

Treport

Ingegneria del Software A.S. 2024 - 2025

Tommaso Ascolani, Riccardo D'Incau, Marco Zanatta

Documento di analisi e progettazione

Indice

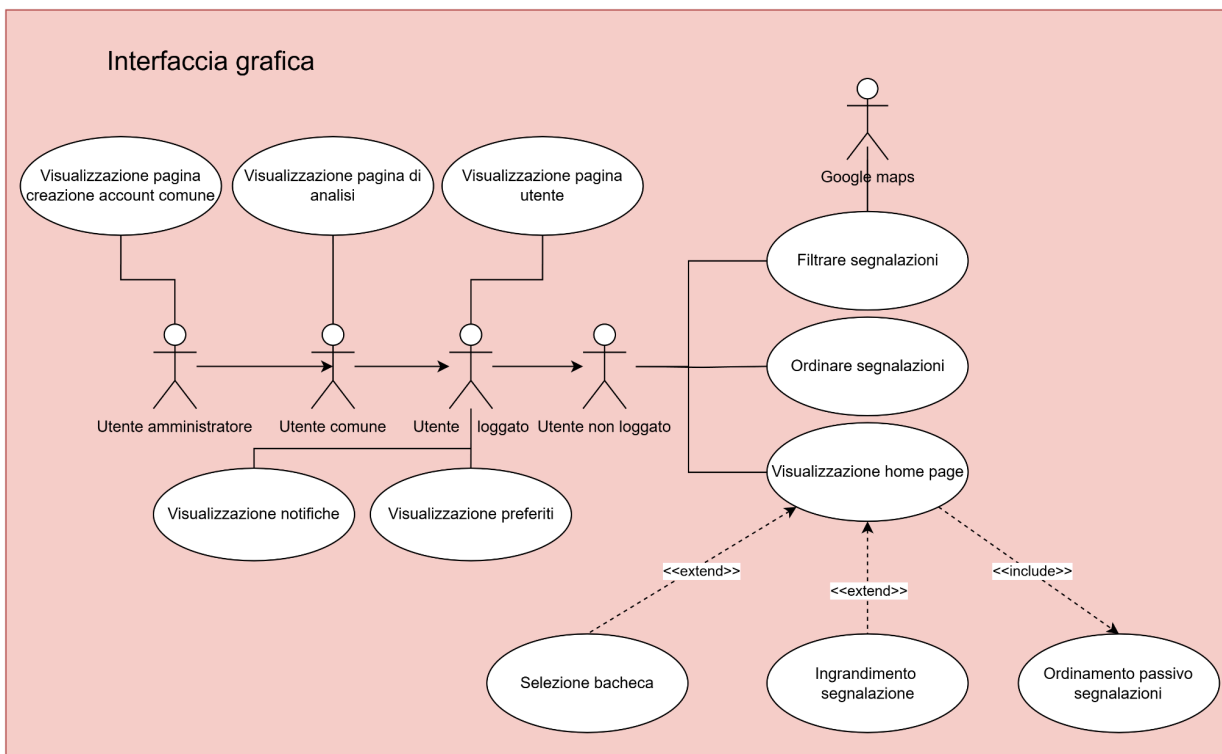
Indice.....	2
1. Requisiti funzionali.....	3
Interfaccia grafica.....	3
Gestione account.....	7
Segnalazioni.....	10
2. Analisi dei componenti.....	14
a. Definizione dei componenti.....	14
b. Diagramma dei componenti.....	23
3. Diagramma delle classi.....	24
4. Dal class diagram alle API.....	25

1. Requisiti funzionali

Nel presente capitolo vengono riportati i requisiti funzionali (RF) del sistema utilizzando il linguaggio naturale e Use Case Diagram (UCD) scritti in UML. Il sistema è stato suddiviso in 3 sezioni: interfaccia grafica, gestione account e segnalazioni. Per ogni sezione è presente una parte dello use case diagram e una descrizione per ogni use case presente nel pezzo del diagramma. In ogni descrizione prima vengono elencati i requisiti funzionali che riguardano lo use case e poi viene specificato il nome dello use case a cui si fa riferimento.

Interfaccia grafica

RF7 Preview segnalazione, RF8 Segnalazione ingrandita, RF10 Notifica segnalazioni, RF11 Selezione bacheca, RF12 Filtro e riordinamento segnalazioni, RF13 Ordinamento passivo segnalazioni, RF14 Pagina utente, RF15 Sezione preferiti, RF16 Analisi segnalazione, RF17 Creazione account comune



Use case RF7, RF8, RF11, RF13: Visualizzazione home page

Riassunto

All'accesso al sito sono visibili le preview delle segnalazioni attive, e i pulsanti per la selezione della bacheca.

Descrizione

1. All'accesso al sito gli utenti vedono le preview delle segnalazioni scelte dall'algoritmo di ordinamento passivo.
2. Una volta nella home page gli utenti possono cambiare bacheca **[Estensione 1]** e ingrandire una segnalazione cliccando sulla preview della stessa **[Estensione 2]**.

Estensioni

1. Un utente può selezionare tramite il tasto corrispondente quale bacheca visualizzare tra bacheca attive, prese in carico e archiviate.
2. Un utente può accedere alla pagina segnalazione ingrandita per vedere interamente il contenuto della segnalazione e la sezione commenti.

Use case RF10: Visualizzazione notifiche

Riassunto

Ad un utente loggato vengono mostrate le notifiche in una sezione apposita della homepage.

Descrizione

1. L'utente loggato visualizza le notifiche relative alle segnalazioni con cui questo ha interagito.
2. Quando una di queste viene cliccata, l'utente viene reindirizzato alla pagina segnalazione ingrandita della segnalazione della notifica.

Use case RF12: Filtro segnalazioni/Ordinamento segnalazioni

Riassunto

Un utente può: riordinare la lista di preview di segnalazioni mostrate e/o filtrarla.

Descrizione

1. L'utente può scegliere i criteri con cui riordinare la lista di preview.
2. L'utente può scegliere i filtri da applicare alla lista di preview. **[Estensione 1]**
3. L'utente può effettuare una ricerca per parola chiave.

Al termine di una di queste azioni la lista verrà in automatico aggiornata.

Estensioni

1. L'utente può scegliere di applicare un filtro per posizione anche selezionando una posizione tramite Google Maps

Use case RF15: Visualizzazione preferiti

Riassunto

Ad un utente loggato vengono mostrati i preferiti in una sezione apposita della homepage.

Descrizione

1. L'utente loggato visualizza i preferiti, ovvero la raccolta dei filtri recentemente usati o più usati dall'utente.
2. Quando un filtro dei preferiti verrà scelto, esso verrà applicato in maniera automatica alla lista di preview di segnalazioni attualmente mostrata.

Use case RF14: Visualizzazione pagina utente

Riassunto

Ogni utente loggato può accedere alla propria pagina utente.

Descrizione

1. Per gli utenti loggati sarà visibile un pulsante per accedere alla pagina utente nella homepage.
2. Dopo aver cliccato il pulsante gli utenti accedono alla pagina utente e possono visualizzare le loro informazioni sull'account e le segnalazioni con cui l'utente ha interagito o che ha creato.

Use case RF16: Visualizzazione pagina di analisi

Riassunto

Gli utenti comune possono accedere alla pagina di analisi dei dati raccolti tramite l'applicazione.

Descrizione

1. Per gli utenti comune sarà visibile un pulsante per accedere alla pagina di analisi nella home page.
2. Una volta nella pagina è possibile visualizzare le statistiche basate sui dati dell'applicazione.

Use case RF17: Visualizzazione pagina creazione account comune

Riassunto

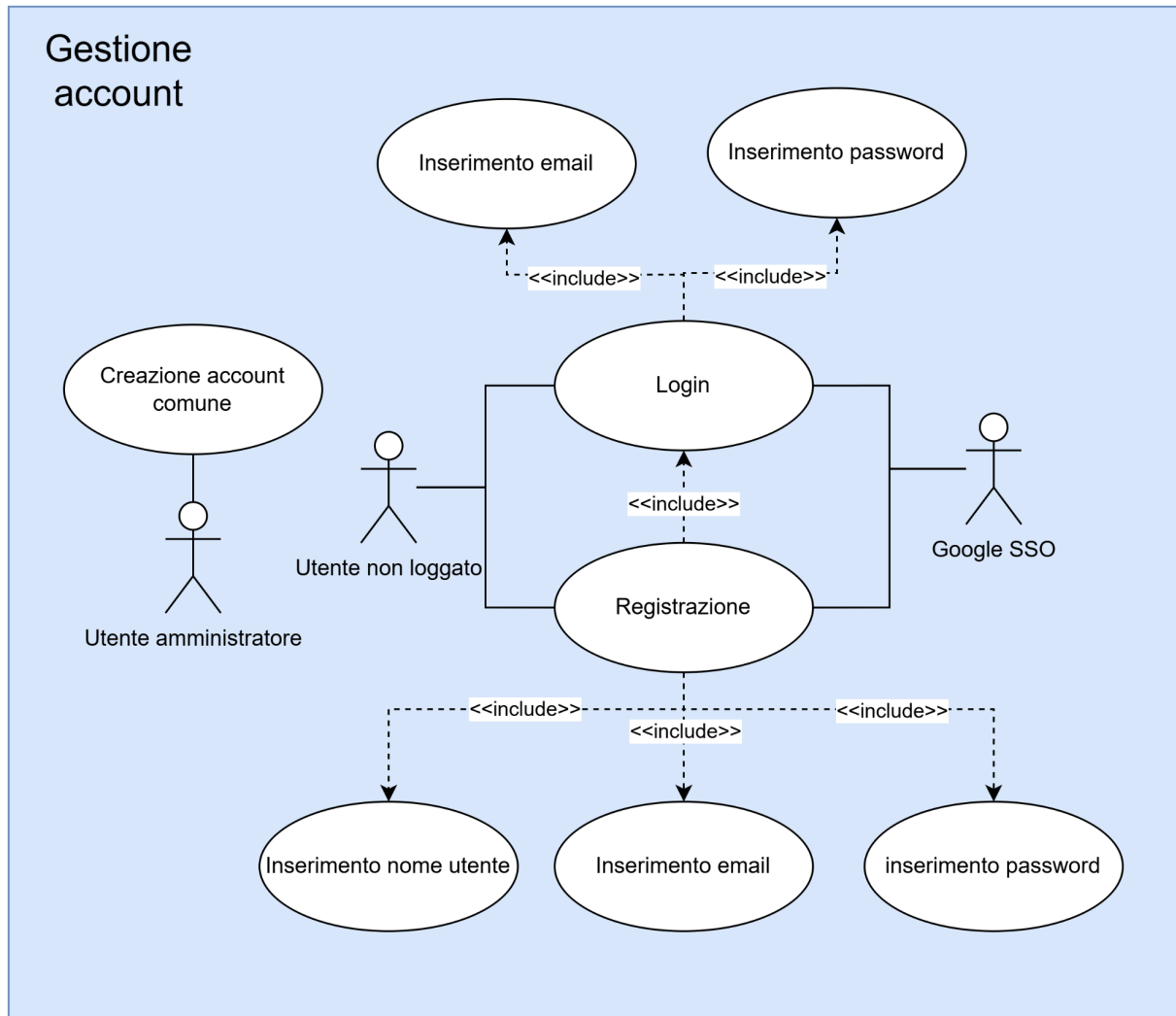
Per gli utenti amministratori è disponibile la pagina di creazione account comune.

Descrizione

1. Nella homepage agli utenti amministratore è visibile un pulsante per accedere alla pagina.
2. Una volta nella pagina l'utente può compilare i campi richiesti.
3. Dopo aver confermato l'operazione di creazione appare un messaggio che conferma il corretto salvataggio del nuovo account.

Gestione account

RF1 Login, RF2 Registrazione, RF17 Creazione account comune



Use case RF17: Creazione account comune

Riassunto

Gli utenti amministratore possono creare un account utente comune.

Descrizione

1. Agli utenti amministratore sarà visibile un pulsante per accedere alla pagina di creazione account comune.
2. Una volta arrivato alla pagina l'utente amministratore può inserire i campi richiesti (nome utente, email **[Eccezione 1]** e password).
3. Premendo il tasto di conferma l'account viene creato.

Eccezioni

1. La mail non deve essere collegata ad alcun account esistente altrimenti viene visualizzato un messaggio di errore.

Use case RF1: Login

Riassunto

Un utente non loggato può effettuare il login.

Descrizione

1. Per gli utenti non loggati nella homepage è visibile un pulsante che porta alla pagina di login.
2. Una volta nella pagina l'utente può inserire email e password per accedere alla piattaforma. **[Estensione 1]**.
3. Cliccando il pulsante apposito l'utente viene autenticato **[Eccezione 1]** e viene riportato alla pagina precedente al login.

Eccezioni

1. Se le credenziali non sono valide o l'autenticazione tramite Google fallisce la procedura di autenticazione viene cancellata e viene visualizzato un messaggio di errore.

Estensioni

1. L'utente può cliccare sul pulsante per l'accesso con Google per effettuare l'accesso tramite Google.

Use case RF2: Registrazione

Riassunto

L'utente non loggato effettua la registrazione inserendo le sue credenziali oppure attraverso autenticazione Google.

Descrizione

1. L'utente non loggato arriva alla pagina di registrazione
2. L'utente non loggato inserisce le credenziali desiderate (Nome utente, email, password) **[Estensione 1]**.
3. Se le credenziali sono valide **[Eccezione 1]** l'utente viene registrato.
4. In automatico l'utente viene autenticato con le credenziali appena create.
5. L'utente viene reindirizzato alla pagina in cui si trovava precedentemente a quella di login.

Eccezione

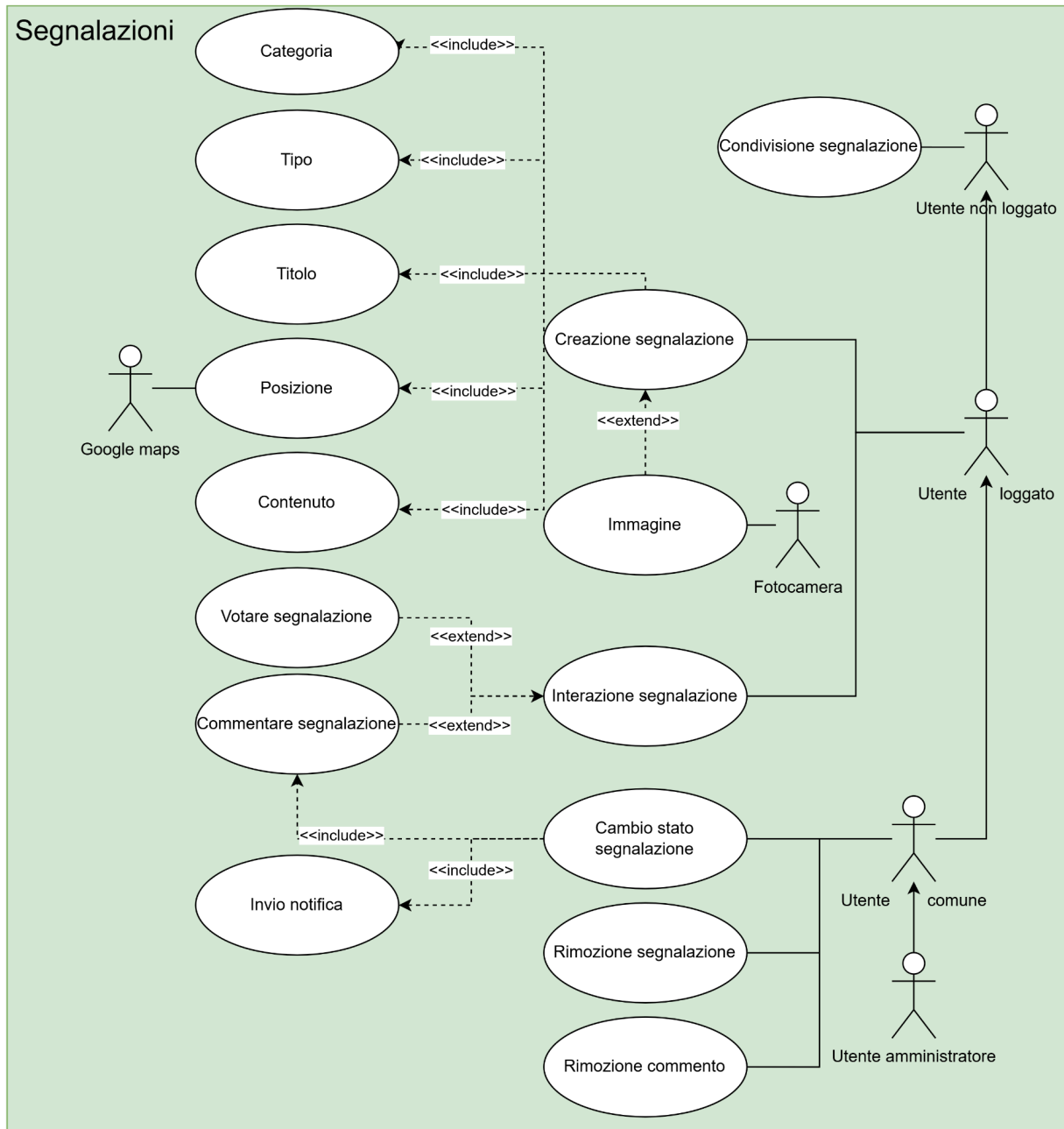
1. Nel caso che le credenziali non siano corrette o l'autenticazione Google fallisca viene visualizzato un messaggio di errore e si interrompe la procedura di autenticazione.

Estensioni

1. L'utente può scegliere di registrarsi tramite Google con l'apposito pulsante.

Segnalazioni

RF3 Commenti, RF4 votazione segnalazione, RF5 Condivisione segnalazione, RF6 Creazione segnalazione, RF8 Segnalazione ingrandita, RF9 Cambio di stato della segnalazione, RF10 Notifica segnalazione, RF18 Moderazione commenti e segnalazioni



Use case RF6: Creazione segnalazione

Riassunto

Gli utenti loggati possono creare una segnalazione.

Descrizione

1. Cliccando sull'apposito bottone di creazione segnalazione, l'utente **[Eccezione 1]** viene reindirizzato alla pagina per la creazione di una segnalazione.
2. Nel form l'utente deve inserire il titolo, il contenuto, il tipo, la categoria e la posizione **[Estensione 1]** inerenti alla segnalazione
3. L'utente può allegare alla segnalazione un'immagine relativa al problema segnalato **[Estensione 2]**.

Eccezioni

1. Se l'utente non è loggato verrà reindirizzato alla pagina di login.

Estensioni

1. L'utente può scegliere se inserire manualmente la posizione (Google maps) o delegare al gps del dispositivo.
2. Per farlo può scattare una foto tramite la fotocamera.

Use case RF3, RF8: Commentare segnalazione

Riassunto

Gli utenti loggati possono interagire con le segnalazioni commentandole.

Descrizione

1. L'utente **[Eccezione 1]** che si trova nella pagina segnalazione ingrandita può premere sulla barra di testo per inserire un commento.
2. Nella barra di testo l'utente può scrivere il commento relativo alla segnalazione.
3. Con l'apposito pulsante l'utente può confermare la pubblicazione del commento.

[Estensione 1]

Eccezioni

1. Se l'utente non è loggato verrà reindirizzato alla pagina di login.

Estensioni

1. I commenti pubblicati da un utente comune sono evidenziati.

Use case RF4: Votare segnalazione

Riassunto

Gli utenti loggati possono votare.

Descrizione

1. L'utente **[Eccezione 1]** può votare la segnalazione tramite l'apposito bottone nella preview segnalazione oppure nella pagina segnalazione ingrandita.

Eccezioni

1. Se l'utente non è loggato verrà reindirizzato alla pagina di login.

Use case RF9: Cambio stato segnalazione

Riassunto

L'utente comune può cambiare lo stato della segnalazione.

Descrizione

1. L'utente comune può cliccare sull'apposito tasto di cambio stato segnalazione nella pagina segnalazione ingrandita per far apparire i possibili stati da poter scegliere.
2. L'utente comune può scegliere quale stato far diventare quello della segnalazione.
3. L'utente comune è obbligato ad inserire un commento prima di confermare il cambio di stato **[Eccezione 1]**.

[Estensione 1]

Eccezioni

1. Se l'utente comune non inserisce un commento viene visualizzato un messaggio di errore e viene annullato il cambio di stato.

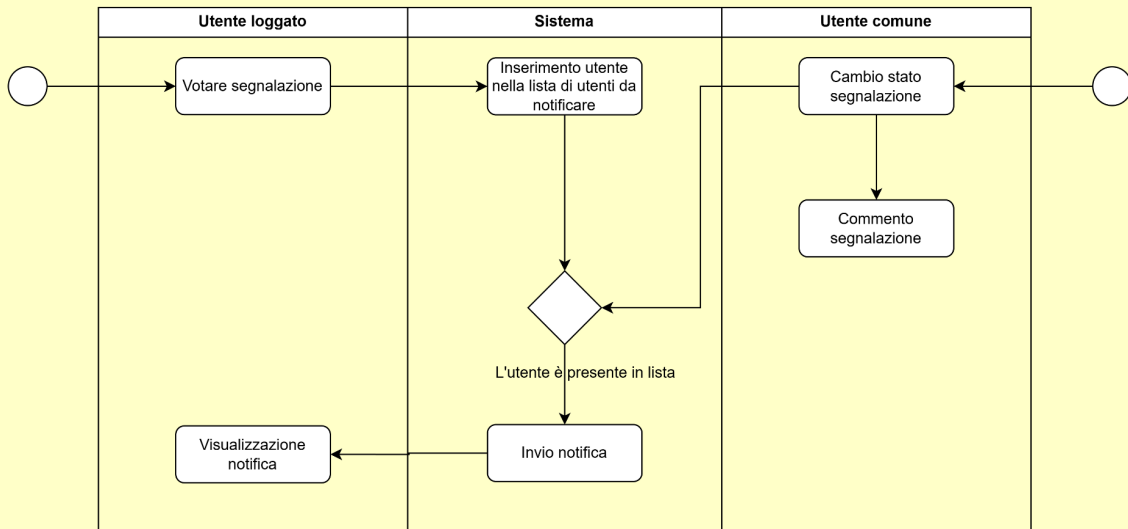
Estensioni

1. Quando lo stato di una segnalazione viene cambiato, chi ha pubblicato quella segnalazione e chi l'ha votata riceverà una notifica che informerà del cambiamento avvenuto.

Use case RF10: Invio notifica

Di seguito un activity diagram che descrive il flusso delle operazioni riguardante l'invio delle notifiche.

Activity diagram sistema notifiche



Use case RF18: Rimozione segnalazione/Rimozione commento

Riassunto

Un utente comune può eliminare una segnalazione o un commento.

Descrizione

1. Cliccando sull'apposito pulsante un utente comune può eliminare una segnalazione dalla pagina segnalazione ingrandita.
2. Cliccando sull'apposito pulsante un utente comune può eliminare un commento dalla pagina segnalazione ingrandita.

Use case RF5: Condivisione segnalazione

Riassunto

Un utente può condividere una segnalazione.

Descrizione

1. Cliccando il pulsante di condivisione sulla preview segnalazione o sulla pagina segnalazione ingrandita un utente può condividere una segnalazione.
2. Dopo aver cliccato il pulsante appare una lista di piattaforme utilizzabili per la condivisione.

2. Analisi dei componenti

Nel presente capitolo viene presentata l'architettura in termini di componenti (CMP) interni al sistema definiti sulla base dei requisiti analizzati in precedenza. Viene poi adottato l'uso del diagramma dei componenti per rappresentare l'interconnessione tra i vari componenti, identificando quindi le interfacce tra questi e verso sistemi esterni.

a. Definizione dei componenti

CMP1. Gestione autenticazione

Descrizione

Questo componente si occupa di gestire l'autenticazione di un utente alla piattaforma. Fornisce una pagina di login per l'accesso. Per accedere si possono usare le credenziali generate durante la fase di registrazione (Email e password) oppure si può effettuare l'accesso tramite autenticazione Google.

Interfaccia richiesta - Autorizzazione autenticazione da Google

Questa interfaccia aspetta la conferma di autenticazione da parte di Google quando questa viene scelta.

Interfaccia richiesta - Credenziali accesso

Sono le credenziali utilizzate per l'accesso alla piattaforma, in particolare sono richiesti email e password.

Interfaccia richiesta - Autenticazione con registrazione

Questa interfaccia accetta una conferma di avvenuta registrazione per automaticamente autenticare un utente che si è appena registrato.

Interfaccia fornita - Richiesta autenticazione verso Google

Questa interfaccia reindirizza l'utente verso l'autenticazione Google.

Interfaccia fornita - Autenticazione

Fornisce i dati di un utente che ha correttamente effettuato la fase di login. Per esempio fornisce: nome utente, id utente e livello utente.

CMP2. Gestione registrazione

Descrizione

Questo componente permette agli utenti non loggati di registrare un nuovo account. Per registrare un nuovo account possono essere inserite le credenziali desiderate o si può effettuare una registrazione tramite Google. Gli utenti così creati saranno di tipo utente loggato (Cioè non di tipo utente comune). In alternativa può essere creato un utente

comune attraverso l'interazione con il componente creazione utente comune.

Interfaccia richiesta - Dati nuovo utente

Vengono richiesti i dati necessari per registrare un nuovo utente (Nome utente, email e password).

Interfaccia richiesta - Autorizzazione registrazione da Google

Aspetta la conferma di una corretta registrazione con Google quando questa viene scelta, inoltre riceve i dati necessari per creare l'account.

Interfaccia richiesta - Nuovo utente comune

Riceve un utente comune da registrare nel sistema.

Interfaccia fornita - Richiesta registrazione verso Google

Reindirizza l'utente alla pagina di registrazione di Google.

Interfaccia fornita - Autenticazione da registrato

Fornisce i dati necessari per verificare la corretta registrazione di un utente e i dati dell'utente necessari per identificarlo.

Interfaccia fornita - Nuovo utente

Distribuisce tutti i dati di un utente appena registrato.

CMP3. Gestione database

Descrizione

Questo componente gestisce le interazioni tra il sistema e il database, prendendo i dati dal sistema trasformandoli in dati adeguati per il database e viceversa.

Interfaccia richiesta - Aggiornamenti lista preferiti

Prende una lista di aggiornamenti dei preferiti di un utente e li salva nel database.

Interfaccia richiesta - Lista notifiche aggiornate

Richiede una nuova lista di notifiche di un utente da sostituire a quella presente nel database.

Interfaccia richiesta - Nuova segnalazione

Accetta una nuova segnalazione da aggiungere al database

Interfaccia richiesta - Nuove notifiche

Richiede una notifica e una lista di utenti e aggiunge la notifica ad ogni lista notifiche degli utenti specificati.

Interfaccia richiesta - Nuovo stato segnalazione

Prende una modifica dello stato di una segnalazione e la salva nel database.

Interfaccia richiesta - Dati segnalazione aggiornati

Accetta i nuovi dati di una segnalazione dopo un'interazione da parte di un utente e li salva. I dati che possono cambiare sono il numero di voti oppure un commento che è stato aggiunto/rimosso.

Interfaccia richiesta - Segnalazione da eliminare

Prende una segnalazione da eliminare. Quando questa viene ricevuta viene cancellata dal database e non è più visualizzabile.

Interfaccia fornita - Lista preferiti

Fornisce la lista dei preferiti salvata nel database di un utente loggato.

Interfaccia fornita - Lista segnalazioni

Fornisce la lista di tutte le segnalazioni presenti nel database.

Interfaccia fornita - Lista notifiche

Restituisce la lista di tutte le notifiche destinate ad un determinato utente che non sono ancora state eliminate.

Interfaccia fornita - Segnalazione

Rappresenta tutti i dati di una segnalazione richiesta.

Interfaccia fornita - Dati utente

Fornisce tutti i dati di un utente richiesto.

CMP4. Visualizzazione homepage

Descrizione

Questo componente gestisce l'interfaccia grafica che viene mostrata all'utente prendendo i dati dagli altri componenti.

Interfaccia richiesta - Lista preferiti utente

Richiede la lista dei preferiti di un utente. Viene richiesta solo se l'utente è loggato.

Interfaccia richiesta - Lista segnalazioni filtrate

Prende la lista di segnalazioni filtrate e ordinate secondo i criteri dell'utente.

Interfaccia richiesta - Lista notifiche utente

Accetta la lista di notifiche di un utente. Viene richiesta solo se l'utente è loggato.

CMP5. Gestione preferiti

Descrizione

Gestisce la lista dei preferiti di un utente aggiornandola mano a mano che un utente specifica dei filtri. Neecessita di un utente loggato per funzionare in modo tale da poter collegare le modifiche ad un account.

Interfaccia richiesta - Autenticazione

I dati necessari per identificare un utente nel database.

Interfaccia richiesta - Lista preferiti

Richiede la lista di preferiti di un utente

Interfaccia richiesta - Filtri utilizzati

La lista dei filtri che sono stati utilizzati da un utente. Serve per costruire la lista dei preferiti di un utente.

Interfaccia fornita - Lista preferiti utente

Fornisce la lista dei preferiti di un utente loggato

Interfaccia fornita - Aggiornamenti lista preferiti

Fornisce gli aggiornamenti da fare alla lista dei preferiti di un utente loggato rispetto all'ultima lista di preferiti dell'utente ottenuta.

CMP6. Filtro e ordinamento segnalazioni**Descrizione**

Prende una lista di segnalazioni e ci applica filtri e ordinamenti creando una nuova lista.

Interfaccia richiesta - Filtri

I filtri da applicare alla lista di segnalazioni. I filtri richiesti possono riguardare la categoria, tipo o posizione. Inoltre è possibile accettare una stringa per tenere solo le segnalazioni che contengono questa stringa nel loro contenuto.

Interfaccia richiesta - Tipo ordinamento

Accetta quale tipo di ordinamento è da applicare alla lista di segnalazioni. Gli ordinamenti possibili sono: per data, per numero di voti oppure per numero di interazioni.

Interfaccia richiesta - Lista segnalazioni

La lista a cui applicare filtri e ordinamenti.

Interfaccia fornita - Filtri utilizzati

La lista di filtri e ordinamenti utilizzati da un utente.

Interfaccia fornita - Lista segnalazioni filtrate

Restituisce la lista iniziale con i filtri e ordinamenti applicati ad essa.

CMP7. Creazione segnalazione

Descrizione

Si occupa di raccogliere i dati necessari per creare una nuova segnalazione e ci aggiunge i metadati richiesti (Per esempio la data).

Interfaccia richiesta - Autenticazione

I dati necessari per verificare che l'utente che richiede di creare una nuova segnalazione sia autenticato. Richiede anche i dati dell'utente necessari per creare la segnalazione.

Interfaccia richiesta - Dati segnalazione

Raccoglie i dati della segnalazione. In particolare i dati richiesti sono: titolo, posizione, immagine (opzionale), contenuto, tipo e categoria.

Interfaccia richiesta - Posizione Google Maps

Accetta la posizione fornita da Google Maps se questa viene richiesta.

Interfaccia fornita - Richiesta posizione Google Maps

Reindirizza l'utente verso Google Maps per selezionare una posizione.

Interfaccia fornita - Nuova segnalazione

Fornisce tutti i dati della segnalazione appena creata.

CMP8. Cambio stato segnalazione

Descrizione

Gestisce il cambio di stato di una segnalazione assicurandosi che l'utente sia autenticato come utente comune e che abbia inserito un commento.

Interfaccia richiesta - Autenticazione

Raccoglie i dati necessari per verificare che l'utente sia correttamente autenticato e che sia un utente comune.

Interfaccia richiesta - Nuovo stato

È il nuovo stato impostato dall'utente.

Interfaccia richiesta - Conferma commento

Prende i dati necessari per assicurarsi sia stato creato un commento sul cambio stato da parte dell'utente comune. Dopo aver ricevuto la conferma il cambio stato può essere confermato.

Interfaccia richiesta - Segnalazione

Prende la segnalazione a cui va cambiato lo stato.

Interfaccia fornita - Nuovo stato segnalazione

Fornisce lo stato aggiornato da salvare di una segnalazione.

Interfaccia fornita - Segnalazione aggiornata

Da tutti i dati della segnalazione a cui è stato cambiato lo stato.

Interfaccia fornita - Richiesta commento

Richiede l'inserimento obbligatorio di un commento per effettuare il cambio stato di una segnalazione.

CMP9. Creazione notifica

Descrizione

Crea la notifica dopo il cambio stato di una segnalazione e la aggiunge alle notifiche degli utenti che hanno votato la segnalazione.

Interfaccia richiesta - Segnalazione aggiornata

Prende i dati della segnalazione a seguito di un cambio di stato. In particolare i dati richiesti sono: titolo della segnalazione, nuovo stato della segnalazione e lista di utenti che hanno votato la segnalazione.

Interfaccia richiesta - Nuove notifiche

Fornisce una nuova notifica e la lista degli utenti che hanno votato la segnalazione.

CMP10. Gestione interazione segnalazione

Descrizione

Raccoglie le interazioni e aggiorna le segnalazioni. Ha delle interfacce che operano insieme alla componente di cambio stato per assicurarsi che i commenti del cambio stato vengano correttamente pubblicati.

Interfaccia richiesta - Autenticazione

Raccoglie i dati necessari ad assicurarsi che l'utente che richiede l'interazione sia correttamente autenticato.

Interfaccia richiesta - Voto

Raccoglie una eventuale aggiunta o rimozione di un voto ad una segnalazione da parte di un utente.

Interfaccia richiesta - Commento

Prende un commento scritto da un utente riguardo una segnalazione

Interfaccia richiesta - Segnalazione

Richiede la segnalazione con cui l'utente vuole interagire. Questa comprende anche i commenti relativi alla stessa.

Interfaccia richiesta - Richiesta commento

Aspetta una richiesta di commento obbligatorio.

Interfaccia fornita - Conferma commento

Specifica se un utente comune ha correttamente scritto un commento durante una procedura di cambio di stato.

Interfaccia fornita - Dati segnalazione aggiornati

Fornisce gli aggiornamenti sui dati di una segnalazione dopo una corretta interazione dell'utente. I cambiamenti sui dati possono essere: aggiunta/rimozione di un voto, aggiunta di un commento.

CMP11. Gestione notifiche**Descrizione**

Si occupa di manipolare la lista di notifiche di un utente. In particolare cancella le notifiche visualizzate e fornisce la lista quando necessario.

Interfaccia richiesta - Autenticazione

Raccoglie i dati necessari per verificare che l'utente sia correttamente autenticato e prende l'identificativo utente.

Interfaccia richiesta - Lista notifiche

Richiede la lista di tutte le notifiche di un utente.

Interfaccia fornita - Lista notifiche utente

Restituisce la lista attuale delle notifiche di un utente.

CMP12. Creazione utente comune**Descrizione**

Si occupa di ricevere i dati necessari per la creazione di un utente comune e di verificare che l'utente che ne richiede la creazione sia autorizzato (Sia un utente amministratore). La creazione vera e propria dell'account è delegata al componente di gestione registrazione.

Interfaccia richiesta - Autenticazione

Prende tutti i dati necessari per verificare che l'utente che richiede la creazione dell'account abbia le autorizzazioni richieste.

Interfaccia richiesta - Dati nuovo utente comune

Raccoglie i dati necessari per la creazione di un utente comune. I dati richiesti sono: nome utente, email e password.

Interfaccia fornita - Nuovo utente comune

Fornisce un utente da registrare nel sistema.

CMP13. Gestione condivisione**Descrizione**

Quando un utente vuole condividere una segnalazione questo componente si occupa di crearla. Indipendentemente dal metodo scelto nella condivisione devono essere presenti titolo e un pezzo del contenuto.

Interfaccia richiesta - Segnalazione

Richiede la segnalazione che l'utente vuole condividere.

Interfaccia richiesta - Metodo di condivisione

Richiede all'utente quale metodo di condivisione si vuole usare (Per esempio: whatsapp o facebook).

Interfaccia fornita - Condivisione

Fornisce la condivisione con formato diverso in base al metodo richiesto dall'utente.

CMP14. Moderazione commenti e segnalazioni**Descrizione**

Permette agli utenti autenticati come utente comune di eliminare una segnalazione o il commento contenuto in una segnalazione.

Interfaccia richiesta - Autenticazione

Riceve i dati necessari per verificare che l'utente sia un utente comune.

Interfaccia richiesta - Segnalazione da eliminare

Riceve la segnalazione che l'utente vuole eliminare

Interfaccia richiesta - Commento da eliminare

Riceve il commento che l'utente vuole eliminare

Interfaccia richiesta - Segnalazione

Fa riferimento alla segnalazione che si vuole eliminare o alla segnalazione che contiene il commento da eliminare.

Interfaccia fornita - Dati segnalazione aggiornati

In caso di rimozione di un commento viene restituita la segnalazione togliendo il commento rimosso.

Interfaccia fornita - Segnalazione da eliminare

In caso di rimozione di una segnalazione viene fornita la segnalazione da rimuovere.

CMP15. Visualizzazione pagina utente

Descrizione

Mostra la pagina utente, raccogliendo i dati dell'utente. Inoltre raccoglie tutte le segnalazioni dal database e ne seleziona quelle create dall'utente e quelle con cui l'utente ha interagito.

Interfaccia richiesta - Lista segnalazioni

Richiede la lista di tutte le segnalazioni nel database. In questa lista vengono cercate tutte le segnalazioni create dall'utente o con cui esso ha interagito.

Interfaccia richiesta - Dati utente

Prende i dati dell'utente dal database (Per esempio email e nome utente).

Interfaccia richiesta - Autenticazione

Prende i dati necessari per verificare qual utente è correntemente autenticato e dunque quale la pagina utente a chi si riferisce.

CMP16. Visualizzazione pagina di analisi

Descrizione

Mostra all'utente la pagina di analisi in cui vengono mostrate tutte le statistiche sulle segnalazioni raccolte. Solo agli utenti comune è consentito accedere alla pagina.

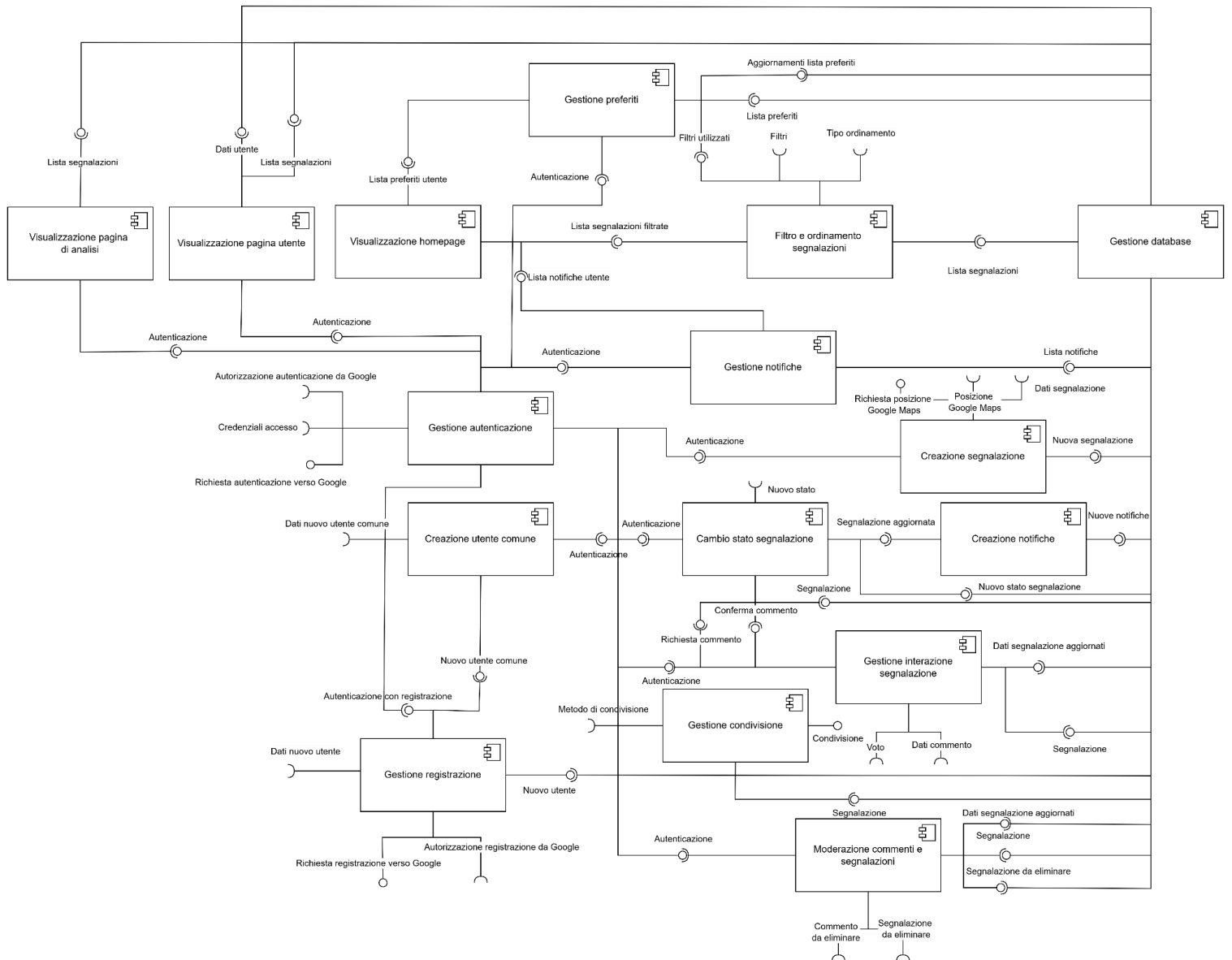
Interfaccia richiesta - Autenticazione

Prende tutti i dati necessari per verificare che l'utente che vuole visualizzare la pagina sia un utente comune.

Interfaccia richiesta - Lista segnalazioni

Prende la lista di tutte le segnalazioni. Utilizza questa lista per creare le statistiche da mostrare.

b. Diagramma dei componenti

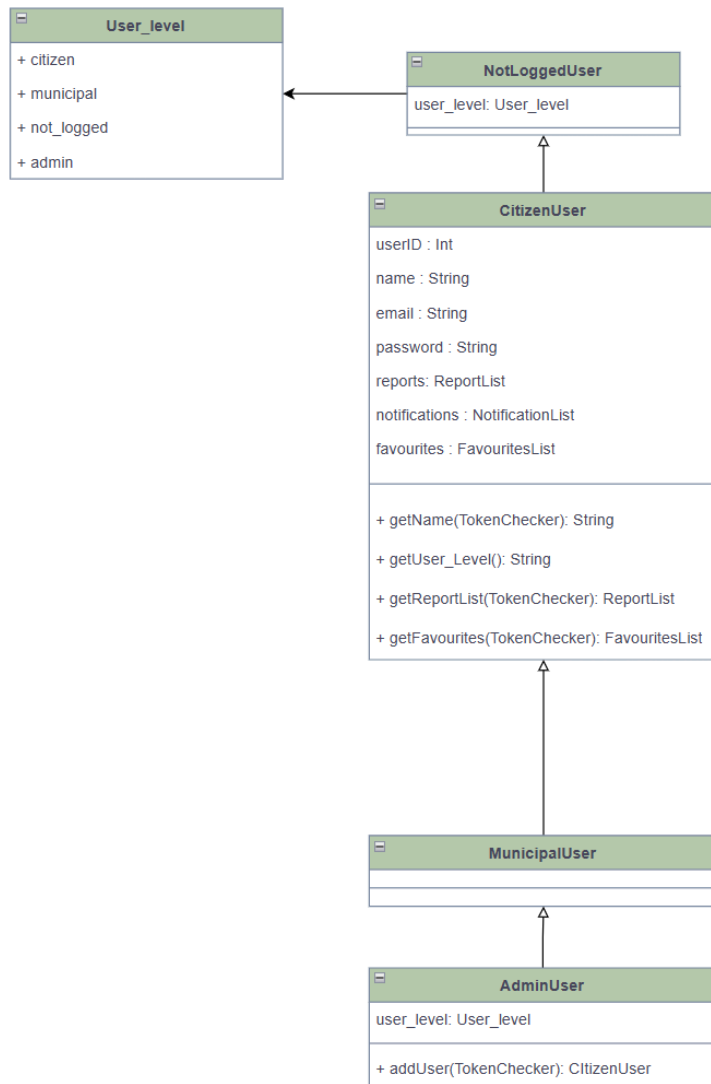


3. Diagramma delle classi

In questa sezione sono presentate le classi previste nell'ambito del progetto Treport. Le classi sono derivate dai componenti del component diagram. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi che identificano i dati gestiti dalla classe e una lista di metodi che definiscono le operazioni previste all'interno della classe. Ogni classe può essere anche associata ad altre classi e, tramite questa associazione, è possibile fornire informazioni su come le classi si relazionano tra loro. Per ogni gruppo di classi è stata fornita una descrizione e l'immagine della sezione del class diagram.

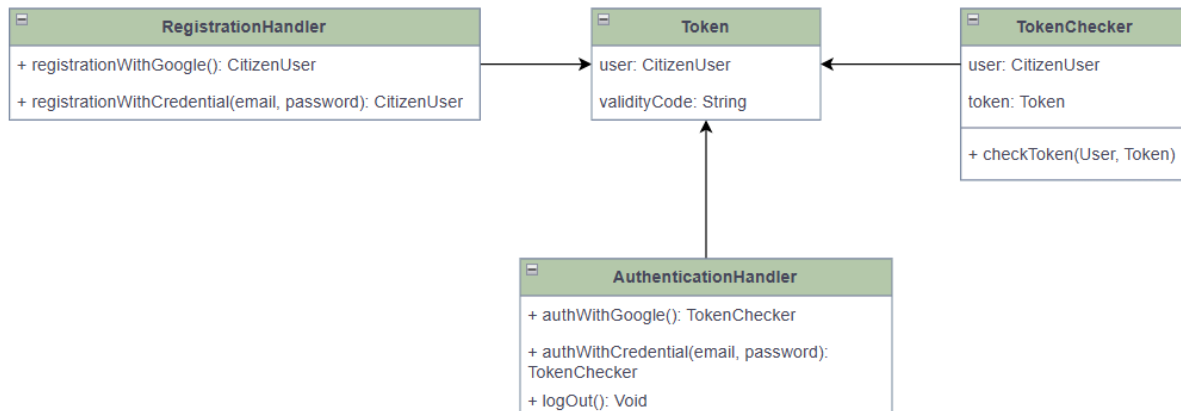
NotLoggedUser, CitizenUser, MunicipalUser, AdminUser, User_level

Molti degli attori individuati nel class diagram sono stati rappresentati da una gerarchia di classi, che comprende NotLoggedUser, CitizenUser, MunicipalUser e AdminUser. L'utente cittadino e l'utente comune non differiscono per i metodi definiti nelle loro classi, i quali comprendono principalmente la gestione dei propri dati. L'utente amministratore è l'unico che può utilizzare il metodo addUser() per creare utenti comune, mentre l'utente non loggato può solo utilizzare il metodo getUserLevel(). Le classi sono gestite in questo modo: NotLoggedUser è superclasse di CitizenUser, che è superclasse di MunicipalUser, e quest'ultima è a sua volta superclasse di AdminUser.



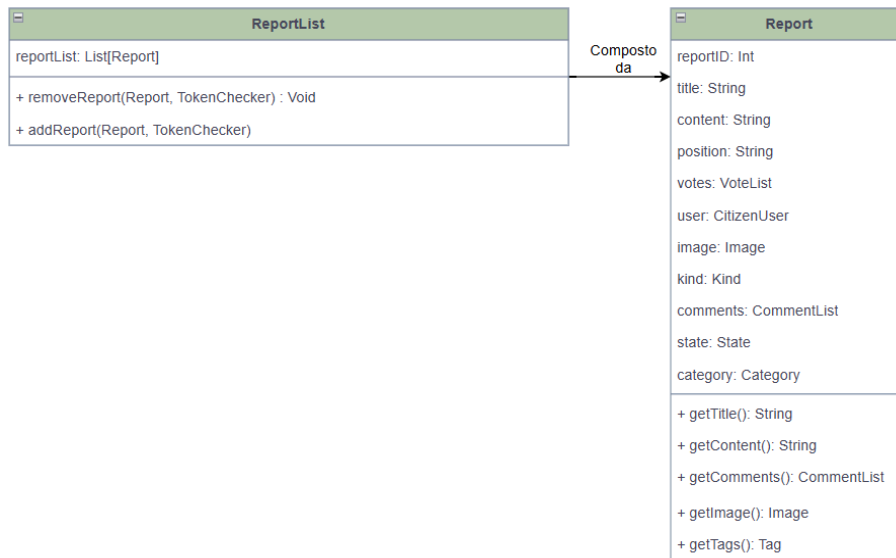
AuthenticationHandler, RegistrationHandler, Token, TokenChecker

La registrazione e l'autenticazione degli utenti sono delegate alle classi `RegistrationHandler` e `AuthenticationHandler`. `RegistrationHandler` permette di registrare un utente sia tramite GoogleSSO sia tramite inserimento di credenziali. `AuthenticationHandler` permette di accedere tramite autenticazione Google o inserimento di credenziali, permette di effettuare il log out e verifica l'accesso di un utente tramite un token che verrà controllato dall'apposita classe `TokenChecker`, la quale verifica la validità del token e la corretta associazione ad un utente.



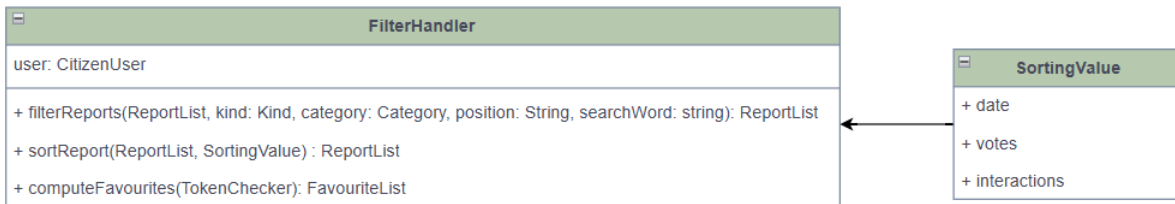
Report, ReportList

La gestione delle segnalazioni è affidata alle classi **Report** e **ReportList**. La classe **Report** definisce le caratteristiche di una segnalazione e ne permette il reperimento. La classe **ReportList** viene invece utilizzata come database di segnalazioni e ne permette l'aggiunta e l'eliminazione basata sulla validità dell'accesso tramite il **TokenChecker**.



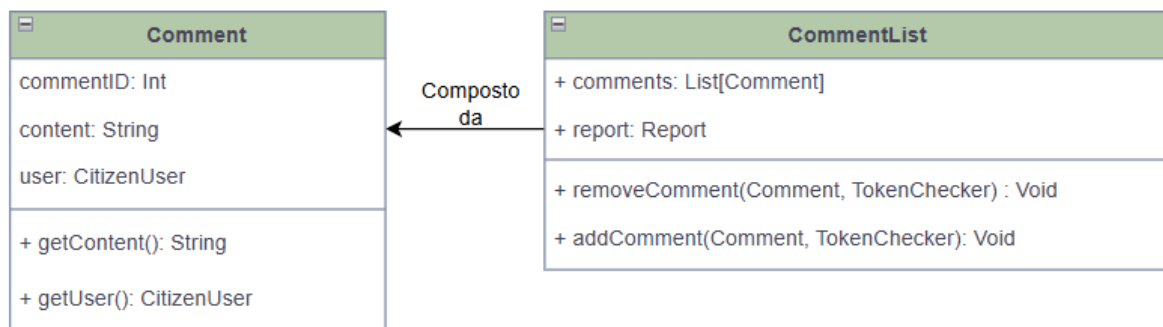
Filter Handler, Sorting Value

La classe **FilterHandler** viene utilizzata per gestire il filtraggio e l'ordinamento delle segnalazioni fornendo i metodi appositi, e appoggiandosi alla classe dell'enumerativo **SortingValue** che raggruppa tutti i possibili elementi per cui le segnalazioni possono essere ordinate. Gli ultimi filtri scelti dall'utente vengono poi aggiunti ai preferiti.



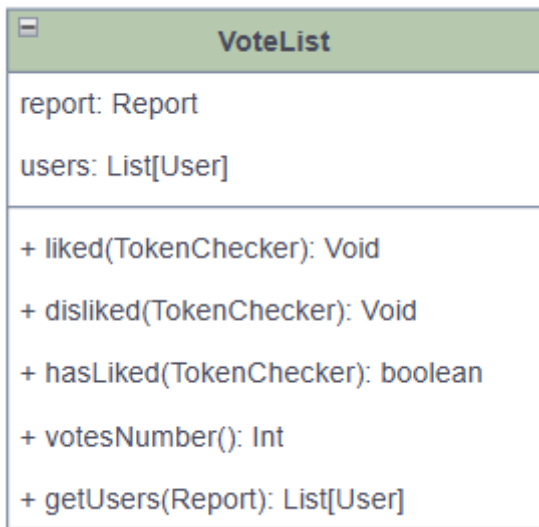
Comment, CommentList

La sezione dei commenti viene diretta dalle classi Comment e CommentList. Come per le segnalazioni in precedenza, la classe Comment definisce la struttura dei commenti e ne permette l'ottenimento del contenuto e dell'autore, mentre la classe CommentList costituisce il database di commenti relativi ad ogni segnalazione e ne permette l'aggiunta e la rimozione se autorizzati tramite TokenChecker.



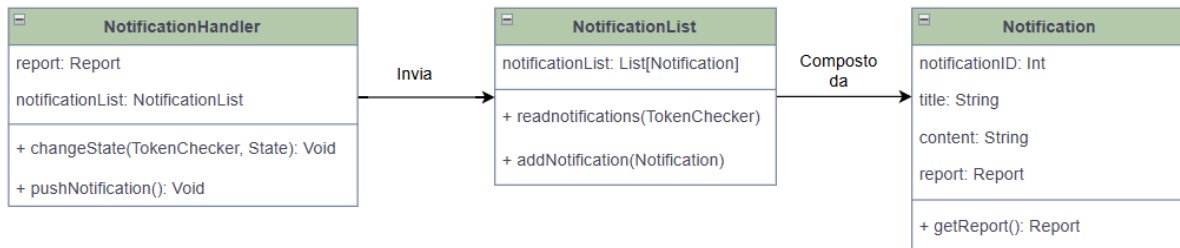
VoteList

La classe VoteList racchiude tutti gli utenti che hanno votato una determinata segnalazione, e permette di votare o togliere il voto se autorizzati (aggiungendo o togliendo l'utente dalla lista).



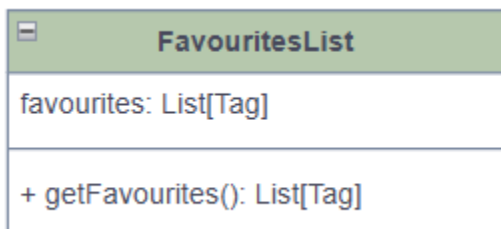
Notification, NotificationList, NotificationHandler

La classe `Notification` definisce la struttura di una notifica e permette di ottenere la segnalazione a cui la notifica si riferisce. `NotificationList`, come per `ReportList` e `CommentList` costituisce l'insieme di tutte le notifiche (non relative ad uno specifico utente), e i suoi metodi permettono di aggiungere notifiche o leggerle se autorizzati. L'invio delle notifiche vero e proprio è controllato dalla classe `NotificationHandler`, che gestisce sia il cambio di stato di una segnalazione che l'invio automatico della notifica riguardante il cambio di stato a tutti gli utenti che hanno votato la segnalazione (ottenuti direttamente tramite l'attributo "votes" di `Report`).



FavouritesList

La classe `FavouritesList` contiene gli ultimi filtri utilizzati dall'utente.



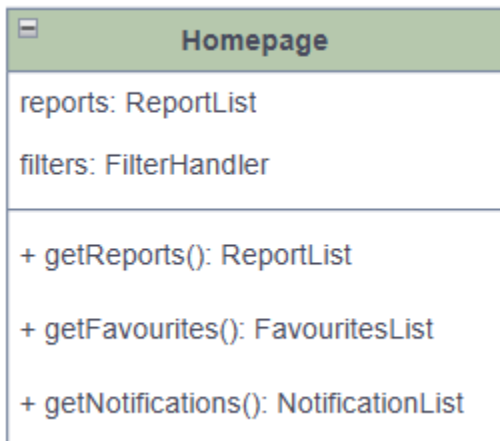
UserPage

La classe UserPage raccoglie informazioni sulle interazioni dell'utente, come una raccolta delle segnalazioni pubblicate, una delle segnalazioni votate, e le informazioni personali dell'utente.



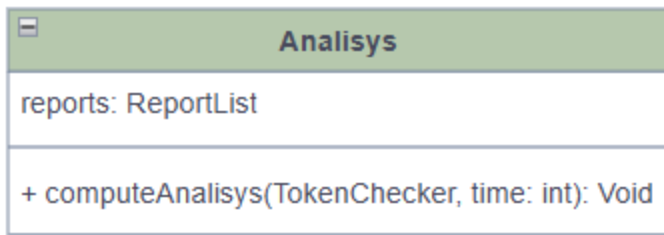
Homepage

La classe Homepage raccoglie e gestisce tutte le informazioni visibili nella homepage della webapp, come la lista di segnalazioni, i preferiti e le notifiche (le ultime due sono ottenute direttamente dall'attributo "user").



Analisis

La classe Analisis si occupa di effettuare alcune operazioni di analisi dati che verranno richieste dall'utente comune a partire da un insieme di segnalazioni.

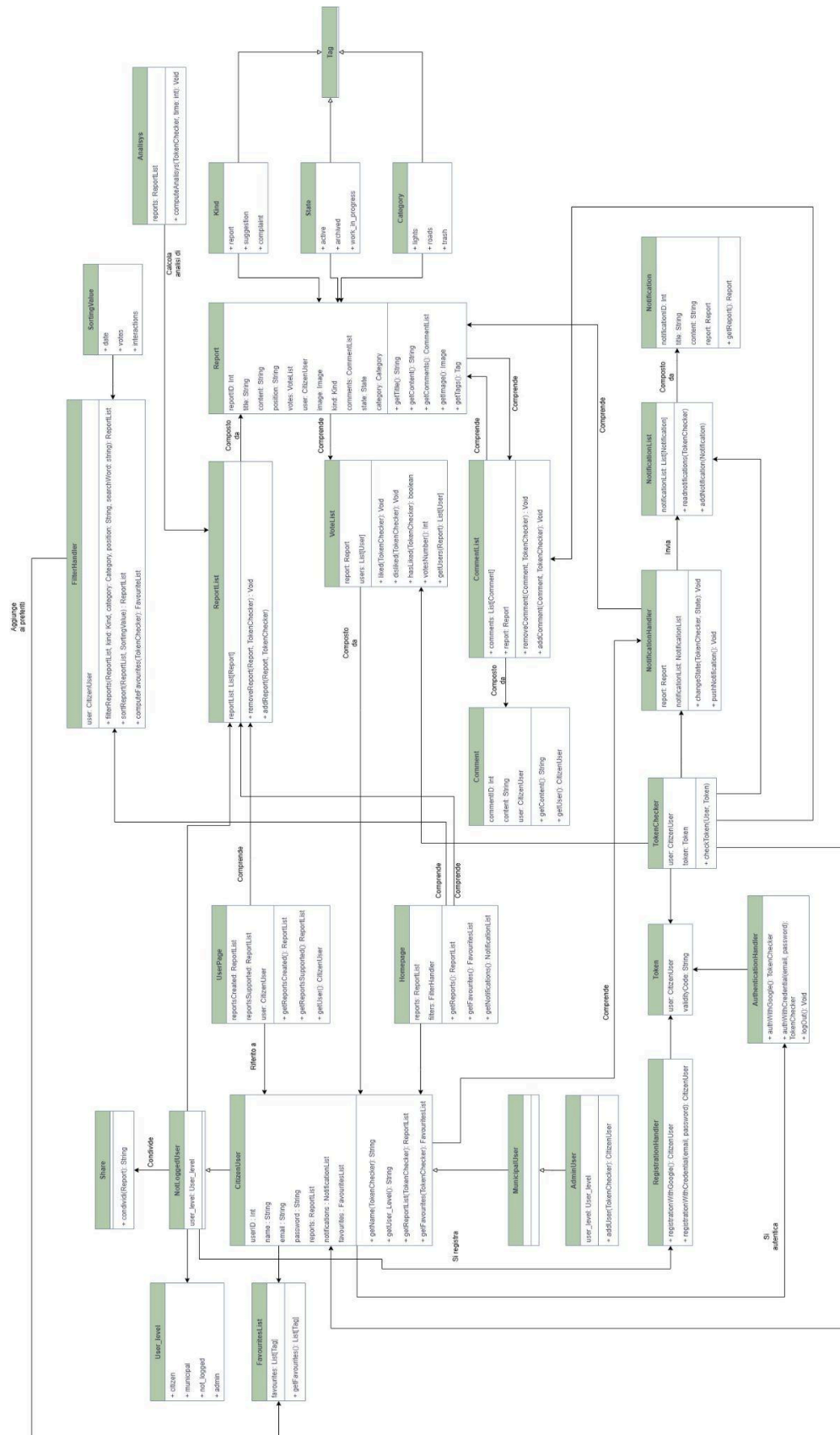


Share

La classe Share gestisce la condivisione delle segnalazioni, permettendo di inviare il link per una segnalazione anche senza aver effettuato l'accesso.



Diagramma complessivo



4. Dal class diagram alle API

In questa sezione è presentata una tabella che vuole spiegare il passaggio dal class diagram alle API. Di seguito alcune precisazioni riguardo lo schema.

In generale la classe TokenChecker è strettamente collegata all'autenticazione, quindi, almeno che non sia specificato diversamente, per ogni metodo che richiede un TokenChecker la richiesta HTTP deve contenere il token di autenticazione JWT nell'header.

I preferiti non sono stati implementati nel backend e neanche nelle API, quindi nella tabella sono riportati solo dei valori ipotetici di messaggi HTTP.

Nella tabella sono riportati solo le risposte in caso di successo, ma per ogni chiamata HTTP sono previsti codici di errore, che però non forniscono informazioni aggiuntive e quindi non vale la pena riportare.

L'autenticazione e la registrazione con Google non sono state implementate.

Classe	Metodo	HTTP method	URI + query params	Request	Response	Note
CitizenUser	getName(TokenChecker): String	GET	/api/users/:id		status: 200 body: { "i"": 0, "name": "string", "email": "string", "user_level": "citizen", "reports": ["string"] }	
	getUser_Level(): String					
	getReportList(TokenChecker): ReportList	GET	/api/reports?user=userId		status: 200 [{ "i"": 0, "title": "string", "content": "string", "user": "string", "votes": 0, "position": "string", "kind": "report", "category": "lights", "state": "active", "comments": ["string"] }]	
	getFavourites(TokenChecker): FavouritesList					Utilizza il motodo getFavourites della FavouritesList
	AdminUser	addUser(TokenChecker): User	POST	/api/users	{ "name": "string", "email": "string", "user_level": "municipal", "password": "string" }	status: 200 headers: { "location": "string" }
RegistrationHandler	registrationWithGoogle(): User					
	registrationWithCredential(email, password): User	POST	/api/users	{ "name": "string", "email": "string", "user_level": "citizen", "password": "string" }	status: 200 headers: { "location": "string" }	A differenza del AdminUser può creare solo utenti cittadino e non richiede un token (o l'autenticazione)
AuthenticationHandler	authWithGoogle(): TokenChecker					
	authWithCredential(email, password): TokenChecker	POST	/api/authentication	{ "email": "string", "password": "string" }	status: 200 { "token": "string" }	Ritorna un token che certifica che l'utente si è autenticato
	logOut(): Void					Gestito in automatico dal frontend cancellando il token
TokenChecker	checkToken(User, Token)					Controlla che il token fornito sia valido. Viene fatto in automatico nel backend ogni volta che è richiesta una autorizzazione basata sui token
CondivisioneHandler	condividi(Report): String					Gestito tramite il frontend
ReportList	removeReport(Report, TokenChecker) : Void	DELETE	/api/reports/:reportId		status: 204	Richiede un token per verificare che l'utente sia un utente comune
	addReport(Report, TokenChecker)	POST	/api/reports	{ "title": "string", "content": "string", "position": "string", "kind": "report", "category": "lights", "state": "active" }	status:201 headers: { "location": "string" }	Solo agli utenti loggati è permesso aggiungere segnalazioni, richiede quindi un token
			POST	/api/reports/:reportId/image	image/*	status:201 headers: { "location": "string" }

Classe	Metodo	HTTP method	URI + query params	Request	Response	Note
Report	getTitle(): String	GET	/api/reports		<pre>status: 200 { "id": 0, "title": "string", "content": "string", "user": "string", "votes": 0, "position": "string", "kind": "report", "category": "lights", "state": "active", "comments": ["string"] }</pre>	
	getContent(): String					
	getTags(): Tag					
	getComments(): CommentList	GET	/api/reports/:reportId/comments		<pre>status: 200 [{ "id": 0, "content": "string", "user": "string" }]</pre>	
	getImage(): Image	GET	/api/reports/:reportId/image		<pre>status: 200 image/*</pre>	
VoteList	liked(TokenChecker): Void	PUT	/api/reports/:reportId/votes	{ "liked": "true" }	0	Aggiunge un voto alla segnalazione con id: reportId. Ritorna il numero di voti totali della segnalazione.
	disliked(TokenChecker): Void	PUT	/api/reports/:reportId/votes	{ "liked": "false" }	0	Aggiunge un voto alla segnalazione con id: reportId. Ritorna il numero di voti totali della segnalazione.
	hasLiked(TokenChecker): boolean	GET	/api/reports/:reportId/votes		{ "hasLiked": true }	Viene utilizzato un token in modo che un utente non possa vedere cosa hanno votato altri utenti.
	votesNumber(): Int	GET	/api/reports/:reportId		<pre>status: 200 { "id": 0, "title": "string", "content": "string", "user": "string", "votes": 0, "position": "string", "kind": "report", "category": "lights", "state": "active", "comments": ["string"] }</pre>	
	getUsers(Report): List<User>					Non esiste un API per questo metodo perché questa non è un'informazione che dovrebbe essere pubblica
CommentList	removeComment(Comment, TokenChecker): Void	DELETE	/api/reports/:reportId/comments/:commentId		status: 204	
	addComment(Comment, TokenChecker): Void	POST	/api/reports/:reportId/comments/:commentId	{ "content": "string" }	<pre>status: 201 headers: { "location": "string" }</pre>	
Comment	getContent(): String	GET	/api/reports/:reportId/comments/:commentId		<pre>{ "id": 0, "content": "string", "user": "string" }</pre>	
NotificationList	readNotifications(TokenChecker)	GET	/api/users/:userId/notifications		<pre>[{ "id": 0, "title": "string", "content": "string", "report": "string" }]</pre>	Richiede un token checker perché un utente può leggere solo le sue notifiche, quindi va effettuato un controllo.
	addNotification(Notification)					Viene gestito in automatico dal backend dopo il push di una notifica

Classe	Metodo	HTTP method	URI + query params	Request	Response	Note
NotificationHandler	changeState(TokenChecker, State): Void	PUT	/api/reports/:reportId	{ "state": "active" }	status: 200	
	pushNotification(): Void					La notifica viene automaticamente inviata dal backend quando avviene un cambio di stato
Notification	getReport(): Report	GET	/api/users/: userId/notifications/: notificationId		{ "id": 0, "title": "string", "content": "string", "report": "string", }	In questo caso il campo report si riferisce ad un identificativo di una segnalazione. In caso si desideri leggere l'intero report sarebbe necessario usare il metodo GET /reports/:reportId, in cui reportId è l'identificativo ottenuto con questa chiamata
FilterHandler	filterReports(ReportList, kind: Kind, category: Category, position: String, searchWord: string): ReportList	GET	/api/reports?state={state} &kind={kind}&category={category}&position={position}		[{ "id": 0, "title": "string", "content": "string", "user": "string", "votes": 0, "position": "string", "kind": "report", "category": "lights", "state": "active", "comments": ["string"] }]	Vengono mandati al backend una lista di filtri e viene ritornata la lista già filtrata
	sortReport(ReportList, SortingValue) : ReportList					L'ordinamento avviene nel frontend, con le segnalazioni già ricevute dal backend.
	computeFavourites(TokenChecker): FavouriteList	PUT	/api/users/:userId/favourites	{ "type": "string", "value": "string" }		
Analysis	computeAnalysis(TokenChecker, time: int): Void					Le statistiche vengono generate nel frontend prendendo tutte le segnalazioni disponibili nel database
FavouritesList	getFavourites(): List[Tag]	GET	/api/users/:id/favourites		status: 200 [{ "type": "string", "value": "string" }]	
ProfiloUtente	getReportsCreated(): ReportList					E' equivalente al metodo getReportList (TokenChecker): ReportList di CitizenUser
	getReportsSupported(): ReportList					Questo metodo non è implementato nelle API
	getUser(): User					Si riferisce all'eventuale utente salvato nel frontend della sessione corrente
Homepage	getReports(): ReportList					Utilizza il metodo filterReports di filterHandler
	getFavourites(): FavouriteList					Usa il metodo getFavourites dell'utente
	getNotifications(): NotificationList					E' equivalente al metodo readNotifications di NotificationHandler se esiste una sessione collegata ad un utente