ev.getData());
    assertEquals(seg.getOra(), ev.getOra());
    assertEquals(StatoSegnalazione.ACCEPTEATA,
seg.getStato());
}
}

public class TestCercaEventi{

private DatabaseMock db;
private UtenteController u_controller;

@BeforeAll

```

```
public void setup(){

    db = newDatabaseMock();
    u_controller = new UtenteController();

    Integer idEvento = 1233;
    TipoEvento tipo_evento = new TipoEvento("Caduta
alberi");
    GravitaEvento gravita = GravitaEvento.GIALLO;
    String descrizione = "alberi bloccano la carreggiata";
    String luogo = "viale Panzacchi";
    LocalTime ora = LocalTime.of(10,0,0);
    LocalDate data = LocalDate.of(2025, 10, 12);

    Evento e1 = new Evento(idEvento, tipo_evento, gravita,
descrizione, luogo, ora, data);

    idEvento = 1348;
    tipo_evento = TipoEvento.TERREMOTO;
    gravita = GravitaEvento.ROSSO;
    descrizione = "Scosse di terremoto"
    luogo = "Bologna e provincia";
    ora = LocalTime.of(12,30,0);
    data = LocalDate.of(2025,06, 05);

    Evento e2 = new Evento(idEvento, tipo_evento, gravita,
descrizione, luogo, ora, data);

    //simulazione interazione con DB
    db.addEvento(e1);
    db.addEvento(e2);

}

@Test
public void TestRicerca(){

    Integer idEvento = 1233;
```

```
        List<Evento> listaEventi =
u_controller.elencaEventiAttuali();
        assertEquals(idEvento,
listaEventi.get(0).getIdEvento());
    }

    @Test
    public void TestFiltrati{

        List<Evento> listaEventi =
u_controller.elencaEventiAttuali();
        List<Evento> listaEventi_filtrata =
u_controller.filtrareEventi(new
FiltroTipoEvento(TipoEvento.CADUTA_ALBERI))
        assertEquals(listaEventi_filtrata.size(), 1);

        assertEquals(listaEventi_filtrata.get(0).getTipoEvento(),
TipoEvento.CADUTA_ALBERI);

    }
}

public class SegnalazioneEventi{

    private UtenteController u_controller;
    private FocusSegnalazioneController
controller_segnalazione;

    @BeforeAll
    public void setup(){

        u_controller = new UtenteController();
        controller_Segnalazioni = new
FocusSegnalazioneController();
    }

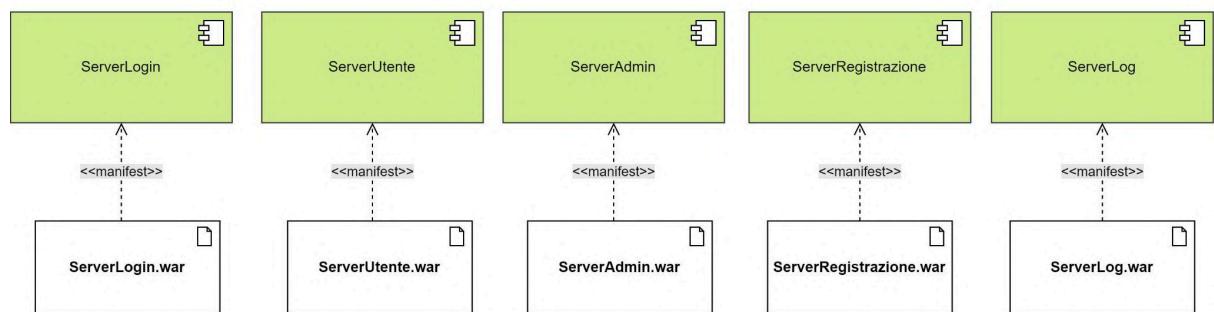
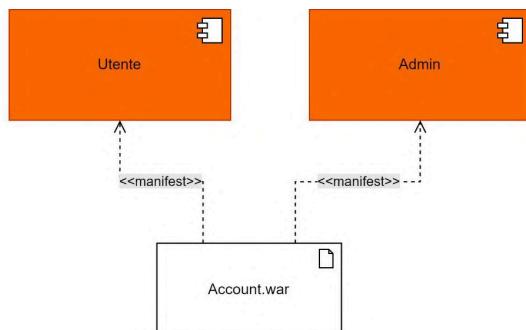
    @Test
```

```
public void testNuovaSegnalazione () {  
  
    TipoEvento tipo_evento = new TipoEvento("Incendio");  
    String descrizione = "incendio in un centro  
commerciale";  
    String luogo = "viale Orsolini 4";  
    Utente autore= new utente ("Rosanna", "Bastoni",  
"ros07", false);  
    Immagine fotoSegnalazione = new  
Immagine("fotoIncendio.jpg");  
    List<Immagine> listaFoto = new LinkedList<>();  
    listaFoto.add(fotoSegnalazione);  
    SegnalazioneEvento s = new  
SegnalazioneEvento(autore,luogo, descrizione, tipo_evento,  
fotosegnalazione);  
    u_controller.effettuaSegnalazione(s);  
  
    Integer idSegnalazione = s.getId();  
    Segnalazione s1 =  
controller_segnalazione.focusSegnalazione(idSegnalazione);  
    assertEquals(luogo, s1.getluogo());  
    assertEquals(autore, s1.getautore());  
    assertEquals(descrizione, s1.getDescrizione());  
    assertEquals(tipo_evento, s1.getTipoEvento());  
}  
  
}
```

Progettazione del deployment

Deployment del sistema

Artefatti



Deployment type-level

