UNIVERSITÀ DI TRENTO

Life Manager

LifeManager

Corso di Ingegneria del software

Luca Boschiero, Mauro Meneghello, Nicola Turniano

23 aprile 2023

Gruppo T10
Deliverable 2

Mauro Meneghello	217564	mauro.meneghello@studenti.unitn.it
Luca Boschiero	217460	luca.boschiero@studenti.unitn.it
Nicola Turniano	271457	nicola.turniano@studenti.unitn.it

Parte I
Analisi dei componenti

Nel seguente capitolo analizzeremo i componenti interni che costituiranno il nostro sistema, andando a definire una prima architettura al software LifeManager, e le relazioni tra questi componenti.
In particolare, un componente rappresenta un'entità autonoma all'interno di un sistema o sottosistema; esso ha una o più interfacce (fornite o richieste), e gli elementi al suo interno sono nascosti ed inaccessibili eccetto che attraverso i mezzi forniti dalle sue interfacce.
Il risultato di questa analisi costituirà i vari *Component Diagrams* riportati nelle figure sottostanti; inoltre per ogni singolo componente sarà descritto lo scopo e le sue interfacce.
Raggrupperemo d'ora in poi nel termine Utente sia l'utente autenticato che l'utente non autenticato.

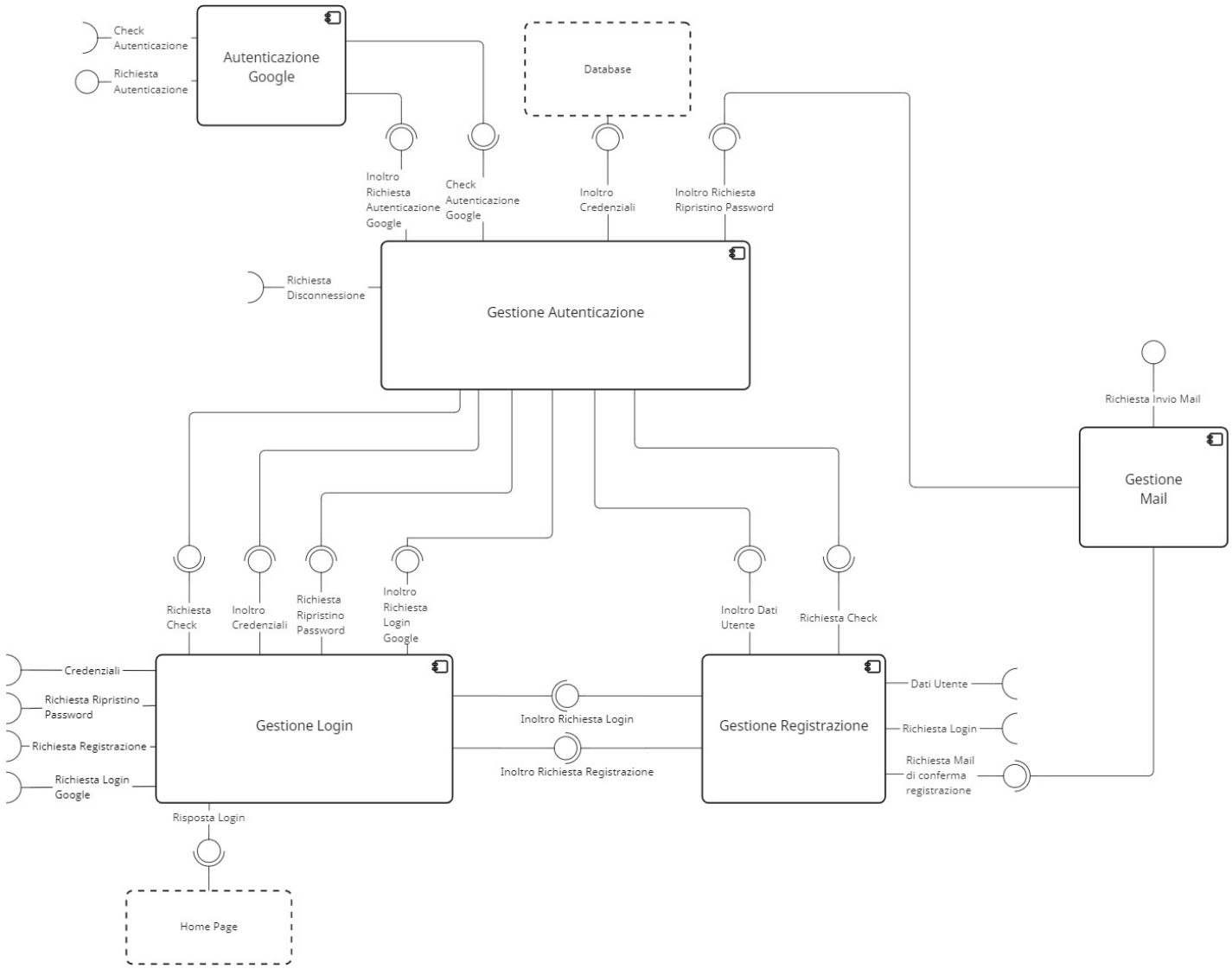

Figura 1: Component diagram Login, Registrazione, Autenticazione

1 - Gestione Login

Descrizione

Questo componente permette di interfacciarsi direttamente con l'utente che desidera effettuare il login con le proprie credenziali. In particolare, verranno raccolti i dati dell'utente (credenziali) o eventuali richieste (recupero password) e inoltrati ad altri componenti interni per elaborarli.

Interfacce richieste

- Credenziali: il componente riceve dall'utente username (o email) e password per accedere all'account.
- Richiesta check: il componente riceve dal componente 'Gestione autenticazione' l'esito del controllo della correttezza delle credenziali inserite dall'utente.
- Richiesta ripristino password: il componente riceve dall'utente la richiesta di ripristino della password.
- Richiesta login Google: il componente riceve dall'utente la richiesta di effettuare l'autenticazione con Google.
- Richiesta registrazione: il componente riceve dall'utente la richiesta di registrarsi e creare un nuovo account perchè non dispone ancora delle credenziali per effettuare il login.
- Inoltro richiesta login: il componente riceve la richiesta di effettuare il login dal componente 'Gestione registrazione'.

Interfacce fornite

- Risposta login: il componente restituisce l'esito dell'operazione di autenticazione, se positivo permetterà la visualizzazione della home page tramite il componente 'Home Page' .
- Inoltro richiesta registrazione: il componente fornisce al componente 'Gestione registrazione' la richiesta di effettuare una nuova registrazione .
- Inoltro credenziali: il componente fornisce al componente 'Gestione autenticazione' le credenziali ricevute dall'utente per effettuarne i controlli.
- Richiesta ripristino password: il componente fornisce al componente 'Gestione autenticazione' la richiesta dell'utente di ripristinare la password.
- Inoltro richiesta login Google: il componente dornisce al componente 'Gestione autenticazione' la richiesta dell'utente di effettuare il login con Google.

2 - Gestione registrazione

Descrizione

Questo componente permette di interfacciarsi con l'utente per permettere la registrazione al sistema e la creazione di un nuovo account. I dati raccolti verranno smistati agli altri componenti che li elaboreranno.

Interfacce richieste

- Dati utente: il componente, per permettere all'utente di registrarsi, richiede l'inserimento di nome, cognome, username, email e la password (ripetuta due volte).
- Richiesta registrazione: il componente riceve dal componente 'Gestione login' la richiesta dell'utente di effettuare la registrazione.
- Richiesta check: il componente riceve dal componente 'Gestione autenticazione' l'esito del controllo sui dati inseriti dall'utente.
- Richiesta login: il componente riceve dall'utente la richiesta di effettuare il login.

Interfacce fornite

- Richiesta mail conferma registrazione: il componente fornisce al componente 'Gestione mail' la richiesta di inviare una mail all'utente per confermare l'avvenuta registrazione.
- Inoltro richiesta login: il componente fornisce al componente 'Gestione login' la richiesta dell'utente di effettuare il login.
- Inoltro dati utente: il componente fornisce al componente 'Gestione autenticazione' i dati inseriti dall'utente affinche essi vengano verificati.

3 - Gestione autenticazione

Descrizione

Questo componente è il responsabile della verifica dei dati inviati (dati di login o registrazione) e quindi dell'autenticazione degli utenti al sistema, oltre che dell'aggiornamento del database in caso di nuove registrazioni e la disconnessione dell'utente quando richiesto.

Interfacce richieste

- Inoltro dati utente: il componente riceve dal componente 'Gestione registrazione' i dati inseriti dall'utente in fase di registrazione.
- Inoltro credenziali: il componente riceve dal componente 'Gestione login' i dati inseriti dall'utente in fase di login (le credenziali).
- Richiesta ripristino password: il componente riceve dal componente 'Gestione login' la richiesta dell'utente di recuperare la password.
- Richiesta login Google: il componente riceve dal componente 'Gestione login' la richiesta dell'utente di effettuare il login con google invece che con le credenziali.
- Check credenziali: il componente riceve dal componente 'Database' le informazioni per effettuare il controllo sulle credenziali inserite, in modo da permettere o negare l'autenticazione.
- Check autenticazione Google: il componente riceve dal componente 'Autenticazione Google' l'esito dell'autenticazione con Google.
- Richiesta disconnessione: il componente riceve dal componente 'Gestione impostazioni' la richiesta di disconnettere l'utente dal sistema.

Interfacce fornite

- Risposta check: il componente fornisce al componente 'Gestione registrazione' l'esito del check sui dati inseriti dall'utente.
- Richiesta check: il componente fornisce al componente 'Gestione login' l'esito del check sulle credenziali inserite dall'utente..
- Inoltro richiesta ripristino password: il componente fornisce al componente 'Gestione mail' la richiesta di inviare una mail all'utente per permettergli di recuperare la password.
- Inoltro richiesta autenticazione Google: il componente fornisce al componente 'Autenticazione Google' la richiesta di effettuare il login con Google.
- Inoltro credenziali: il componente fornisce al componente 'Database' i dati inseriti dall'utente in fase di registrazione e le sue credenziali affinchè vengano salvate nel database.

4 - Autenticazione Google

Descrizione

Questo componente interfaccia il nostro sistema con il sistema di auteticazione di Google e permette all'utente di accedere con Google invece che con le credenziali. In particolare, gestisce le richieste di autenticazione con Google e gli esiti di tali richieste.

Interfacce richieste

- Inoltro richiesta autenticazione Google: il componente riceve dal componente 'Gestione autenticazione' la richiesta di autenticare l'utente tramite Google.
- Check autenticazione: il componente richiede l'esito dell'autenticazione ai servizi di Google.

Interfacce fornite

- Richiesta autenticazione: il componente invia la richiesta di autenticazione ai servizi di Google, che si occuperanno di autenticare l'utente.
- Check autenticazione Google: il componente fornisce l'esito dell'autententicazione arrivatogli da Google al componente 'Gestione autenticazione'.

5 - Gestione mail

Descrizione

Questo componente interfaccia il nostro sistema con Gmail e permette di inviare delle notifiche (email) all'utente per permettergli di recuperare la password, confermare la registrazione e notificare l'approssimarsi di un evento.

Interfacce richieste

- Inoltro richiesta ripristino password: il componente riceve dal componente 'Gestione autenticazione' la richiesta di inviare una mail all'utente per permettergli di ripristinare la propria password.
- Richiesta mail conferma registrazione: il componente riceve dal componente 'Gestione login' la richiesta di inviare una mail all'utente per confermare la registrazione.
- Richiesta invio notifica: il componente riceve dal componente 'Gestione eventi' la richiesta di ricevere una notifica per segnalare l'approssimarsi di un proprio evento salvato.

Interfacce fornite

- Richiesta invio mail: il componente invia la richiesta a Gmail, che si occuperanno di inviare la mail all'utente.

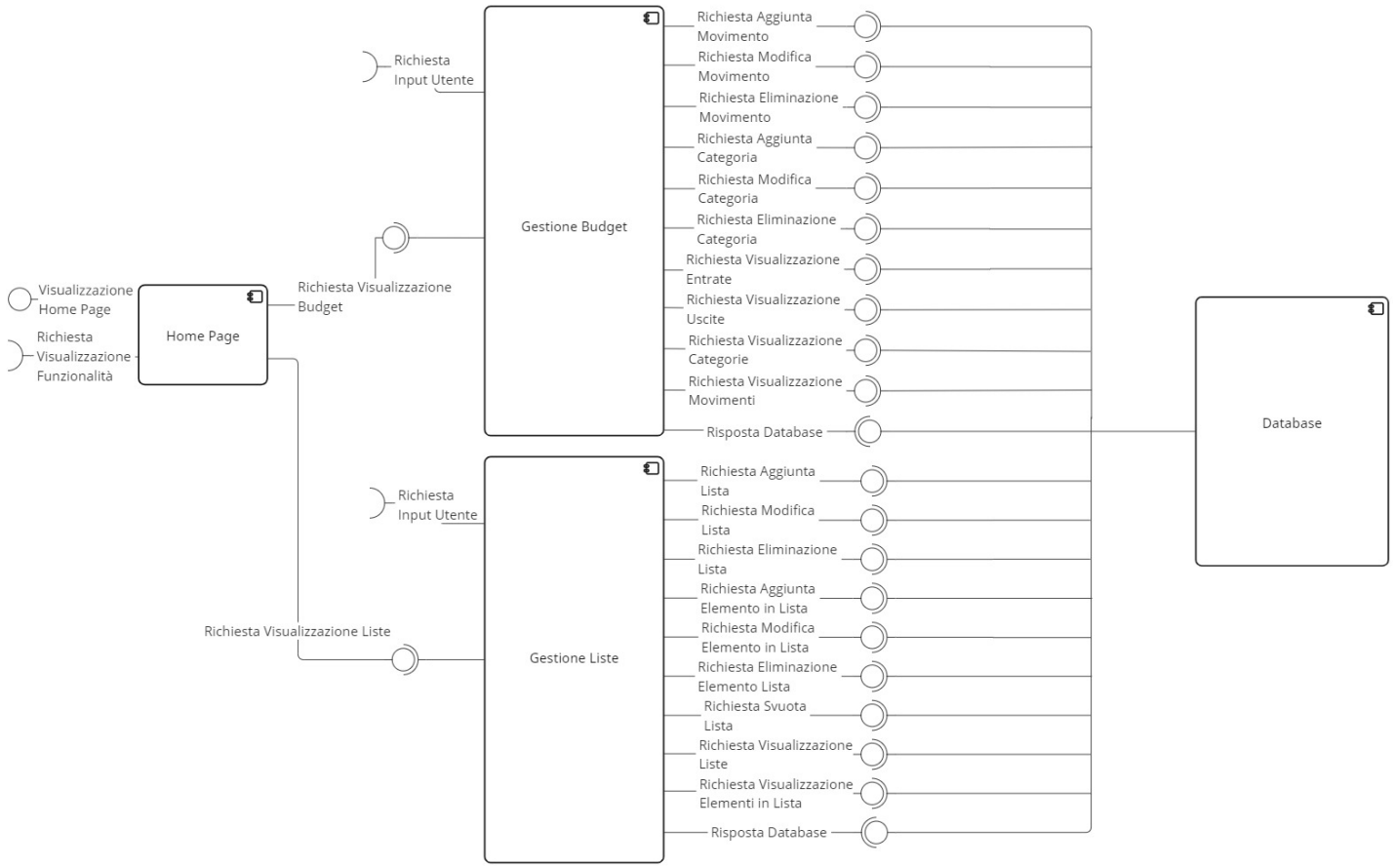

Figura 2: Component diagram Home Page, Budget, Liste

6 - Home page

Descrizione

Questo componente interfaccia il nostro sistema con l'utente permettendogli, una volta effettuato il login, di visualizzare e accedere alle varie funzionalità dell'applicazione.

Interfacce richieste

- Risposta login: il componente richiede al componente 'Gestione login' l'esito dell'autenticazione dell'utente.
- Richiesta visualizzazione funzionalità: il componente riceve dall'utente la richiesta di accedere a una determinata funzionalità del sistema e di visualizzarne (o eventualmente poi andare a modificarne) le informazioni. Per semplicità e chiarezza questa interfaccia è unica per tutte le funzionalità, altrimenti la versione più completa (ma più confusionaria e meno leggibile) prevederebbe 8 interfacce, una per ogni funzionalità (budget, liste, eventi, mappa, ricette, carte fedeltà, impostazioni e profilo).

Interfacce fornite

- Visualizzazione home page: il componente permette all'utente di visualizzare l'home page con tutte le funzionalità.
- Richiesta visualizzazione budget: il componente fornisce al componente 'Gestione budget' la richiesta dell'utente di accedere alla funzionalità budget e visualizzarne le informazioni.

- Richiesta visualizzazione liste: il componente fornisce al componente 'Gestione liste' la richiesta dell'utente di accedere alla funzionalità liste e visualizzarne le informazioni.
- Richiesta visualizzazione eventi: il componente fornisce al componente 'Gestione eventi' la richiesta dell'utente di accedere alla funzionalità eventi e visualizzarne le informazioni.
- Richiesta visualizzazione ricette: il componente fornisce al componente 'Gestione ricette' la richiesta dell'utente di accedere alla funzionalità ricette e visualizzarne le informazioni.
- Richiesta visualizzazione mappa: il componente fornisce al componente 'Gestione mappa' la richiesta dell'utente di accedere alla funzionalità mappa e visualizzarne le informazioni.
- Richiesta visualizzazione carte fedeltà: il componente fornisce al componente 'Gestione carte fedeltà' la richiesta dell'utente di accedere alla funzionalità carte fedeltà e visualizzarne le informazioni.
- Richiesta visualizzazione profilo: il componente fornisce al componente 'Gestione profilo' la richiesta dell'utente di accedere alla funzionalità relativa al profilo e visualizzarne le informazioni.
- Richiesta visualizzazione impostazioni: il componente fornisce al componente 'Gestione impostazioni' la richiesta dell'utente di accedere alla funzionalità relativa alle impostazioni e visualizzarne le informazioni.

7 - Gestione budget

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità budget e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione del proprio budget e dei propri movimenti.

Interfacce richieste

- Richiesta visualizzazione budget: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità budget e di visualizzare il proprio budget e i propri movimenti (sia complessivamente che classificati per categoria, oppure visualizzare solo le entrate o le uscite).
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare i propri movimenti. Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi ai movimenti che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano il budget e i movimenti (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta aggiunta movimento: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire un nuovo movimento, insieme ai dati inseriti dall'utente legati al nuovo movimento, affichè vengano scritti sul database creando tale nuovo movimento.
- Richiesta modifica movimento: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare un movimento esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale movimento.
- Richiesta eliminazione movimento: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare un movimento esistente.
- Richiesta aggiunta categoria: il componente fornisce al componente 'Database' la richiesta dell'utente di aggiungere una nuova categoria di movimento, insieme ai dati inseriti dall'utente relativi a tale categoria.

- Richiesta modifica categoria: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare una categoria di movimento esistente, insieme ai dati inseriti dall'utente relativi a tale categoria.
- Richiesta eliminazione categoria: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare una categoria di movimento esistente.
- Richiesta visualizzazione movimenti: il componente fornisce al componente 'Database' una richiesta per ottenere i dati relativi al budget (tutti i movimenti).
- Richiesta visualizzazione entrate: il componente fornisce al componente 'Database' una richiesta per ottenere i dati relativi ai soli movimenti in entrata.
- Richiesta visualizzazione uscite: il componente fornisce al componente 'Database' una richiesta per ottenere i dati relativi ai soli movimenti in uscita.
- Richiesta visualizzazione categoria: il componente fornisce al componente 'Database' una richiesta per ottenere i dati relativi alla categoria di movimento richiesta dall'utente.

8 - Gestione liste

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità liste e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione delle proprie liste di interesse.

Interfacce richieste

- Richiesta visualizzazione liste: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità liste e di visualizzarne le informazioni.
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare le proprie liste (e gli elementi all'interno delle liste). Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi a liste o elementi che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano le liste (in base a quanto richiesto al database tramite le interfacce fornite).
- Richiesta aggiunta ingredienti a lista spesa: il componente riceve dal componente 'Gestione ricette' la richiesta di aggiungere elementi (che corrispondono agli ingredienti della ricetta) alla lista della spesa.

Interfacce fornite

- Richiesta aggiunta lista: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire una nuova lista, insieme ai dati inseriti dall'utente legati alla nuova lista, affichè vengano scritti sul database creando tale nuova lista.
- Richiesta modifica lista: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare una lista esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale lista.
- Richiesta eliminazione lista: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare una lista esistente.
- Richiesta aggiunta elemento in lista: il componente fornisce al componente 'Database' la richiesta dell'utente di aggiungere un nuovo elemento in una lista, insieme ai dati inseriti dall'utente relativi a tale elemento.

- Richiesta modifica elemento in lista: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare un elemento esistente di una lista, insieme ai dati inseriti dall'utente relativi a tale elemento.
- Richiesta eliminazione elemento in lista: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare un elemento esistente di una lista.
- Richiesta visualizzazione liste: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare l'elenco delle liste esistenti.
- Richiesta visualizzazione elementi in lista: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare gli elementi di una specifica lista.
- Richiesta svuota lista: il componente fornisce al componente 'Database' una richiesta per eliminare gli elementi della lista di cui non ha più interesse.

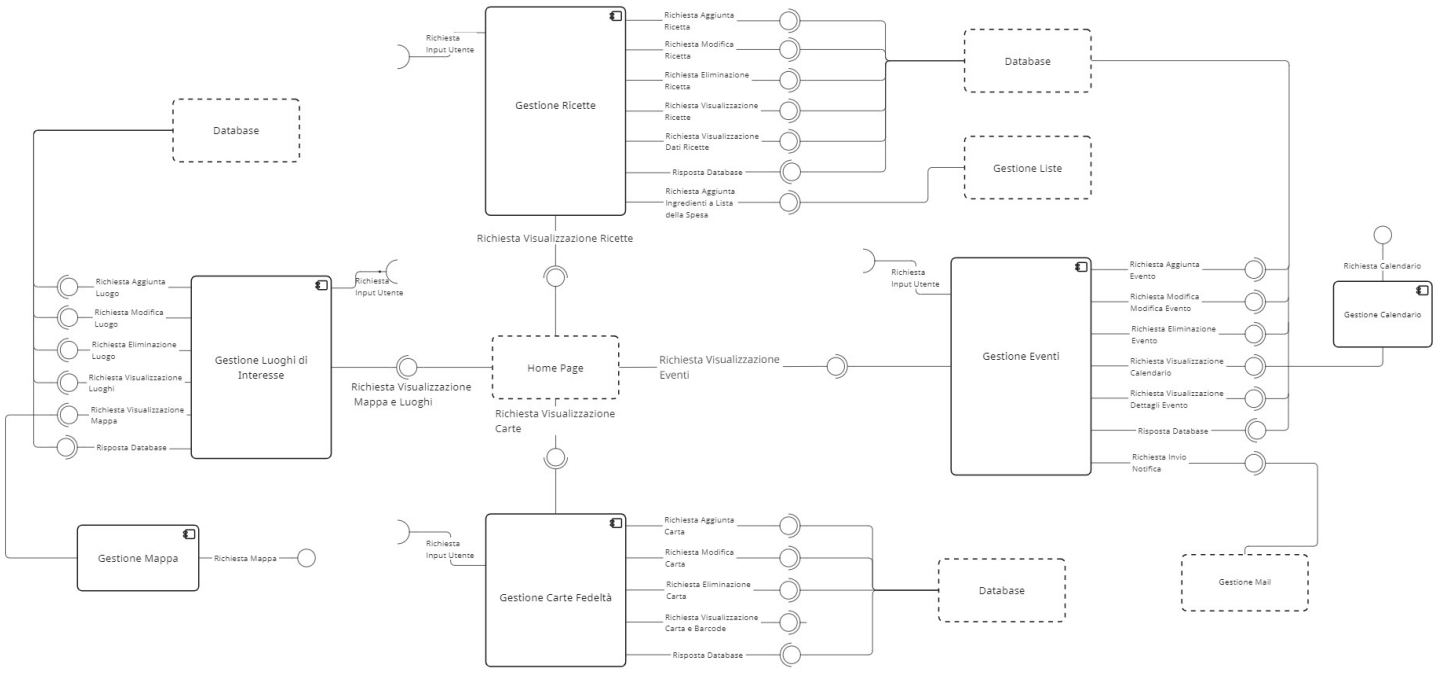

Figura 3: Component diagram Ricette, Luoghi di Interesse, Eventi, Carte Fedeltà

9 - Gestione ricette

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità ricette e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione delle proprie ricette.

Interfacce richieste

- Richiesta visualizzazione ricette: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità ricette e di visualizzarne le informazioni.
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare le proprie ricette. Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi alla ricetta che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano le ricette (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta aggiunta ricetta: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire una nuova ricetta, insieme ai dati inseriti dall'utente legati alla nuova ricetta, affichè vengano scritti sul database creando tale nuova ricetta.
- Richiesta modifica ricetta: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare una ricetta esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale ricetta.
- Richiesta eliminazione ricetta: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare una ricetta esistente.
- Richiesta aggiunta ingredienti a lista spesa: il componente fornisce al componente 'Database' una richiesta per inserire gli ingredienti che compongono la ricetta direttamente nella lista della spesa.

- Richiesta visualizzazione ricette: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare l'elenco delle ricette esistenti.
- Richiesta visualizzazione dati ricetta: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare i dati relativi ad una specifica ricetta.
- Richiesta aggiunta ingredienti a lista spesa:il componente fornisce al componente 'Gestione liste' la richiesta di inserire tutti o alcuni ingredienti alla lista della spesa.

10 - Gestione eventi

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità eventi e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione dei propri eventi/attività.

Interfacce richieste

- Richiesta visualizzazione eventi: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità eventi e di visualizzarne le informazioni.
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare i propri eventi. Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi agli eventi che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano gli eventi (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta aggiunta evento: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire un nuovo evento, insieme ai dati inseriti dall'utente legati al nuovo evento, affichè vengano scritti sul database creando tale nuovo evento.
- Richiesta modifica evento: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare un evento esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale evento.
- Richiesta eliminazione evento: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare un evento esistente.
- Richiesta visualizzazione calendario: il componente fornisce al componente 'Gestione calendario' una richiesta visualizzare il calendario con segnati gli eventi salvati dall'utente.
- Richiesta visualizzazione dettagli evento: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare i dati relativi ad un specifico evento.
- Richiesta invio notifica: il componente fornisce al componente 'Gestione mail' la richiesta di inviare una email all'utente per segnalare l'approssimarsi di un evento/attività di cui esso vuole essere ricordato.

11 - Gestione mappe/luoghi di interesse

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità mappe/luoghi di interesse e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione dei propri luoghi di interesse.

Interfacce richieste

- Richiesta visualizzazione mappa e luoghi: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità mappa e di visualizzare le informazioni relative ai luoghi di interesse.
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare i propri luoghi di interesse. Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi ai luoghi che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano i luoghi di interesse (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta aggiunta luogo: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire un nuovo luogo di interesse, insieme ai dati inseriti dall'utente legati al nuovo luogo, affichè vengano scritti sul database creando tale nuovo luogo.
- Richiesta modifica luogo: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare un luogo di interesse esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale luogo.
- Richiesta eliminazione luogo: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare un luogo di interesse esistente.
- Richiesta visualizzazione mappa: il componente fornisce al componente 'Gestione mappa' una richiesta visualizzare la mappa con segnati i luoghi di interesse salvati dall'utente.
- Richiesta visualizzazione dettagli luogo: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare i dati relativi ad un specifico luogo.

12 - Gestione carte fedeltà

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità carte fedeltà e in base alle richieste dell'utente permette la visualizzazione, gestione, aggiunta e rimozione delle proprie carte fedeltà.

Interfacce richieste

- Richiesta visualizzazione carte: il componente riceve dal componente 'Home page' la richiesta di accedere alla funzionalità carte fedeltà e di visualizzare le informazioni relative alle carte.
- Richiesta input utente: il componente riceve dall'utente la richiesta di aggiungere, rimuovere o modificare le proprie carte fedeltà. Inoltre questa interfaccia si occupa di richiedere all'utente di inserire i dati relativi alla carta fedeltà che intende aggiungere o modificare).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano i luoghi di interesse (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta aggiunta carta: il componente fornisce al componente 'Database' la richiesta dell'utente di inserire una nuova carta fedeltà, insieme ai dati inseriti dall'utente legati alla nuova carta, affichè vengano scritti sul database creando tale carta.

- Richiesta modifica carta: il componente fornisce al componente 'Database' la richiesta dell'utente di modificare una carta fedeltà esistente, insieme ai dati inseriti dall'utente legati a tale modifica, affichè vengano scritti sul database modificando tale carta.
- Richiesta eliminazione carta: il componente fornisce al componente 'Database' la richiesta dell'utente di eliminare una carta fedeltà esistente.
- Richiesta visualizzazione carta e barcode: il componente fornisce al componente 'Database' una richiesta per ottenere e visualizzare i dati relativi ad una specifica carta (compreso il barcode generato a partire dal numero della carta).

13 - Gestione mappa

Descrizione

Questo componente interfaccia il nostro sistema con OpenStreetMap per permettere la visualizzazione della mappa su cui segnalare i luoghi di interesse salvati dall'utente.

Interfacce richieste

- Richiesta visualizzazione mappa: il componente riceve dal componente 'Gestione luoghi di interesse' la richiesta visualizzare la mappa.

Interfacce fornite

- Richiesta mappa: il componente fornisce la richiesta a OpenStreetMap, che si occuperà di visualizzare la mappa.

14 - Gestione calendario

Descrizione

Questo componente interfaccia il nostro sistema con Google Calendar per permettere la visualizzazione del calendario con gli eventi salvati dall'utente.

Interfacce richieste

- Richiesta visualizzazione calendario: il componente riceve dal componente 'Gestione eventi' la richiesta visualizzare il calendario.

Interfacce fornite

- Richiesta mappa: il componente fornisce la richiesta a Google Calendar, che si occuperà di visualizzare il calendario.

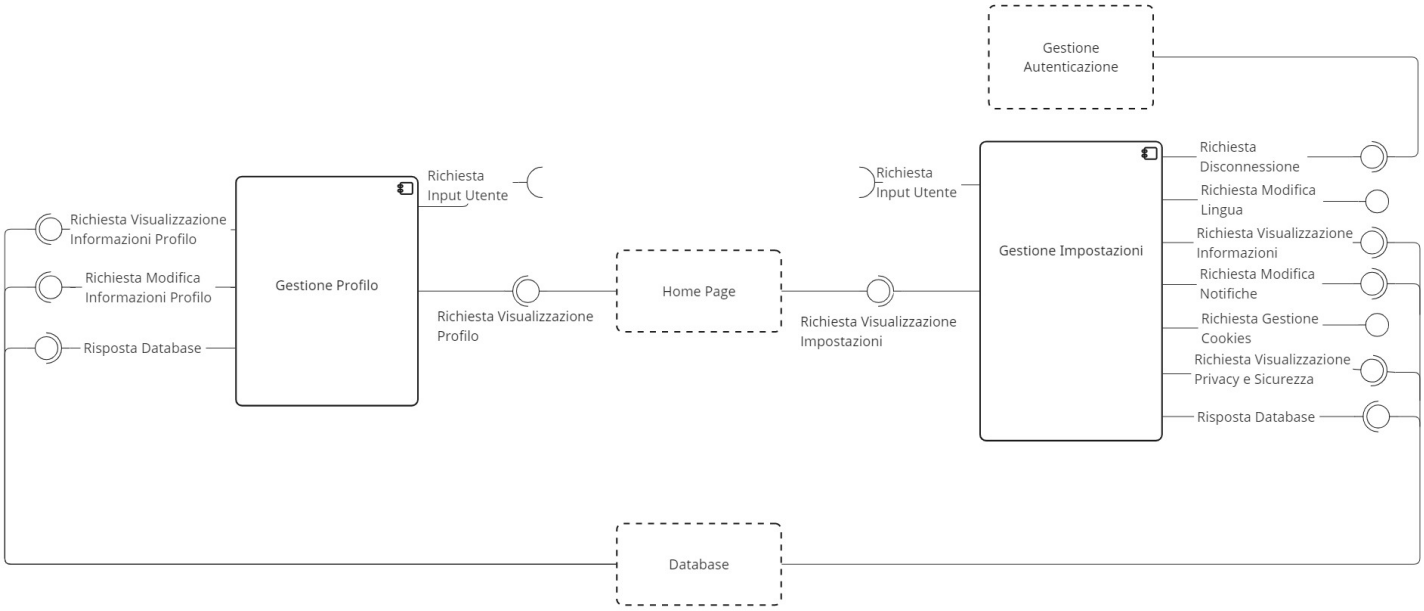

Figura 4: Component diagram Profilo, Impostazioni

15 - Gestione Profilo

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alla funzionalità del profilo e in base alle richieste dell'utente permette la visualizzazione e gestione delle informazioni relative al profilo.

Interfacce richieste

- Richiesta visualizzazione profilo: il componente riceve dal componente 'Home page' la richiesta di accedere alla sezione profilo e di visualizzare le informazioni relative al proprio profilo.
- Richiesta input utente: il componente, per permettere all'utente di modificare le informazioni associate il proprio profilo, richiede l'inserimento di uno o più dati relativi al profilo (nome, cognome, username, email, password).
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano il profilo (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta visualizzazione informazioni profilo: il componente fornisce al componente 'Database' la richiesta di visualizzare e quindi ricevere le informazioni relative al proprio profilo.
- Richiesta modifica informazioni profilo: il componente fornisce al componente 'Database' la richiesta di modificare le informazioni ruguardanti il proprio profilo (nome, cognome, username, email, password).

16 - Gestione impostazioni

Descrizione

Questo componente è responsabile di tutti gli aspetti relativi alle impostazioni del sistema e in base alle richieste dell'utente permette la visualizzazione e gestione delle informazioni relative alle impostazioni.

Interfacce richieste

- Richiesta visualizzazione impostazioni: il componente riceve dal componente 'Home page' la richiesta di accedere alla sezione impostazioni e di visualizzare le preferenze relative alle impostazioni.
- Richiesta input utente: il componente, per permettere all'utente di modificare le preferenze riguardo le impostazioni del sistema, richiede di inserire le proprie modifiche o preferenze riguardanti notifiche (attive o meno), lingua, cookies.
- Risposta database: il componente riceve dal componente 'Database' le informazioni che riguardano le impostazioni (in base a quanto richiesto al database tramite le interfacce fornite).

Interfacce fornite

- Richiesta disconnessione: il componente fornisce al componente 'Gestione autenticazione' la richiesta dell'utente disconnettersi dal sistema.
- Richiesta modifica lingua: il componente fornisce la richiesta dell'utente di modificare la lingua con cui sono visualizzate tutte le funzionalità e i dati del sistema.
- Richiesta visualizzazione informazioni: il componente fornisce al componente 'Database' la richiesta dell'utente di visualizzare le informazioni del sistema e i contatti degli sviluppatori.
- Richiesta modifica notifiche: il componente fornisce al componente 'Database' una richiesta per attivare/disattivare l'invio di notifiche da parte del sistema verso l'utente.
- Richiesta gestione cookies: il componente fornisce una richiesta per permettere all'utente di salvare le proprie preferenze riguardanti i cookies.
- Richiesta visualizzazione privacy e sicurezza: il componente fornisce al componente 'Database' una richiesta per visualizzare le informazioni riguardanti l'informativa sulla privacy e il trattamento dei dati personali.

17 - Database

Descrizione

Questo componente è responsabile dell'interfacciamento con il database e permette di effettuare richieste per leggere dal database (ottenere dati che verranno visualizzati dal sistema) oppure per scriverci (modifcarne le informazioni in base alla richieste dell'utente al sistema).

Interfacce richieste

- Inoltro dati utente/credenziali: il componente riceve dal componente 'Gestione autenticazione' la richiesta di scrivere le credenziali e i dati inseriti dall'utente in fase di registrazione nel database.
- Richiesta aggiunta movimento: il componente riceve dal componente 'Gestione budget' la richiesta di aggiungere un nuovo movimento al database con in dati inseriti dall'utente.
- Richiesta modifica movimento: il componente riceve dal componente 'Gestione budget' la richiesta di modificare un movimento esistente nel database con in dati inseriti dall'utente.
- Richiesta eliminazione movimento: il componente riceve dal componente 'Gestione budget' la richiesta di eliminare un movimento dal database.
- Richiesta aggiunta categoria: il componente riceve dal componente 'Gestione budget' la richiesta di aggiungere una nuova categoria di movimenti al database con in dati inseriti dall'utente.
- Richiesta modifica categoria: il componente riceve dal componente 'Gestione budget' la richiesta di modificare una categoria di movimenti esistente nel database con in dati inseriti dall'utente.

- Richiesta eliminazione categoria: il componente riceve dal componente 'Gestione budget' la richiesta di eliminare una categoria di movimenti dal database.
- Richiesta visualizzazione entrate: il componente riceve dal componente 'Gestione budget' la richiesta di ottenere i dati relativi ai soli movimenti in entrata affinchè possano essere visualizzati.
- Richiesta visualizzazione uscite: il componente riceve dal componente 'Gestione budget' la richiesta di ottenere i dati relativi ai soli movimenti in uscita affinchè possano essere visualizzati
- Richiesta visualizzazione categorie: il componente riceve dal componente 'Gestione budget' la richiesta di ottenere i dati relativi ai movimenti riguardanti una specifica categoria affinchè possano essere visualizzati.
- Richiesta visualizzazione movimenti: il componente riceve dal componente 'Gestione budget' la richiesta di ottenere i dati relativi a tutti i movimenti affinchè possano essere visualizzati.
- Richiesta aggiunta lista: il componente riceve dal componente 'Gestione liste' la richiesta di aggiungere una nuova lista al database con in dati inseriti dall'utente.
- Richiesta modifica lista: il componente riceve dal componente 'Gestione liste' la richiesta di modificare una lista esistente nel database con in dati inseriti dall'utente.
- Richiesta eliminazione lista: il componente riceve dal componente 'Gestione liste' la richiesta di eliminare una lista dal database.
- Richiesta aggiunta elemento lista: il componente riceve dal componente 'Gestione liste' la richiesta di aggiungere un nuovo elemento ad una lista con in dati inseriti dall'utente.
- Richiesta modifica elemento lista: il componente riceve dal componente 'Gestione liste' la richiesta di modificare un elemento esistente in una lista con in dati inseriti dall'utente.
- Richiesta eliminazione elemento lista: il componente riceve dal componente 'Gestione liste' la richiesta di eliminare un elemento di una lista.
- Richiesta svuota lista: il componente riceve dal componente 'Gestione liste' la richiesta di rimuovere tutti gli elementi della lista che sono contrassegnati (cioè che non servono più all'utente).
- Richiesta visualizzazione liste: il componente riceve dal componente 'Gestione liste' la richiesta di ottenere l'elenco delle liste esistenti affinchè possano essere visualizzate.
- Richiesta visualizzazione elementi in lista: il componente riceve dal componente 'Gestione liste' la richiesta di ottenere i dati relativi agli elementi di una specifica lista affinchè possano essere visualizzati.
- Richiesta aggiunta ricetta: il componente riceve dal componente 'Gestione ricette' la richiesta di aggiungere una nuova ricetta al database con in dati inseriti dall'utente.
- Richiesta modifica ricetta: il componente riceve dal componente 'Gestione ricette' la richiesta di modificare una ricetta esistente nel database con in dati inseriti dall'utente.
- Richiesta eliminazione ricetta: il componente riceve dal componente 'Gestione ricette' la richiesta di eliminare una ricetta dal database.
- Richiesta visualizzazione ricette: il componente riceve dal componente 'Gestione ricette' la richiesta di ottenere l'elenco delle ricette esistenti affinchè possano essere visualizzate.
- Richiesta visualizzazione dati ricetta: il componente riceve dal componente 'Gestione ricette' la richiesta di ottenere i dati relativi ad una specifica ricetta affinchè possano essere visualizzati.
- Richiesta aggiunta luogo: il componente riceve dal componente 'Gestione luoghi di interesse' la richiesta di aggiungere un nuovo luogo al database con in dati inseriti dall'utente.
- Richiesta modifica luogo: il componente riceve dal componente 'Gestione luoghi di interesse' la richiesta di modificare un luogo di interesse esistente nel database con in dati inseriti dall'utente.

- Richiesta eliminazione luogo: il componente riceve dal componente 'Gestione luoghi di interesse' la richiesta di eliminare un luogo dal database.
- Richiesta visualizzazione luoghi: il componente riceve dal componente 'Gestione luoghi di interesse' la richiesta di ottenere i dati relativi ai luoghi di interesse salvati affinchè possano essere visualizzati.
- Richiesta aggiunta carta: il componente riceve dal componente 'Gestione carte fedeltà' la richiesta di aggiungere una nuova carta fedeltà al database con in dati inseriti dall'utente.
- Richiesta modifica carta: il componente riceve dal componente 'Gestione carte fedeltà' la richiesta di modificare una carta fedeltà esistente nel database con in dati inseriti dall'utente.
- Richiesta eliminazione carta: il componente riceve dal componente 'Gestione carte fedeltà' la richiesta di eliminare una carta fedeltà dal database.
- Richiesta visualizzazione carta e barcode: il componente riceve dal componente 'Gestione carte fedeltà' la richiesta di ottenere i dati relativi ad una carta fedeltà affinchè possano essere visualizzati (con associato il barcode calcolato a partire dal numero della carta).
- Richiesta aggiunta evento: il componente riceve dal componente 'Gestione eventi' la richiesta di aggiungere un nuovo evento al database con in dati inseriti dall'utente.
- Richiesta modifica evento: il componente riceve dal componente 'Gestione eventi' la richiesta di modificare un evento esistente nel database con in dati inseriti dall'utente.
- Richiesta eliminazione evento: il componente riceve dal componente 'Gestione eventi' la richiesta di eliminare un evento dal database.
- Richiesta visualizzazione dettagli evento: il componente riceve dal componente 'Gestione eventi' la richiesta di ottenere i dati relativi ad un evento affinchè possano essere visualizzati.
- Richiesta visualizzazione informazioni profilo: il componente riceve dal componente 'Gestione profilo' la richiesta di ottenere i dati relativi al profilo dell'utente affinchè possano essere visualizzati.
- Richiesta modifica informazioni profilo: il componente riceve dal componente 'Gestione profilo' la richiesta di modificare le informazioni relative al profilo dell'utente nel database con in dati inseriti.
- Richiesta visualizzazione informazioni: il componente riceve dal componente 'Gestione impostazioni' la richiesta di ottenere i dati relativi alle informazioni di sistema e ai contatti degli sviluppatori affinchè possano essere visualizzati.
- Richiesta modifica notifiche: il componente riceve dal componente 'Gestione impostazioni' la richiesta di modificare la preferenza dell'utente riguardo la ricezione o meno delle notifiche.
- Richiesta visualizzazione privacy e sicurezza: il componente riceve dal componente 'Gestione impostazioni' la richiesta di ottenere le informazioni relative al trattamento dei dati personali affinchè possano essere visualizzati.

Interfacce fornite

- Risposta database: il componente fornisce ai vari componenti che hanno richiesto la lettura di dati dal database tali dati affinchè possano essere visualizzati.

Parte II
Analisi delle classi

Nel seguente capitolo analizzeremo le classi che costituiranno il sistema, andando a definire una prima architettura alle classi su cui LifeManager si basa, e le relazioni tra queste.
In particolare, una classe è sostanzialmente la descrizione di un gruppo di oggetti con proprietà (attributi), comportamento (operazioni), relazioni e semantica comuni. Una relazione è una connessione tra cose (cioè classi, interfacce, componenti, package), che fornisce un pathway (strada) per la comunicazione fra oggetti.
Il risultato di questa analisi costituirà il *Class Diagram*.

1 Utente

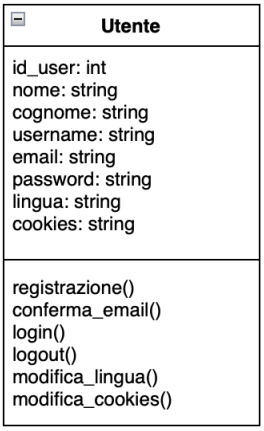

Figura 5: Classe Utente

La classe Utente è la classe principale in cui vengono memorizzate tutte le informazioni relative all'utente e le impostazioni che vengono salvate.

Attributi

- `id_user`: valore intero che identifica univocamente l'utente nell'intero database.
- `nome`: stringa contenente il nome anagrafico dell'utente.
- `cognome`: stringa contenente il cognome anagrafico dell'utente.
- `username`: stringa contenente il nome che l'utente ha scelto per farsi identificare univocamente all'interno del sistema.
- `email`: stringa contenente l'indirizzo di posta elettronica dell'utente.
- `password`: stringa contenente la password cifrata dell'utente.
- `lingua`: stringa contenente la lingua che l'utente ha scelto per operare nel sistema.
- `cookies`: stringa contenente valori che identificano i cookies che l'utente ha scelto di accettare o non accettare, secondo un ordine preciso.

Metodi

- `registrazione`: metodo che viene invocato alla registrazione che comporta l'inserimento nel database dell'utente.
- `conferma_email`: metodo che viene invocato quando, dopo aver effettuato la registrazione, l'utente riceve una mail in cui attraverso un link, attiva il profilo.
- `login`: metodo che viene invocato ogni qual volta che l'utente deve effettuare il login. Attraverso questo metodoo, l'utente anonimo diviene utente autenticato.
- `logout`: metodo che viene invocato quando l'utente intende uscire dall'applicazione. Con questo metodo l'utente autenticato diventa utente anonimo.
- `modifica_lingua`: metodo che viene invocato quando l'utente, che si trova nella sezione relativa alle impostazioni, desidera modificare la lingua del sistema.
- `modifica_cookies`: metodo che viene invocato quando l'utente, che si trova nella sezione relativa alle impostazioni, desidera modificare la lista dei cookies accettati, per usufruire del sistema.

Object Constraint Language (OCL)

- ```
  context Utente
  inv NomeLunghezza:  self.nome.size() >= 1 and self.nome.size() <= 100
  ```
- ```
  context Utente
  inv CognomeLunghezza:  self.cognome.size() >= 1 and self.cognome.size <= 100
  ```
- ```
  context Utente
  inv UsernameLunghezza:  self.username.size() >= 1 and self.username.size <= 100
  ```
- ```
  context Utente::registrazione()
  pre self.nome->notEmpty
  pre self.cognome->notEmpty
  pre self.username->notEmpty
  pre self.email->notEmpty
  pre self.password->notEmpty
  ```
- ```
  context Utente
  inv PasswordLunghezza:  self.password.size() >= 8
  ```
- ```
  context Utente::modifica_lingua()
  post self.lingua = NuovaLingua
  ```
- ```
  context Utente::modifica_cookies()
  post self.cookies = NuoviCookies
  ```
  ```

2 Data

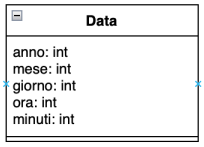

Figura 6: Classe Data

La classe Data è la classe in cui è mostrata la struttura di ogni data presente nel sistema.

Attributi

- `anno`: intero contenente l'anno.
- `mese`: intero contenente il mese.
- `giorno`: intero contenente il giorno.
- `ora`: intero contenente l'ora (formato 24 ore).
- `minuti`: intero contenente i minuti.

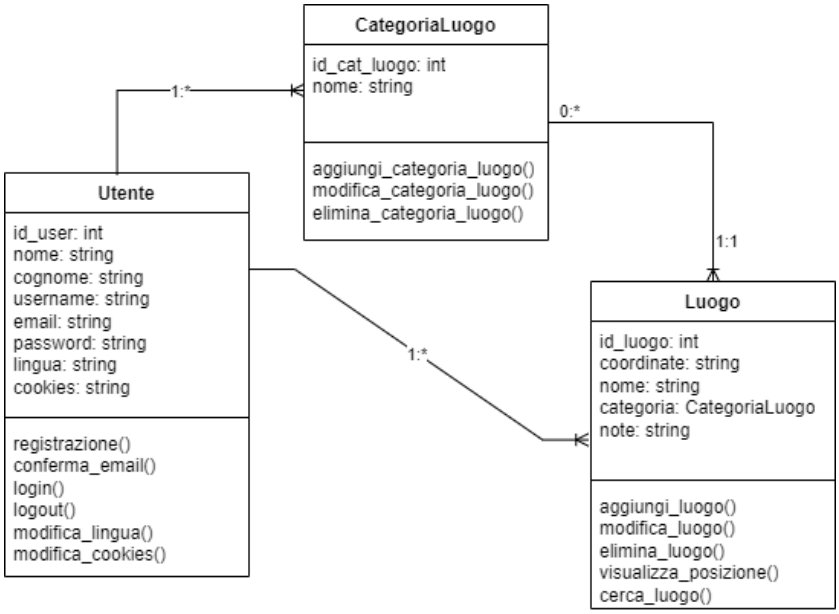

Figura 7: Classi Luogo e CategoriaLuogo

3 Categoria Luogo

La classe CategoriaLuogo permette di descrivere gli oggetti che rappresentano le categorie con cui l'utente classifica i propri luoghi e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_cat_luogo`: intero contenente un numero progressivo che identifica la precisa categoria.
- `nome`: stringa contenente il nome che l'utente sceglie di dare alla categoria.

Metodi

- `aggiungi_categoria_luogo`: metodo che l'utente invoca quando deve creare una nuova categoria. Il sistema, attraverso tal funzione, la inserisce nel database.
- `modifica_categoria_luogo`: metodo che l'utente invoca quando desidera modificare il nome di una categoria.
- `elimina_categoria_luogo`: metodo che l'utente invoca quando desidera cancellare una categoria. L'eliminazione non comporta la cancellazione dei luoghi associati alla categoria.

Object Constraint Language (OCL)

- `context CategoriaLuogo inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context CategoriaLuogo::aggiungi_categoria_luogo()
  pre self.nome->notEmpty()
  ```

4 Luogo

La classe Luogo permette di descrivere gli oggetti che rappresentano i luoghi dei quali l'utente vuole tener traccia e descrive le operazione possibili per ogni oggetto di questo tipo.
  ```

Attributi

- `id_luogo`: intero contenente un numero progressivo che identifica il preciso luogo.
- `coordinate`: stringa contenente le coordinate a cui si riferisce un luogo.
- `nome`: stringa contenente il nome che l'utente desidera dare al luogo.
- `categoria`: stringa contenente il nome della categoria a cui il luogo è associato
- `note`: stringa contenente le note che l'utente desidera inserire.

Metodi

- `aggiungi_luogo`: metodo che l'utente invoca quando deve inserire un nuovo luogo. Il sistema, attraverso tal funzione, lo inserisce nel database.
- `modifica_luogo`: metodo che l'utente invoca quando desidera modificare qualunque attributo del luogo.
- `elimina_luogo`: metodo che l'utente invoca quando desidera eliminare un luogo.
- `visualizza_posizione`: metodo che l'utente invoca quando desidera visualizzare la posizione (attrzverso un segnaposto) sulla mappa.
- `cerca_luogo`: metodo che l'utente invoca quando desidera cercare un luogo.

Object Constraint Language (OCL)

- `context Luogo inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context Luogo::aggiungi_luogo()
  pre self.nome->notEmpty()
  pre self.categoria->notEmpty()
  ```
  ```

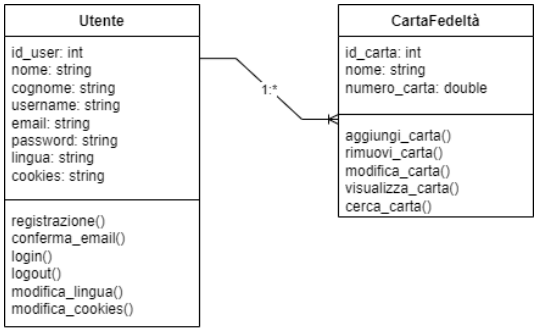

Figura 8: Classe Carta fedeltà

5 Carta fedeltà

La classe CartaFedeltà permette di descrivere gli oggetti che rappresentano le carte fedeltà dell'utente e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_carta`: intero, identificativo univoco della carta fedeltà.
- `nome`: stringa contenente il nome che l'utente intende associare alla carta.
- `numero_carta`: intero contenente il numero della carta (codice a barre).

Metodi

- `aggiungi_carta`: metodo che l'utente invoca quando deve inserire una nuova carta fedeltà. Il sistema, attraverso tal funzione, la inserisce nel database.
- `modifica_carta`: metodo che l'utente invoca quando desidera modificare qualunque attributo della carta fedeltà.
- `elimina_carta`: metodo che l'utente invoca quando desidera eliminare una carta fedeltà.
- `visualizza_carta`: metodo che l'utente invoca quando desidera visualizzare il codice a barre correlato alla carta.
- `cerca_carta`: metodo che l'utente invoca quando desidera cercare una carta.

Object Constraint Language (OCL)

- `context CartaFedeltà inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context CartaFedeltà::aggiungi_carta()
  pre self.nome->notEmpty()
  pre self.numero_carta->notEmpty()
  ```
  ```

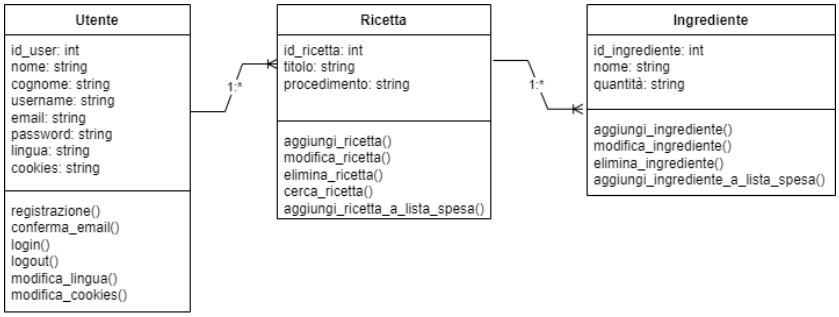

Figura 9: Classi Ricetta e Ingrediente

6 Ricetta

La classe Ricetta permette di descrivere gli oggetti che rappresentano le ricette dell'utente e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_ricetta`: intero contenente un numero progressivo che identifica la precisa ricetta.
- `titolo`: stringa contenente il titolo che l'utente sceglie di dare alla ricetta.
- `procedimento`: stringa contenente un testo che comprende il procedimento ed eventuali note.

Metodi

- `aggiungi_ricetta`: metodo che l'utente invoca quando deve creare una nuova ricetta. Il sistema, attraverso tal funzione, la inserisce nel database.
- `modifica_ricetta`: metodo che l'utente invoca quando desidera modificare una ricetta.
- `elimina_ricetta`: metodo che l'utente invoca quando desidera cancellare una ricetta. L'eliminazione comporta anche l'eliminazione degli ingredienti ad essa associati.
- `cerca_ricetta`: metodo che l'utente invoca quando desidera cercare una ricetta.
- `aggiungi_ricetta_a_lista_spesa`: metodo che l'utente invoca quando desidera aggiungere tutti gli ingredienti alla propria lista della spesa.

Object Constraint Language (OCL)

- `context Ricetta inv TitoloLunghezza:  self.titolo.size() >= 1`
- ```
  context Ricetta::aggiungi_ricetta()
  pre self.titolo->notEmpty()
  ```

7 Ingrediente

La classe Ingrediente permette di descrivere gli oggetti che rappresentano gli ingredienti di una ricetta e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_ingrediente`: intero contenente l'identificativo dell'ingrediente.
- `nome`: stringa contenente l'ingrediente.
- `quantità`: stringa contenente la quantità dell'ingrediente nel contesto della ricetta
  ```

Metodi

- `aggiungi_ingrediente`: metodo che l'utente invoca quando deve inserire un nuovo ingrediente.
- `modifica_ingrediente`: metodo che l'utente invoca quando desidera modificare le informazioni relative ad un ingrediente.
- `elimina_ingrediente`: metodo che l'utente invoca quando desidera eliminare un ingrediente.
- `aggiungi_ingrediente_a_lista_spesa`: metodo che l'utente invoca quando desidera aggiungere l'ingrediente alla propria lista della spesa.

Object Constraint Language (OCL)

- `context Ingrediente inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context Ingrediente::aggiungi_ingrediente()
  pre self.nome->notEmpty()
  ```
  ```

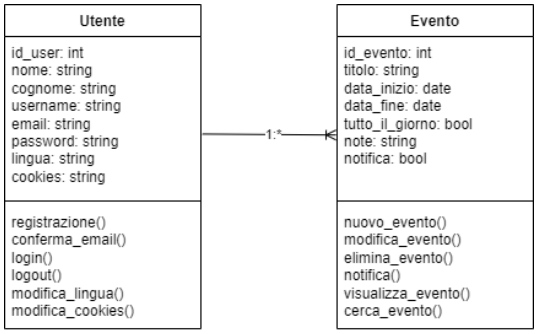

Figura 10: Classe Evento

8 Evento

La classe Evento permette di descrivere gli oggetti che rappresentano gli eventi e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_evento`: intero identificativo dell'evento specifico.
- `titolo`: stringa contenente il titolo che l'utente desidera associare ad un evento.
- `data_inizio`: data (secondo il formato dato dalla classe Data) in cui l'evento inizia.
- `data_fine`: data (secondo il formato dato dalla classe Data) in cui l'evento termina.
- `tutto_il_giorno`: booleano che identifica se l'evento è relativo ad una giornata intera (da mezzanotte a mezzanotte).
- `note`: stringa contenente le note che l'utente desidera inserire.
- `notifica`: booleano in cui è memorizzata la scelta dell'utente in merito al ricevere la notifica.

Metodi

- `nuovo_evento`: metodo che l'utente invoca quando deve inserire un nuovo evento.
- `modifica_evento`: metodo che l'utente invoca quando desidera modificare qualunque attributo dell'evento.
- `elimina_evento`: metodo che l'utente invoca quando desidera eliminare un evento.
- `visualizza_evento`: metodo che l'utente invoca quando desidera visualizzare l'evento.
- `cerca_evento`: metodo che l'utente invoca quando desidera cercare un evento.
- `notifica`: metodo invocato quando l'utente riceve la notifica di un evento.

Object Constraint Language (OCL)

- `context Evento inv self.data_fine > self.data_fine`
- `context Evento inv NomeLunghezza:  self.titolo.size() >= 1`
- ```
  context Evento::aggiungi_evento()
  pre self.titolo->notEmpty()
  ```
  ```

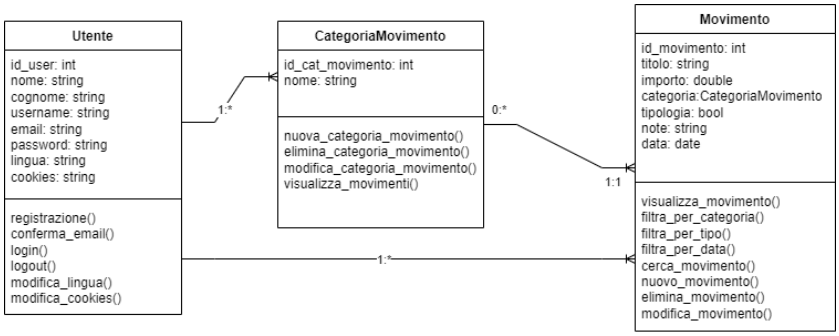

Figura 11: Classi Movimento e CategoriaMovimento

9 Categoria Movimento

La classe CategoriaMovimento permette di descrivere gli oggetti che rappresentano le categorie con cui l'utente classifica i propri movimenti e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_cat_movimento`: intero contenente un numero progressivo che identifica la precisa categoria.
- `nome`: stringa contenente il nome che l'utente sceglie di dare alla categoria.

Metodi

- `nuova_categoria_movimento`: metodo che l'utente invoca quando deve creare una nuova categoria. Il sistema, attraverso tal funzione, la inserisce nel database.
- `modifica_categoria_movimento`: metodo che l'utente invoca quando desidera modificare il nome di una categoria.
- `elimina_categoria_movimento`: metodo che l'utente invoca quando desidera cancellare una categoria. L'eliminazione non comporta la cancellazione dei movimenti associati alla categoria.
- `visualizza_movimenti`: metodo consente all'utente di visualizzare tutti i movimenti associati alla categoria.

Object Constraint Language (OCL)

- `context CategoriaMovimento inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context CategoriaMovimento::nuova_categoria_movimento()
  pre self.nome->notEmpty()
  ```

10 Movimento

La classe Movimento permette di descrivere gli oggetti che rappresentano i movimenti e descrive le operazioni possibili per ogni oggetto di questo tipo.

Attributi

- `id_movimento`: intero contenente un numero progressivo che identifica il preciso movimento.
- `titolo`: stringa contenente il titolo del movimento.
- `importo`: valore numerico associato all'importo.
  ```

- `categoria`: categoria a cui il movimento è associato.
- `tipologia`: booleano contenente la tipologia del moovimento (entrata o uscita).
- `note`: stringa contenente le note che l'utente desidera inserire.
- `data`: data (secondo il formato dato dalla classe Data) associata al movimento.

Metodi

- `nuovo_movimento`: metodo che l'utente invoca quando deve inserire un nuovo movimento.
- `modifica_movimento`: metodo che l'utente invoca quando desidera modificare qualunque attributo del movimento.
- `elimina_movimento`: metodo che l'utente invoca quando desidera eliminare un movimento.
- `visualizza_movimento`: metodo che l'utente invoca quando desidera mostrare i dati relativi al movimento
- `cerca_moovimento`: metodo che l'utente invoca quando desidera cercare un movimento.
- `filtra_per_categoria`: metodo che l'utente invoca quando desidera mostrare tutti i movimenti di una specifica categoria.
- `filtra_per_tipo`: metodo che l'utente invoca quando desidera mostrare tutti i movimenti di un tipo specifico (tutte le entrate o tutte le uscite).
- `filtra_per_data`: metodo che l'utente invoca quando desidera mostrare tutti i movimenti dell'ultimo mese.

Object Constraint Language (OCL)

- `context Movimento inv TitoloLunghezza:  self.titolo.size() >= 1`
- ```
  context Movimento::nuovo_movimento()
  pre self.titolo->notEmpty()
  pre self.importo->notEmpty()
  ```
  ```

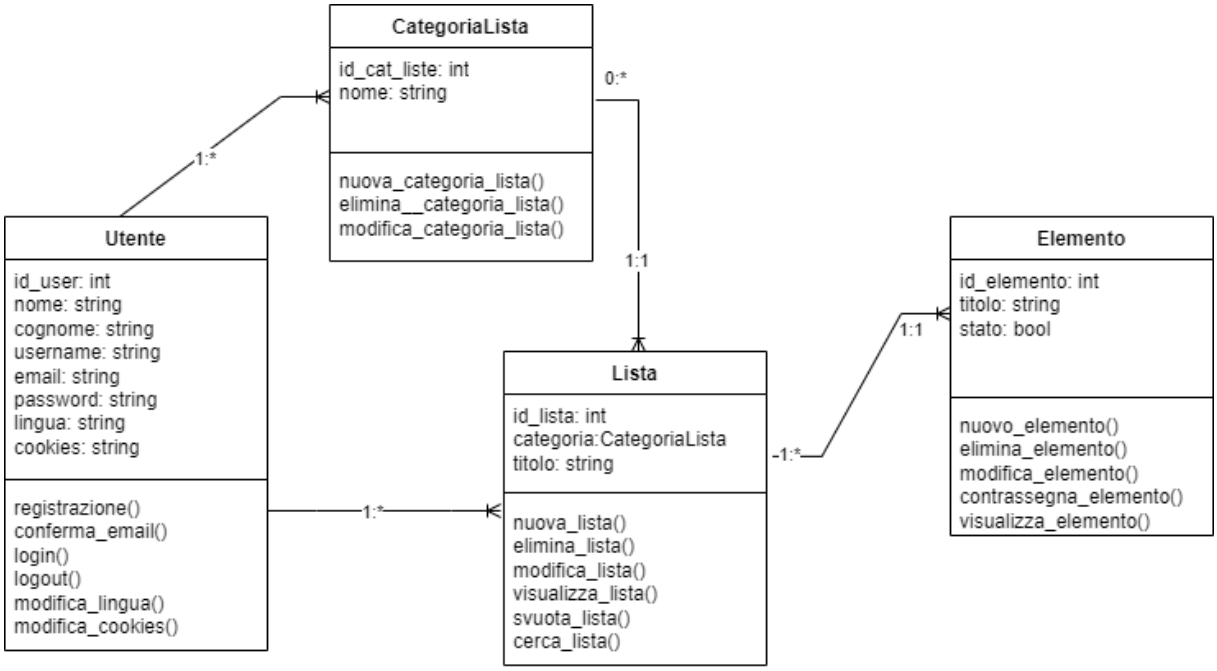

Figura 12: Classi CategoriaLista, Lista ed Elemento

11 Categoria Lista

La classe CategoriaLista permette di descrivere gli oggetti che rappresentano le categorie con cui l'utente classifica le proprie liste e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_cat_lista`: intero contenente un numero progressivo che identifica la precisa categoria.
- `nome`: stringa contenente il nome che l'utente sceglie di dare alla categoria.

Metodi

- `nuova_categoria_lista`: metodo che l'utente invoca quando deve creare una nuova categoria.
- `modifica_categoria_lista`: metodo che l'utente invoca quando desidera modificare il nome di una categoria.
- `elimina_categoria_lista`: metodo che l'utente invoca quando desidera cancellare una categoria. L'eliminazione non comporta la cancellazione delle liste associate alla categoria.

Object Constraint Language (OCL)

- `context CategoriaLista inv NomeLunghezza:  self.nome.size() >= 1`
- ```
  context CategoriaLIsta::nuova_categoria_lista()
  pre self.nome->notEmpty()
  ```

12 Lista

La classe Lista permette di descrivere gli oggetti che rappresentano le liste con cui l'utente classifica i propri luoghi e descrive le operazione possibili per ogni oggetto di questo tipo.
  ```

Attributi

- `categoria`: categoria a cui è associata la lista.
- `id_lista`: intero contenente un numero progressivo che identifica la lista.
- `titolo`: stringa contenente il titolo della lista.

Metodi

- `nuova_lista`: metodo che l'utente invoca quando deve inserire una nuova lista.
- `modifica_lista`: metodo che l'utente invoca quando desidera modificare una lista.
- `elimina_lista`: metodo che l'utente invoca quando desidera eliminare una lista. L'eliminazione comporta anche la cancellazione degli elementi associati.
- `visualizza_lista`: metodo che l'utente invoca quando desidera mostrare la lista e tutti gli elementi.
- `cerca_lista`: metodo che l'utente invoca quando desidera cercare una lista.
- `svuota_lista`: metodo che l'utente invoca quando desidera eliminare tutti gli elementi della lista senza però cancellare la lista.

Object Constraint Language (OCL)

- `context Lista inv TitoloLunghezza: self.titolo.size() >= 1`
- ```
  context Lista::nuova_lista()
  pre self.titolo->notEmpty()
  ```

13 Elemento

La classe Elemento permette di descrivere gli oggetti che rappresentano gli elementi della lista e descrive le operazione possibili per ogni oggetto di questo tipo.

Attributi

- `id_elemento`: intero contenente un numero progressivo che identifica l'elemento.
- `titolo`: stringa contenente il titolo dell'elemento.
- `stato`: booleano che identifica lo stato dell'elemento (contrassegnato o non contrassegnato).

Metodi

- `nuovo_elemento`: metodo che l'utente invoca quando deve inserire un nuovo elemento.
- `modifica_elemento`: metodo che l'utente invoca quando desidera modificare un elemento.
- `elimina_elemento`: metodo che l'utente invoca quando desidera eliminare un elemento.
- `visualizza_elemento`: metodo che l'utente invoca quando desidera mostrare l'elemento.
- `contrassegna_elemento`: metodo che l'utente invoca quando desidera contrassegnare un elemento.
  ```

Object Constraint Language (OCL)

- `context Elemento inv TitoloLunghezza:  self.titolo.size() >= 1`
- `context Elemento::nuovo_elemento()`
  `pre self.titolo->notEmpty()`
- `context Elemento::contrassegna_elemento()`
  `post self.stato = true`

Mauro Meneghello - matricola 217564 - mauro.meneghello@studenti.unitn.it
Luca Boschiero - matricola 217460 - luca.boschiero@studenti.unitn.it
Nicola Turniano - matricola 217457 - nicola.turniano@studenti.unitn.it