

GESTRI – Gestionale Rifiuti Industriali

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione



UNIVERSITÀ POLITECNICA DELLE MARCHE

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione

TESINA DI:

RELATORI:

Prof. Domenico Ursino
Prof. Davide Traini

Tarek Naja
Davide Ronchini
Marco Sambughi
Sara Vaccaro

Indice

DESCRIZIONE DEL PROGETTO	1
GLOSSARIO DEI TERMINI	3
GESTIONE DEI REQUISITI	5
Requisiti Funzionali	5
Requisiti Non Funzionali	8
Tabella MoSCoW dei Requisiti.	9
Diagrammi dei Casi d’Uso	11
Matrice di Mapping	52
ANALISI	53
Diagrammi delle Classi di Analisi	53
Package di Analisi	53
Classi di Analisi: Utente	54
Classi di Analisi: Mezzo	55
Classi di Analisi: Attività	56
Classi di Analisi: Documento	57
Diagrammi di Sequenza	58
Login Utente	58
Registrazione Utente	59
CRUD Attività	60
Trova Documento	62
Filtra Documento	63
Diagrammi di Attività	64
Login Utente	64
Registrazione Utente	65
CRUD Attività	66
Trova Documento	67
Filtra Documento	68
PROGETTAZIONE	69
Diagrammi delle Classi di Progettazione	69
Package di Progettazione	69
Classi di Progettazione: Utente	70
Classi di Progettazione: Mezzo	71
Classi di Progettazione: Attività	72
Classi di Progettazione: Documento	73
Classi di Progettazione: Controllers	74
Diagramma dei Componenti.	75
Diagrammi delle Macchine a Stati	76
Macchine a Stati: Attività	76
Macchine a Stati: Mezzo	77
Diagramma Entità-Relazione	78

IMPLEMENTAZIONE	79
Diagramma di Deployment	79
Tecnologie Utilizzate	80
Tecnologie Backend	80
Tecnologie Frontend	80
Database	80
Tecnologie Version Control System	81
Altre tecnologie utilizzate	81
Mockup	82
Verifica e Validazione del Software: Unit Tests	85
Test Manuali	85
Unit Test	85

Descrizione del Progetto

Panoramica Generale

Il progetto consiste nello sviluppo di **GESTRI**, un gestionale per il trasporto di rifiuti industriali, pensato per supportare le aziende nella pianificazione, monitoraggio e tracciabilità delle operazioni. L'obiettivo è offrire uno strumento intuitivo così da semplificare la gestione delle attività quotidiane, riducendo errori e tempi di coordinamento.

Il sistema consente agli utenti di interagire secondo il proprio ruolo. Gli Operatori hanno la possibilità di aggiornare i dati di base e modificare la propria anagrafica, garantendo che le informazioni personali e operative siano sempre corrette. Lo Staff dispone di strumenti avanzati che gli permettono di creare nuove attività, assegnare i mezzi di trasporto più idonei, coordinare le operazioni e supervisionare l'andamento complessivo. I Clienti, oltre a commissionare il servizio, possono a loro volta creare richieste di attività e consultare lo stato delle operazioni in corso. Le attività rappresentano il fulcro del sistema e racchiudono tutte le informazioni necessarie al loro svolgimento: tipologia di rifiuto, scadenze e mezzi di trasporto assegnati. Ogni attività è tracciata passo dopo passo, così da fornire una visione aggiornata e affidabile sia agli operatori sia ai clienti.

Un ruolo fondamentale è ricoperto dalla gestione documentale. Ogni trasporto è accompagnato da un FIR (Formulario Identificazione Rifiuto), documento obbligatorio che certifica il percorso del rifiuto dal produttore fino alla destinazione finale, e da un CER (Codice Europeo del Rifiuto), che identifica in maniera univoca la tipologia del rifiuto trattato. L'integrazione digitale di questi documenti assicura tracciabilità completa, riduzione degli errori di compilazione e conformità alle normative vigenti.

In sintesi, GESTRI permette di centralizzare e automatizzare l'intero processo di gestione dei rifiuti industriali: gli utenti possono creare e seguire attività, i mezzi vengono assegnati in modo mirato, i documenti vengono caricati e archiviati in modo sicuro, e tutte le operazioni restano tracciabili e verificabili. Il risultato è una piattaforma che rende il servizio più efficiente, trasparente e affidabile, migliorando sia il lavoro delle aziende che l'esperienza dei clienti.

01 XX XX Rifiuti derivanti da prospezione, estrazione da miniere e cave, nonché da trattamenti fisici e chimici di minerali	02 XX XX Rifiuti da agricoltura, orticoltura, acquacoltura, silvicolto, caccia e pesca	03 XX XX Rifiuti della lavorazione del legno e della produzione di carta e cartone	04 XX XX Rifiuti dell'industria conciaria e tessile	05 XX XX Rifiuti della raffinazione del petrolio, gas naturale, trattamento pirotecnico del carbone
06 XX XX Rifiuti dei processi chimici inorganici	07 XX XX Rifiuti dei processi chimici organici	08 XX XX Rifiuti da produzione, formulazione, fornitura e uso di rivestimenti (vernici, adesivi, sigillanti, inchiostri)	09 XX XX Rifiuti da industria fotografica	10 XX XX Rifiuti da processi termici
11 XX XX Rifiuti da trattamenti chimici superficiali e rivestimenti di metalli e altri materiali	12 XX XX Rifiuti dalla lavorazione e trattamento meccanico dei metalli	13 XX XX Oli esauriti e residui di carburanti liquidi	14 XX XX Solventi organici, refrigeranti e propellenti di scarto	15 XX XX Rifiuti da imballaggi, assorbenti, materiali filtranti, stracci protettivi
16 XX XX Rifiuti non specificati altrimenti	17 XX XX Rifiuti delle operazioni di costruzione e demolizione	18 XX XX Rifiuti da attività sanitarie e veterinarie	19 XX XX Rifiuti da impianti di trattamento rifiuti, acque reflue e bonifiche	20 XX XX Rifiuti urbani e simili

Categorie principali dei Codici CER

Gestione dei Dati

La gestione dei dati rappresenta un elemento centrale del progetto. Il sistema è stato progettato per garantire un'organizzazione chiara e strutturata delle informazioni, supportata da procedure di aggiornamento e validazione che ne assicurano l'affidabilità nel tempo.

Particolare attenzione è rivolta ai principi di integrità e consistenza, così da mantenere i dati coerenti e privi di ridondanze inutili.

In questo modo, la piattaforma garantisce un flusso informativo ordinato, sicuro e facilmente consultabile, a beneficio sia degli operatori interni sia dei clienti.

Architettura del Sistema

L'architettura di GESTRI è concepita secondo un approccio modulare, così da favorire la scalabilità, la manutenzione e l'evoluzione futura della piattaforma. I principali componenti possono essere riassunti come segue:

- **Interfaccia Utente (Front-end):** fornisce un ambiente semplice e intuitivo per l'interazione degli utenti, adattato alle diverse tipologie di attori (Operatori, Staff, Clienti). È progettata per garantire chiarezza, usabilità e accesso rapido alle funzionalità principali.
- **Logica Applicativa (Back-end):** rappresenta il cuore del sistema, gestisce i processi interni e applica le regole di business. Qui risiedono i meccanismi come quelli di registrazione, monitoraggio delle attività, generazione dei documenti e gestione dei permessi degli utenti.
- **Gestione dei Dati (Database):** si occupa della memorizzazione strutturata delle informazioni. La progettazione segue criteri di integrità e consistenza, sfruttando relazioni tra entità per ridurre ridondanze e facilitare le operazioni di ricerca e consultazione.
- **Servizi di Supporto:** includono funzionalità trasversali come autenticazione, controlli di sicurezza e strumenti di reportistica.

Questa architettura a più livelli permette di separare in modo chiaro presentazione, logica e dati, garantendo così robustezza, flessibilità e una più agevole gestione del sistema nel tempo.

Accesso

La gestione delle credenziali assicura una netta distinzione tra i profili e rafforza il livello di sicurezza, prevenendo usi impropri e garantendo che ciascun utente operi esclusivamente nell'ambito delle proprie competenze.

Glossario dei Termini

TERMINI	DESCRIZIONE	TIPO	SINONIMI
Account	Area personale dell'utente.	TECNICO	Profilo
Login	Processo con cui un utente viene identificato e entra nella piattaforma.	TECNICO	Accesso
Sistema	Insieme organizzato di componenti interconnessi che lavorano insieme per svolgere una o più funzioni definite.	TECNICO	Gestionale, Programma, Sito
Utente	Ruolo generico da cui derivano Cliente e Operatore. Ha la capacità di accedere al sistema.	TECNICO	-
Cliente	Soggetto che commissiona il servizio e consulta documenti e stato delle attività.	BUSINESS	-
Operatore	Soggetto incaricato di eseguire operazioni pratiche di carico/scarico e compilazione documenti. Estende le funzionalità dello Staff.	BUSINESS	-
Staff	Personale amministrativo che gestisce l'organizzazione dei turni, le assenze e le attività complessive.	BUSINESS	-
Attività	Operazione di carico o scarico di rifiuti, con assegnazione di operatori, mezzi e documenti di tipo FIR.	BUSINESS	-
Mezzo	Veicolo utilizzato per il trasporto dei rifiuti.	BUSINESS	Veicolo
CER	Sequenze numeriche, composte da 6 cifre riunite in coppie, volte ad identificare un rifiuto, di norma, in base al processo produttivo da cui è originato.	BUSINESS	Catalogo Europeo dei Rifiuti

TERMINE	DESCRIZIONE	TIPO	SINONIMI
FIR	Documento obbligatorio che accompagna il trasporto dei rifiuti per garantirne la tracciabilità, riportando informazioni sul produttore, sul rifiuto, sul trasportatore e sull'impianto di destinazione.	BUSINESS	Formulario di Identificazione Rifiuti
Turno	Periodo temporale in cui un operatore è assegnato a un'attività.	BUSINESS	-

Gestione dei requisiti

Nella presente sezione si analizzano in modo approfondito i requisiti del sistema, procedendo alla loro suddivisione secondo le seguenti categorie:

- Requisiti funzionali, che definiscono le specifiche funzionalità che il sistema è tenuto a fornire.
- Requisiti non funzionali, che individuano i vincoli e le qualità che il sistema deve soddisfare, quali ad esempio l'usabilità, la sicurezza e le prestazioni.

Requisiti Funzionali

Area: Gestione Utente

RF1: Registrazione Utente

Il sistema dovrà permettere la registrazione di nuovi utenti (Clienti, Operatori, Staff) inserendo i dati necessari.

RF2: Login Utente

Il sistema dovrà permettere all'Utente di autenticarsi tramite email e password.

RF3: Recupero Credenziali

Il sistema dovrà permettere il recupero delle credenziali tramite procedura (es. email).

RF4: Visualizza Utente

Il sistema dovrà permettere all'Utente di visualizzare i propri dati anagrafici e le informazioni correlate (ruolo, turni, assenze, attività); lo Staff dovrà poter visualizzare i dati di altri Utenti per scopi di gestione e supervisione.

RF5: Modifica Ruolo

Il sistema dovrà permettere allo Staff di assegnare e modificare il ruolo dell'Utente (Operatore e Staff).

RF6: Modifica Utente

Il sistema dovrà permettere la modifica delle informazioni anagrafiche e dei dati di contatto di un Utente.

RF7: Elimina Utente

Il sistema dovrà permettere la cancellazione di un account Utente e la gestione coerente dei dati associati.

Area: Gestione Assenza

RF8: CRUD Assenza

Il sistema dovrà permettere allo Staff di registrare e monitorare le assenze del personale.

Area: Gestione Attività

RF9: CRUD Attività

Il sistema dovrà permettere la creazione, lettura, aggiornamento e cancellazione delle attività di carico/-scarico rifiuti.

RF10: Assegnazione Operatore

Il sistema dovrà permettere l'assegnazione di uno o più Operatori a ciascuna attività, con controllo delle disponibilità.

RF11: Assegnazione Mezzo

Il sistema dovrà permettere l'assegnazione del mezzo rimorchio all'attività in base al tipo di rifiuto e al carico previsto.

RF12: Assegnazione Documento

Il sistema dovrà permettere l'assegnazione del documento di tipo FIR all'attività in base al tipo di rifiuto e al carico previsto.

Area: Gestione Documento**RF13: CRUD Documento FIR**

Il sistema dovrà permettere la creazione, lettura, aggiornamento e cancellazione del documento (es. Formulario di Identificazione Rifiuto - FIR associato a un'attività assegnata oppure Documento Corso sulla Sicurezza associato all'Operatore).

RF14: Associa Operatore

Il sistema dovrà permettere l'assegnazione dell'Operatore al documento di tipo non FIR.

RF15: Filtra Documento

Il sistema dovrà permettere il filtraggio e la ricerca dei documenti tramite criteri multipli.

RF16: Notifica Rinnovo Corso Sicurezza

Il sistema dovrà inviare notifiche per il rinnovo dei corsi di sicurezza agli Operatori interessati prima della scadenza.

RF17: Notifica Scadenza Documento

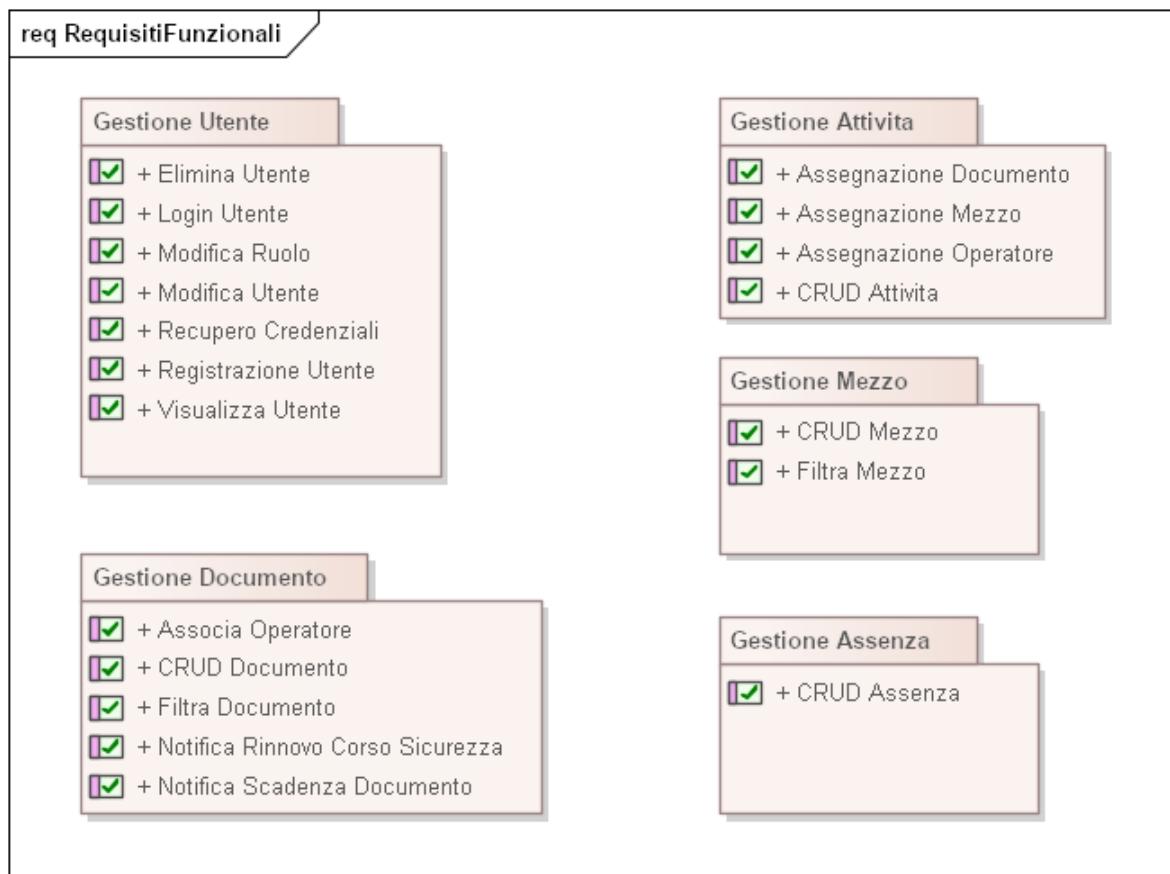
Il sistema dovrà inviare notifiche relative alle scadenze dei documenti (es. FIR, consegne, ritiri) agli Operatori interessati.

Area: Gestione Mezzo**RF18: CRUD Mezzo**

Il sistema dovrà permettere la creazione, lettura, aggiornamento e cancellazione dei mezzi, inclusi dati tecnici.

RF19: Filtra Mezzo

Il sistema dovrà permettere il filtraggio e la ricerca dei mezzi per attributi rilevanti.



Requisiti Non Funzionali

Area: Gestione Tecnologie

RNF1: Implementazione in Python 3

Il sistema dovrà essere implementato utilizzando Python 3.

RNF2: Utilizzo Database Relazionale

Il sistema dovrà utilizzare un database relazionale per la gestione persistente dei dati.

RNF3: Verifica Email

Il sistema dovrà prevedere la verifica dell'indirizzo email degli utenti durante la registrazione.

RNF4: Recupero Credenziali Tramite Email

Il sistema dovrà permettere il recupero delle credenziali tramite email.

Area: Gestione UI/UX

RNF5: Visualizzazione Attività con Calendario

Il sistema dovrà offrire una visualizzazione delle attività tramite un calendario integrato.

RNF6: Interfaccia Responsive

Il sistema dovrà presentare un'interfaccia responsive, fruibile da dispositivi con diverse risoluzioni.

RNF7: Notifica Scadenza Compilazione Documento

Il sistema dovrà inviare notifiche relative alla scadenza della compilazione dei documenti entro 24 ore quando mancano 10 giorni alla scadenza.

RNF8: Notifica Eliminazione Documento

Il sistema dovrà inviare notifiche 30 giorni prima dell'eliminazione di un documento al termine del periodo minimo di archiviazione di 3 anni.

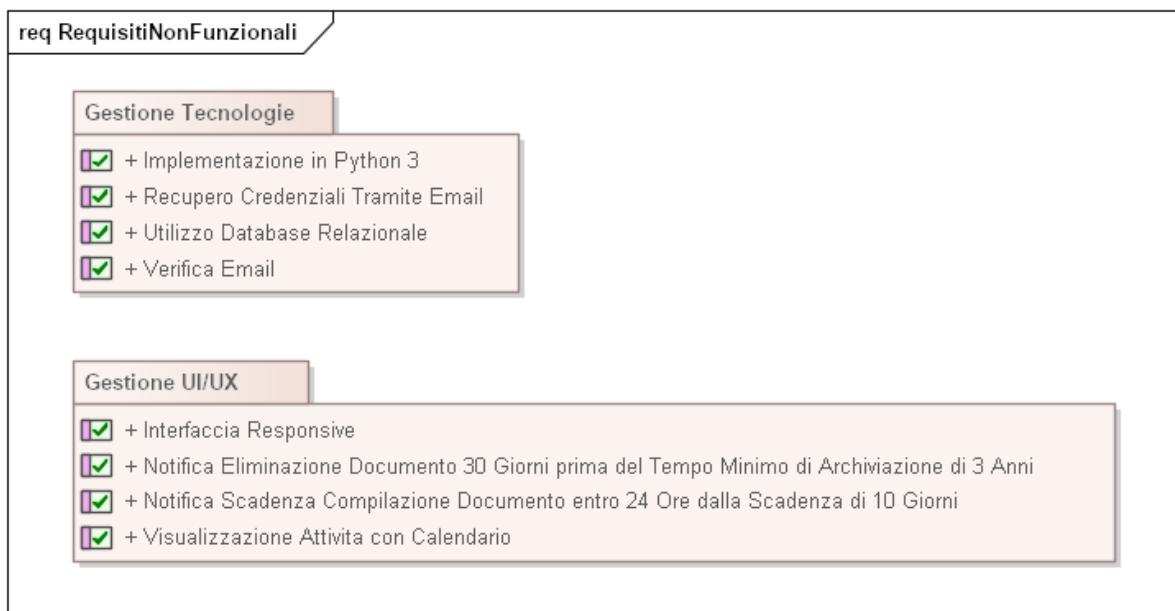


Tabella MoSCoW dei Requisiti

Di seguito la matrice MoSCoW che classifica i requisiti del sistema secondo le priorità: M (Must-have), S (Should-have), C (Could-have), W (Want-have).

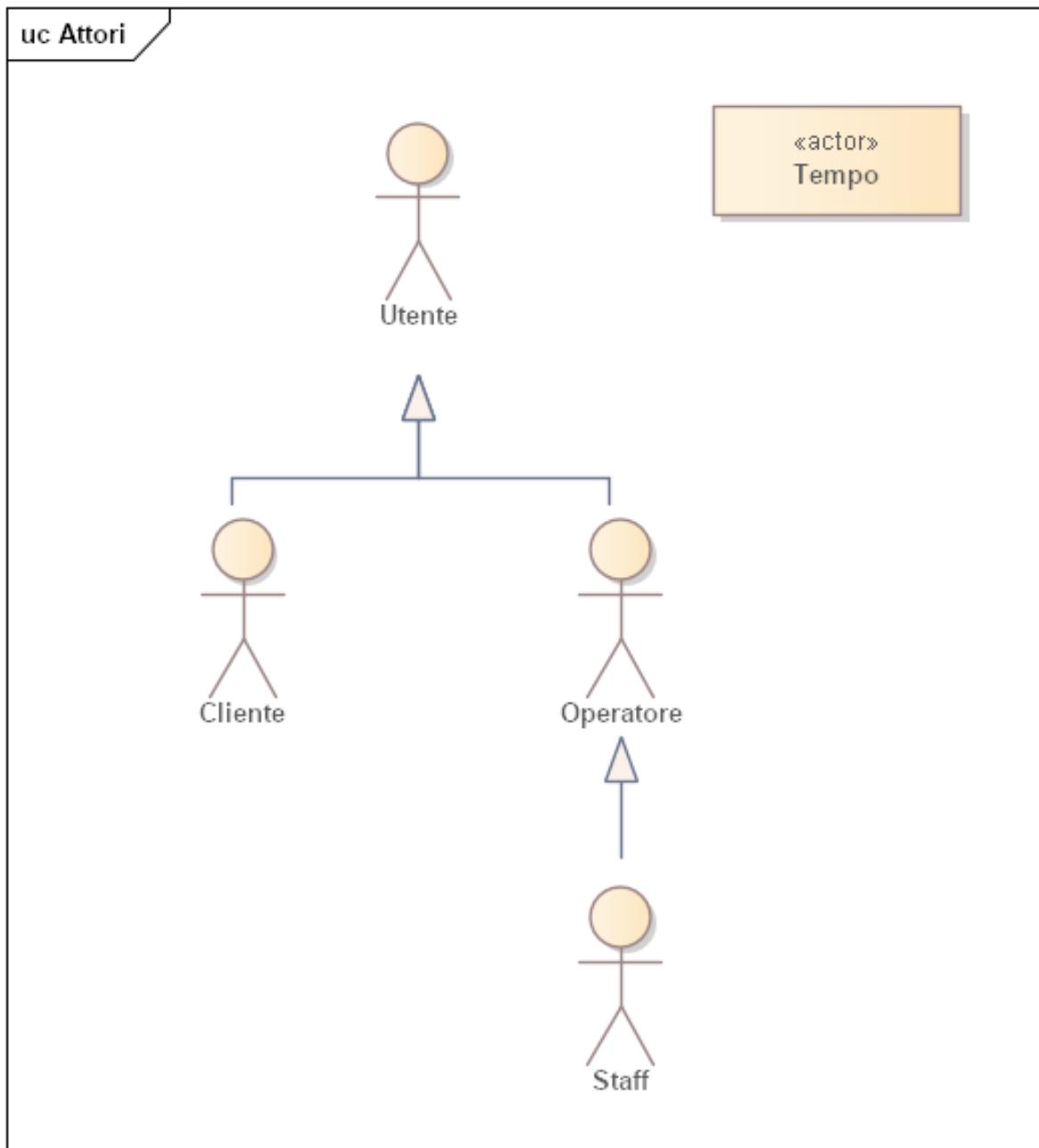
Area	Requisito	Priorità MoSCoW
Gestione Utente	RF1: Registrazione Utente	Must
Gestione Utente	RF2: Login Utente	Must
Gestione Utente	RF3: Recupero Credenziali	Should
Gestione Utente	RF4: Visualizza Utente	Must
Gestione Utente	RF5: Modifica Ruolo	Should
Gestione Utente	RF6: Modifica Utente	Should
Gestione Utente	RF7: Elimina Utente	Must
Gestione Assenza	RF8: CRUD Assenza	Should
Gestione Attività	RF9: CRUD Attività - carico/scarico	Must
Gestione Attività	RF10: Assegnazione Operatore	Must
Gestione Attività	RF11: Assegnazione Mezzo	Must
Gestione Documento	RF12: Assegnazione Documento	Must
Gestione Documento	RF13: CRUD Documento FIR	Must
Gestione Documento	RF14: Associa Operatore	Must
Gestione Documento	RF15: Filtra Documento	Should
Gestione Documento	RF16: Notifica Rinnovo Corso Sicurezza	Could
Gestione Documento	RF17: Notifica Scadenza Documento	Should
Gestione Mezzo	RF18: CRUD Mezzo	Must

Area	Requisito	Priorità MoSCoW
Gestione Mezzo	RF19: Filtra Mezzo	Should
Gestione Tecnologie	RNF1: Implementazione in Python 3	Must
Gestione Tecnologie	RNF2: Utilizzo Database Relazionale	Must
Gestione Tecnologie	RNF3: Verifica Email	Should
Gestione Tecnologie	RNF4: Recupero Credenziali tramite Email	Should
Gestione UI/UX	RNF5: Visualizzazione Turni con Calendario	Could
Gestione UI/UX	RNF6: Interfaccia Responsive	Should
Gestione UI/UX	RNF7: Notifica Scadenza Compilazione Documento	Should
Gestione UI/UX	RNF8: Notifica Eliminazione Documento	Want

Diagrammi dei Casi d'Uso

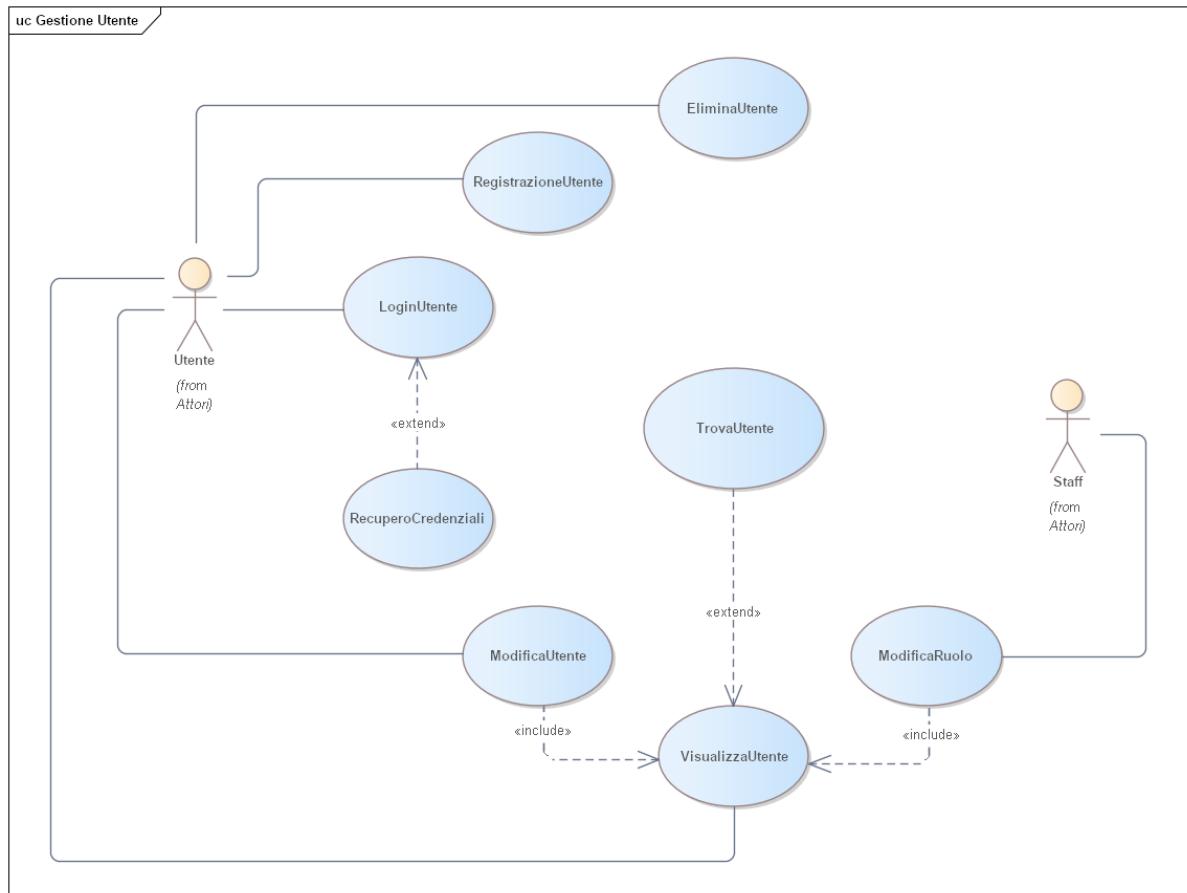
Diagramma degli Attori

Il diagramma individua gli attori coinvolti nel sistema e visualizza le connessioni di ereditarietà tra di essi.



Gestione Utente

Il seguente diagramma descrive le relazioni tra attori e casi d'uso per la gestione degli Utenti. Racchiude la possibilità di registrare, modificare ed eliminare un Utente, effettuare il login e recuperare le credenziali in caso di smarrimento. Sono previsti, inoltre, casi d'uso per la modifica del ruolo e per la consultazione dei dati tramite le operazioni di visualizzazione e ricerca degli Utenti.



Caso d'uso: RegistrazioneUtente	
ID	1
Breve descrizione	Lo Staff o il Cliente si registra nel sistema, con un ruolo che dipende dal punto di accesso
Attori primari	Cliente, Staff
Attori secondari	Nessuno
Precondizioni	Nessuna
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. If l'Utente è un Cliente che accede alla pagina di registrazione pubblica, allora <ol style="list-style-type: none"> 1.1. Il sistema mostra il modulo di registrazione 1.2. Il Cliente inserisce le proprie informazioni (nome, cognome, email, password) 1.3. While le credenziali inserite del Cliente non sono valide <ol style="list-style-type: none"> 1.1. Il sistema richiede al Cliente di inserire le sue informazioni 1.4. Il sistema valida i dati inseriti 1.5. Il sistema crea un nuovo account con il ruolo di Cliente
Postcondizioni	<ol style="list-style-type: none"> 1. È stato creato un nuovo account Utente con ruolo Cliente
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If l'Utente è un membro dello Staff che accede alla pagina di gestione utenti interna, allora <ol style="list-style-type: none"> 1.1. Il sistema mostra un modulo di creazione Utente 1.2. Il membro dello Staff inserisce le informazioni del nuovo Utente (nome, cognome, email, password) e seleziona il ruolo desiderato (Operatore o Staff) 1.3. While le credenziali inserite dell'Utente non sono valide <ol style="list-style-type: none"> 1.1. Il sistema richiede allo Staff di inserire le sue informazioni 1.4. Il sistema valida i dati inseriti 1.5. Il sistema crea un nuovo account con il ruolo specificato (Operatore o Staff)

Caso d'uso: LoginUtente	
ID	2
Breve descrizione	Permette all'Utente di accedere al proprio account
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'Utente deve essere già registrato nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Utente inserisce le credenziali e seleziona "Accedi" <i>Extend RecuperaCredenziali</i> 2. While le credenziali inserite dall'Utente non sono valide <ul style="list-style-type: none"> 2.1. Il sistema richiede all'Utente di inserire email e password 2.2. Il sistema valida le credenziali fornite 3. Il sistema autentica l'Utente e lo reindirizza alla pagina principale
Postcondizioni	<ul style="list-style-type: none"> 1. L'Utente ha effettuato il login al proprio account e può accedere alle funzionalità autorizzate
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: RecuperaCredenziali	
ID	3
Breve descrizione	L'Utente recupera le credenziali del proprio account in caso di smarrimento o dimenticanza della password
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente ha un account registrato nel sistema 2. L'Utente visualizza la schermata di login
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Utente seleziona l'opzione "Password dimenticata?" 2. While l'Utente non ha inserito l'email <ol style="list-style-type: none"> 2.1. Il sistema richiede all'Utente di inserire la propria email 2.2. Il sistema valida la correttezza e l'esistenza dell'email nel sistema 3. Il sistema genera una procedura di recupero (es. invio di un link sicuro o generazione temporanea di una nuova password) e invia le istruzioni all'indirizzo email fornito 4. L'Utente riceve la mail, segue la procedura indicata e utilizza la nuova password o il link per ripristinare l'accesso al proprio account
Postcondizioni	<ol style="list-style-type: none"> 1. L'Utente ha ricevuto le istruzioni per recuperare l'accesso al proprio account 2. L'Utente può accedere nuovamente al sistema utilizzando la nuova password o il link fornito
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: VisualizzaUtente	
ID	4
Breve descrizione	L'Utente visualizza i propri dati anagrafici e le informazioni
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente è stato autenticato dal sistema
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Utente seleziona l'opzione di visualizzazione del proprio profilo 2. Il sistema mostra i dati anagrafici e le informazioni relative all'Utente
Postcondizioni	<ol style="list-style-type: none"> 1. All'Utente sono mostrati i propri dati
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If Lo Staff richiede la visualizzazione <i>Extend TrovaUtente</i> <ol style="list-style-type: none"> 1.1. Il sistema mostra l'elenco degli Operatori registrati 1.2. Lo Staff seleziona un Operatore dall'elenco 1.3. Il sistema mostra i dati anagrafici completi e le informazioni aggiuntive

Caso d'uso: TrovaUtente	
ID	5
Breve descrizione	Lo Staff esegue la ricerca di uno o più Utenti
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff inserisce il criterio di ricerca nella barra dedicata 2. Il sistema ricerca l'Utente che soddisfa i criteri desiderati dallo Staff 3. If Il sistema trova uno o più utenti <ul style="list-style-type: none"> 3.1. For Each Utente trovato <ul style="list-style-type: none"> 3.1.1. Il sistema mostra le informazioni base dell'Utente 4. Else Il sistema comunica allo staff che non sono stati trovati Utenti che soddisfano i criteri specificati
Postcondizioni	Nessuna
Sequenza degli eventi alternativa	Nessuna

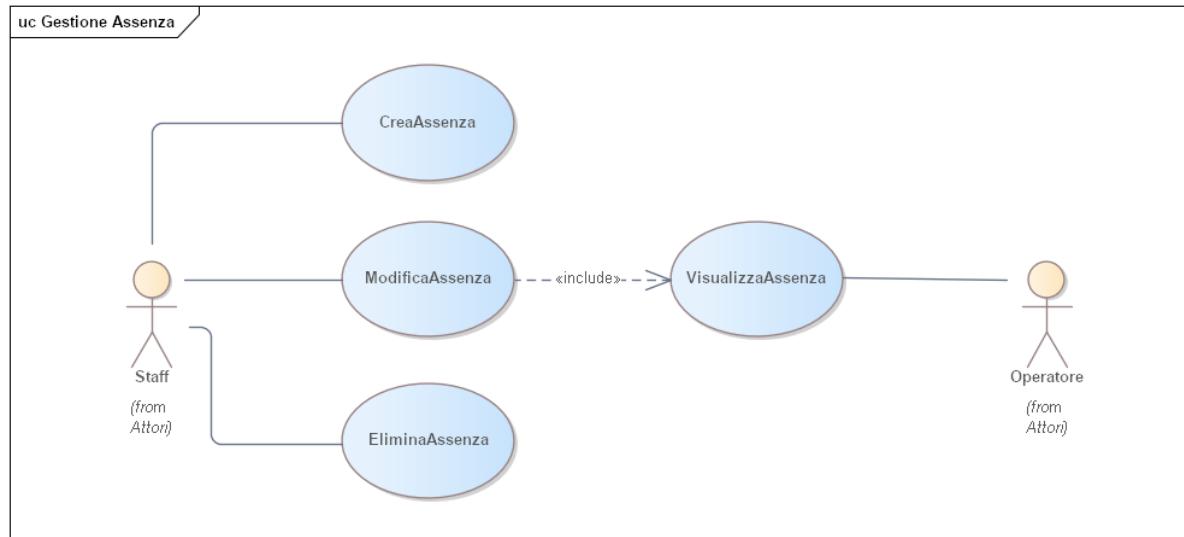
Caso d'uso: ModificaRuolo	
ID	6
Breve descrizione	Lo Staff modifica il ruolo dell'Operatore
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è autenticato 2. Lo Staff visualizza la sezione interna per la gestione degli utenti
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'Operatore da modificare 2. Lo Staff modifica il ruolo dell'Operatore 3. If la modifica è confermata e i dati sono validi <ul style="list-style-type: none"> 3.1. Il sistema aggiorna il ruolo dell'Operatore 3.2. Include <i>VisualizzaUtente</i>
Postcondizioni	<ul style="list-style-type: none"> 1. Il ruolo dell'Operatore è stato modificato
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: ModificaUtente	
ID	7
Breve descrizione	L'Utente modifica i propri dati anagrafici
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente è stato autenticato dal sistema
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Utente seleziona l'opzione di modifica del profilo 2. L'Utente modifica i dati desiderati 3. If I dati inseriti sono validi <ol style="list-style-type: none"> 3.1. Il sistema aggiorna i dati modificati 3.2. Include (<i>VisualizzaUtente</i>) 4. Else <ol style="list-style-type: none"> 4.1. Il sistema mostra un messaggio di errore e richiede una nuova modifica
Postcondizioni	<ol style="list-style-type: none"> 1. I dati anagrafici dell'Utente vengono aggiornati nel sistema
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If È lo Staff che richiede la modifica <ol style="list-style-type: none"> 1.1. Il sistema mostra l'elenco di tutti gli Operatori registrati 1.2. Lo Staff seleziona l'Operatore da modificare dall'elenco 1.3. Lo Staff modifica i dati desiderati 1.4. If I dati inseriti sono validi <ol style="list-style-type: none"> 1.4.1. Il sistema aggiorna i dati modificati 1.4.2. Include (<i>VisualizzaUtente</i>) 1.5. Else <ol style="list-style-type: none"> 1.5.1. Il sistema mostra un messaggio di errore e richiede una nuova modifica

Caso d'uso: EliminaUtente	
ID	8
Breve descrizione	L'Utente elimina il proprio account
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente è stato autenticato dal sistema
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Utente seleziona l'opzione di eliminazione del profilo 2. If L'Utente conferma l'operazione <ol style="list-style-type: none"> 2.1. Il sistema elimina l'Utente e mostra un messaggio di conferma 3. Else <ol style="list-style-type: none"> 3.1. Il sistema annulla l'operazione e non apporta modifiche
Postcondizioni	<ol style="list-style-type: none"> 1. If confermato <ol style="list-style-type: none"> 1.1. L'Utente selezionato viene rimosso dal sistema
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If È lo Staff che richiede l'eliminazione <ol style="list-style-type: none"> 1.1. Il sistema mostra l'elenco di tutti gli Operatori registrati 1.2. Lo Staff seleziona l'Operatore da eliminare dall'elenco 1.3. Lo Staff seleziona l'opzione di eleminazione 1.4. If Lo Staff conferma l'operazione <ol style="list-style-type: none"> 1.4.1. Il sistema elimina l'Utente e mostra un messaggio di conferma 1.5. Else <ol style="list-style-type: none"> 1.5.1. Il sistema annulla l'operazione

Gestione Assenza

Il seguente diagramma descrive le realazioni tra attori e casi d'uso per la gestione delle assenze del personale, come: aspettativa, ferie, malattia, maternità e permesso. Racchiude la possibilità di creazione, modifica ed eliminazione di assenze. Sono previsti, inoltre, casi d'uso per la consultazione di esse tramite le operazioni di visualizzazione.



Caso d'uso: CreaAssenza	
ID	9
Breve descrizione	Lo Staff crea una nuova assenza (aspettativa, ferie, malattia, maternità, permesso) per un Operatore
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Lo Staff è autenticato. 2. L'Operatore per cui viene registrata l'assenza esiste
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'operazione di creazione dell'assenza 2. Il sistema richiede i dati dell'assenza 3. Lo Staff inserisce i dati 4. If I dati sono validi <ol style="list-style-type: none"> 4.1. Il sistema registra l'assenza 4.2. Il sistema mostra un messaggio di conferma 5. Else <ol style="list-style-type: none"> 5.1. Il sistema mostra un errore e richiede la correzione
Postcondizioni	<ol style="list-style-type: none"> 1. L'assenza è registrata nel sistema.
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If Lo Staff non conferma l'operazione <ol style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche.

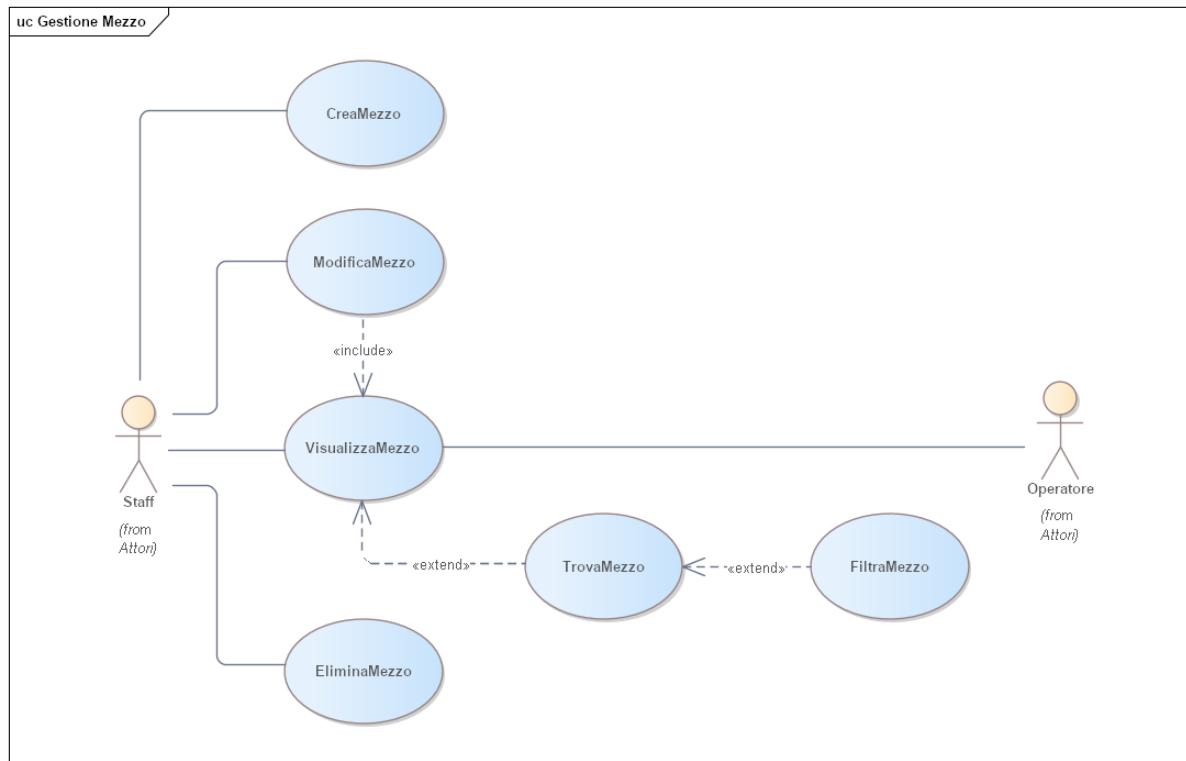
Caso d'uso: VisualizzaAssenza	
ID	10
Breve descrizione	L'Operatore visualizza le assenze registrate
Attori primari	Operatore
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none">1. L'Operatore è autenticato.2. Esistono assenze registrate.
Sequenza degli eventi principale	<ol style="list-style-type: none">1. Il caso d'uso inizia quando l'Operatore seleziona l'assenza da visualizzare2. Il sistema mostra i dettagli dell'assenza selezionata
Postcondizioni	<ol style="list-style-type: none">1. L'Operatore visualizza le informazioni sulle assenze.
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: ModificaAssenza	
ID	11
Breve descrizione	Lo Staff modifica i dati di un'assenza esistente
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è autenticato. 2. L'assenza esiste nel sistema.
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'assenza da modificare. 2. Lo Staff aggiorna i dati. 3. If I dati modificati sono validi <ul style="list-style-type: none"> 3.1. Il sistema aggiorna l'assenza. 3.2. Include (<i>VisualizzaAssenza</i>) 4. Else <ul style="list-style-type: none"> 4.1. Il sistema mostra un errore e non applica le modifiche.
Postcondizioni	<ul style="list-style-type: none"> 1. L'assenza è aggiornata nel sistema.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If Lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche.

Caso d'uso: EliminaAssenza	
ID	12
Breve descrizione	Lo Staff elimina un'assenza registrata.
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è autenticato. 2. L'assenza esiste nel sistema.
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di eliminazione dell'assenza 2. Il sistema elimina l'assenza e mostra un messaggio di conferma
Postcondizioni	<ul style="list-style-type: none"> 1. L'assenza è eliminata.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If Lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche.

Gestione Mezzo

Il seguente diagramma descrive le relazioni tra attori e casi d'uso per la gestione dei mezzi. Racchiude la possibilità di creazione, modifica ed eliminazione dei mezzi. Sono previsti, inoltre, casi d'uso per la consultazione di essi tramite le operazioni di visualizzazione, ricerca e filtraggio.



Caso d'uso: CreaMezzo	
ID	13
Breve descrizione	Lo Staff crea un nuovo mezzo nel sistema
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di creazione di un nuovo mezzo 2. Il sistema richiede l'inserimento dei dati 3. Lo Staff inserisce i dati 4. If i dati inseriti sono validi e non duplicati <ol style="list-style-type: none"> 4.1. Il sistema registra il nuovo mezzo 4.2. Il sistema mostra un messaggio di conferma 5. Else <ol style="list-style-type: none"> 5.1. Il sistema mostra un messaggio di errore e richiede la correzione dei dati
Postcondizioni	<ol style="list-style-type: none"> 1. Il nuovo mezzo è registrato nel sistema
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ol style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche.

Caso d'uso: VisualizzaMezzo	
ID	14
Breve descrizione	L'Operatore visualizza i dati di un mezzo.
Attori primari	Operatore
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'Operatore è stato autenticato dal sistema 2. Il mezzo da visualizzare esiste nel sistema.
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Operatore seleziona il mezzo da visualizzare 2. Il sistema mostra i dati del mezzo
Postcondizioni	<ul style="list-style-type: none"> 1. I dati del mezzo selezionato vengono visualizzati.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If Lo Staff richiede la visualizzazione <i>Extend TrovaMezzo</i> <ul style="list-style-type: none"> 1.1. Lo Staff seleziona il mezzo da visualizzare 1.2. Il sistema mostra i dati del mezzo

Caso d'uso: TrovaMezzo	
ID	15
Breve descrizione	Lo Staff esegue la ricerca di uno o più mezzi
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	1. Lo Staff è stato autenticato dal sistema
Sequenza degli eventi principale	1. Il caso d'uso inizia quando lo Staff inserisce il singolo criterio di ricerca nella barra dedicata Extend FiltraMezzo 2. While il criterio di ricerca non è vuoto 2.1. Il sistema trova i mezzi corrispondenti 2.2. If viene trovato almeno un mezzo 2.2.1. Il sistema mostra l'elenco dei mezzi corrispondenti. 2.3. Else 2.3.1. Il sistema mostra il messaggio "Nessun mezzo trovato"
Postcondizioni	1. Lo Staff visualizza l'elenco dei mezzi corrispondenti ai criteri
Sequenza degli eventi alternativa	Nessuna

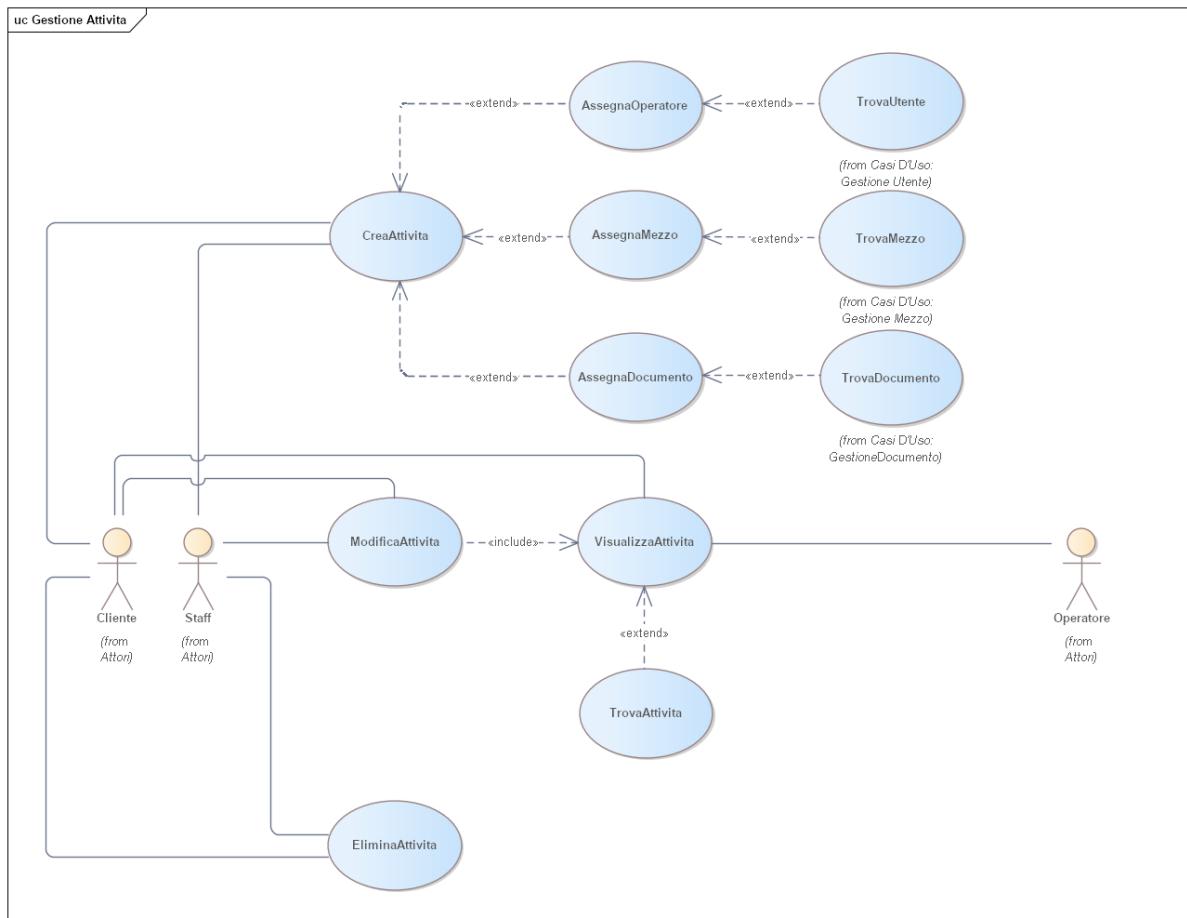
Caso d'uso: FiltraMezzo	
ID	16
Breve descrizione	Lo Staff filtra i mezzi in base a criteri avanzati
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema.
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff applica uno o più filtri durante l'azione di ricerca del mezzo 2. If esistono mezzi che rispettano i filtri <ul style="list-style-type: none"> 2.1. Il sistema mostra l'elenco dei mezzi filtrati 3. Else <ul style="list-style-type: none"> 3.1. Il sistema mostra "Nessun mezzo corrisponde ai criteri di filtro"
Postcondizioni	<ul style="list-style-type: none"> 1. Lo Staff visualizza l'elenco dei mezzi filtrati
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: ModificaMezzo	
ID	17
Breve descrizione	Lo Staff modifica i dati di un mezzo esistente
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema 2. Il mezzo da modificare esiste nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona il mezzo da modificare 2. Lo Staff modifica i dati del mezzo 3. If i dati inseriti sono validi <ul style="list-style-type: none"> 3.1. Il sistema aggiorna i dati del mezzo 3.2. Include (<i>VisualizzaMezzo</i>) 4. Else <ul style="list-style-type: none"> 4.1. Il sistema mostra un messaggio di errore e richiede una nuova modifica
Postcondizioni	<ul style="list-style-type: none"> 1. Il mezzo è aggiornato nel sistema
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: EliminaMezzo	
ID	18
Breve descrizione	Lo Staff elimina un mezzo dal sistema
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema 2. Il mezzo da eliminare esiste nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di eliminazione del mezzo 2. Il sistema elimina il mezzo e mostra un messaggio di conferma
Postcondizioni	<ul style="list-style-type: none"> 1. Il mezzo selezionato viene rimosso dal sistema.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If Lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche.

Gestione Attività

Il seguente diagramma descrive le relazioni tra attori e casi d'uso per la gestione delle attività. Racchiude la possibilità di creazione, modifica ed eliminazione delle attività, con assegnazione di operatori e mezzi associati. Sono previsti, inoltre, casi d'uso per la consultazione di esse tramite le operazioni di visualizzazione e ricerca.



Caso d'uso: CreaAttivita	
ID	19
Breve descrizione	Il Cliente o lo Staff crea una nuova attività di gestione dei rifiuti
Attori primari	Cliente, Staff
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Il Cliente o lo Staff è autenticato
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente o lo Staff seleziona l'opzione di creazione di una nuova attività 2. Il sistema richiede i dati dell'attività 3. Il Cliente o lo Staff inserisce i dati <i>Extend AssegnaOperatore</i> <i>Extend AssegnaMezzo</i> <i>Extend AssegnaDocumento</i> 4. If i dati sono validi <ol style="list-style-type: none"> 4.1. Il sistema registra la nuova attività 4.2. Il sistema mostra un messaggio di conferma 5. Else <ol style="list-style-type: none"> 5.1. Il sistema mostra un errore e richiede la correzione
Postcondizioni	<ol style="list-style-type: none"> 1. L'attività è creata nel sistema
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If l'operazione non è confermata <ol style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: AssegnaOperatore	
ID	20
Breve descrizione	Lo Staff assegna un Operatore ad un'attività
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'attività esiste 2. L'Operatore è registrato nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di assegnazione di un Operatore all'attività <i>Extend TrovaUtente</i> 2. Lo Staff seleziona un Operatore da assegnare all'attività 3. Il sistema registra l'assegnazione
Postcondizioni	<ul style="list-style-type: none"> 1. L'Operatore è assegnato all'attività.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: AssegnaMezzo	
ID	21
Breve descrizione	Lo Staff assegna un mezzo ad un'attività
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'attività esiste 2. Il mezzo è registrato nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di assegnazione di un mezzo all'attività Extend TrovaMezzo 2. Lo Staff seleziona un mezzo da assegnare all'attività 3. If Il mezzo è disponibile <ul style="list-style-type: none"> 3.1. Il sistema registra l'assegnazione 4. Else <ul style="list-style-type: none"> 4.1. Il sistema mostra il messaggio d'errore “Il mezzo non è disponibile”
Postcondizioni	<ul style="list-style-type: none"> 1. Il mezzo è associato all'attività
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: AssegnaDocumento	
ID	22
Breve descrizione	Lo Staff assegna un documento di tipo FIR ad un'attività
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'attività esiste 2. Il documento di tipo FIR è registrato nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di assegnazione di un documento all'attività <i>Extend TrovaDocumento</i> 2. Lo Staff seleziona un documento di tipo FIR da assegnare all'attività 3. Il sistema registra l'assegnazione
Postcondizioni	<ul style="list-style-type: none"> 1. Il documento di tipo FIR è assegnato all'attività.
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: VisualizzaAttivita	
ID	23
Breve descrizione	L'Utente visualizza l'attività selezionata
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente è autenticato
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di visualizzazione dell'attività <i>Extend TrovaAttivita</i> 2. Lo Staff seleziona un'attività da visualizzare 3. Il sistema mostra le informazioni sull'attività selezionata
Postcondizioni	Nessuna
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If Il Cliente o l'Operatore, possono selezionare solo le attività a loro associate, o da loro create <i>Extend TrovaAttivita</i> 2. Lo Staff seleziona un'attività da visualizzare 3. Il sistema mostra le informazioni sull'attività selezionata

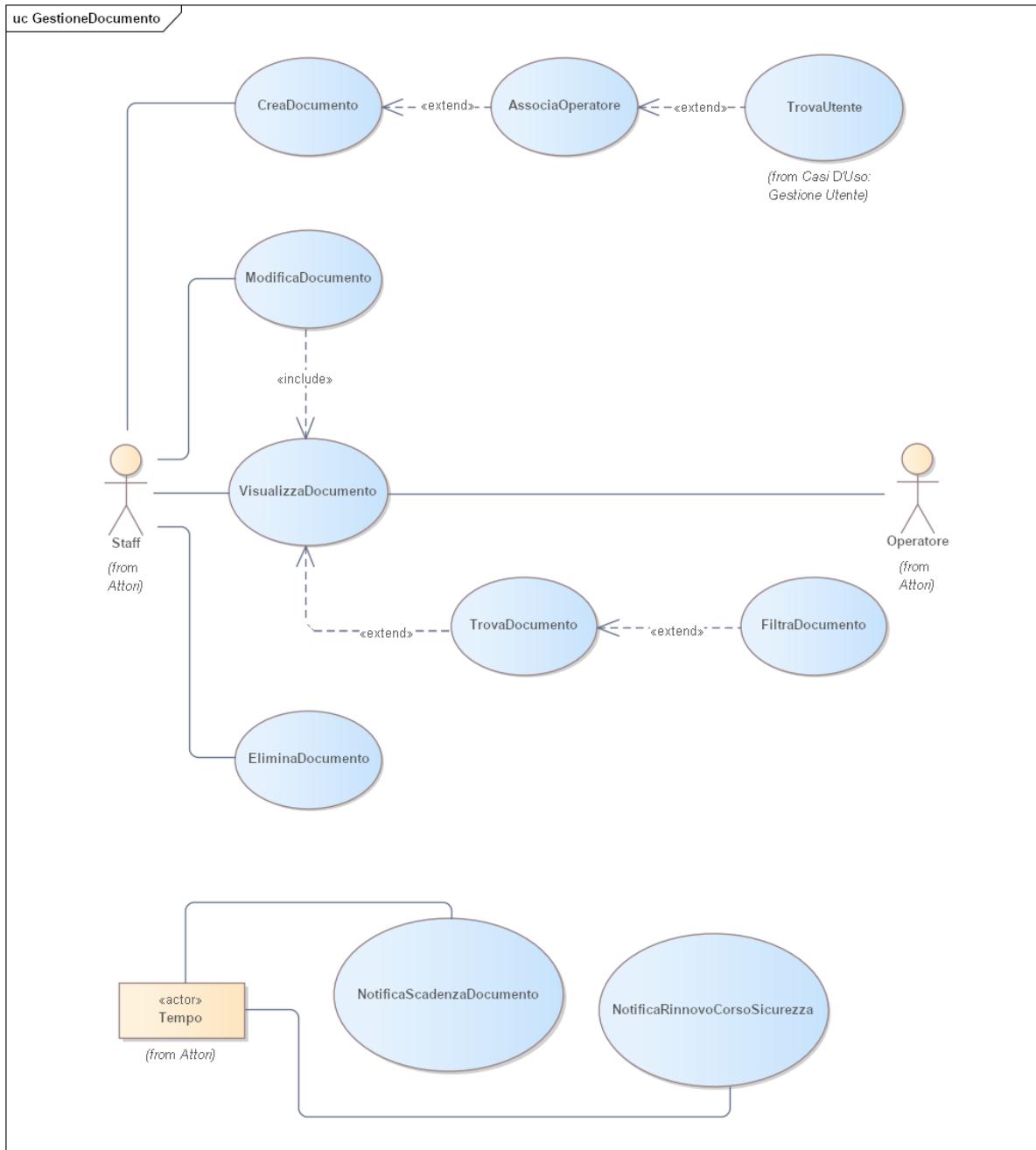
Caso d'uso: TrovaAttivita	
ID	24
Breve descrizione	L'Utente cerca un'attività
Attori primari	Utente
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. L'Utente è autenticato
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff inserisce il singolo criterio di ricerca nella barra dedicata 2. Il sistema esegue la ricerca 3. If l'attività esiste <ol style="list-style-type: none"> 3.1. For Each attività trovata <ol style="list-style-type: none"> 3.1.1. Il sistema mostra i risultati dell'attività 4. Else <ol style="list-style-type: none"> 4.1. Il sistema mostra "Nessuna attività trovata"
Postcondizioni	Nessuna
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If Il Cliente o l'Operatore inseriscono i criteri di ricerca nella barra dedicata <ol style="list-style-type: none"> 1.1. If l'attività esiste <ol style="list-style-type: none"> 1.1.1. Il sistema mostra i risultati dell'attività da lui creata, o a lui associata 1.2. Else <ol style="list-style-type: none"> 1.1. Il sistema mostra "Nessuna attività trovata"

Caso d'uso: ModificaAttivita	
ID	25
Breve descrizione	Il Cliente o lo Staff modifica un'attività esistente
Attori primari	Cliente, Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Il Cliente o lo Staff è autenticato 2. L'attività esiste
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente o lo Staff seleziona l'opzione di modifica dell'attività 2. Lo Staff aggiorna i dati dell'attività 3. If i dati sono validi <ul style="list-style-type: none"> 3.1. Il sistema registra le modifiche 3.2. Include (<i>VisualizzaAttivita</i>) 4. Else <ul style="list-style-type: none"> 4.1. Il sistema mostra un messaggio di errore
Postcondizioni	<ul style="list-style-type: none"> 1. L'attività è aggiornata nel sistema
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If l'operazione non è confermata <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Caso d'uso: EliminaAttivita	
ID	26
Breve descrizione	Il Cliente o lo Staff elimina un'attività esistente
Attori primari	Cliente, Staff
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Il Cliente o lo Staff è autenticato 2. L'attività esiste
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente o lo Staff seleziona l'opzione di eliminazione dell'attività 2. Il sistema elimina l'attività selezionata e mostra un messaggio di conferma
Postcondizioni	<ol style="list-style-type: none"> 1. L'attività è eliminata
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If l'operazione non è confermata <ol style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e non apporta modifiche

Gestione Documento

Il seguente diagramma descrive le relazioni tra attori e casi d'uso per la gestione dei documenti. Racchiude la possibilità di creazione, modifica ed eliminazione dei documenti. Sono previsti, inoltre, casi d'uso per la consultazione di essi tramite le operazioni di visualizzazione, ricerca e filtraggio. I vari documenti e il rinnovo corsi per la sicurezza sul lavoro vengono notificati in caso di scadenza.



Caso d'uso: CreaDocumento	
ID	27
Breve descrizione	Lo Staff crea un nuovo documento nel sistema
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di creazione del documento 2. Il sistema richiede i dati del documento 3. Lo Staff inserisce i dati <i>Extend AssociaOperatore</i> 4. If i dati sono validi e non duplicati <ol style="list-style-type: none"> 4.1. Il sistema registra il documento 5. Else <ol style="list-style-type: none"> 5.1. Il sistema segnala un errore e richiede la correzione
Postcondizioni	<ol style="list-style-type: none"> 1. Il documento è registrato nel sistema
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ol style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e nessun documento viene creato

Caso d'uso: AssociaOperatore	
ID	28
Breve descrizione	Lo Staff un Operatore ad un documento di tipo non FIR
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è stato autenticato dal sistema 2. Il documento di tipo non FIR esiste 3. L'Operatore è registrato nel sistema
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di assegnazione di un documento di tipo non FIR ad Operatore <i>Extend TrovaUtente</i> 2. Lo Staff seleziona un Operatore da associare al documento di tipo non FIR 3. Il sistema registra l'assegnazione
Postcondizioni	<ul style="list-style-type: none"> 1. L'Operatore è associato al documento di tipo non FIR
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: VisualizzaDocumento	
ID	29
Breve descrizione	L'Operatore visualizza i dati di un documento
Attori primari	Operatore
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Operatore è autenticato 2. Il documento esiste
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Operatore seleziona l'opzione di visualizzazione del documento <i>Extend TrovaDocumento</i> 2. L'Operatore seleziona un documento 3. Il sistema mostra i dati del documento selezionato
Postcondizioni	Nessuna
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: TrovaDocumento	
ID	30
Breve descrizione	L'Operatore ricerca uno o più documenti
Attori primari	Operatore
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'Operatore è autenticato.
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Operatore inserisce il singolo criterio di ricerca nella barra dedicata <i>Extend FiltraDocumento</i> 2. If il documento esiste <ul style="list-style-type: none"> 2.1. For Each documento trovato <ul style="list-style-type: none"> 2.1.1. Il sistema mostra le informazioni base del documento 3. Else <ul style="list-style-type: none"> 3.1. Il sistema mostra "Nessun documento trovato"
Postcondizioni	Nessuna
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: FiltraDocumento	
ID	31
Breve descrizione	L'Operatore filtra i documenti in base a criteri avanzati
Attori primari	Operatore
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. L'Operatore è autenticato
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando l'Operatore applica filtri avanzati 2. If esistono documenti che rispettano i filtri <ul style="list-style-type: none"> 2.1. Il sistema mostra l'elenco dei documenti filtrati 3. Else <ul style="list-style-type: none"> 3.1. Il sistema mostra "Nessun documento corrisponde ai criteri di filtro"
Postcondizioni	<ul style="list-style-type: none"> 1. L'Operatore visualizza i documenti filtrati
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: ModificaDocumento	
ID	32
Breve descrizione	Lo Staff modifica i dati di un documento esistente
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è autenticato 2. Il documento esiste
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di modifica del documento 2. Lo Staff modifica i dati del documento 3. If i dati modificati sono validi <ul style="list-style-type: none"> 3.1. Il sistema aggiorna il documento 3.2. Include (<i>VisualizzaDocumento</i>) 4. Else <ul style="list-style-type: none"> 4.1. Il sistema mostra un messaggio di errore
Postcondizioni	<ul style="list-style-type: none"> 1. I dati del documento risultano aggiornati
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e il documento non viene modificato

Caso d'uso: EliminaDocumento	
ID	33
Breve descrizione	Lo Staff elimina un documento dal sistema
Attori primari	Staff
Attori secondari	Nessuno
Precondizioni	<ul style="list-style-type: none"> 1. Lo Staff è autenticato 2. Il documento esiste
Sequenza degli eventi principale	<ul style="list-style-type: none"> 1. Il caso d'uso inizia quando lo Staff seleziona l'opzione di eliminazione del documento 2. Il sistema elimina il documento selezionato e mostra un messaggio di conferma
Postcondizioni	<ul style="list-style-type: none"> 1. Il documento è eliminato
Sequenza degli eventi alternativa	<ul style="list-style-type: none"> 1. If lo Staff non conferma l'operazione <ul style="list-style-type: none"> 1.1. Il sistema annulla l'operazione e il documento viene eliminato

Caso d'uso: NotificaRinnovoCorsoSicurezza	
ID	34
Breve descrizione	Il sistema invia le notifiche relative al rinnovo dei corsi sulla sicurezza
Attori primari	Tempo
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> Esistono corsi sulla sicurezza con data di scadenza registrata
Sequenza degli eventi principale	<ol style="list-style-type: none"> Il caso d'uso inizia quando al raggiungimento della data predefinita, l'attore Tempo innesca il controllo dei corsi in scadenza For Each corso in scadenza <ol style="list-style-type: none"> Il sistema invia la notifica all'Operatore o allo Staff interessato
Postcondizioni	<ol style="list-style-type: none"> Le notifiche di rinnovo dei corsi sulla sicurezza sono state inviate
Sequenza degli eventi alternativa	Nessuna

Caso d'uso: NotificaScadenzaDocumento	
ID	35
Breve descrizione	Il sistema invia la notifica per la scadenza dei documenti
Attori primari	Tempo
Attori secondari	Nessuno
Precondizioni	<ol style="list-style-type: none"> 1. Esistono documenti con data di scadenza registrata
Sequenza degli eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando al raggiungimento della data pre-definita, l'attore Tempo innesca il controllo dei documenti in scadenza 2. For Each ogni documento in scadenza <ol style="list-style-type: none"> 2.1. Il sistema invia la notifica allo Staff o all'Operatore
Postcondizioni	<ol style="list-style-type: none"> 1. Le notifiche di scadenza dei documenti sono state inviate
Sequenza degli eventi alternativa	Nessuna

Matrice di Mapping

		Target		
			Source	
	Gestione Assenza::CRUD_Assenza		Casi D'Uso: Gestione Assenza::CreaAssenza	↑
	Gestione Attivita::Assegnazione Documento		Casi D'Uso: Gestione Assenza::EliminaAssenza	↑
	Gestione Attivita::Assegnazione Mezzo		Casi D'Uso: Gestione Assenza::ModificaAssenza	↑
	Gestione Attivita::Assegnazione Operatore		Casi D'Uso: Gestione Assenza::VisualizzaAssenza	↑
	Gestione Attivita::CRUD Attivita		Casi D'Uso: Gestione Attivita::AssegnaDocumento	↑
	Gestione Documento::Associa Operatore		Casi D'Uso: Gestione Attivita::AssegnaMezzo	↑
	Gestione Documento::CRUD Documento		Casi D'Uso: Gestione Attivita::AssegnaOperatore	↑
	Gestione Documento::Filtro Documento		Casi D'Uso: Gestione Attivita::CreaAttivita	↑
	Gestione Documento::Notifica Rinnovo Corso Sicurezza		Casi D'Uso: Gestione Attivita::EliminaAttivita	↑
	Gestione Documento::Notifica Scadenza Documento		Casi D'Uso: Gestione Attivita::ModificaAttivita	↑
	Gestione Mezzo::CRUD Mezzo		Casi D'Uso: Gestione Attivita::TrovaAttivita	↑
	Gestione Mezzo::Filtro Mezzo		Casi D'Uso: Gestione Attivita::VisualizzaAttivita	↑
	Gestione Utente::Elimina Utente		Casi D'Uso: Gestione Mezzo::CreaMezzo	↑
	Gestione Utente::Login Utente		Casi D'Uso: Gestione Mezzo::EliminaMezzo	↑
	Gestione Utente::Modifica Ruolo		Casi D'Uso: Gestione Mezzo::FiltraMezzo	↑
	Gestione Utente::Modifica Utente		Casi D'Uso: Gestione Mezzo::ModificaMezzo	↑
	Gestione Utente::Recupero Credenziali		Casi D'Uso: Gestione Mezzo::TrovaMezzo	↑
	Gestione Utente::Registrazione Utente		Casi D'Uso: Gestione Mezzo::VisualizzaMezzo	↑
			Casi D'Uso: Gestione Utente::EliminaUtente	↑
			Casi D'Uso: Gestione Utente::LoginUtente	↑
			Casi D'Uso: Gestione Utente::ModificaRuolo	↑
			Casi D'Uso: Gestione Utente::ModificaUtente	↑
			Casi D'Uso: Gestione Utente::RecuperoCredenziali	↑
			Casi D'Uso: Gestione Utente::RegistrazioneUtente	↑
			Casi D'Uso: Gestione Utente::TrovaUtente	↑
			Casi D'Uso: Gestione Utente::VisualizzaUtente	↑
	GestioneDocumento::AssociaOperatore		Casi D'Uso: GestioneDocumento::CreaDocumento	↑
	GestioneDocumento::EliminaDocumento		Casi D'Uso: GestioneDocumento::EliminaDocumento	↑
	GestioneDocumento::FiltroDocumento		Casi D'Uso: GestioneDocumento::FiltroDocumento	↑
	GestioneDocumento::ModificaDocumento		Casi D'Uso: GestioneDocumento::ModificaDocumento	↑
	GestioneDocumento::NotificaRinnovoCorsoSicurezza		Casi D'Uso: GestioneDocumento::NotificaRinnovoCorsoSicurezza	↑
	GestioneDocumento::NotificaScadenzaDocumento		Casi D'Uso: GestioneDocumento::NotificaScadenzaDocumento	↑
	GestioneDocumento::TrovaDocumento		Casi D'Uso: GestioneDocumento::TrovaDocumento	↑
	GestioneDocumento::VisualizzaDocumento		Casi D'Uso: GestioneDocumento::VisualizzaDocumento	↑

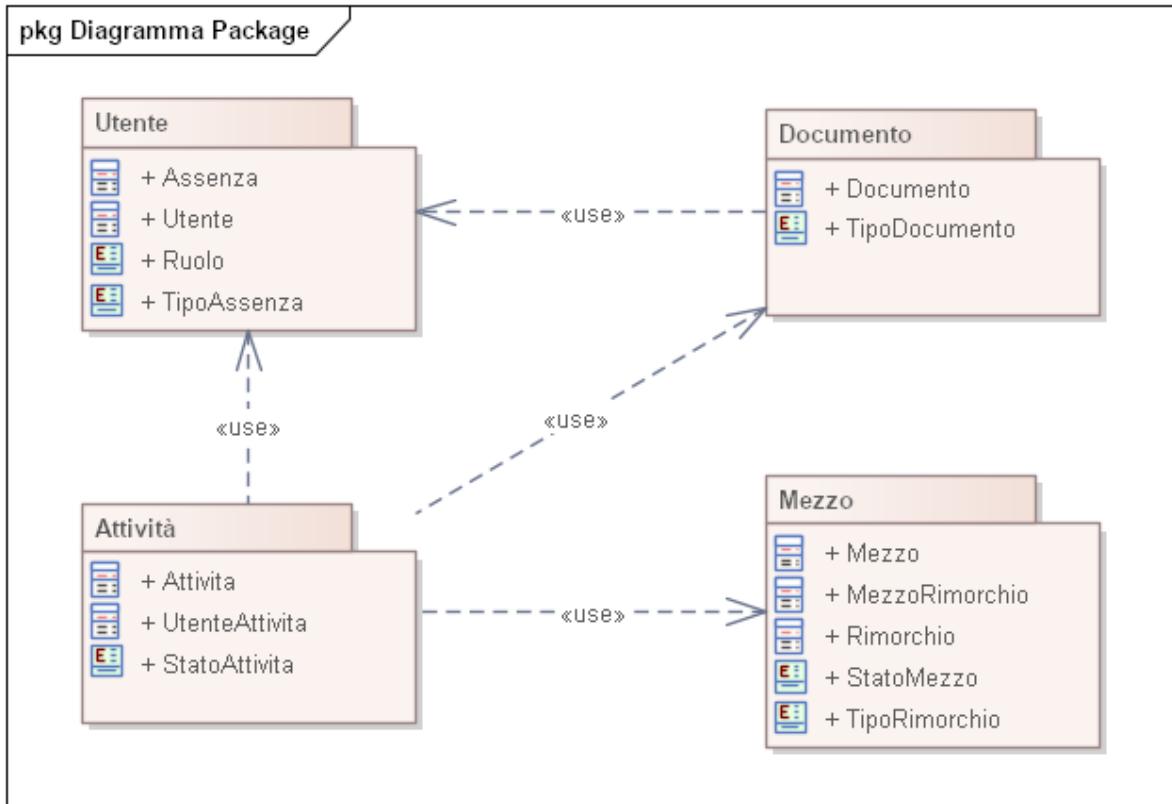
Analisi

Nella presente sezione si raccolgono i principali diagrammi UML utilizzati per modellare il sistema. I diagrammi delle classi, che rappresentano le entità principali e le loro relazioni. I diagrammi di sequenza, che mostrano l'ordine delle interazioni tra gli oggetti. Infine i diagrammi di attività, che descrivono il flusso delle operazioni e dei processi all'interno del sistema.

Diagrammi delle Classi di Analisi

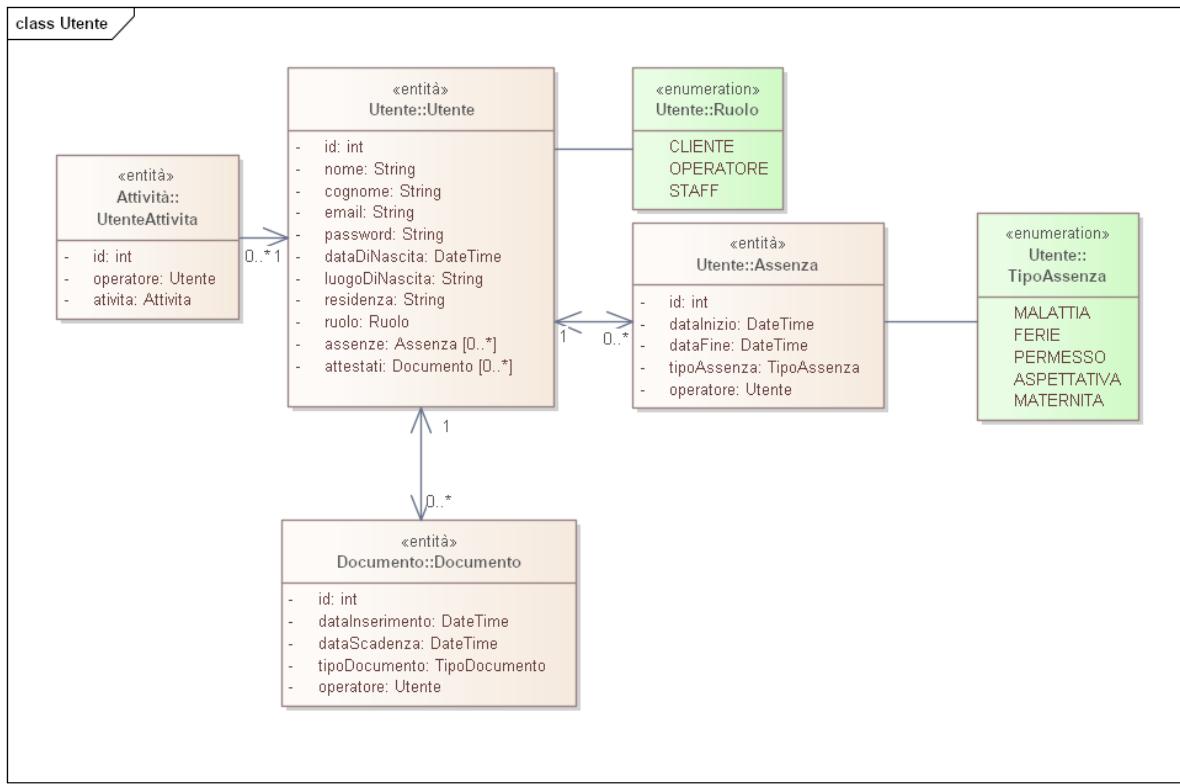
Package di Analisi

Il diagramma mostra la struttura dei package del sistema e le dipendenze tra i moduli principali, evidenziando come le classi siano organizzate per responsabilità.



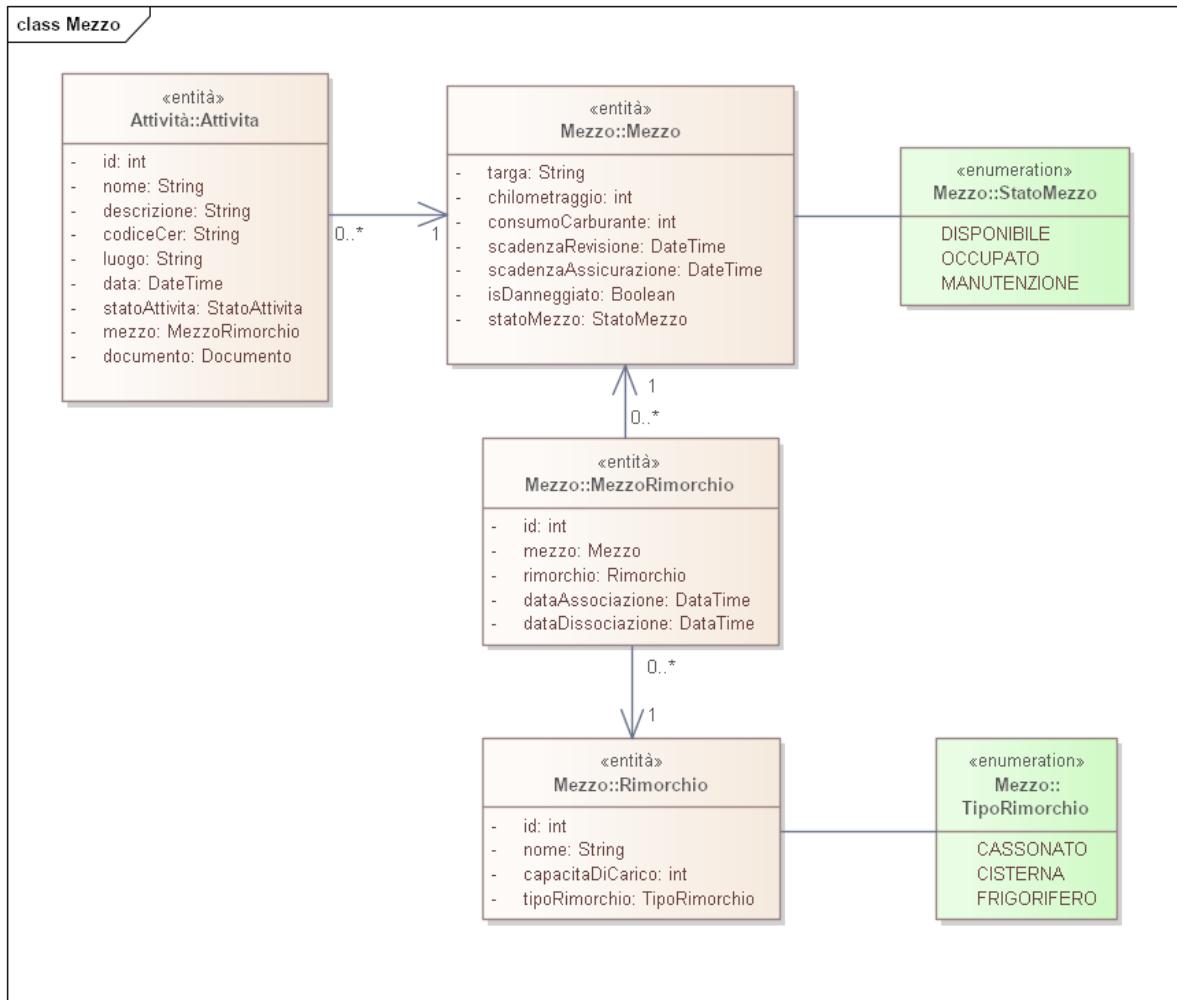
Classi di Analisi: Utente

Il diagramma evidenzia le relazioni tra la classe Utente e le altre classi del sistema, mettendo in luce associazioni, dipendenze e responsabilità condivise tra i diversi ruoli.



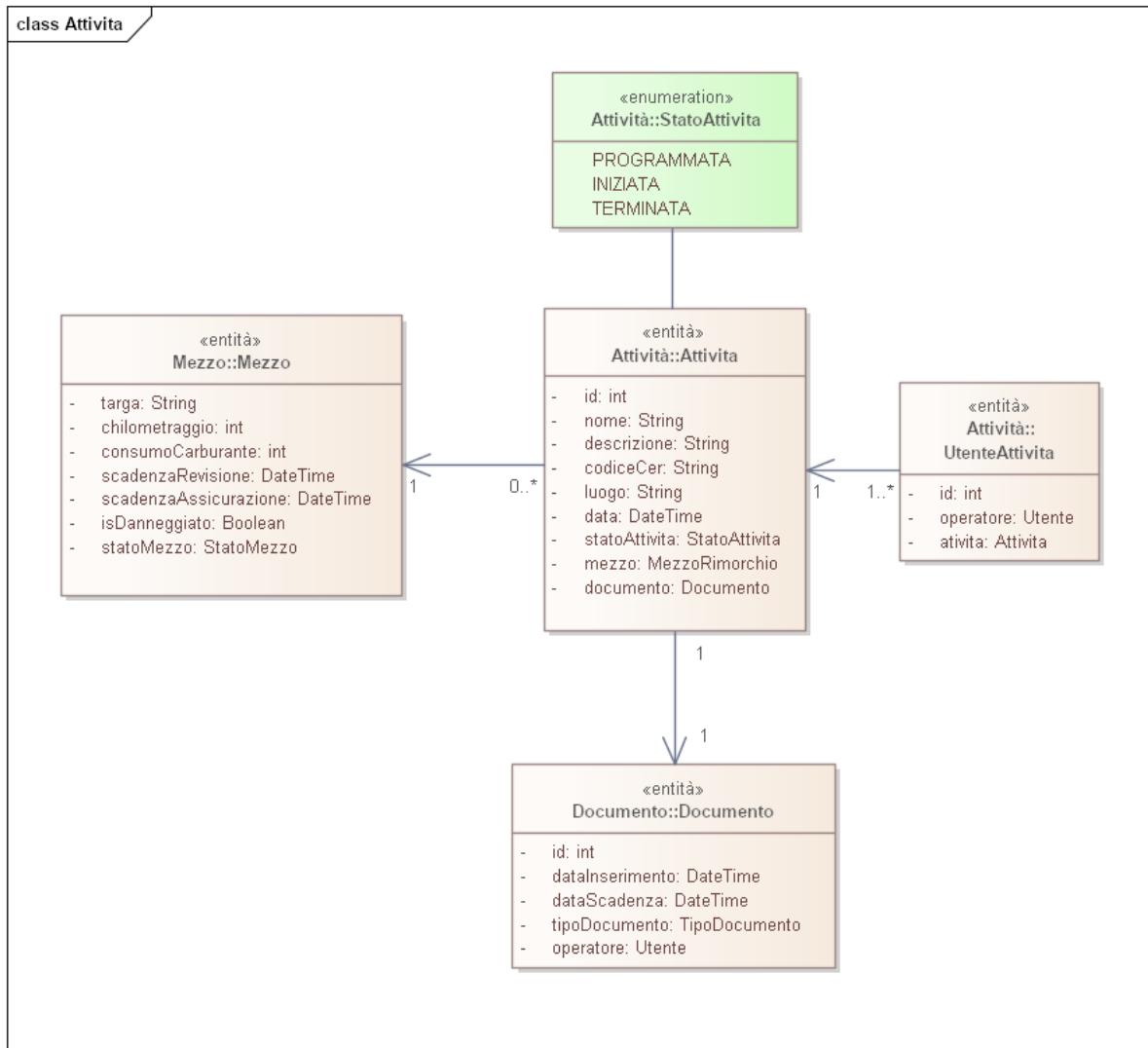
Classi di Analisi: Mezzo

Il diagramma mette in evidenza le classi che modellano i mezzi, le loro proprietà e le relazioni con le attività e le informazioni di manutenzione.



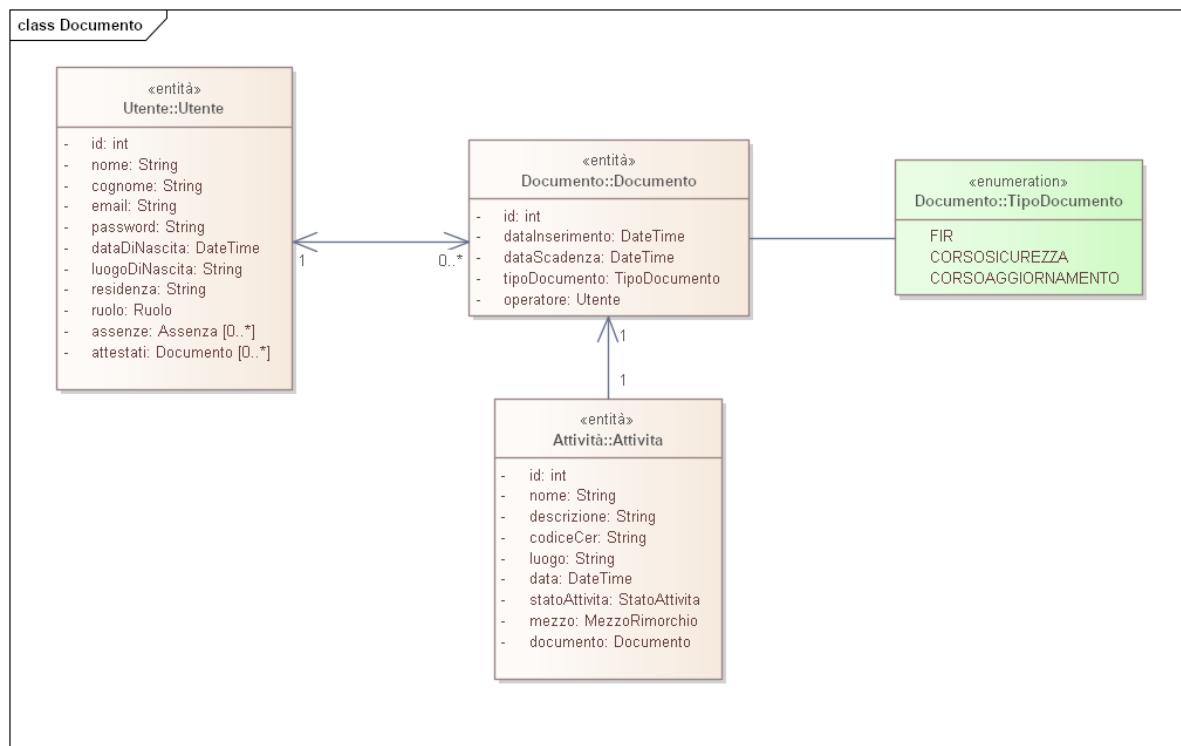
Classi di Analisi: Attività

Il diagramma descrive le classi coinvolte nella gestione delle attività di carico/scarico, mostrando attributi, operazioni e collegamenti con operatori e mezzi.



Classi di Analisi: Documento

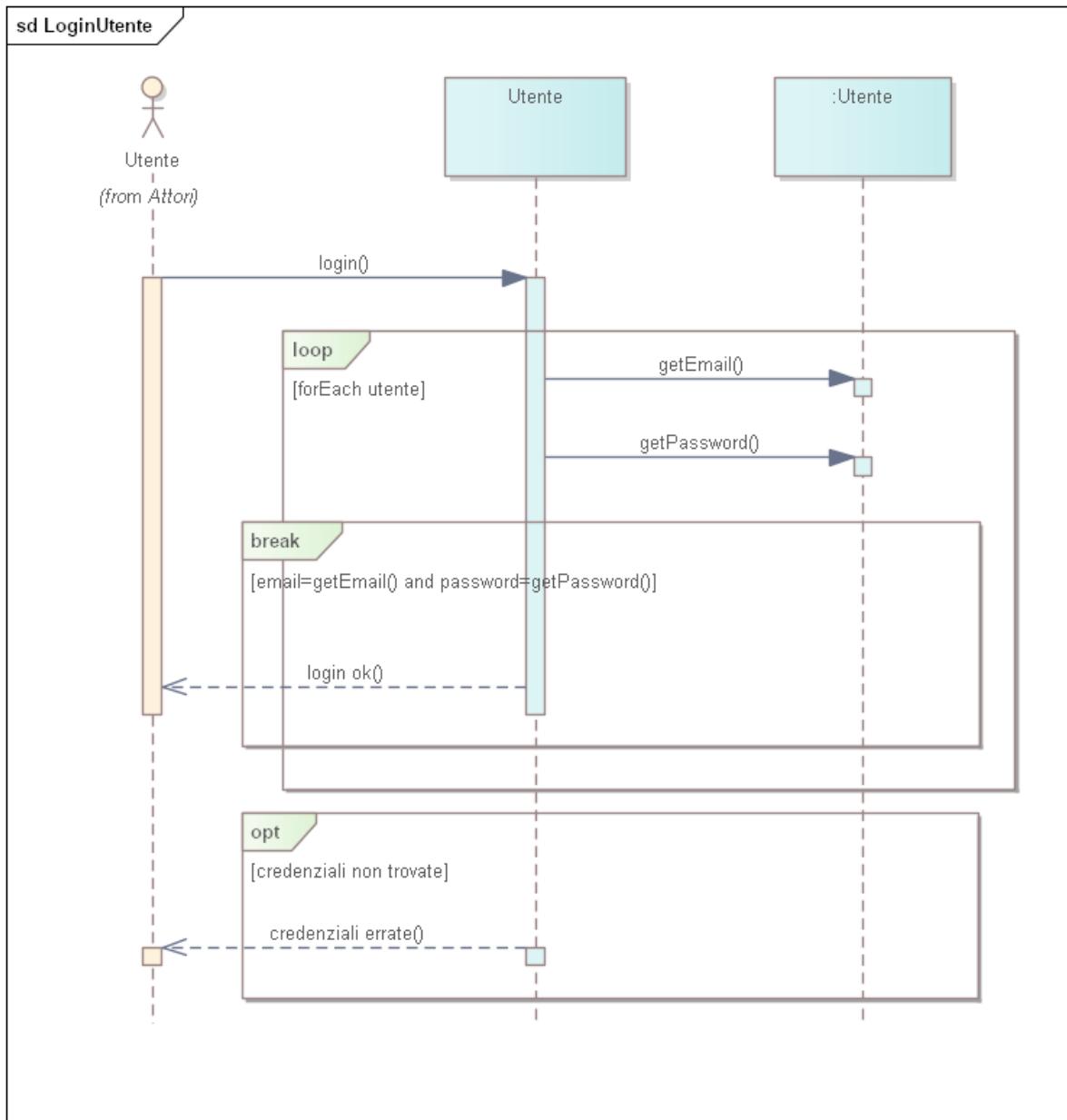
Il diagramma illustra la struttura delle classi che rappresentano i documenti (es. FIR), con le relazioni e le responsabilità nel ciclo di vita dei documenti.



Diagrammi di Sequenza

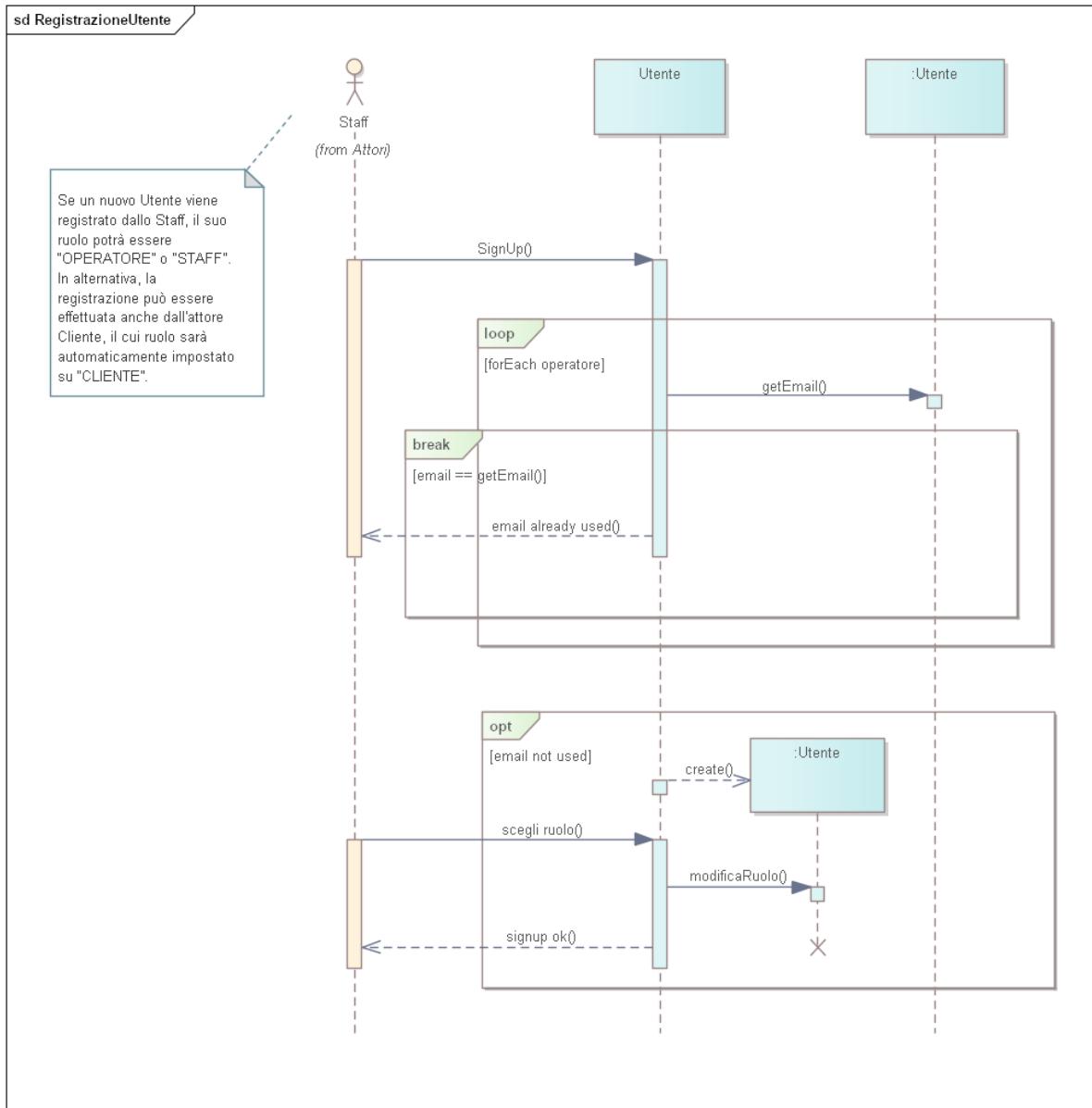
Login Utente

Il diagramma di sequenza seguente mette in evidenza il processo di accesso dell'utente al sistema: l'utente inserisce le proprie credenziali che vengono inoltrate a un servizio di autenticazione esterno per la validazione; in caso di esito positivo viene stabilita la sessione utente e l'applicazione reindirizza l'utente alla vista corrispondente al suo ruolo (ad esempio Staff, Operatore o Cliente). In caso di autenticazione fallita, il sistema restituisce un messaggio di errore e richiede un nuovo tentativo.



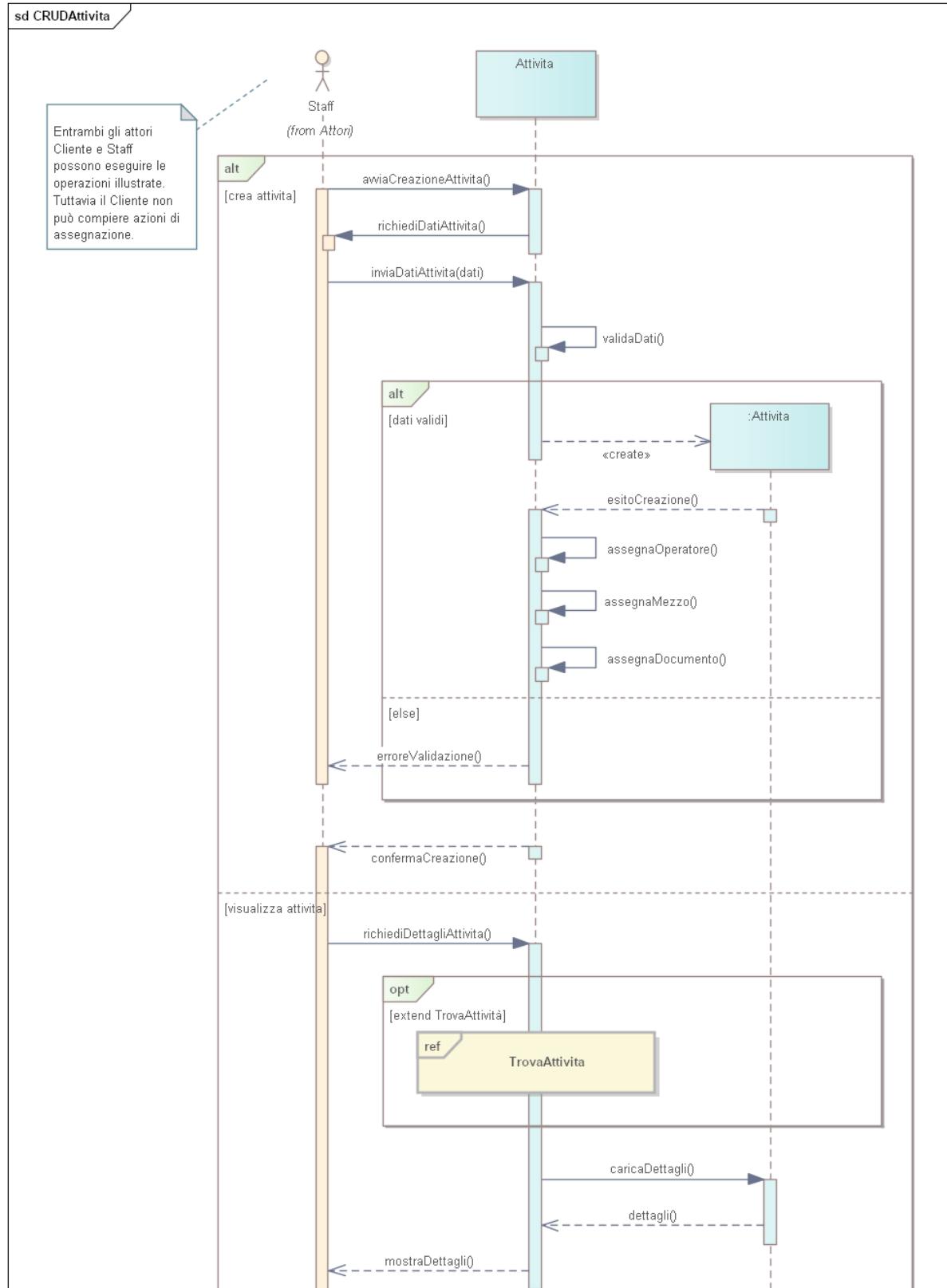
Registrazione Utente

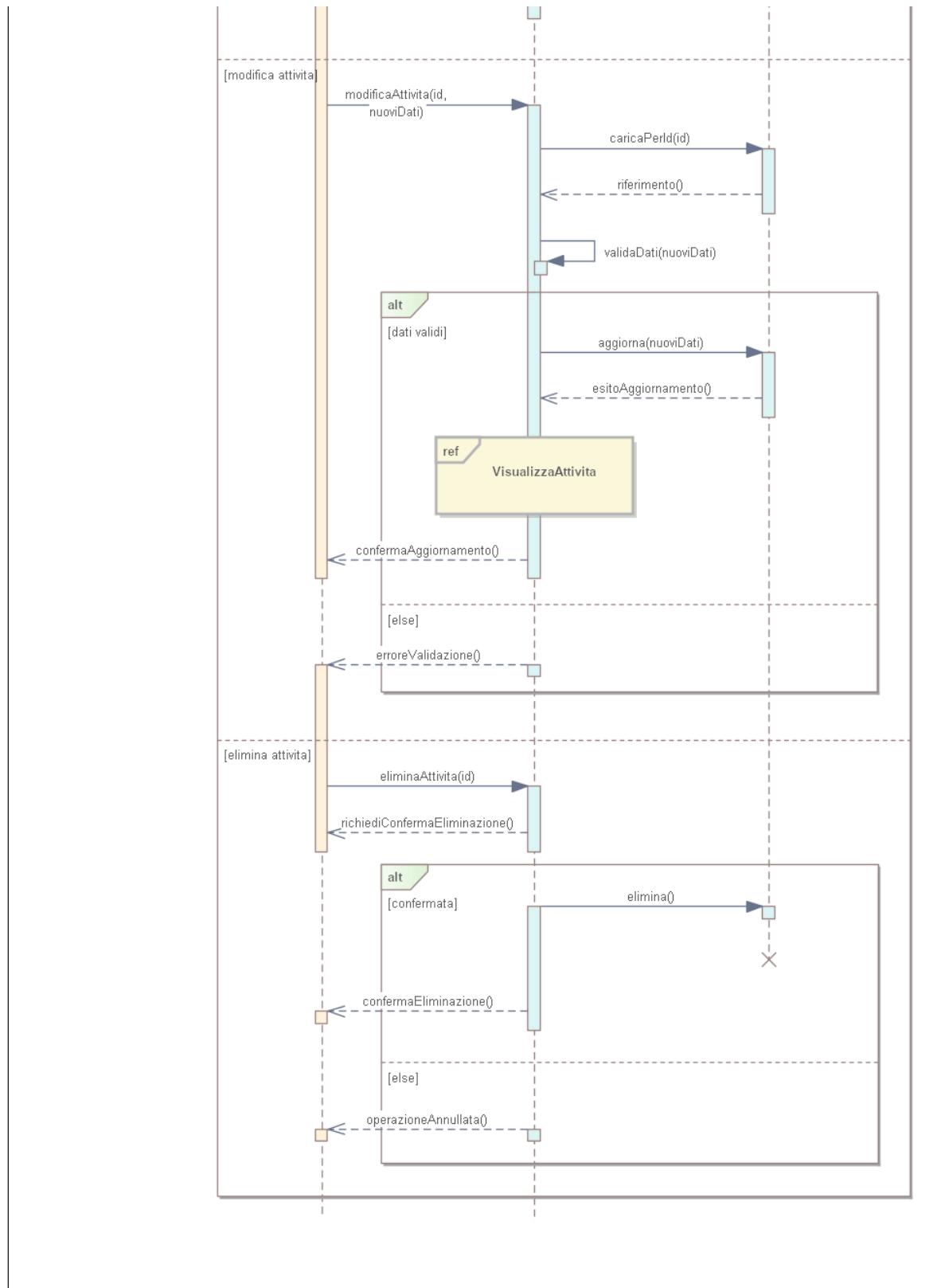
Il diagramma di sequenza mostra il processo di registrazione di un nuovo utente al sistema. Viene evidenziato il controllo sull'unicità dell'email fornita e, in caso positivo, la creazione del relativo account. In alternativa, se l'email risulta già utilizzata, il sistema restituisce un messaggio di errore.



CRUD Attività

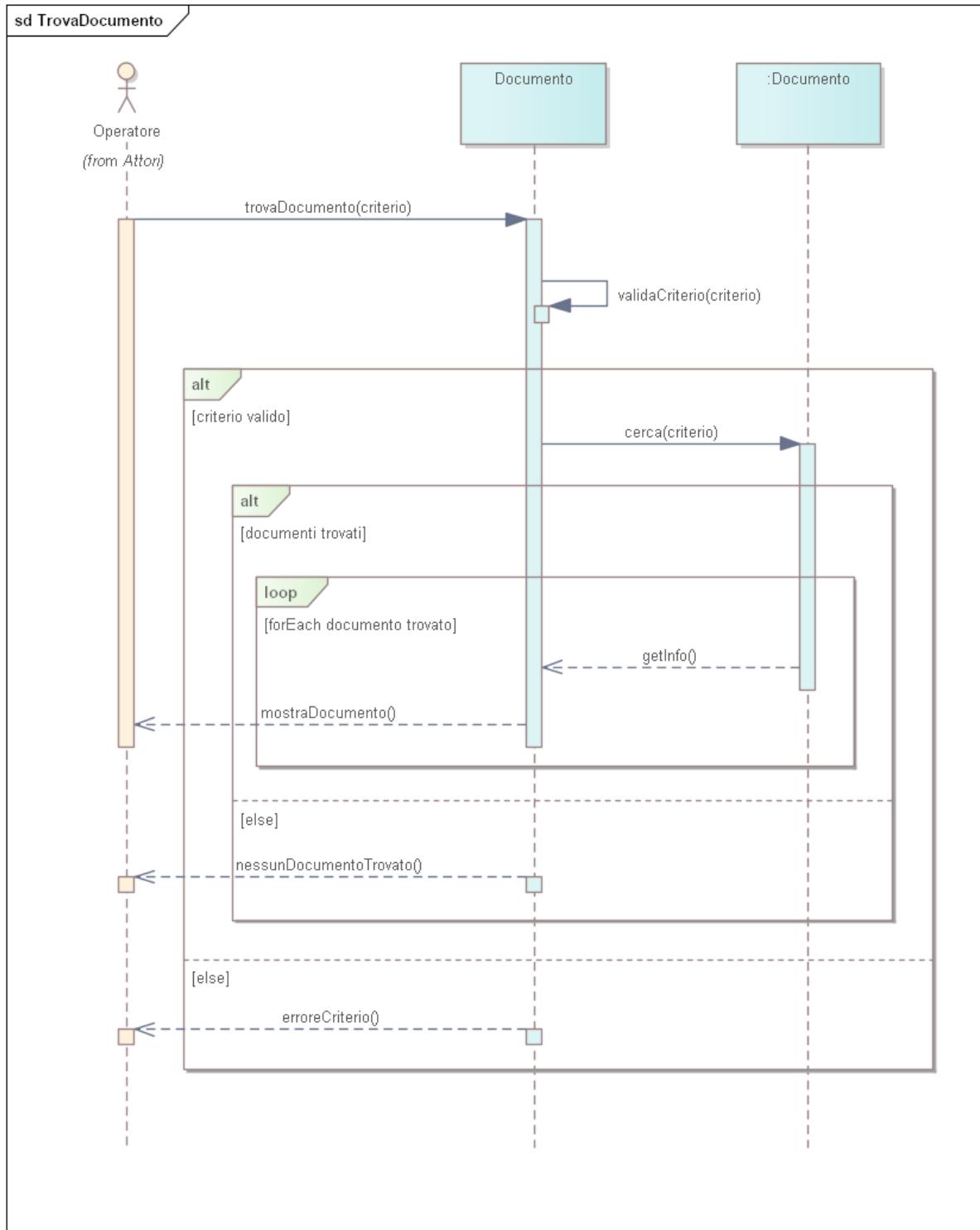
Il seguente diagramma di sequenza illustra le operazioni principali di gestione delle attività, comprendendo creazione, visualizza, modifica e eliminazione. Mostra come l'attore interagisce con il sistema per eseguire ciascuna di queste funzioni, evidenziando i passaggi di validazione dei dati e l'aggiornamento coerente delle informazioni da parte del sistema.





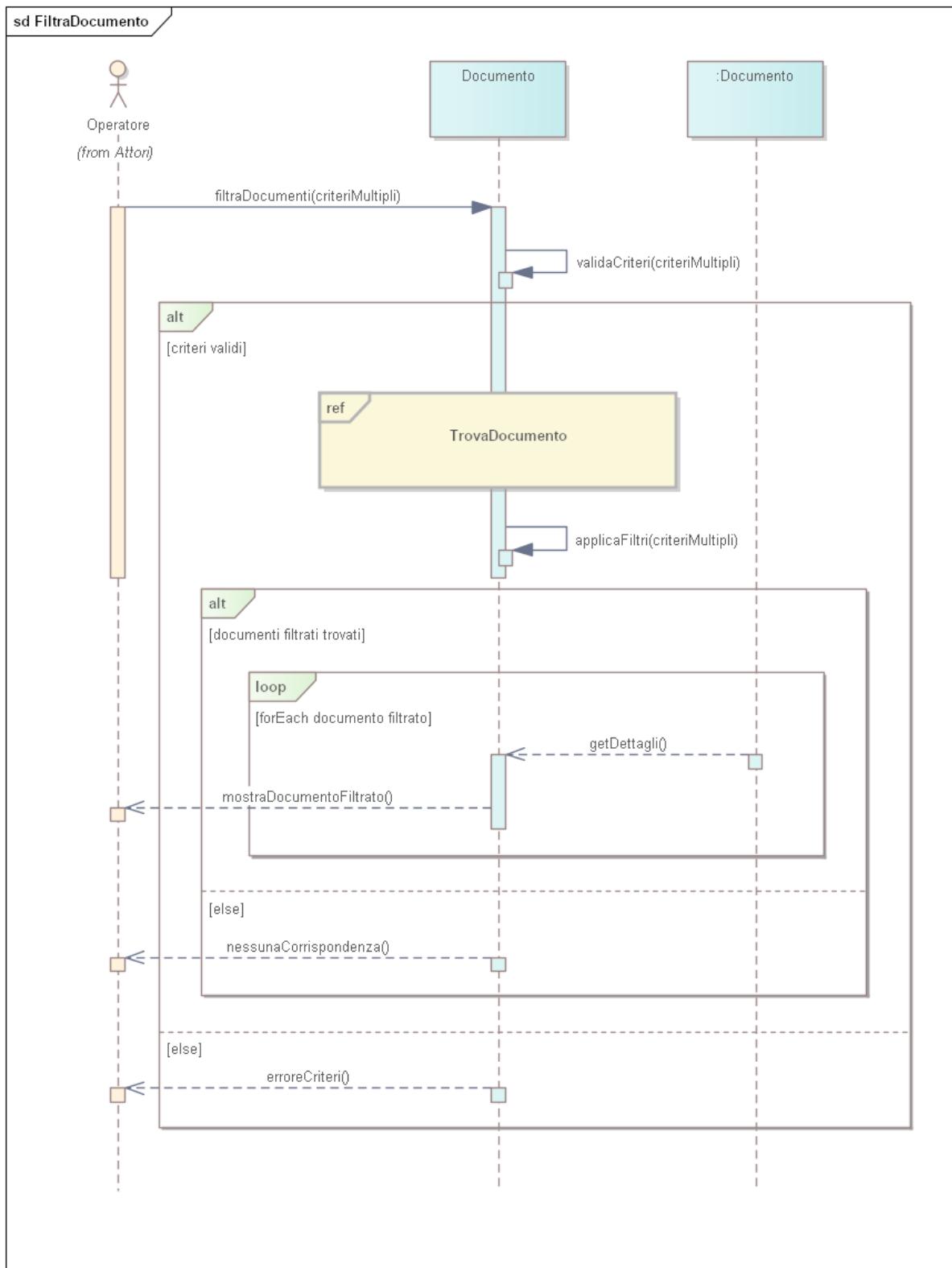
Trova Documento

Il diagramma di sequenza illustra le interazioni tra l'operatore e il sistema per la ricerca di un documento tramite un criterio semplice. Viene evidenziata la validazione del criterio inserito e, se valido, la restituzione dei risultati corrispondenti. In alternativa, il sistema notifica l'assenza di documenti oppure segnala un errore se il criterio non è valido.



Filtra Documento

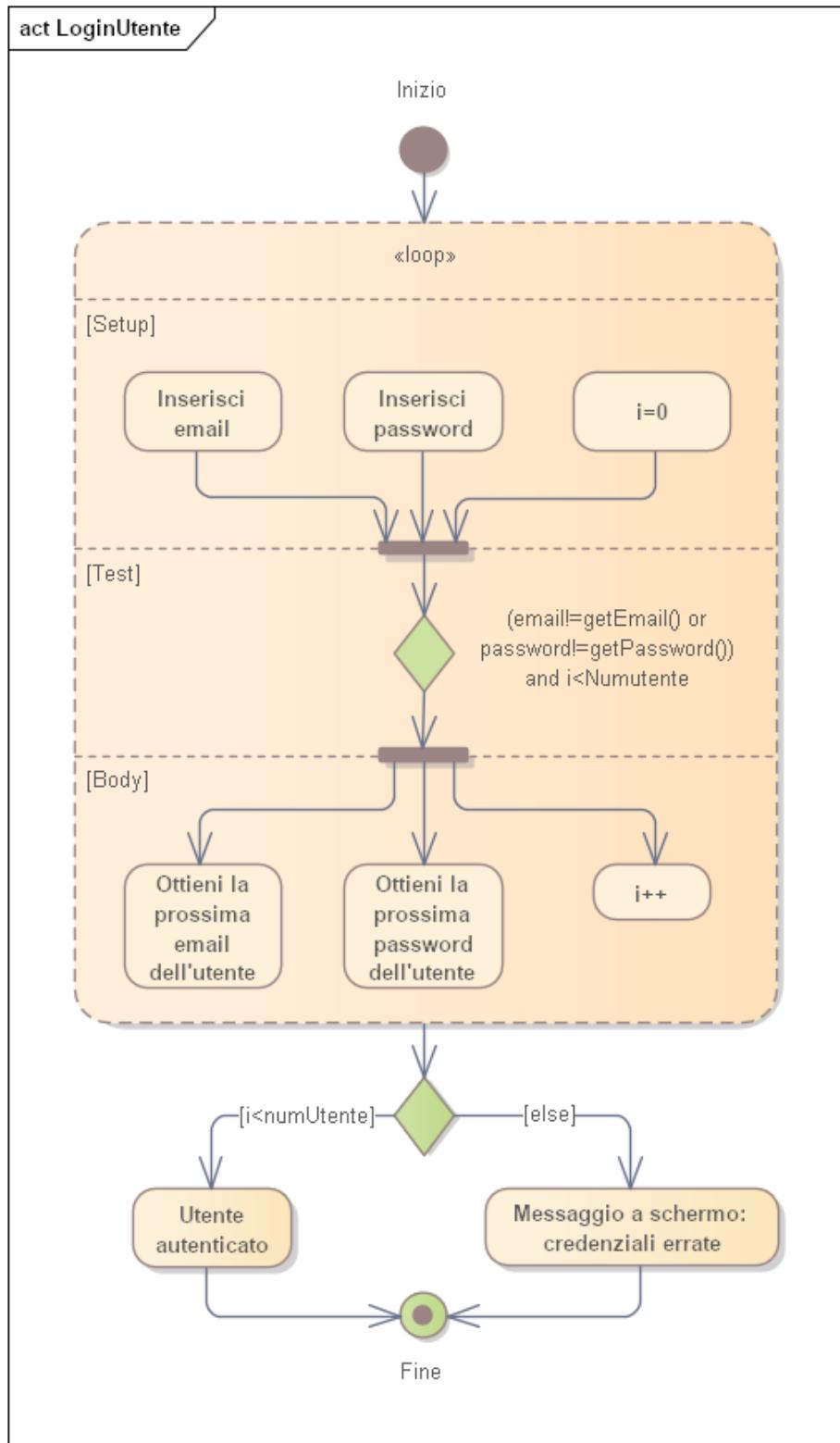
Il diagramma di sequenza mostra il processo di filtraggio avanzato dei documenti, che estende il caso d'uso TrovaDocumento. Dopo la validazione dei criteri multipli, viene richiamata la logica di TrovaDocumento e successivamente applicati i filtri per restringere i risultati. L'operatore riceve così l'elenco dei documenti filtrati o, in alternativa, un messaggio di assenza di corrispondenze; in caso di criteri non validi, viene segnalato un errore.



Diagrammi di Attività

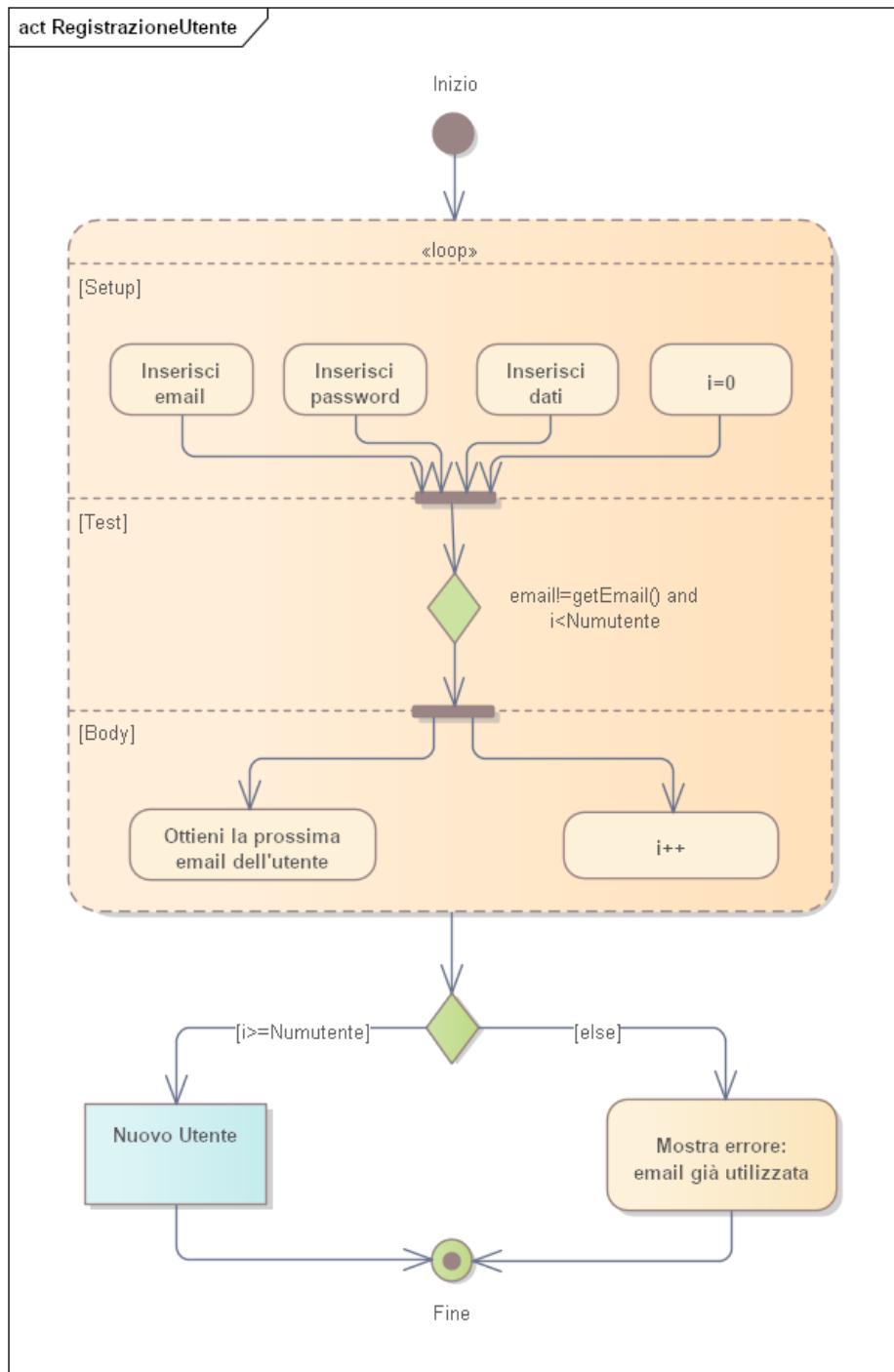
Login Utente

Il diagramma rappresenta il processo di autenticazione di un utente. Dopo l'inserimento delle credenziali, il sistema le valida e consente l'accesso se corrette; altrimenti richiede un nuovo tentativo mostrando un messaggio di errore.



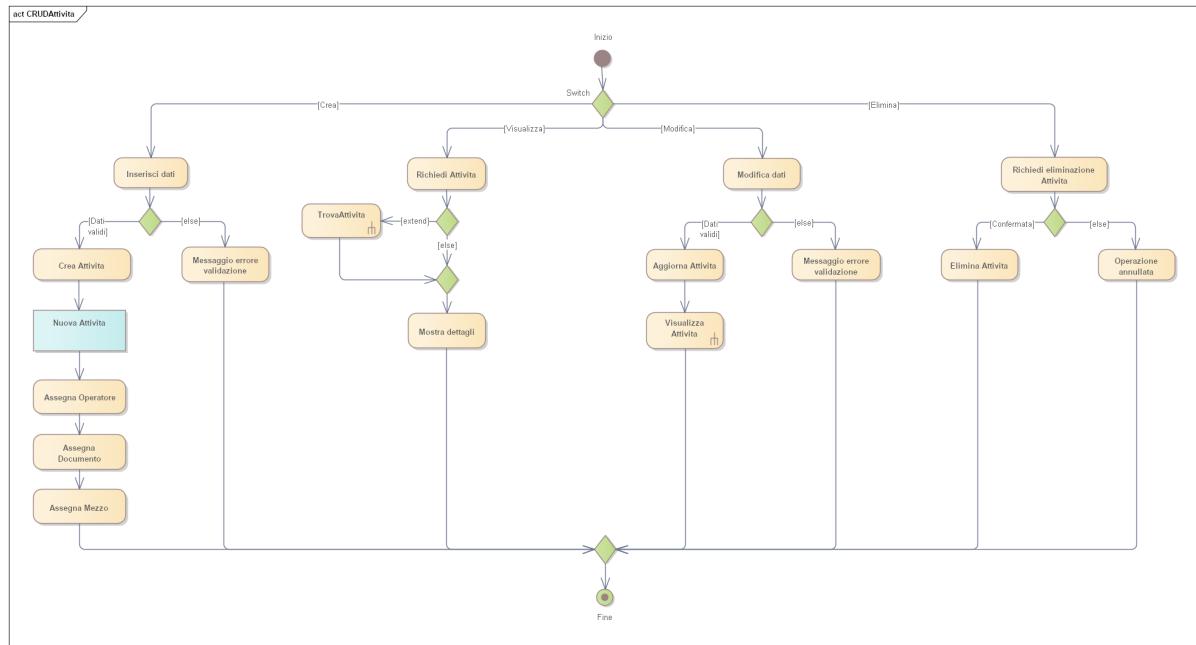
Registrazione Utente

Il diagramma mostra il processo di registrazione di un nuovo utente, sia da parte di un cliente sia da parte dello staff. Dopo l'inserimento dei dati, il sistema effettua la validazione e controlla l'unicità dell'email. In caso di errori o email già esistente, l'utente deve ripetere l'inserimento finché i dati non sono corretti. Solo dopo i controlli di validazione dell'email viene creato l'account con il ruolo appropriato e confermata l'avvenuta registrazione.



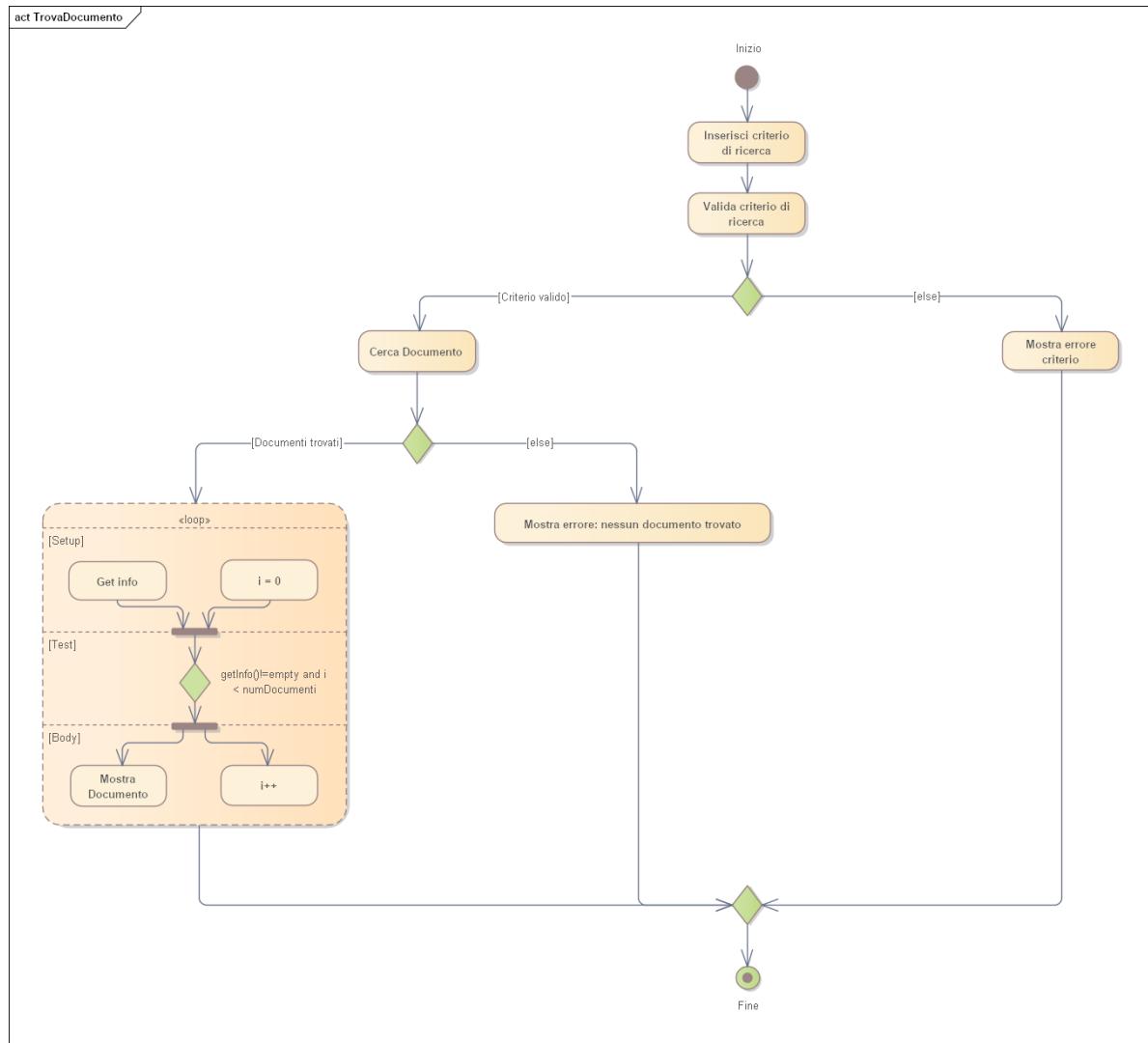
CRUD Attività

Il diagramma illustra la gestione completa delle attività attraverso operazioni di creazione, visualizzazione, modifica ed eliminazione. Le operazioni includono controlli di validazione e, se necessario, conferme da parte dell'utente. In fase di creazione sono previsti anche flussi quali l'assegnazione di un operatore e di un mezzo. Il diagramma evidenzia quindi le alternative e le condizioni che regolano il ciclo di vita di un'attività.



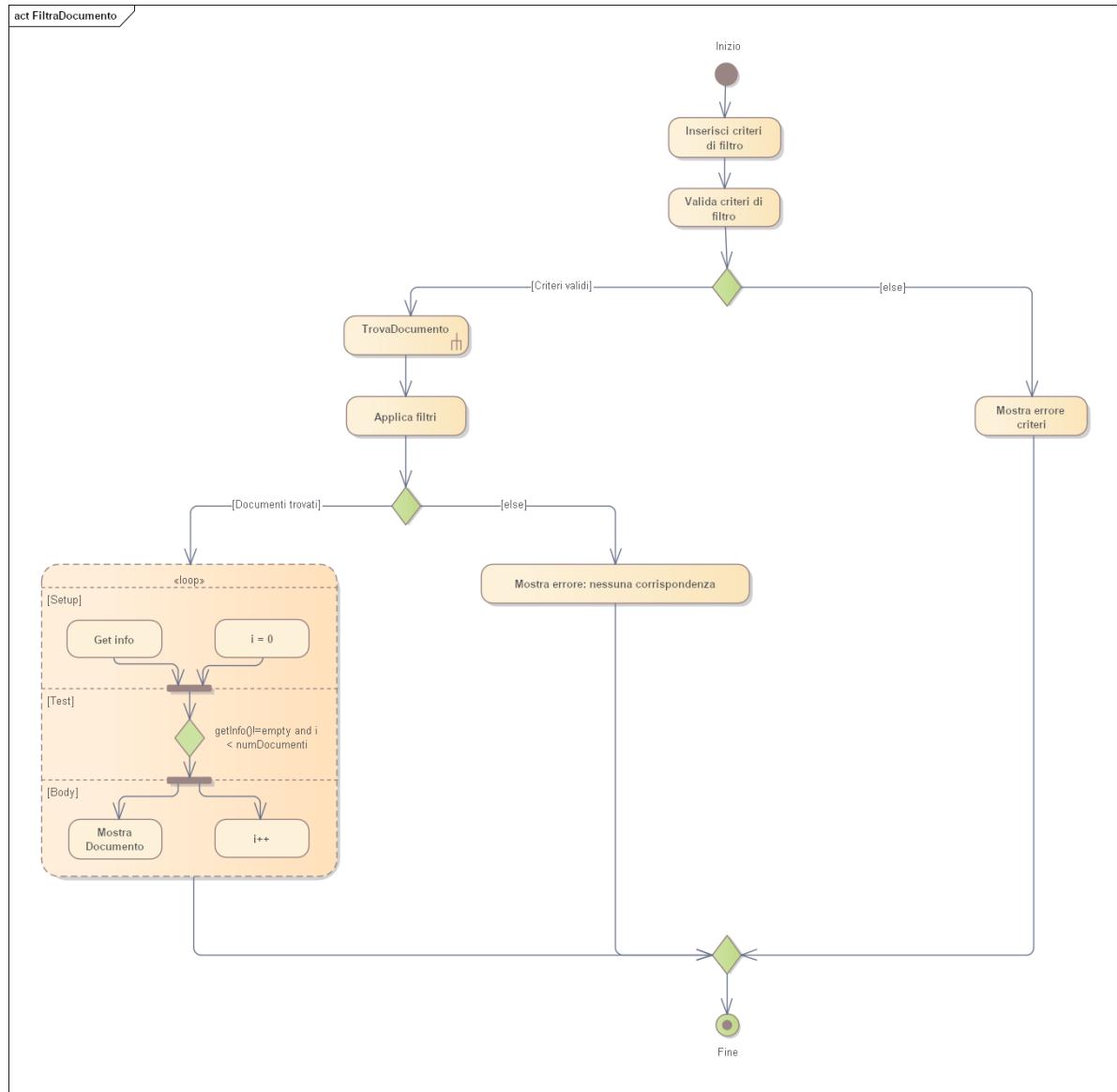
Trova Documento

Ricerca semplice basata su un singolo criterio: il sistema valida il criterio, restituisce l'elenco dei documenti trovati (iterando su ciascuno) o segnala l'assenza di risultati / l'errore di validazione.



Filtra Documento

Filtraggio avanzato che estende la ricerca base: dopo la validazione dei criteri multipli viene eseguita la logica di TrovaDocumento e, se ci sono risultati base, si applicano i filtri avanzati per ottenere l'elenco finale o l'assenza di corrispondenze.



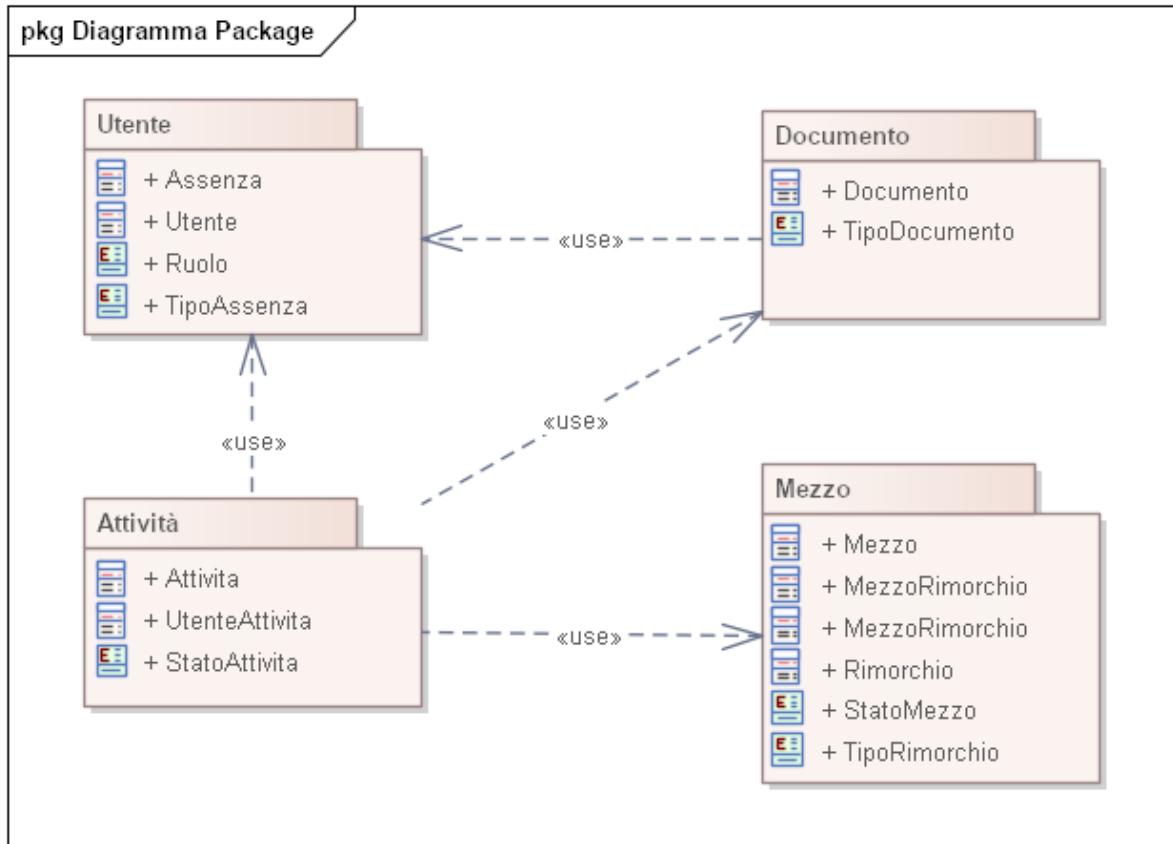
Progettazione

Nella presente sezione si presentano i diagrammi utilizzati per definire la struttura e l'architettura del sistema. I diagrammi delle classi di progettazione, che dettagliano le classi con attributi e metodi. Il diagramma dei componenti, che mostra l'organizzazione dei moduli software. I diagrammi delle macchine a stati, che descrivono i comportamenti e le transizioni degli oggetti. Infine, il diagramma ER, che rappresenta la struttura del database e le relazioni tra le entità.

Diagrammi delle Classi di Progettazione

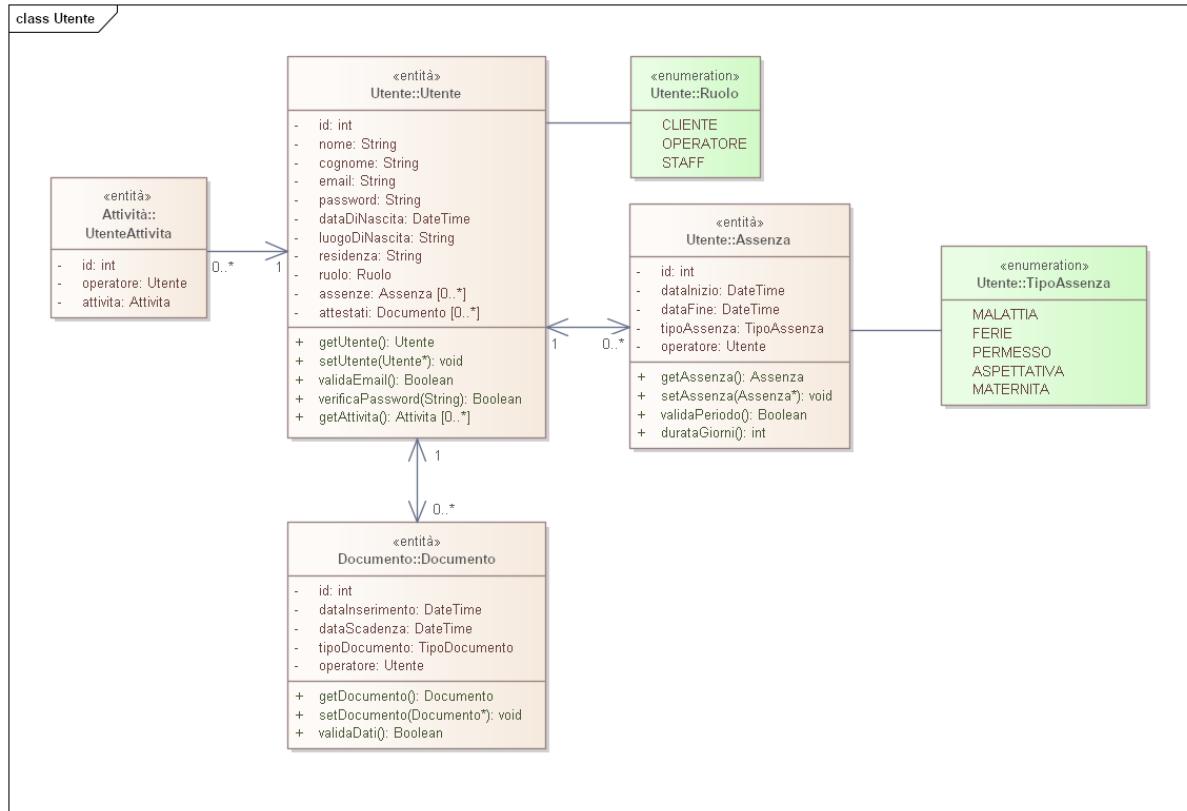
Package di Progettazione

Il diagramma mostra la struttura dei package del sistema e le dipendenze tra i moduli principali, evidenziando come le classi siano organizzate per responsabilità.



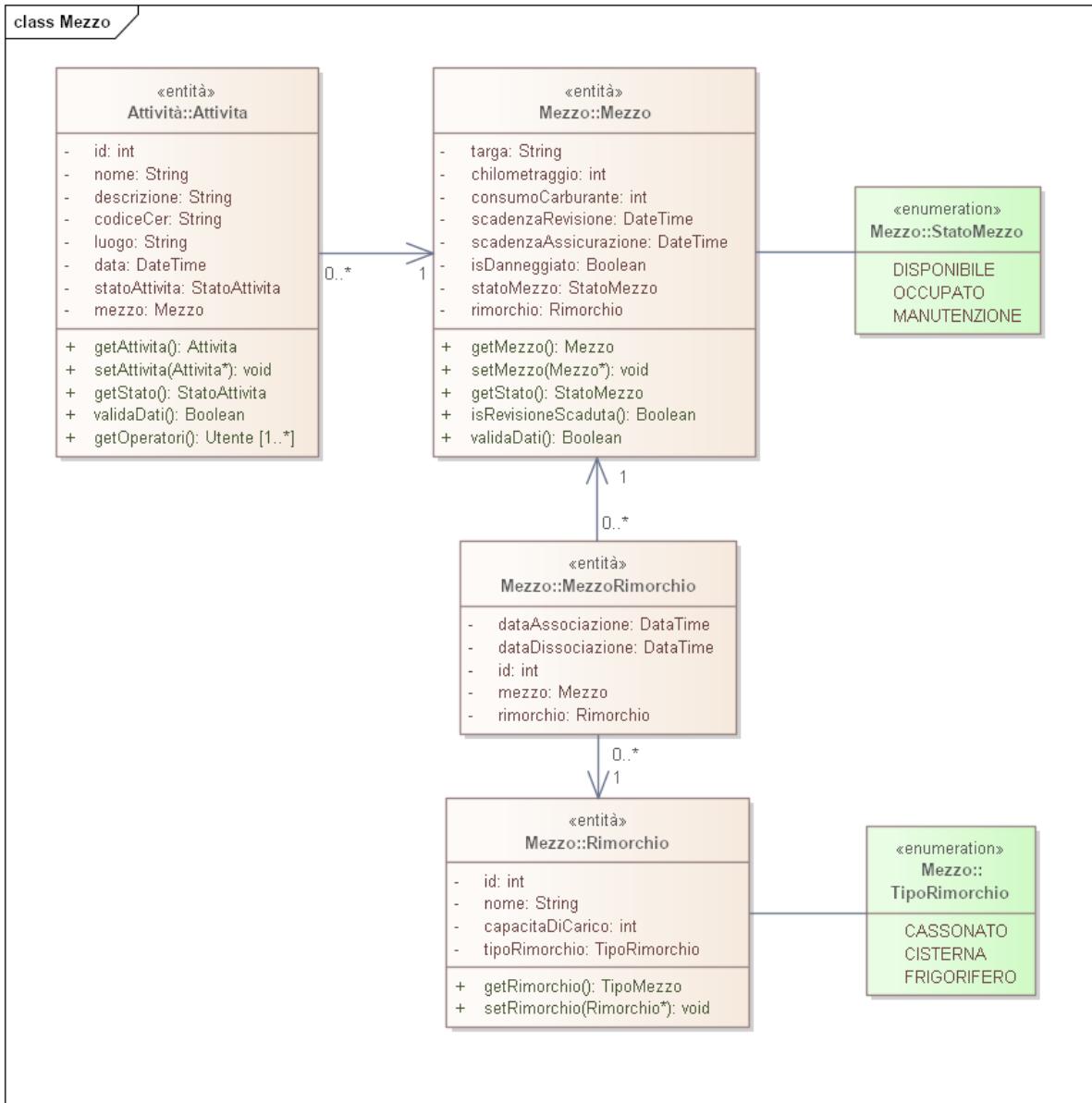
Classi di Progettazione: Utente

Il diagramma delle classi di Utente e Assenza è una rappresentazione che riprende ed espande l'analisi iniziale. Le classi Utente e Assenza sono state arricchite con i dettagli di progettazione necessari per l'implementazione, inclusi attributi e metodi per la gestione e l'autenticazione.



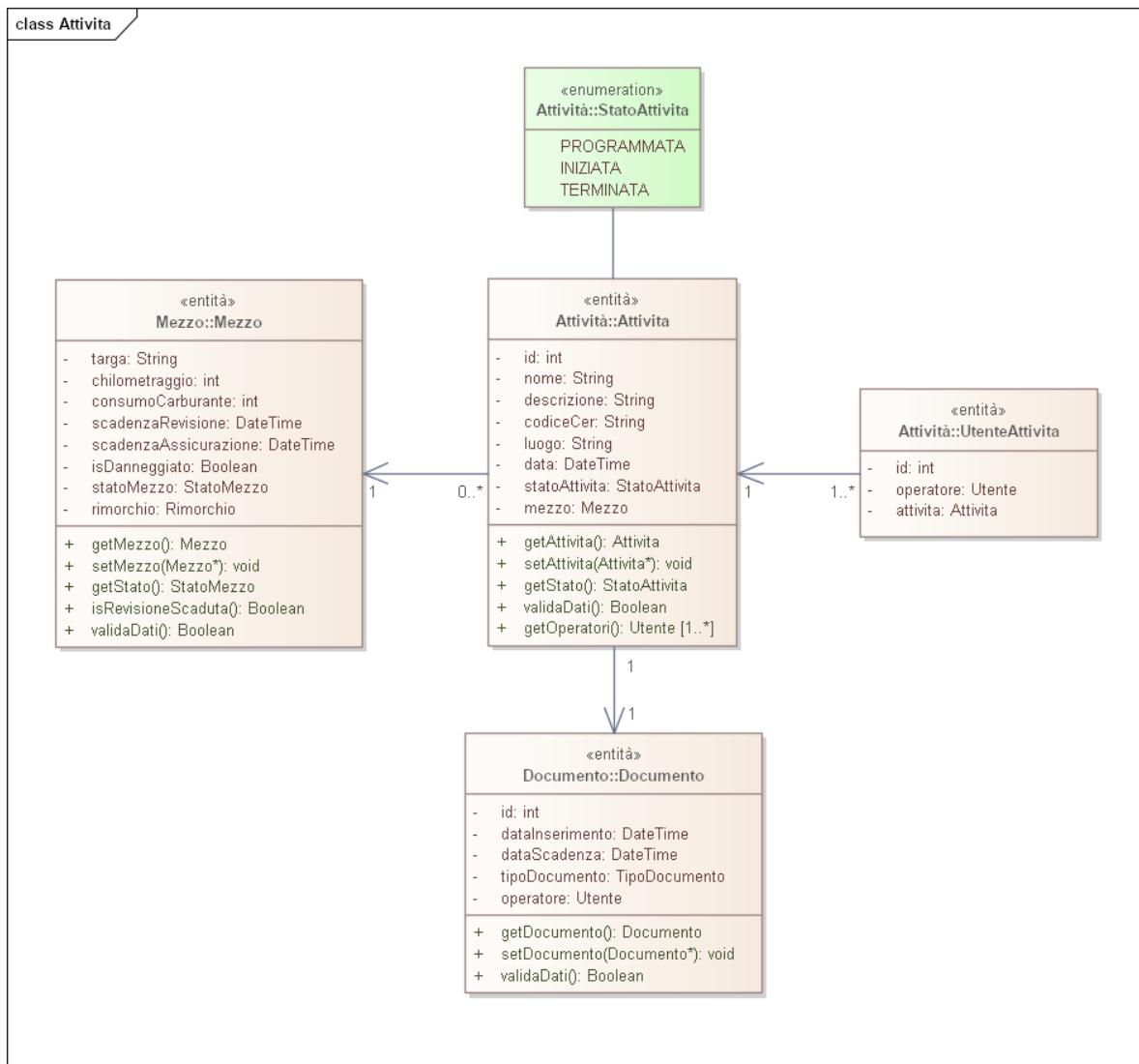
Classi di Progettazione: Mezzo

Il diagramma delle classi di Mezzo riprende il modello di analisi e lo raffina per la fase di progettazione. Le classi Mezzo e la classe associata Rimorchio sono state arricchite con i dettagli necessari per l'implementazione, come gli attributi specifici e i metodi di gestione. Le associazioni con l'enumerazione per lo stato del veicolo rendono il modello una base solida per lo sviluppo.



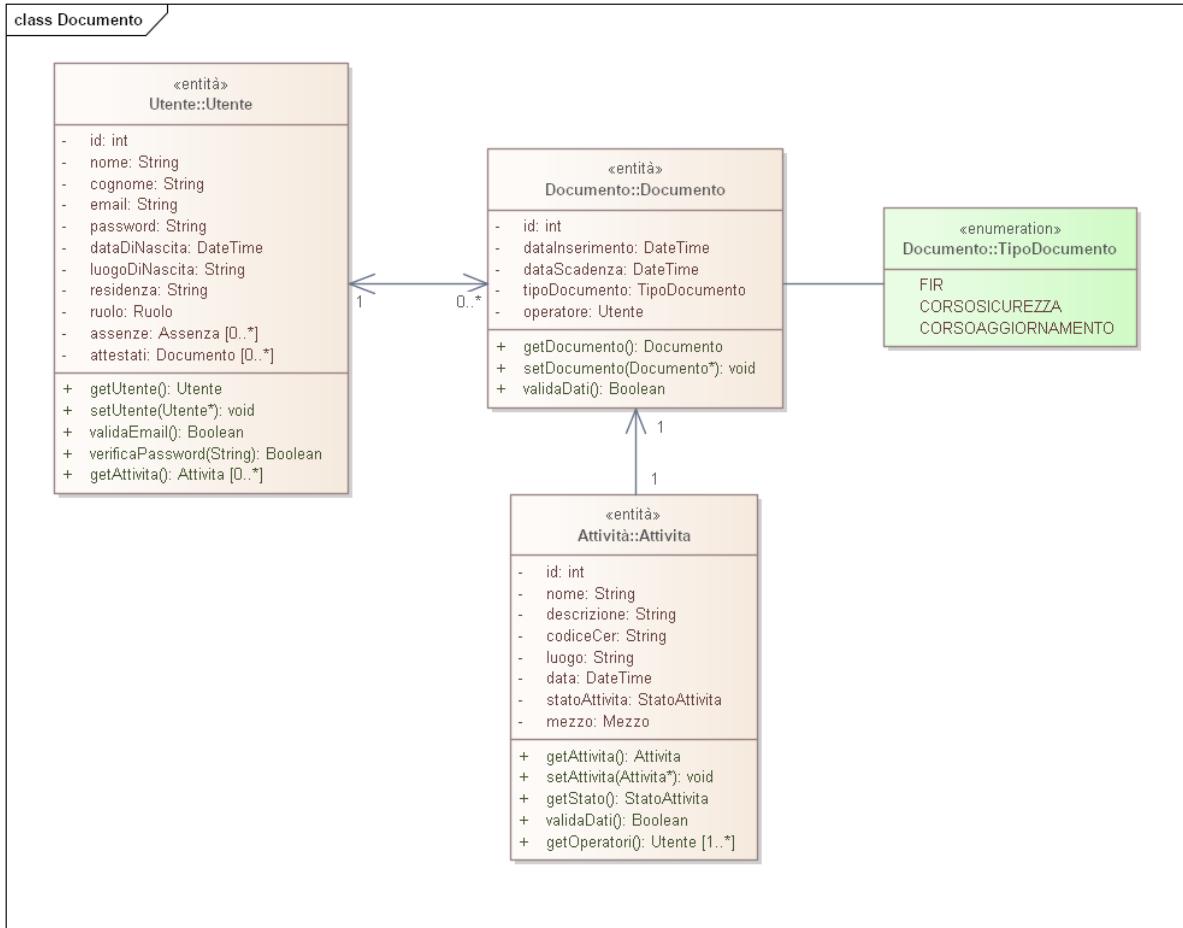
Classi di Progettazione: Attività

Il diagramma delle classi di Attività è stato perfezionato per la progettazione. La classe Attività è stata arricchita con dettagli specifici e metodi di gestione. L'introduzione della classe intermedia UtenteAttività risolve il legame molti a molti (many-to-many) tra Utente e Attività, rendendo il modello pronto per l'implementazione.



Classi di Progettazione: Documento

Il diagramma delle classi di Documento è un'evoluzione del modello di analisi. La classe Documento è stata arricchita con i dettagli di progettazione necessari, inclusi attributi e metodi per la gestione. Le associazioni con l'enumerazione per il tipo di documento rendono il modello una base solida per lo sviluppo.



Classi di Progettazione: Controllers

Questo diagramma delle classi si discosta dagli altri, concentrandosi sull'architettura del software piuttosto che sul modello dati. Mostra i controller del sistema (es. UtenteController, MezzoController, ecc.), che fungono da intermediari tra l'interfaccia utente e le entità. Le relazioni tra i controller e le rispettive entità (es. Utente, Mezzo, ecc.) sono indicate con una dipendenza (<use>), evidenziando che i controller utilizzano le entità per svolgere le loro funzioni. Questo modello illustra la separazione dei compiti a un livello più alto, cruciale per l'organizzazione del codice e la manutenzione.

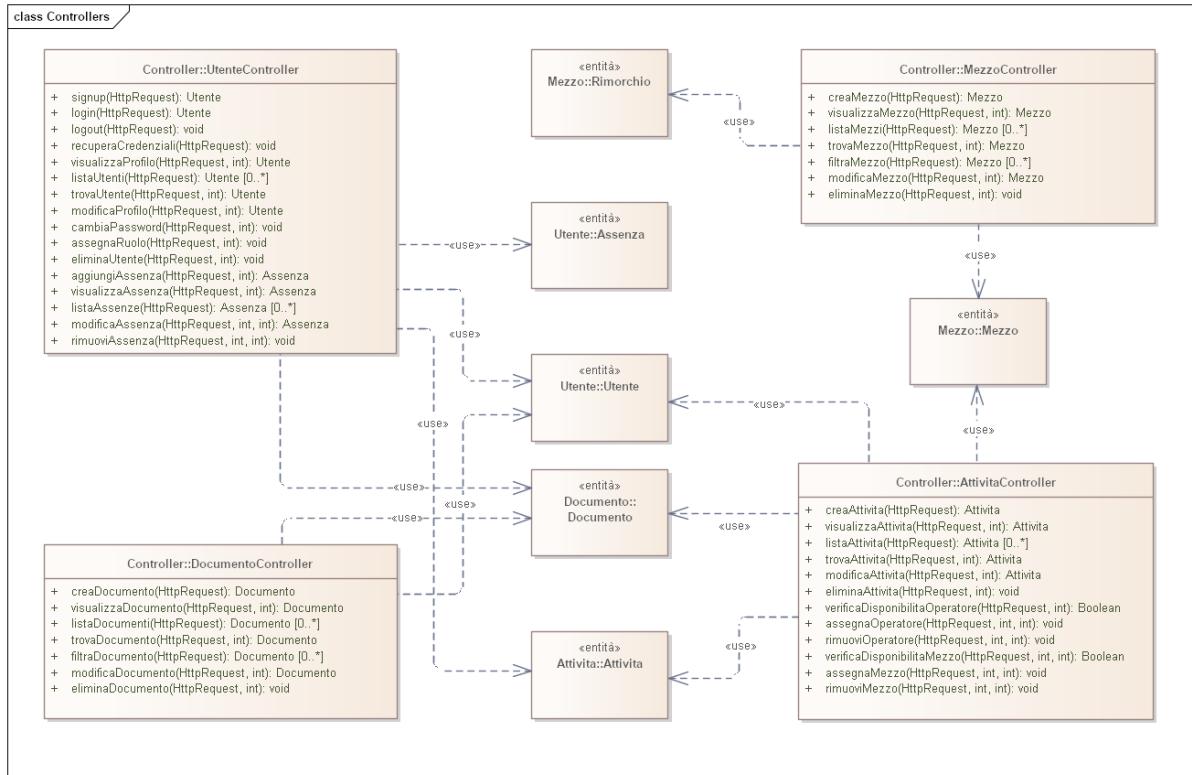
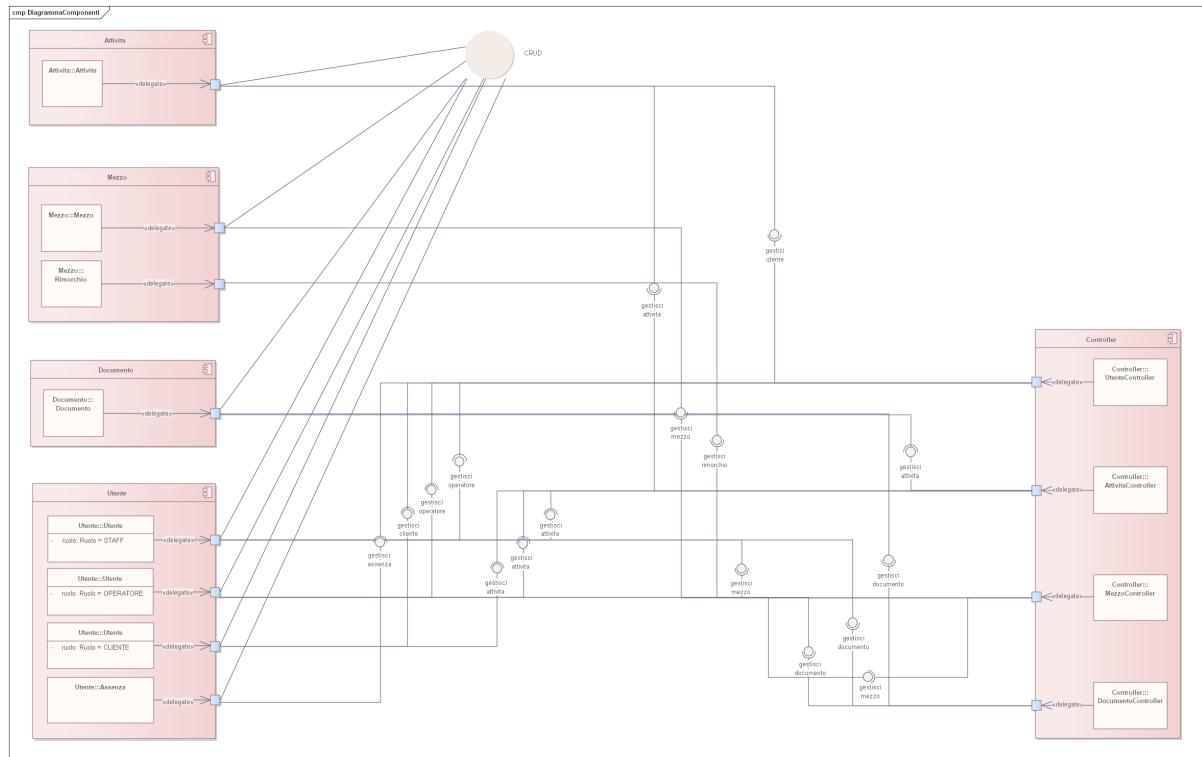


Diagramma dei Componenti

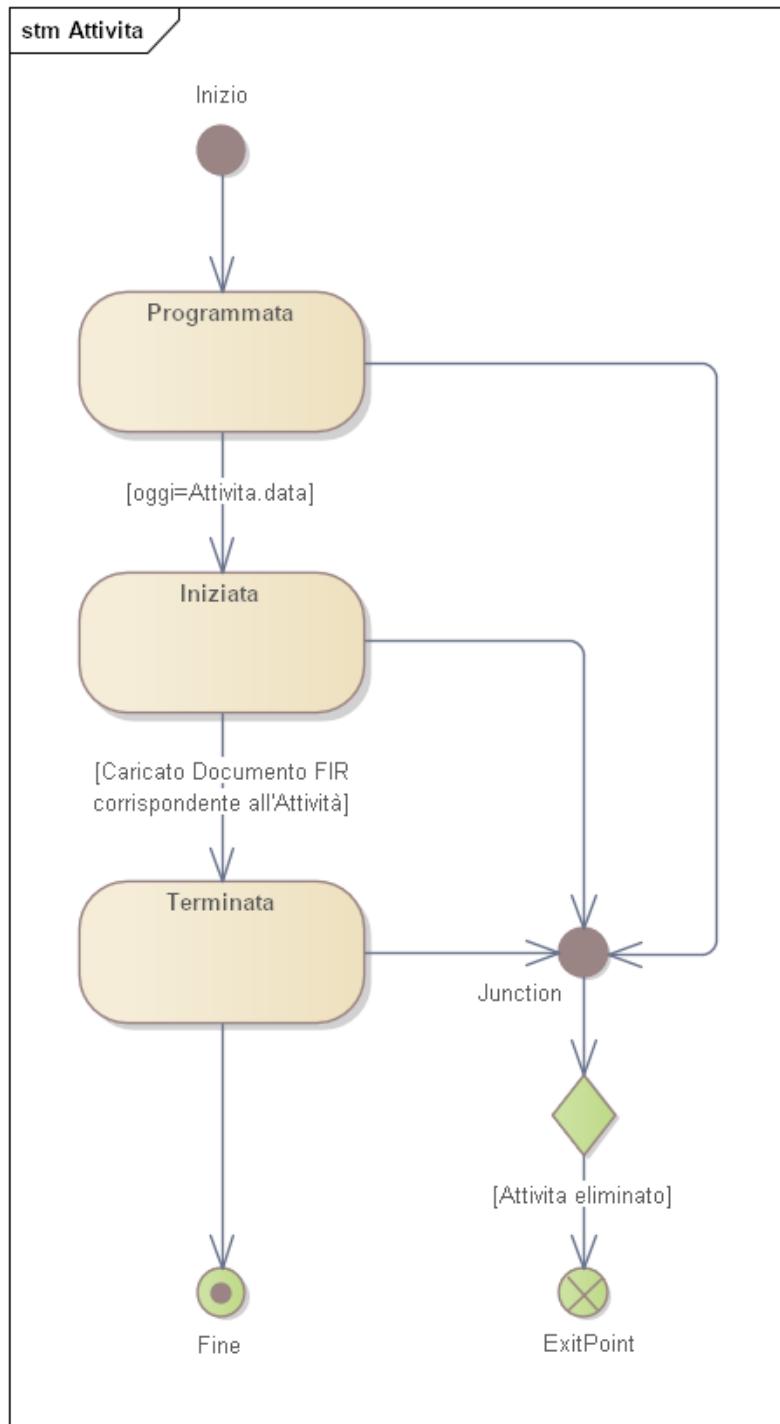
Questo diagramma illustra l'architettura di alto livello del sistema, focalizzandosi sulle dipendenze tra i suoi componenti. Si evidenzia che le entità di dominio, come Utente, Attività, Mezzo e Documento, non interagiscono direttamente tra loro, ma delegano la loro logica a un insieme di Controller. I controller (UtenteController, AttivitaController, MezzoController, DocumentoController) espongono delle interfacce che sono utilizzate per separare la logica applicativa dalla gestione dei dati.



Diagrammi delle Macchine a Stati

Macchine a Stati: Attività

Questo diagramma di stato modella il ciclo di vita di un'attività. L'attività inizia nello stato Programmata e passa allo stato Iniziata solo quando la sua data coincide con quella odierna. L'attività si sposta quindi nello stato Terminata solo dopo che il documento "FIR" (Foglio di Itinerario e Registro) corrispondente è stato caricato. Il diagramma mostra un flusso sequenziale ma, come in tutti i diagrammi di stato, un'uscita dal sistema può verificarsi in qualsiasi momento. Se un'attività viene eliminata, il suo stato si conclude, rappresentato nel diagramma dal simbolo Exit, che indica la terminazione del ciclo di vita dell'oggetto.



Macchine a Stati: Mezzo

Questo diagramma di stato modella il ciclo di vita completo di un mezzo. Il mezzo inizia nello stato Disponibile e passa allo stato Occupato quando viene assegnato a un'attività. Al termine dell'attività, il mezzo ritorna allo stato Disponibile, pronto per una nuova assegnazione. Se il mezzo si danneggia o la condizione scadenzaRevisione < oggi viene verificata come vera, il mezzo entra nello stato di Manutenzione, per poi tornare nuovamente Disponibile una volta che i lavori sono stati completati. In ogni momento, se il mezzo viene eliminato dal sistema, il suo stato si conclude, come rappresentato dal simbolo ExitPoint nel diagramma UML, che indica la terminazione del ciclo di vita dell'oggetto.

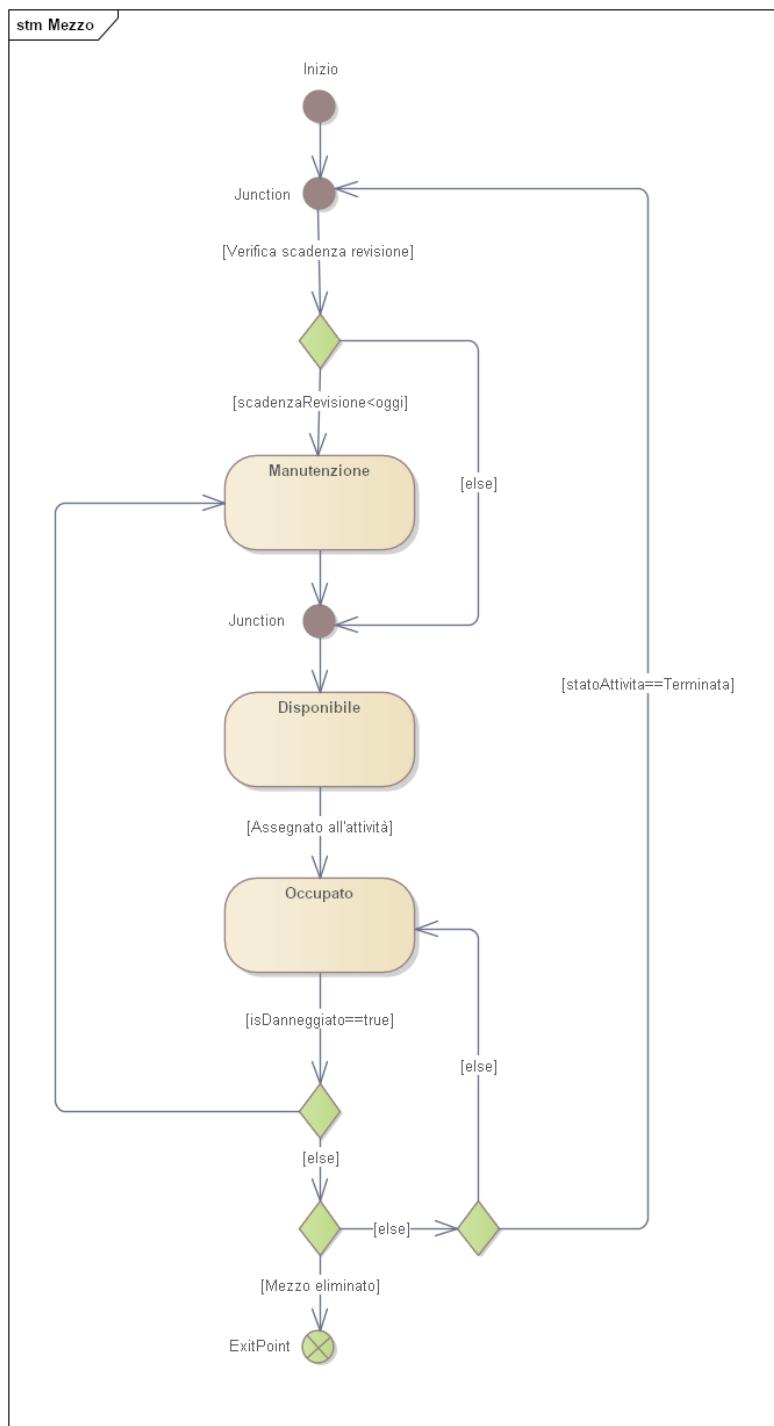
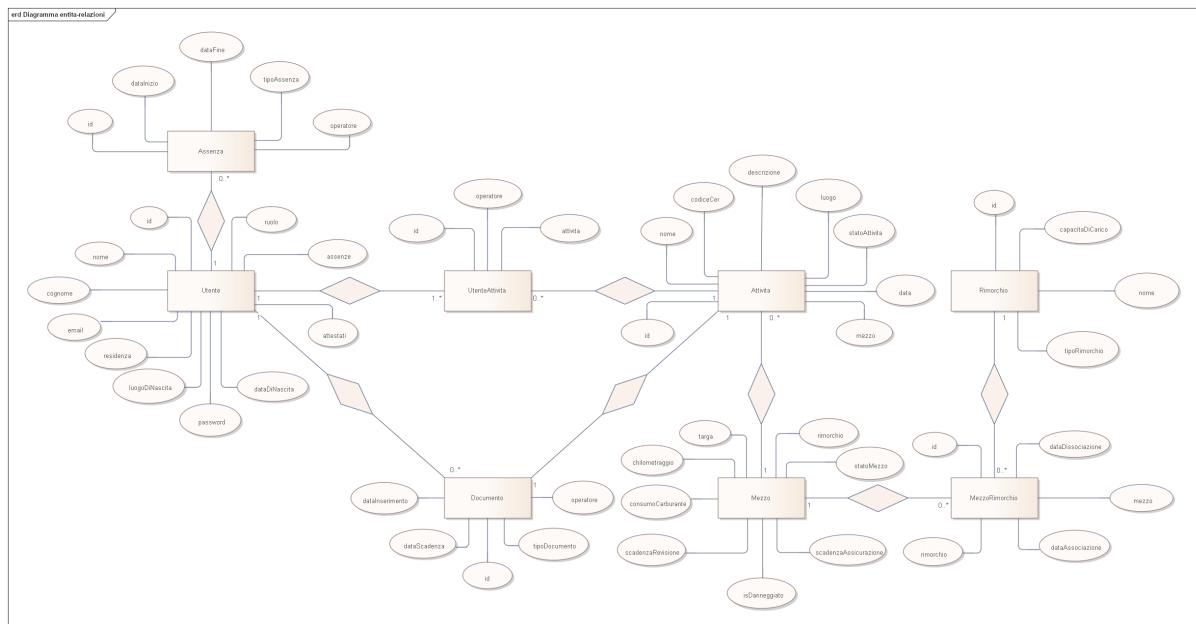


Diagramma Entità-Relazione

Il seguente diagramma ER espone la struttura del sistema gestionale, mostrando entità come utenti, assenze, documenti, mezzi e rimorchi, insieme ai loro attributi e collegamenti.

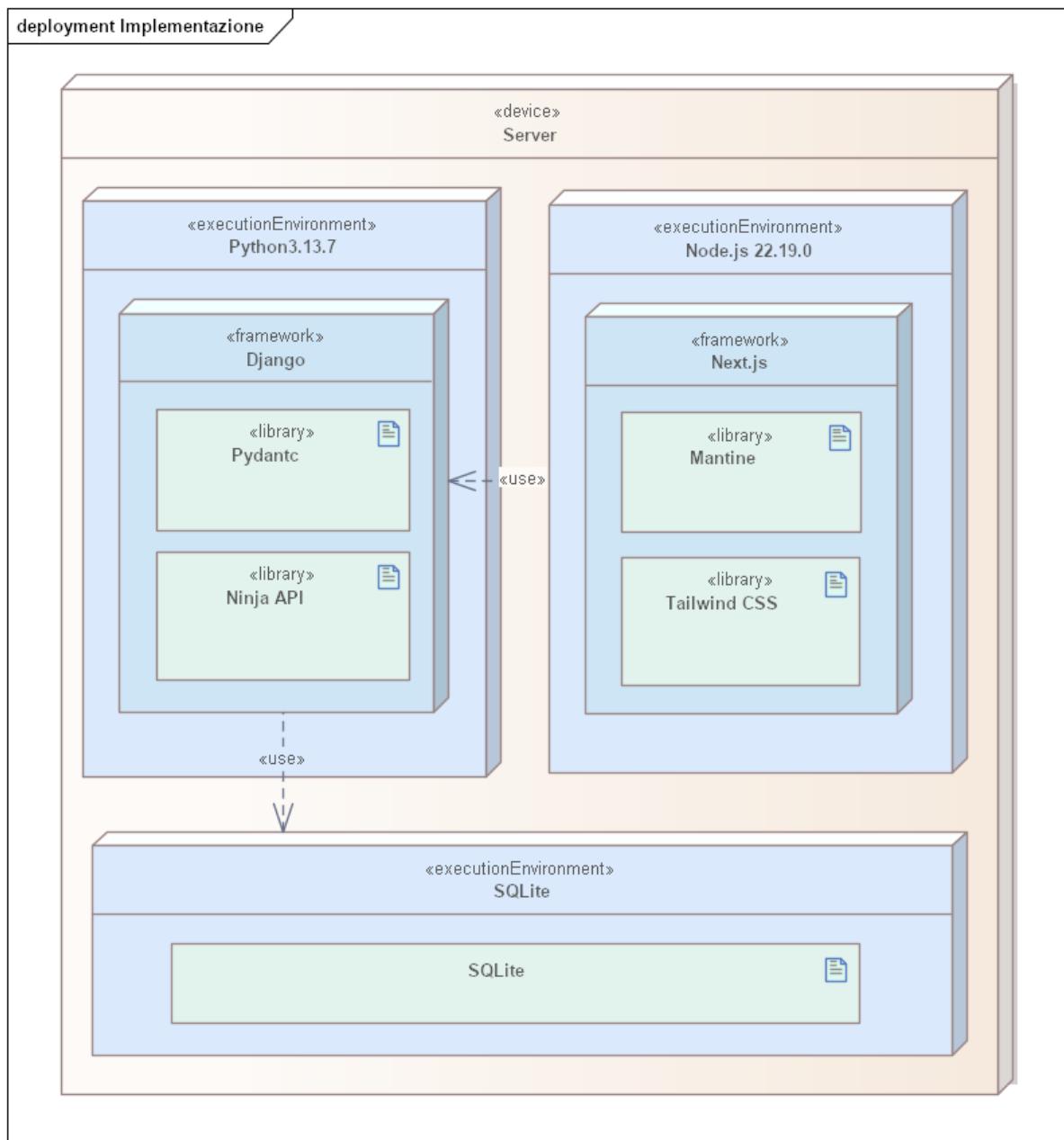


Implementazione

La seguente sezione è dedicata al flusso di lavoro dell'implementazione, nella quale sono stati inseriti: il diagramma di deployment, una descrizione delle tecnologie utilizzate, il mockup ed infine la descrizione unit test.

Diagramma di Deployment

Il diagramma rappresenta il deployment dell'applicazione, mostrando come le diverse componenti (frontend, backend e database) sono distribuite in container separati e come comunicano tra loro.



Tecnologie Utilizzate

Tecnologie Backend

Django

Django è un framework web di alto livello, sviluppato in Python, pensato per semplificare e velocizzare la realizzazione di applicazioni web. La scelta di questo utilizzo è dovuta ai suoi numerosi vantaggi, tra cui la rapidità di sviluppo, l'elevato livello di sicurezza e la ricchezza di funzionalità integrate che riducono la necessità di codice ridondante.

Django Ninja API

Django Ninja è un framework per la creazione di API basato su Django. Consente di sviluppare endpoint RESTful in maniera rapida, con un'ottima integrazione al modello dati e una gestione automatica della documentazione (OpenAPI/Swagger). La scelta di questo utilizzo è motivata dalla possibilità di ridurre il codice boilerplate garantendo al contempo elevate performance e la solidità dell'ecosistema Django.

Pydantic

Pydantic è una libreria Python per la validazione e la serializzazione dei dati tramite modelli tipizzati. È stata utilizzata in combinazione con Django Ninja per garantire la correttezza dei dati in ingresso e in uscita dalle API. La scelta di questo utilizzo è dovuta alla sua precisione, che permette di intercettare errori legati ai dati in modo semplice e veloce.

Tecnologie Frontend

Next.js

Next.js è un framework frontend basato su React che offre funzionalità avanzate come il rendering lato server (SSR) e la generazione statica delle pagine. La scelta di questo utilizzo è volta a garantire un'ottima ottimizzazione delle performance e una migliore indicizzazione SEO, oltre che per la sua flessibilità nello sviluppo di interfacce web moderne.

Mantine

Mantine è una libreria di componenti UI moderna e altamente personalizzabile, utilizzata all'interno del progetto con Next.js. La scelta di questo utilizzo è stata fatta per la sua ampia collezione di componenti pronti all'uso che ha semplificato la realizzazione di un'interfaccia utente coerente ed esteticamente gradevole, riducendo i tempi di sviluppo.

Tailwind CSS

Tailwind CSS è un framework CSS basato su classi utility, che consente di stilizzare rapidamente le interfacce in modo modulare e scalabile. La scelta di questo utilizzo è motivata dal fatto che, nel nostro progetto, è stato installato ed è utilizzato anche da Mantine per la personalizzazione grafica, permettendo un controllo dettagliato sul design senza sacrificare la produttività.

Database

SQLite

SQLite è un database relazionale leggero e serverless, integrato direttamente nell'applicazione senza necessità di un processo server separato. La scelta di questo utilizzo è motivata dalla sua capacità di garantire elevate performance e dalla semplicità di gestione, grazie all'assenza di configurazioni complesse. Il database è contenuto in un singolo file, facilitando la portabilità e la gestione dei dati. Inoltre, supporta transazioni ACID, assicurando integrità e affidabilità anche in caso di crash o interruzioni.

Tecnologie Version Control System

Git

Git è un sistema di controllo di versione che consente di monitorare le modifiche apportate ai file di progetto nel tempo. Grazie a Git, è stato possibile suddividere il carico di lavoro tra i membri del team, favorendo lo sviluppo simultaneo delle funzionalità del progetto.

Github

GitHub è una piattaforma di hosting per progetti software che utilizza il sistema di controllo di versione distribuito Git. È stato impiegato per facilitare lo sviluppo parallelo delle funzionalità del progetto, garantendo la sincronizzazione continua del codice e fornendo un backup sicuro e centralizzato.

Altre tecnologie utilizzate

Enterprise Architect

Enterprise Architect è uno strumento di modellazione e progettazione visiva basato su OMG UML. Nel progetto è stato utilizzato per modellare l'architettura del sito web e supportare l'implementazione dei modelli durante l'intero ciclo di vita dello sviluppo dell'applicazione.

Mockup

Il mockup dell'applicativo GESTRi è stato realizzato con Figma, uno strumento per la progettazione e la prototipazione di interfacce utente. Ogni mockup rappresenta la versione progettuale dell'interfaccia, definendo struttura, elementi grafici e funzionalità prima della fase di sviluppo.



Figura 1: Mockup login

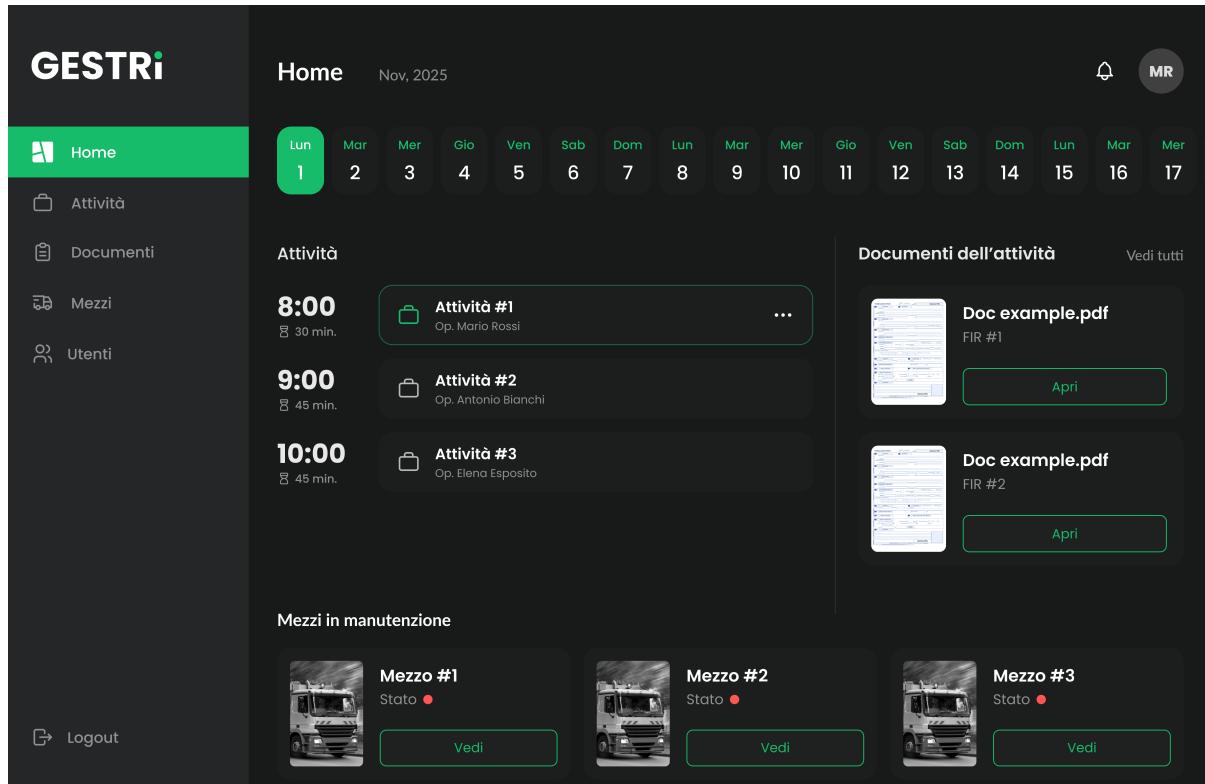


Figura 2: Mockup Home

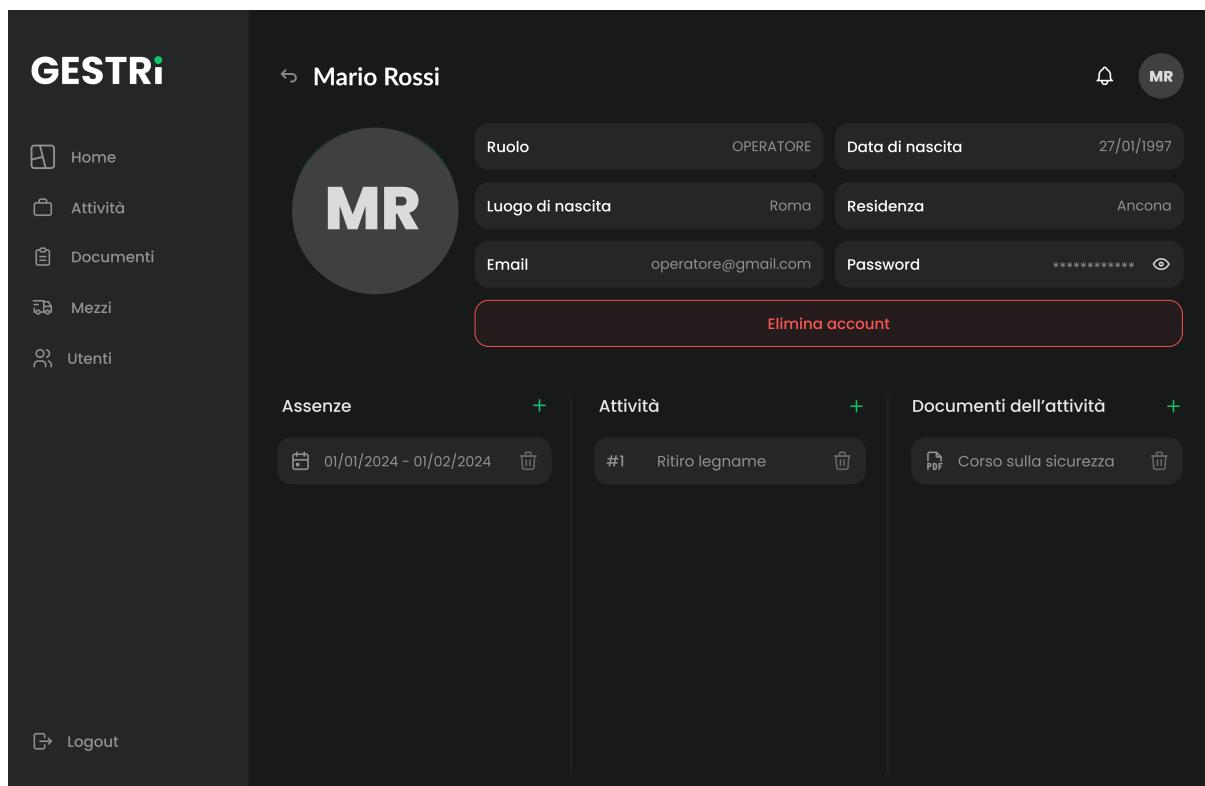


Figura 3: Mockup profilo

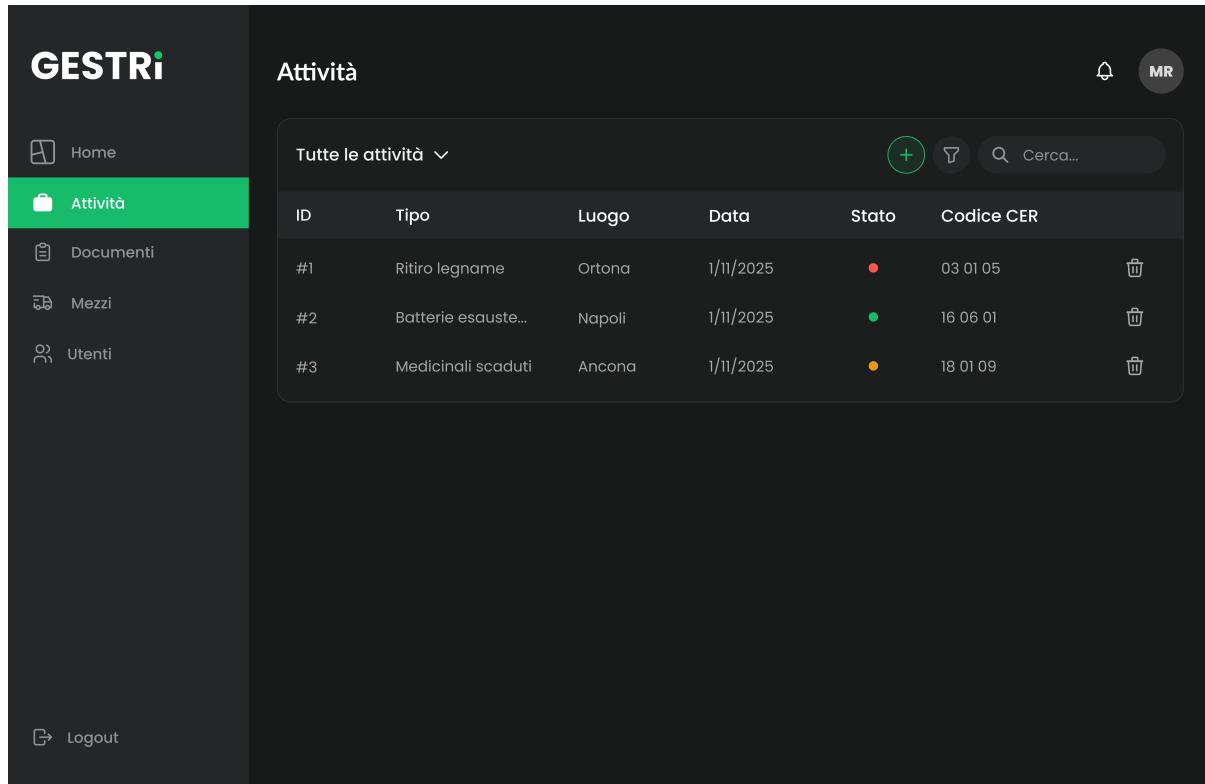


Figura 4: Mockup Attività

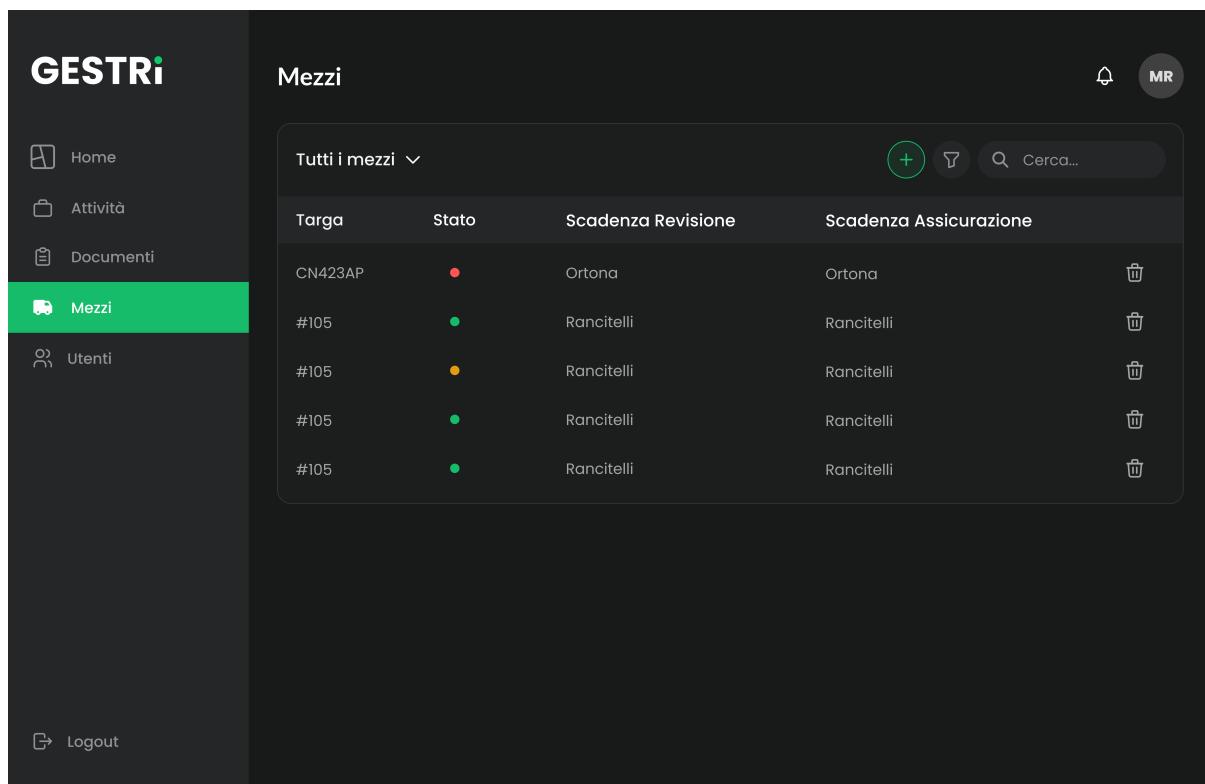


Figura 5: Mockup Mezzo

Verifica e Validazione del Software: Unit Tests

I seguenti test rappresentano l'insieme delle attività di verifica implementate per garantire il corretto funzionamento del progetto. Sono stati adottati due approcci complementari: test automatici lato codice e test manuali delle API.

Test Manuali

Per verificare le API è stata utilizzata l'applicazione Postman, uno strumento che consente di inviare richieste HTTP e analizzare le risposte. Questi test non rientrano nel diagramma degli Unit Test, ma rappresentano un'importante componente di validazione dell'integrazione e del comportamento delle API lato client-server.

Unit Test

La sezione presenta i test unitari eseguiti sul modulo Attività di GESTRi, con l'obiettivo di verificarne la correttezza funzionale e assicurare l'affidabilità del relativo codice.

```

1  class AttivitaIntegrationTests(TestCase):
2      def setUp(self):
3          self.client = Client()
4
5
6          # create users
7          self.staff = Utente.objects.create(email='staff@example.com', ruolo=
8              Ruolo.STAFF)
9          self.staff.set_password('pass')
10         self.staff.save()
11
12
13         self.operatore = Utente.objects.create(email='op@example.com', ruolo=
14             Ruolo.OPERATORE)
15         self.operatore.set_password('pass')
16         self.operatore.save()
17
18
19         self.cliente = Utente.objects.create(email='cliente@example.com', ruolo=
20             Ruolo.CLIENTE)
21         self.cliente.set_password('pass')
22         self.cliente.save()
23         # create activities
24         # att_assigned will be assigned to operator (for operatore visibility
25             test)
26         self.att_assigned = Attivita.objects.create(titolo='Assigned',
27             descrizione='DescA', utente_creatore=self.staff)
28         UtenteAttivita.objects.create(attivita=self.att_assigned, utente=self.
29             operatore)
30
31
32         # att_unassigned is owned by cliente and initially has no operator (for
33             association test)
34         self.att_unassigned = Attivita.objects.create(titolo='ClienteAtt',
35             descrizione='DescC', utente_creatore=self.cliente)
36
37
38         # att_other is another activity to ensure staff sees everything
39         self.att_other = Attivita.objects.create(titolo='Other', descrizione='
40             Desc2', utente_creatore=self.staff)
41
42
43     def _login_get_headers(self, email, password='pass'):
44         payload = json.dumps({'email': email, 'password': password})

```

```
36     resp = self.client.post('/api/utenti/login', data=payload, content_type='
37         application/json')
38     self.assertEqual(resp.status_code, 200)
39     tokens = resp.json()
40     access = tokens.get('access')
41     return {'HTTP_AUTHORIZATION': f'Bearer {access}'}
42
43 def _login_get_headers_and_token(self, email, password='pass'):
44     """Return both Django test client headers dict and raw access token
45     string."""
46     payload = json.dumps({'email': email, 'password': password})
47     resp = self.client.post('/api/utenti/login', data=payload, content_type='
48         application/json')
49     self.assertEqual(resp.status_code, 200)
50     tokens = resp.json()
51     access = tokens.get('access')
52     headers = {'HTTP_AUTHORIZATION': f'Bearer {access}'}
53     return headers, access
54
55
56 def test_staff_list_sees_all(self):
57     headers = self._login_get_headers('staff@example.com')
58     resp = self.client.get('/api/attivita/', **headers)
59     self.assertEqual(resp.status_code, 200)
60     data = resp.json()
61     ids = {a.get('id') for a in data}
62     self.assertIn(self.att_assigned.id, ids)
63     self.assertIn(self.att_unassigned.id, ids)
64     self.assertIn(self.att_other.id, ids)
65
66
67 def test_operatore_list_sees_assigned(self):
68     headers = self._login_get_headers('op@example.com')
69     resp = self.client.get('/api/attivita/', **headers)
70     self.assertEqual(resp.status_code, 200)
71     data = resp.json()
72     ids = {a.get('id') for a in data}
73     self.assertIn(self.att_assigned.id, ids)
74     self.assertNotIn(self.att_other.id, ids)
75
76
77 def test_cliente_create_and_detail(self):
78     headers = self._login_get_headers('cliente@example.com')
79     payload = json.dumps({
80         "titolo": "Nuova",
81         "descrizione": "desc",
82         "data": None
83     })
84     resp = self.client.post('/api/attivita/', data=payload, content_type='
85         application/json', **headers)
86     self.assertEqual(resp.status_code, 200)
87     data = resp.json()
88     self.assertIn('id', data)
89     new_id = data.get('id')
90
91
92     # detail
93     resp2 = self.client.get(f'/api/attivita/{new_id}', **headers)
94     self.assertEqual(resp2.status_code, 200)
95     d2 = resp2.json()
96     self.assertEqual(d2.get('id'), new_id)
```

```

95
96     def test_cliente_associate_operator(self):
97         headers = self._login_get_headers('cliente@example.com')
98         payload = json.dumps({"operatore_id": self.operatore.id})
99         resp = self.client.post(f'/api/attivita/{self.att_unassigned.id}/associa-
100             -operatore', data=payload, content_type='application/json', **headers
101         )
102         self.assertEqual(resp.status_code, 200)
103         data = resp.json()
104         self.assertIn('success', data)
105
106
107     def test_cliente_cannot_see_other_cliente_activity(self):
108         # create another cliente and activity
109         other_cliente = Utente.objects.create(email='other_cliente@example.com',
110             ruolo=Ruolo.CLIENTE)
111         other_cliente.set_password('pass')
112         other_cliente.save()
113
114
115         other_att = Attivita.objects.create(titolo='OtherClienteAtt',
116             descrizione='DescX', utente_creatore=other_cliente)
117
118         headers = self._login_get_headers('cliente@example.com')
119         resp = self.client.get('/api/attivita/', **headers)
120         self.assertEqual(resp.status_code, 200)
121         data = resp.json()
122         ids = {a.get('id') for a in data}
123         self.assertNotIn(other_att.id, ids)
124
125
126     def test_operatore_not_assigned_cannot_get_attivita_documento(self):
127         # create a document on att_unassigned (created by cliente) but operator
128             is not assigned
129         from documento.models import Documento
130         doc = Documento.objects.create(tipoDocumento='FIR', operatore=self.staff
131             )
132         self.att_unassigned.documenti.add(doc)
133         self.att_unassigned.save()
134         headers = self._login_get_headers('op@example.com')
135         resp = self.client.get(f'/api/attivita/{self.att_unassigned.id}/
136             documento', **headers)
137         # unauthorized access should return 403 with a 'detail' key
138         self.assertEqual(resp.status_code, 403)
139         self.assertIn('detail', resp.json())
140
141
142     def test_list_by_date_cliente_and_operatore_filters(self):
143         """Verify /api/attivita/by-date/{data} filters by role correctly.
144
145             - cliente sees only activities they created on that date
146             - operatore sees only activities they are assigned to on that date
147             """
148
149         from datetime import datetime, timedelta
150
151
152         # Create three activities for the same date
153         target_date = datetime.now()
154         date_str = target_date.strftime('%Y-%m-%d')

```

```

151     # activity created by cliente (self.cliente)
152     att_cliente = Attivita.objects.create(
153         titolo='C1', descrizione='desc', utente_creatore=self.cliente, data=
154             target_date
155     )
156
157     # activity created by staff
158     att_staff = Attivita.objects.create(
159         titolo='S1', descrizione='desc', utente_creatore=self.staff, data=
160             target_date
161     )
162
163     # activity created by staff but assigned to operatore
164     att_assigned_for_op = Attivita.objects.create(
165         titolo='OP1', descrizione='desc', utente_creatore=self.staff, data=
166             target_date
167     )
168     UtenteAttivita.objects.create(attivita=att_assigned_for_op, utente=self.
169         operatore)
170
171     # Cliente should only see their activity when requesting by date
172     headers_cliente = self._login_get_headers('cliente@example.com')
173     resp_c = self.client.get(f'/api/attivita/by-date/{date_str}', **
174         headers_cliente)
175     self.assertEqual(resp_c.status_code, 200)
176     data_c = resp_c.json()
177     ids_c = {a.get('id') for a in data_c}
178     self.assertIn(att_cliente.id, ids_c)
179     self.assertNotIn(att_staff.id, ids_c)
180     self.assertNotIn(att_assigned_for_op.id, ids_c)
181
182     # Operatore should only see the activity they are assigned to
183     headers_op = self._login_get_headers('op@example.com')
184     resp_o = self.client.get(f'/api/attivita/by-date/{date_str}', **
185         headers_op)
186     self.assertEqual(resp_o.status_code, 200)
187     data_o = resp_o.json()
188     ids_o = {a.get('id') for a in data_o}
189     self.assertIn(att_assigned_for_op.id, ids_o)
190     self.assertNotIn(att_staff.id, ids_o)
191     self.assertNotIn(att_cliente.id, ids_o)
192
193     def test_by_date_after_switching_users_no_cached_results(self):
194         """Simulate: cliente logs in and requests by-date, then another user
195             logs in and requests by-date.
196
197             This ensures the second user's response is not accidentally the same (
198                 cached) as the first user's.
199
200             from datetime import datetime
201
202             # choose a target date and create two activities: one by cliente, one by
203                 staff
204             target_date = datetime.now()
205             date_str = target_date.strftime('%Y-%m-%d')

```

```

205
206     att_cliente = Attivita.objects.create(
207         titolo='SeqC', descrizione='seq desc C', utente_creatore=self.
208             cliente, data=target_date
209     )
210
211     att_staff = Attivita.objects.create(
212         titolo='SeqS', descrizione='seq desc S', utente_creatore=self.staff,
213             data=target_date
214     )
215     # Step 1: cliente logs in and requests by-date
216     headers_cliente, token_cliente = self._login_get_headers_and_token('
217         cliente@example.com')
218     resp_c = self.client.get(f'/api/attivita/by-date/{date_str}', **
219         headers_cliente)
220     self.assertEqual(resp_c.status_code, 200)
221     data_c = resp_c.json()
222     ids_c = {a.get('id') for a in data_c}
223
224     # cliente should only see their activity
225     self.assertIn(att_cliente.id, ids_c)
226     self.assertNotIn(att_staff.id, ids_c)
227
228     # Step 2: simulate logout by simply obtaining fresh headers for another
229     # user (staff)
230     headers_staff, token_staff = self._login_get_headers_and_token('
231         staff@example.com')
232     resp_s = self.client.get(f'/api/attivita/by-date/{date_str}', **
233         headers_staff)
234     self.assertEqual(resp_s.status_code, 200)
235     data_s = resp_s.json()
236     ids_s = {a.get('id') for a in data_s}
237
238     # staff should see the staff-created activity (and typically all
239     # activities for that date)
240     self.assertIn(att_staff.id, ids_s)
241
242
243     # Ensure the staff view is not identical to the cliente view (no
244     # accidental caching)
245     self.assertNotEqual(ids_c, ids_s)
246
247     # Assert that the Authorization tokens used for cliente and staff are
248     # different
249     self.assertIsNotNone(token_cliente)
250     self.assertIsNotNone(token_staff)
251     self.assertNotEqual(token_cliente, token_staff)
252 }
```

Unit Test 1: Attivita

La sezione presenta i test unitari eseguiti sul modulo Utente di GESTRi, con l'obiettivo di verificarne la correttezza funzionale e assicurare l'affidabilità del relativo codice.

```
1
2
3 class UtenteIntegrationTests(TestCase):
4     def setUp(self):
5         self.client = Client()
6
7
8     def test_register_and_login(self):
9         # Register a new client user
10        payload = json.dumps({
11            "email": "newclient@example.com",
12            "password": "pass",
13            "nome": "New",
14            "cognome": "Client",
15            "ruolo": "CLIENTE"
16        })
17        resp = self.client.post('/api/utenti/register', data=payload, content_type='
18                                application/json')
19        # registration may return 200 or 201 or 400 depending on implementation
20        self.assertIn(resp.status_code, (200, 201, 400))
21
22
23        # Try login (if registration succeeded)
24        login_payload = json.dumps({"email": "newclient@example.com", "password": "
25                                     pass"})
26        resp2 = self.client.post('/api/utenti/login', data=login_payload,
27                               content_type='application/json')
28        # login will return 200 on success or 400 if registration didn't create
29        self.assertIn(resp2.status_code, (200, 400))
30
31 }
```

Unit Test 2: Utente