

Treport

Ingegneria del Software A.S. 2024 - 2025

Tommaso Ascolani, Riccardo D'Incau, Marco Zanatta

Documento di sviluppo

.

Indice

Indice.....	2
1. User stories.....	3
2. User Flow.....	8
3. Web APIs.....	9
4. Implementation.....	10
a. Repository Organization.....	10
b. Branching strategy e organizzazione del lavoro.....	10
c. Dependencies.....	11
d. Database.....	11
e. Testing.....	12
5. FrontEnd.....	13
6. Deployment.....	14

1. User stories

In questa sezione sono presentate le user stories estratte dai requisiti funzionali che abbiamo ritenuto dovessero essere implementati. Per chiarezza viene fornita una lista dei requisiti funzionali non implementati. Questi non sono stati implementati per permetterci di concentrarci sui requisiti fondamentali al funzionamento della web app o eventualmente per mancanza di conoscenze tecniche. I requisiti non implementati sono i seguenti: **RF5 Condivisione segnalazione**, **RF13 Ordinamento passivo segnalazioni**, **RF15 Sezione dei preferiti**(Solo implementazione grafica), **RF16 Analisi segnalazioni**, **RF17 Creazione account comune e RF18 Moderazione commenti e segnalazioni**.

Inoltre i seguenti requisiti funzionali non sono stati interamente implementati: **RF1 Login** (Non implementata: autenticazione Google), **RF2 Registrazione** (Non implementata: registrazione Google), **RF6 Creazione segnalazione** (Non implementato: caricamento immagini, raccolta posizione tramite GPS) e **RF12 Filtro e riordinamento segnalazioni** (Implementato solo ordinamento per numero di voti).

User story 1: Come utente quando apro il sito voglio vedere delle preview di segnalazioni in modo da poter iniziare la navigazione anche senza essere loggato.

Task:

1. Creare la sezione segnalazioni scorrevole
2. Sistema di raccolta delle segnalazioni
3. Mostrarle secondo l'ordine stabilito dall'algoritmo

Test:

- All'accesso sono mostrate le segnalazioni anche senza essere loggati
- È possibile scorrere la sezione segnalazioni

Stima: m

User story 2: Come utente voglio scegliere se effettuare la registrazione tramite credenziali, o tramite autenticazione google, in modo da poter usufruire delle funzionalità che richiedono il login.

Task:

1. Interfaccia che contiene: campo per il nome utente, campo e-mail, campo password e bottone di conferma
2. Bottone per l'autenticazione tramite google
3. Sistema di salvataggio dati
4. Autenticazione "automatica" al sistema
5. Reindirizzamento alla pagina precedente

Test:

- Posso inserire nome utente, e-mail e password.
- Posso scegliere di registrarsi con autenticazione google
- Il sistema salva i miei dati
- Vengo reindirizzato correttamente (loggato)

Stima: s

User story 3: Come utente voglio effettuare il login tramite credenziali, o tramite autenticazione google, in modo da poter accedere alle risorse riservate agli utenti loggati.

Task:

1. Interfaccia che contiene: campo per l'e-mail, campo password e bottone di conferma
2. Bottone per l'autenticazione tramite google
3. Verifica di autenticazione al sistema
4. Reindirizzamento alla pagina precedente

Test:

- Posso inserire e-mail e password
- Posso scegliere se autenticarmi con google
- Se le credenziali di accesso sono corrette l'utente è autenticato
- Vengo reindirizzato correttamente (loggato)

Stima: m

User story 4: Come utente loggato voglio poter creare una nuova segnalazione da una pagina apposita in modo da poter comunicare al comune un problema che ho riscontrato.

Task:

1. Creazione pagina nuova segnalazione
2. Definizione layout dei campi di inserimento
3. Raccolta dei dati dai campi di inserimento
4. Sistema di salvataggio delle segnalazione creata

Test:

- Si può accedere alla pagina nuova segnalazione
- Si può correttamente inserire i dati nei campi di inserimento
- La segnalazione viene automaticamente salvata

Stima: l

User story 5: Come utente-cittadino voglio poter votare una segnalazione in modo da supportarla.

Task:

5. Creazione pulsante voto
6. Aggiornamento voti segnalazione dopo aver premuto il pulsante voto
7. Aggiornamento layout (colore)

Test:

- Sono visibili i pulsanti voto

- Il numero di voti viene correttamente salvato dopo aver premuto il pulsante voto
- Il layout è correttamente aggiornato

Stima: xs

User story 6: Come utente-cittadino voglio poter vedere le segnalazioni a seconda del loro stato e averne un accesso immediato, in modo da poter analizzare le segnalazioni già prese in carico, attive o archiviate.

Task:

1. Creazione sezione “Attive”, “Prese in carico” e “Archivate”
2. Implementazione sistema che, a seconda dello stato selezionato, permetta la visione solo delle segnalazioni in quello stato
3. Creazione bottoni “Attive”, “Prese in carico” e “archivate”

Test:

- I pulsanti relativi allo stato sono sempre visibili e cliccabili all'utente
- Quando cliccati, questi permettono la visione solo delle segnalazioni in quello stato
- Il bottone della bacheca corrente viene selezionato

Stima: s

User story 7: Come utente-comune voglio poter interagire con le segnalazioni commentandole e cambiandone lo stato, in modo da comunicare con l'utente-cittadino e tenerlo aggiornato.

Task:

1. Creazione tasto cambia-stato
2. Campo per inserire il commento (obbligatorio quando si cambia lo stato della segnalazione)
3. Evidenziazione del commento dell'utente-comune
4. Bottone invio
5. Memorizzazione del commento
6. Aggiornamento e memorizzazione del nuovo stato della segnalazione

Test:

- Quando l'utente-comune clicca sul bottone cambia stato può scegliere il nuovo stato della segnalazione
- Il cambio di stato di una segnalazione obbliga a commentare (tempistiche, motivazioni...)
- Quando il bottone invio viene premuto il commento viene memorizzato ed evidenziato e lo stato aggiornato
- La segnalazione con lo stato modificato viene eliminata dalla bacheca precedente e aggiunta a quella relativa al nuovo stato

Stima: m

User story 8: Come utente voglio poter visualizzare una segnalazione in una pagina dedicata per poterla vedere interamente e vedere la sezione commenti.

Task:

1. Creazione sistema di reindirizzamento quando si clicca su una preview di una segnalazione
2. Aggiunta pagina segnalazione ingrandita
3. Disposizione layout dati segnalazione
4. Sistema di raccolta dati segnalazione

Test:

- La pagina segnalazione ingrandita è correttamente raggiungibile cliccando su una preview
- Il layout è correttamente visualizzato e tutte le informazioni sono visibili
- I dati della segnalazione vengono correttamente raccolti e mostrati

Stima: m

User story 9: Come utente-cittadino voglio poter visualizzare e pubblicare commenti in modo da poter esprimere la mia opinione riguardo una segnalazione.

Task:

1. Creazione sezione commenti nella pagina segnalazione ingrandita
2. Selezione dei commenti relativi alla segnalazione
3. Campo per inserire contenuto commento
4. Controllo moderazione commento
5. Bottone di invio e memorizzazione commento

Test:

- Sono visibili tutti i commenti relativi ad una segnalazione
- Posso inserire il contenuto di un commento
- Un commento inappropriato viene bloccato
- Quando il bottone di invio è premuto il commento è memorizzato

Stima: l

User story 10: Come utente loggato voglio ricevere le notifiche di un cambio stato delle segnalazioni che ho supportato in una sezione apposita, in modo da poter rimanere aggiornato sulle segnalazioni che mi interessano.

Task:

1. Creazione layout della sezione notifiche nella homepage
2. Sistema di gestione delle notifiche per ogni utente
3. Procedura di generazione notifiche
4. Sistema di invio notifiche
5. Integrazione del sistema di invio notifiche nella procedura di cambio di stato
6. Raccolta delle notifiche per ogni utente loggato

Test:

- Corretta visualizzazione della bacheca notifiche
- L'utente può eliminare le notifiche dalla sezione notifiche
- Le notifiche vengono generate correttamente in base alla segnalazione di riferimento e all'azione di cambio stato
- Le notifiche vengono inviate solo agli utenti che hanno votato una segnalazione a cui è stato cambiato lo stato

- Ogni utente visualizza correttamente soltanto le sue notifiche

Stima: l

User story 11: Come utente nella homepage voglio poter filtrare ed ordinare le preview delle segnalazioni disponibili in modo da vedere solo quelle che mi interessano.

Task:

1. Disposizione, nella homepage, del layout della barra di ricerca e dei menù a discesa per la scelta del filtro e dell'ordinamento
2. Implementazione del menu di scelta per filtri e ordinamento
3. Implementazione di richiesta della posizione a Google Maps per il filtro per posizione
4. Aggiornare il sistema di raccolta delle segnalazioni da mostrare nella homepage in modo che supporti la ricerca per parola chiave e l'applicazione di eventuali filtri
5. Riordinamento delle segnalazioni nella homepage in base all'ordinamento richiesto

Test:

- La barra di ricerca e i menù per la scelta dei filtri e dell'ordinamento sono correttamente visualizzati
- Il sistema raccoglie correttamente i filtri e l'ordinamento che l'utente vuole applicare
- Vengono mostrate nella homepage solo le segnalazioni che rispettano i filtri scelti
- Le segnalazioni nella homepage sono ordinate secondo l'ordinamento scelto

Stima: s

User story 12: Come utente loggato voglio poter accedere alla pagina utente in modo da potere visualizzare tutte le informazioni sul mio account.

Task:

1. Creazione della nuova pagina utente
2. Aggiunta del bottone per accedere alla pagina utente nella homepage
3. Disposizione layout della pagina
4. Sistema di raccolta dati dell'utente
5. Integrazione dei dati raccolti nella pagina

Test:

- Si può accedere alla pagina utente dalla homepage se si è un utente loggato
- La pagina è correttamente visualizzata
- I dati dell'utente sono correttamente raccolti

Stima: s

User story 13: Come utente amministratore voglio poter creare nuovi account "utente comune" per eventuali nuovi operatori del comune.

Task:

1. Creazione nuova pagina "creazione utente" disponibile solo per gli utenti amministratore

2. Disposizione del layout del bottone di conferma e dei campi di inserimento per: la mail, il nome utente e la password.
3. Sistema per l'aggiunta dei nuovi utenti comune

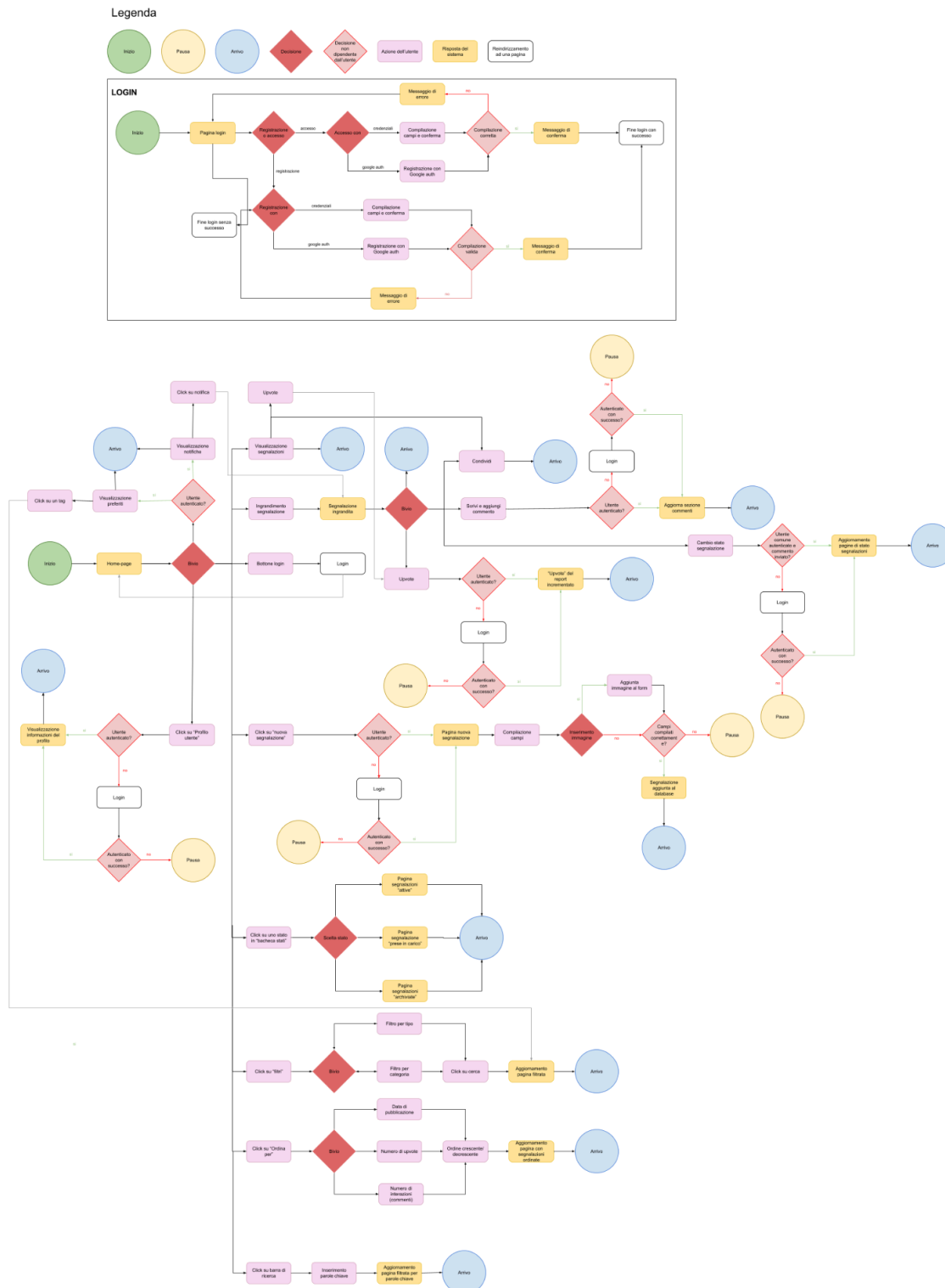
Test:

- La pagina è visualizzabile solo dagli utenti amministratore
- I campi per l'inserimento dei dati raccolgono correttamente le informazioni
- I dati dei nuovi utenti vengono correttamente salvati
- Solo gli utenti amministratore possono salvare nuovi utente comune

Stima: m

2. User Flow

In questa sezione del documento è riportato lo user flow che descrive le azioni possibili per un utente all'interno della web app. La legenda descrive i simboli utilizzati.



3. Web APIs

Le API sono state documentate secondo lo standard openapi3 e sono visibili al link:

https://app.swaggerhub.com/apis-docs/RICCARDODINCAU_1/TreportAPI/v.1.0.5

Le API sono state implementate attraverso il paradigma RESTful API, in particolare sono state individuate 2 risorse principali:

/users. Permette di visualizzare i dati degli utenti. Un utente specifico ha una sotto risorsa: /notifications

/reports. Permette di ottenere le informazioni sulle segnalazioni. Una segnalazione specifica ha tre sotto risorse: /comments, /votes e /image.

È inoltre fornito POST **/authentication** per effettuare l'autenticazione.

Le chiamate che richiedono l'autenticazione (Quindi un JWT) hanno specificato authToken nella sezione security della documentazione (Questo non è visibile da Swagger, ma si può vedere nel file YAML).

La documentazione è disponibile in GitHub nel branch "swagger".

4. Implementation

L'applicazione è stata sviluppata utilizzando le seguenti tecnologie: NodeJS per lo sviluppo del backend, VueJS per lo sviluppo del frontend e MongoDB come database. Queste scelte derivano dagli argomenti e tecnologie affrontate durante il corso.

a. Repository Organization

Il codice del progetto (https://github.com/RiccardoDincau/Progetto_IdS) è organizzato secondo la seguente struttura:

- /app *File backend*
 - /errors *Codice gestione errori*
 - /models *Modelli delle risorse mongoose*
 - app.js *Definizione app express*
 - *.js *Definizione routers express*
- /scripts *Script per codici di setup del backend*
- /vue-project *File frontend (git workspace)*
 - /src
 - /assets *Risorse utilizzate nel frontend (Immagini, CSS, ecc.)*
 - /components *Componenti vue*
 - /pages *Componenti vue che formano le pagine*
 - App.vue *Gestione della pagina da visualizzare*
 - main.js *Chiamata al codice vue*
- index.js *Ingresso backend*
- package.json *Pacchetto npm*

b. Branching strategy e organizzazione del lavoro

Ogni membro del gruppo ha partecipato attivamente in ogni parte del progetto, infatti tutti abbiamo almeno un commit in ogni branch (Swagger, NodeJS, frontend e deploy) del repository. Spesso i commit sono stati fatti da un dispositivo, ma in presenza di tutti i membri.

I branch sono stati così gestiti:

- swagger: in questo branch è presente soltanto la documentazione openapi3 delle nostre API scritta in YAML.
- nodeJS: qui è presente il nostro backend e le suite di test
- frontend: questo è il branch dedicato al frontend, in particolare il frontend è contenuto in una cartella "vue-project" che è stata impostata come git workspace
- deploy: qua è dove avviene il merge tra nodeJS e frontend, questo branch è stato quello che viene automaticamente deployato
- main: in questo branch vengono mergati tutti gli altri, per fornire una panoramica sul progetto

Inoltre esiste un branch per ogni deliverable dove vengono caricati i documenti relativi allo stesso.

c. Dependencies

Il progetto npm del backend si basa sui seguenti moduli esterni:

- cors → supporto nelle chiamate CORS
- express → utilizzato per sviluppare il backend
- jsonwebtoken → utilizzato per autenticazione e autorizzazione
- mongoose → comunicazione con il database

Inoltre il progetto utilizza i seguenti moduli per lo sviluppo:

- dotenv → gestione variabili d'ambiente in ambiente dev da file.env
- jest → testing
- supertest → testing

d. Database

Abbiamo definito alcune strutture dati con l'utilizzo di Mongoose.

Kind: ["report", "suggestion", "complaint"]

Category: ["lights", "road", "trash"]

State: ["active", "work_in_progress", "archived"]

User_level: ["citizen", "admin"]

User:

- name: String
- email: String
- password: String
- user_level: User_level
- reports: [Report]
- notifications: [Notification]

Report:

- title: String
- content: String
- user: User
- votes: [User]
- position: String
- kind: Kind
- category: Category
- state: State

Notification:

- title: String
- content: String
- report: Report

Comment:

- content: String
- report: Report

- user: User

e. Testing

Il testing è stato effettuato con il framework “jest” e la libreria “supertest”. All’interno del documento “app/treport.test.js” è presente il codice che implementa tutti i casi di test elencati qui sotto. Il documento è suddiviso nei vari requisiti funzionali, che raggruppano i singoli test case relativi. Il testing è stato sviluppato utilizzando come database iniziale quello definito da “scripts/testSetup.js”. Dunque, per verificare il suo funzionamento, è necessario prima digitare il comando “npm test setup” e attendere fino a quando il messaggio “setup completed” non appare a schermo. Successivamente, con il comando “npm run test” è possibile eseguire il testing.

Requisito Funzionale	Numero test case	Descrizione Test Case	Test Data	Precondizioni	Dipendenze	Risultato Atteso	Risultato Ricontrato	Note:
RF1	1	Un utente non loggato accede ad un account cittadino	<email> non vuota <password> non vuota	<email> e <password> corrispondono ad un account cittadino esistente	---	All'utente viene consentito l'accesso solo con le autorizzazioni previste all'utente cittadino	Il risultato riscontrato corrisponde a quello atteso	---
RF1	1.1	Un utente non loggato accede ad un account comune	<email> non vuota <password> non vuota	<email> e <password> corrispondono ad un account comune esistente	---	All'utente viene consentito l'accesso solo con le autorizzazioni previste all'utente comune	Il risultato riscontrato corrisponde a quello atteso	---
RF1	1.2	Un utente non loggato accede ad un account amministratore	<email> non vuota <password> non vuota	<email> e <password> corrispondono ad un account amministratore esistente	---	All'utente viene consentito l'accesso solo con le autorizzazioni previste all'utente amministratore	Il risultato riscontrato corrisponde a quello atteso	---
RF1	2	Un utente non loggato accede al suo account con password sbagliata	<email> non vuota <password> non vuota	<email> associata ad un account <password> non legata	---	All'utente non viene consentito l'accesso	Il risultato riscontrato corrisponde a quello atteso	---

				all'account relativo				
RF1	3	Un utente non loggato accede ad un account inesistente	<email> non vuota <password> non vuota	<email> non è collegata a nessun account	---	All'utente non viene consentito l'accesso	Il risultato riscontrato corrisponde a quello atteso	---
RF1	4	Un utente esegue l'accesso non compilando un campo	<email> non vuota / vuota <password> non vuota / vuota	<email>, se presente, non è legata ad alcun account	---	All'utente non viene consentito l'accesso	Il risultato riscontrato corrisponde a quello atteso	---
RF2	1	Un utente cittadino crea un account	<email> non vuota <password> non vuota <nome utente> non vuoto	<email> valida e non legata ad un account già esistente	---	L'account viene correttamente creato nel database e l'utente è automaticamente loggato con le nuove credenziali	Il risultato riscontrato corrisponde a quello atteso	---
RF2	2	Un utente si registra lasciando uno dei campi vuoti	<email> vuota / non vuota <password> vuota / non vuota <nome utente> vuoto / non vuoto	<email>, se presente, è valida e non legata ad alcun account esistente	---	L'account non viene creato non viene effettuata alcuna autenticazione	Il risultato riscontrato corrisponde a quello atteso	---
RF2	3	Un utente si registra inserendo una mail già legata ad un account	<email> non vuota <password> non vuota <nome utente> non vuoto	<email> legata ad un account già esistente	---	L'account non viene creato non viene effettuata alcuna autenticazione	Il risultato riscontrato corrisponde a quello atteso	---
RF3	1	Un utente loggato commenta una segnalazione	<contenuto> non vuoto	Utente loggato	---	Il commento viene correttamente salvato nel database e compare nella sezione commenti della segnalazione	Il risultato riscontrato corrisponde a quello atteso	---
RF3	1.1	Un utente non loggato commenta una segnalazione	<contenuto> non vuoto	Utente non loggato	---	La creazione del commento non viene permessa	Il risultato riscontrato corrisponde a quello atteso	---
RF3	1.2	Un utente comune commenta una segnalazione	<contenuto> non vuoto	Utente comune loggato	---	Il commento viene correttamente salvato nel database e compare nella sezione commenti della segnalazione evidenziato	Il risultato riscontrato corrisponde a quello atteso	---

RF4	1	Un utente cittadino vota una segnalazione	---	Utente cittadino loggato	---	Vengono inserite le informazioni riguardo alla votazione correttamente nel database	Il risultato riscontrato corrisponde a quello atteso	---
RF4	1.1	Un utente comune vota una segnalazione	---	Utente comune loggato	---	Vengono inserite le informazioni riguardo alla votazione correttamente nel database	Il risultato riscontrato corrisponde a quello atteso	---
RF4	2	Un utente non loggato vota una segnalazione	---	Utente non loggato	---	L'utente viene reindirizzato alla pagina login, senza che la votazione venga aggiunta al database	Il risultato riscontrato corrisponde a quello atteso	---
RF6	1	Un utente loggato crea una nuova segnalazione	<titolo> non vuoto <contenuto> non vuoto <immagine> vuota <posizione> non vuota <categoria> non vuota <tipo> non vuota	Utente loggato	---	Viene aggiunta una nuova segnalazione al database	Il risultato riscontrato corrisponde a quello atteso	---
RF6	2	Un utente non loggato crea una nuova segnalazione	<titolo> non vuoto <contenuto> non vuoto <immagine> vuota <posizione> non vuota <categoria> non vuota <tipo> non vuota	Utente non loggato	---	L'utente viene reindirizzato alla pagina di login e non gli viene concesso di creare una nuova segnalazione	Il risultato riscontrato corrisponde a quello atteso	---
RF6	3	Un utente loggato inserisce una descrizione troppo lunga	<titolo> non vuoto <contenuto> con più caratteri del consentito <immagine> vuota <posizione> non vuota <categoria> non vuota <tipo> non vuota	Utente loggato	---	Non viene creata la nuova segnalazione e viene comunicato all'utente	Il risultato riscontrato corrisponde a quello atteso	---

RF6	3.1	Un utente loggato inserisce un titolo troppo lungo	<titolo> con più caratteri del consentito <contenuto> non vuoto <immagine> vuota <posizione> non vuota <categoria> non vuota <tipo> non vuota	Utente loggato	---	Non viene creata la nuova segnalazione e viene comunicato all'utente	Il risultato riscontrato corrisponde a quello atteso	---
RF6	4	Un utente loggato non riempie uno dei campi necessari	<titolo>, <contenuto>, <immagine>, <posizione>, <categoria>, <tipo> non vuoti a parte 1 o più campi	Utente loggato	---	Non viene creata la nuova segnalazione e viene comunicato all'utente	Il risultato riscontrato corrisponde a quello atteso	---
RF9	1	L'utente comune cambia lo stato di una segnalazione	<stato finale> non vuoto	Utente comune loggato <commento> già creato	---	Lo stato della segnalazione viene correttamente cambiato e salvato nel database	Il risultato riscontrato corrisponde a quello atteso	---
RF9	1.1	L'utente comune cambia lo stato di una segnalazione senza aver commentato	<stato finale> non vuoto	Utente comune loggato <commento> non creato	---	Il cambiamento dello stato non viene permesso	Il risultato riscontrato corrisponde a quello atteso	---
RF9	1.2	L'utente cittadino cambia lo stato della segnalazione	<stato finale> non vuoto	Utente cittadino loggato	---	Il cambiamento dello stato non viene permesso	Il risultato riscontrato corrisponde a quello atteso	---
RF10	1	L'utente riceve la notifica del cambiamento di stato dopo aver votato una segnalazione	---	Utente loggato Utente ha votato una segnalazione	L'utente deve aver votato la segnalazione e cambiata al test case 1.1 del RF9	L'utente riceve la notifica quando viene cambiato lo stato	Il risultato riscontrato corrisponde a quello atteso	---
RF10	2	L'utente che non ha votato una segnalazione non deve	---	Utente loggato Utente non ha votato la segnalazione	---	L'utente non riceve la notifica al cambiamento di stato	Il risultato riscontrato corrisponde a quello atteso	---

			ricevere la notifica riguardo il suo cambiamento di stato					
RF10	3	L'utente ha votato ed eliminato il voto della segnalazione di cui dopo viene cambiato lo stato	---	Utente loggato	---	L'utente non riceve la notifica al cambiamento di stato	Il risultato riscontrato corrisponde a quello atteso	---
RF17	1	L'utente amministratore registra un nuovo account comune	<email> non vuota e valida <password> non vuota <username> non vuoto	Utente amministratore e loggato	---	L'account comune viene aggiunto al database Un utente può autenticarsi usando le credenziali registrate	Il risultato riscontrato corrisponde a quello atteso	---
RF17	1.1	L'utente comune registra un nuovo account comune	<email> non vuota e valida <password> non vuota <username> non vuoto	Utente comune loggato	---	L'account non viene creato e viene mostrato un messaggio di errore	Il risultato riscontrato corrisponde a quello atteso	---
RF17	1.2	L'utente amministratore registra un nuovo account usando credenziali già salvate	<email> non vuota e valida <password> non vuota <username> non vuoto	Utente amministratore e loggato <email> esistente	---	L'account non viene creato e viene mostrato un messaggio di errore	Il risultato riscontrato corrisponde a quello atteso	---
RF18	1	L'utente comune elimina un commento	<id commento> non vuoto	Utente comune loggato	---	L'eliminazione viene effettuata con successo	Il risultato riscontrato corrisponde a quello atteso	---
RF18	1.1	L'utente comune elimina una segnalazione	<id segnalazione> non vuota	Utente comune loggato	---	L'eliminazione viene effettuata con successo	Il risultato riscontrato corrisponde a quello atteso	---
RF18	2	L'utente cittadino elimina un commento	<id commento> non vuoto	Utente cittadino loggato	---	L'eliminazione non viene eseguita	Il risultato riscontrato corrisponde a quello atteso	---
RF18	2.1	L'utente cittadino elimina una segnalazione	<id segnalazione> non vuota	Utente cittadino loggato	---	L'eliminazione non viene eseguita	Il risultato riscontrato corrisponde a quello atteso	---

5. FrontEnd

Il FrontEnd di Treport permette agli utenti di visualizzare, aggiungere, modificare e rimuovere dati all'interno dell'applicazione. La struttura comprende una Home page, una pagina di registrazione, una pagina di login, una pagina per la visualizzazione di un'intera segnalazione e dei suoi commenti, una pagina per la creazione di una segnalazione, una pagina profilo e una pagina dedicata all'analisi dei dati.

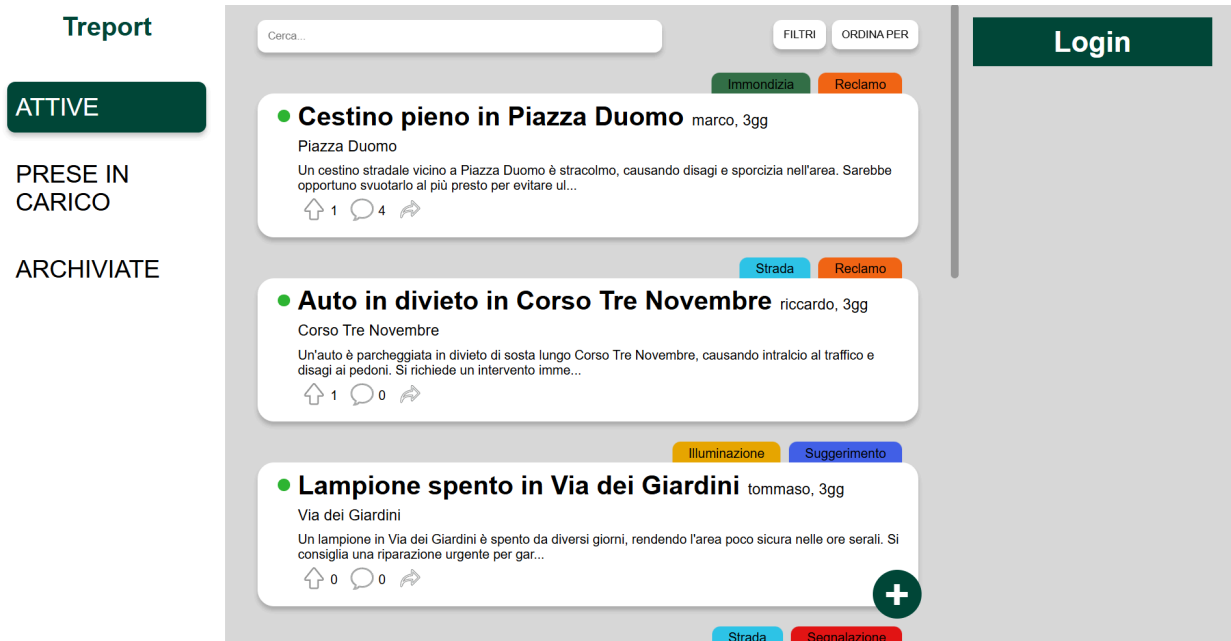
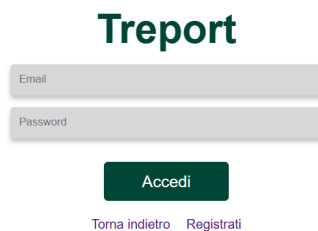


Figura 1

In Figura 1 è mostrata l'home page dell'applicazione. L'utente nella home page può scorrere le preview delle segnalazioni, filtrarle, ordinarle, votarle (se autenticati), condividerle o eliminarle (se si tratta di un utente del comune), può cambiare bacheca di visualizzazione segnalazioni e può accedere a molte delle altre pagine, quali la pagina di creazione segnalazione (se autenticati), quella di segnalazione ingrandita, quella di login e la pagina profilo (se autenticati). Inoltre un utente autenticato visualizza nella home page anche i filtri preferiti e le notifiche ricevute. Se un utente non autenticato prova a svolgere un'azione per cui è richiesta l'autenticazione si troverà a scegliere se autenticarsi o tornare indietro tramite apposito messaggio.



The login form for Treport features a dark green header with the brand name. Below it are two light gray input fields for email and password. A dark green button with white text is centered below the fields. At the bottom, there are two links: one in purple to go back and one in dark green to register.

Treport

Email

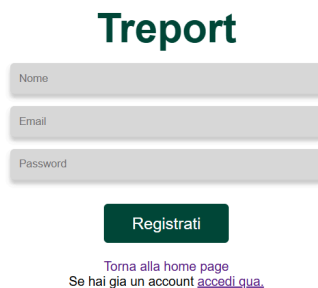
Password

Accedi

[Torna indietro](#) [Registrati](#)

Figura 2

In Figura 2 è mostrata la pagina di login in cui l'utente può autenticarsi inserendo le credenziali di accesso oppure può accedere alla pagina di registrazione.



The registration form for Treport has a dark green header with the brand name. It includes three light gray input fields for name, email, and password. A dark green button with white text is centered below the fields. At the bottom, there is a link in purple to return to the home page and a line of text in dark green with a link in purple to log in if the user already has an account.

Treport

Nome

Email

Password

Registrati

[Torna alla home page](#)
Se hai già un account [accedi qua.](#)

Figura 3

In Figura 3 è mostrata la pagina di registrazione in cui l'utente può creare delle nuove credenziali di accesso.

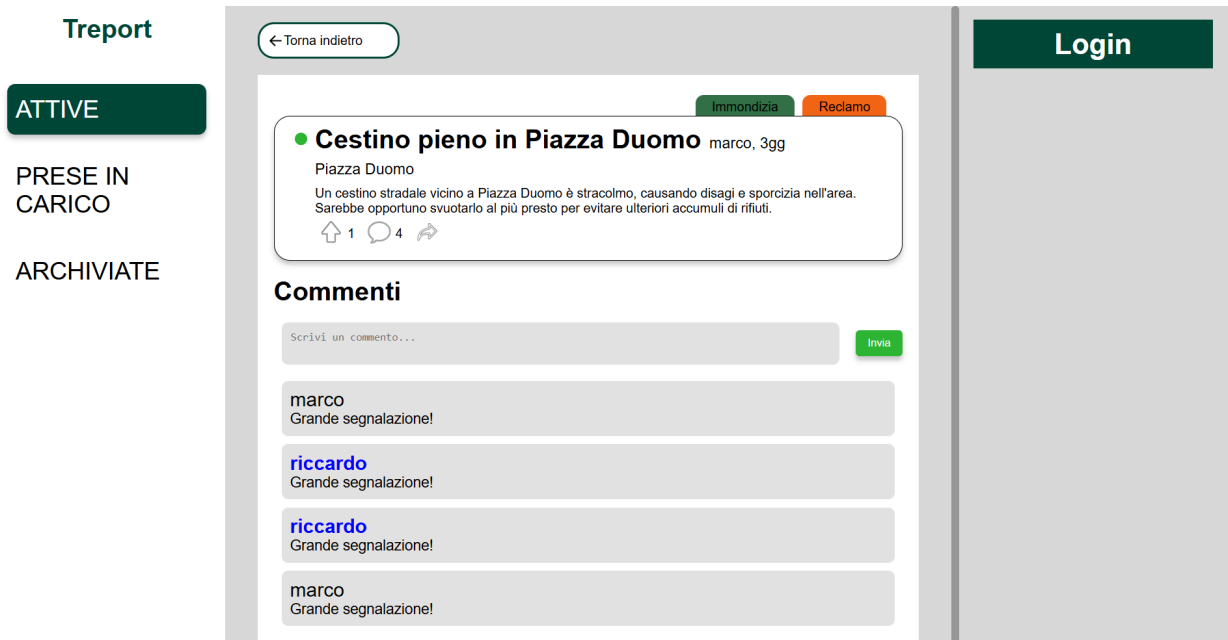


Figura 4

In Figura 4 è mostrata la pagina di segnalazione ingrandita. Qui l'utente può leggere l'intero contenuto della segnalazione e leggere i commenti relativi. Se autenticato, l'utente può anche aggiungere dei commenti. L'utente del comune avrà a disposizione un bottone per cambiare lo stato della segnalazione. Inoltre tutti i bottoni presenti nelle preview della segnalazioni nella home page, quali l'upvote e la condivisione, sono disponibili anche in questa pagina. barre laterali

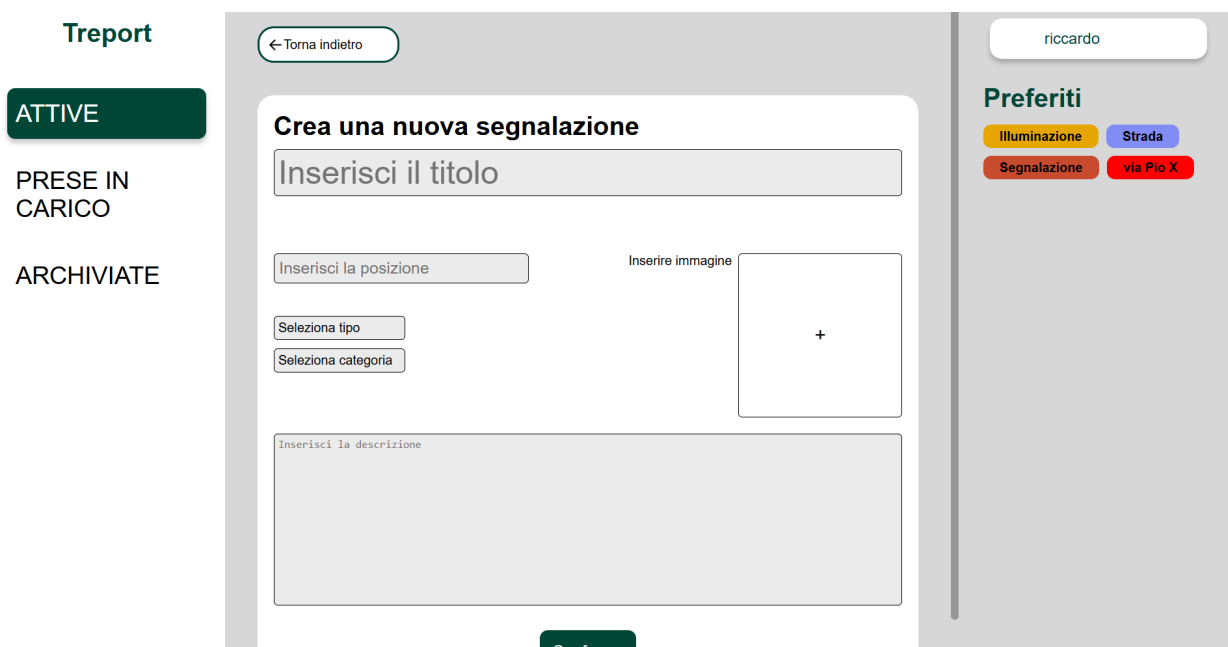


Figura 5

In Figura 5 è mostrata la pagina di creazione di una nuova segnalazione. Qui un utente autenticato può pubblicare una segnalazione inserendo i campi obbligatori come “Titolo”, “Contenuto”, “Categoria”, “Tipo” e quelli facoltativi, come l’immagine.



Figura 6

In Figura 6 è mostrata la pagina profilo di un utente autenticato, la quale mostra i dati dell'utente, le segnalazioni da lui pubblicate e quelle supportate.

6. Deployment

Abbiamo deciso di hostare backend e frontend assieme. Per fare ciò il backend fornisce staticamente il punto di ingresso per il frontend. Un'istanza della nostra applicazione è disponibile al link <https://progetto-ids.onrender.com> (Il sito potrebbe richiedere fino ad un minuto per caricarsi la prima volta, questo è dovuto alle limitazioni del piano gratuito del servizio di host usato). Grazie alle funzionalità offerte dal servizio di host viene fatto il deploy del sito ogni volta che viene aggiunto un commit al branch "deploy" del nostro repository.

Gli account disponibili per il sito sono:

Nome utente	Email	Password	Livello utente
admin	admin@gmail.com	admin	amministratore
comune1	comune1@gmail.com	comune1	comune
comune2	comune2@gmail.com	comune2	comune
cittadino1	cittadino1@gmail.com	cittadino1	cittadino
cittadino2	cittadino2@gmail.com	cittadino2	cittadino

Per problemi con il backend contattare: riccardo.dincau@studenti.unitn.it (Amministratore deploy), marco.zanatta@studenti.unitn.it, tommaso.ascolani@studenti.unitn.it.